

**Administración de la gestión de recursos
en Oracle® Solaris 11.2**

Borrador 2015-01-20-19:44:52+01:00

Copyright © 2004, 2014, Oracle y/o sus filiales. Todos los derechos reservados.

Este software y la documentación relacionada están sujetos a un contrato de licencia que incluye restricciones de uso y revelación, y se encuentran protegidos por la legislación sobre la propiedad intelectual. A menos que figure explícitamente en el contrato de licencia o esté permitido por la ley, no se podrá utilizar, copiar, reproducir, traducir, emitir, modificar, conceder licencias, transmitir, distribuir, exhibir, representar, publicar ni mostrar ninguna parte, de ninguna forma, por ningún medio. Queda prohibida la ingeniería inversa, desensamblaje o descompilación de este software, excepto en la medida en que sean necesarios para conseguir interoperabilidad según lo especificado por la legislación aplicable.

La información contenida en este documento puede someterse a modificaciones sin previo aviso y no se garantiza que se encuentre exenta de errores. Si detecta algún error, le agradeceremos que nos lo comunique por escrito.

Si este software o la documentación relacionada se entrega al Gobierno de EE.UU. o a cualquier entidad que adquiera licencias en nombre del Gobierno de EE.UU. se aplicará la siguiente disposición:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Este software o hardware se ha desarrollado para uso general en diversas aplicaciones de gestión de la información. No se ha diseñado ni está destinado para utilizarse en aplicaciones de riesgo inherente, incluidas las aplicaciones que pueden causar daños personales. Si utiliza este software o hardware en aplicaciones de riesgo, usted será responsable de tomar todas las medidas apropiadas de prevención de fallos, copia de seguridad, redundancia o de cualquier otro tipo para garantizar la seguridad en el uso de este software o hardware. Oracle Corporation y sus filiales declinan toda responsabilidad derivada de los daños causados por el uso de este software o hardware en aplicaciones de riesgo.

Oracle y Java son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

Intel e Intel Xeon son marcas comerciales o marcas comerciales registradas de Intel Corporation. Todas las marcas comerciales de SPARC se utilizan con licencia y son marcas comerciales o marcas comerciales registradas de SPARC International, Inc. AMD, Opteron, el logotipo de AMD y el logotipo de AMD Opteron son marcas comerciales o marcas comerciales registradas de Advanced Micro Devices. UNIX es una marca comercial registrada de The Open Group.

Este software o hardware y la documentación pueden ofrecer acceso a contenidos, productos o servicios de terceros o información sobre los mismos. Ni Oracle Corporation ni sus filiales serán responsables de ofrecer cualquier tipo de garantía sobre el contenido, los productos o los servicios de terceros y renuncian explícitamente a ello. Oracle Corporation y sus filiales no se harán responsables de las pérdidas, los costos o los daños en los que se incurra como consecuencia del acceso o el uso de contenidos, productos o servicios de terceros.

This documentation is in pre-production status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Contenido

Uso de esta documentación	11
1 Introducción a la gestión de recursos	13
Descripción general de la gestión de recursos	13
Clasificación de los recursos	14
Mecanismos de control de administración de recursos	15
Configuración de la gestión de recursos	16
Interacción con zonas no globales	16
Cuándo utilizar la administración de recursos	16
Consolidación de servidores	17
Bases de usuarios diversas y de gran tamaño	17
Configuración del mapa de tareas de gestión de recursos	18
2 Acerca de los proyectos y las tareas	21
Funciones de proyectos y tareas	21
Identificadores de proyecto	22
Determinación de un proyecto predeterminado del usuario	22
Definición de atributos de usuario con los comandos useradd y usermod	23
Base de datos project	23
Subsistema PAM	24
Configuración de servicios de nombres	24
Formato de archivo /etc/project local	25
Configuración del proyecto para NIS	27
Configuración del proyecto para LDAP	27
Identificadores de tareas	27
Comandos utilizados con proyectos y tareas	29
3 Administración de proyectos y tareas	31
Administración del mapa de tareas de proyectos y tareas	31
Ejemplos y opciones de comandos	32

Opciones de comandos utilizadas con proyectos y tareas	32
Uso de cron y su con proyectos y tareas	34
Administración de proyectos	35
▼ Cómo definir un proyecto y ver el proyecto actual	35
▼ Cómo suprimir un proyecto del archivo /etc/project	37
Cómo validar el contenido del archivo /etc/project	38
Cómo obtener información sobre la pertenencia de un proyecto a un grupo	39
▼ Cómo crear una tarea	39
▼ Cómo mover un proceso en ejecución a una nueva tarea	39
Edición y validación de atributos de proyecto	40
▼ Cómo agregar atributos y valores de atributos a los proyectos	40
▼ Cómo eliminar valores de atributo de los proyectos	41
▼ Cómo eliminar un atributo de control de recursos de un proyecto	41
▼ Cómo sustituir atributos y valores de atributos para los proyectos	42
▼ Cómo eliminar los valores existentes para un atributo de control de recursos	42
4 Acerca de la contabilidad ampliada	43
Introducción a las cuentas extendidas	43
Funcionamiento de la contabilidad ampliada	44
Formato ampliable	45
Registros y formato exacct	45
Uso de contabilidad ampliada en un sistema Oracle Solaris con Zones instalado	46
Configuración de contabilidad ampliada	46
Inicio y activación persistente de contabilidad ampliada	46
Registros	47
Comandos utilizados con contabilidad ampliada	47
Interfaz Perl para libexacct	48
5 Administración de tareas de contabilidad ampliada	51
Administración del mapa de tareas de la función de contabilidad ampliada	51
Uso de funciones de cuentas extendidas	52
▼ Cómo activar la contabilidad ampliada para los flujos, los procesos, las tareas y los componentes de red	52
Cómo mostrar el estado de la contabilidad ampliada	53
Cómo ver los recursos de cuentas disponibles	53
▼ Cómo desactivar la contabilidad de procesos, tareas, flujos y gestión de redes	54

Uso de la interfaz de Perl para libexacct	55
Cómo imprimir de forma recurrente el contenido de un objeto exacct	55
Cómo crear un registro de grupo y guardarlo en un archivo	57
Cómo imprimir el contenido de un archivo exacct	57
Ejemplo de salida de Sun::Solaris::Exacct::Object->dump()	58
6 Acerca de los controles de recursos	59
Conceptos de controles de recursos	59
Límites y controles de recursos	60
Comunicación entre procesos y controles de recursos	60
Mecanismos de limitación del control de recursos	60
Mecanismos de atributos de proyecto	61
Configuración de controles de recursos y atributos	61
Controles de recursos disponibles	62
Controles de recursos de toda la zona	65
Compatibilidad con unidades	66
Valores de controles de recursos y niveles de privilegio	67
Acciones locales y globales en valores de controles de recursos	68
Propiedades e indicadores de controles de recursos	70
Aplicación de controles de recursos	71
Supervisión global de los eventos de controles de recursos	72
Aplicación de controles de recursos	72
Actualización temporal de los valores de controles de recursos en un sistema en ejecución	73
Actualización de estados de registro	73
Actualización de controles de recursos	73
Comandos utilizados con controles de recursos	74
7 Administración de las tareas de controles de recursos	75
Administración del mapa de tareas de controles de recursos	75
Configuración de controles de recursos	76
▼ Cómo definir el número máximo de procesos ligeros para cada tarea de un proyecto	76
▼ Cómo definir múltiples controles en un proyecto	77
Uso del comando prctl	78
▼ Cómo utilizar el comando prctl para ver los valores de control de recursos predeterminados	79
▼ Cómo utilizar el comando prctl para ver información de un control de recurso	81

▼ Cómo utilizar <code>prctl</code> para cambiar un valor temporalmente	81
▼ Cómo utilizar <code>prctl</code> para reducir el valor de control de recurso	82
▼ Cómo utilizar <code>prctl</code> para ver, sustituir y verificar el valor de un control en un proyecto	82
Uso de <code>rctladm</code>	83
Cómo utilizar <code>rctladm</code>	83
Uso de <code>ipcs</code>	84
Cómo utilizar <code>ipcs</code>	84
Advertencias de capacidad	84
▼ Cómo determinar si un servidor Web tiene asignada suficiente capacidad de CPU	85
8 Acerca del planificador por reparto equitativo	87
Introducción al programador	87
Definición de recurso compartido de CPU	88
Recursos compartidos de la CPU y estado del proceso	89
Recursos compartidos de la CPU frente al uso	89
Ejemplos de recursos compartidos de la CPU	89
Ejemplo 1: Dos procesos vinculados a la CPU en cada proyecto	90
Ejemplo 2: Proyectos que no compiten	90
Ejemplo 3: No se puede ejecutar un proyecto	91
Configuración de FSS	92
Proyectos y usuarios	92
Configuración de recursos compartidos de la CPU	92
FSS y conjuntos de procesadores	93
FSS y ejemplos de conjuntos de procesadores	94
Combinación de FSS con otras clases de programación	96
Configuración de la clase de programación para el sistema	97
Clase de programación en un sistema con zonas instaladas	97
Comandos utilizados con FSS	97
9 Administración de las tareas del planificador por reparto equitativo	99
Administración del mapa de tareas del planificador de reparto equitativo	99
Supervisión de FSS	100
▼ Cómo supervisar el uso que hacen los proyectos de la CPU del sistema	100
▼ Cómo supervisar el uso de la CPU que hacen los proyectos en los conjuntos de procesadores	101
Configuración de FSS	101
Enumeración de clases de programador en el sistema	101

▼ Cómo convertir FSS en la clase de programador predeterminada	102
▼ Cómo mover manualmente los procesos de la clase TS a la clase FSS	102
▼ Cómo mover manualmente los procesos de todas las clases de usuario a la clase FSS	103
▼ Cómo mover manualmente los procesos de un proyecto a la clase FSS	103
Cómo ajustar los parámetros del programador	103
10 Acerca del control de memoria física mediante el daemon de limitación de recursos	105
Introducción al daemon de límite de recursos	105
Funcionamiento de los límites de recursos	106
Atributo para limitar el uso de memoria física para proyectos	106
Configuración de rcapd	107
Uso del daemon de límite de recursos en un sistema con zonas instaladas	108
Umbral de aplicación de límite de memoria	108
Determinación de valores límite	109
Intervalos de funcionamiento de rcapd	110
Supervisión del uso de recursos con rcapstat	111
Comandos utilizados con rcapd	112
11 Administración de las tareas del daemon de limitación de recursos	115
Configuración de límite del tamaño del conjunto residente	115
▼ Cómo agregar un atributo rcap.max-rss para un proyecto	115
▼ Cómo utilizar el comando projmod para agregar un atributo rcap.max-rss para un proyecto	116
Configuración y uso del mapa de tareas del daemon de limitación de recursos	116
Administración del daemon de límite de recursos con rcapadm	117
▼ Cómo establecer el umbral de aplicación del límite de memoria	117
▼ Cómo configurar intervalos de funcionamiento	117
▼ Cómo activar los límites de recursos	118
▼ Cómo desactivar los límites de recursos	118
▼ Cómo especificar un límite de recursos temporal para una zona	119
Creación de informes con rcapstat	119
Límite de informe e información del proyecto	120
Supervisión del RSS de un proyecto	120
Cómo determinar el tamaño del conjunto de trabajo de un proyecto	121
Informes del uso de la memoria y el umbral de aplicación del límite de la memoria	122

12 Acerca de las agrupaciones de recursos	123
Introducción a las agrupaciones de recursos	124
Introducción a las agrupaciones de recursos dinámicos	125
Activación y desactivación de agrupaciones de recursos y agrupaciones de recursos dinámicos	125
Agrupaciones de recursos utilizadas en zonas	125
Cuándo se utilizan las agrupaciones	126
Estructura de agrupaciones de recursos	127
Contenido de /etc/pooladm.conf	128
Propiedades de agrupaciones	129
Implementación de agrupaciones en un sistema	129
Atributo project.pool	130
SPARC: Agrupaciones de recursos y operaciones de reconfiguración dinámica	130
Creación de configuraciones de agrupaciones	131
Asignación específica de CPU, núcleos centrales y sockets	131
Manipulación directa de la configuración dinámica	132
Descripción general de poold	132
Administración de agrupaciones de recursos dinámicos	133
Objetivos y restricciones de configuración	133
Restricciones de la configuración	133
Objetivos de la configuración	134
Propiedades de poold	137
Funcionalidad poold que se puede configurar	138
Intervalo de supervisión de poold	138
Información de registro de poold	138
Ubicación de registro	141
Administración de registros con logadm	141
Cómo funciona la asignación de recursos dinámicos	141
Acerca de los recursos disponibles	142
Determinación de recursos disponibles	142
Identificación de recursos insuficientes	143
Determinación del uso de recursos	143
Identificación de infracciones del control	143
Determinación de la acción correctiva apropiada	144
Uso de poolstat para supervisar la función de agrupaciones y el uso de los recursos	144
Salida de poolstat	145
Ajuste de los intervalos de funcionamiento de poolstat	146
Comandos utilizados con la función de agrupaciones de recursos	146

13 Creación y administración de las tareas de agrupaciones de recursos	149
Administración del mapa de tareas de agrupaciones de recursos	149
Activación y desactivación de la función de agrupaciones	150
▼ Cómo activar el servicio de agrupaciones de recursos utilizando svcadm	151
▼ Cómo desactivar el servicio de agrupaciones de recursos utilizando svcadm	151
▼ Cómo activar el servicio de agrupaciones de recursos dinámicos utilizando svcadm	151
▼ Cómo desactivar el servicio de agrupaciones de recursos dinámicos utilizando svcadm	154
▼ Cómo activar las agrupaciones de recursos utilizando pooladm	154
▼ Cómo desactivar las agrupaciones de recursos utilizando pooladm	154
Asignación de CPU específica	155
Configuración de agrupaciones	155
▼ Cómo crear una configuración estática	155
▼ Cómo modificar una configuración	157
▼ Cómo asociar una agrupación con una clase de planificación	159
▼ Cómo establecer restricciones de configuración	161
▼ Cómo definir los objetivos de configuración	161
▼ Cómo establecer el nivel de registro de poold	164
▼ Cómo utilizar los archivos de comando con poolcfg	164
Transferencia de recursos	165
▼ Cómo mover CPU entre conjuntos de procesadores	165
Activación y eliminación de configuraciones de agrupaciones	166
▼ Cómo activar una configuración de agrupaciones	166
▼ Cómo validar una configuración antes de confirmarla	166
▼ Cómo eliminar una configuración de agrupaciones	167
Configuración de atributos de agrupaciones y vinculación a una agrupación	167
▼ Cómo vincular procesos a una agrupación	168
▼ Cómo vincular tareas o proyectos a una agrupación	168
▼ Cómo configurar el atributo project.pool para un proyecto	168
▼ Cómo utilizar los atributos project para vincular un proceso a una agrupación diferente	169
Uso de poolstat para registrar estadísticas para los recursos relacionados con las agrupaciones	169
Visualización de la salida predeterminada de poolstat	170
Creación de varios informes en los intervalos especificados	170
Registro de estadísticas del conjunto de recursos	170

14 Ejemplo de configuración de administración de recursos	173
Configuración que consolidar	173
Configuración de consolidación	174
Creación de la configuración	174
Visualización de la configuración	176
 Índice	 181

Uso de esta documentación

- **Descripción general:** describe cómo escribir aplicaciones que particionan y administran los recursos del sistema, como los juegos de procesadores y las clases de programación.
- **Audiencia:** los programadores con experiencia con las interfaces del sistema operativo.
- **Conocimientos necesarios:** conocimientos de interfaces de los sistemas C y de Solaris (u otros similares a Unix).

Biblioteca de documentación del producto

En la biblioteca de documentación, que se encuentra en <http://www.oracle.com/pls/topic/lookup?ctx=E56339>, se incluye información de última hora y problemas conocidos para este producto.

Acceso a My Oracle Support

Los clientes de Oracle disponen de asistencia a través de Internet en el portal My Oracle Support. Para obtener más información, visite <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> o, si tiene alguna discapacidad auditiva, visite <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>.

Comentarios

Envíenos comentarios acerca de esta documentación mediante <http://www.oracle.com/goto/docfeedback>.

BETA
DRAFT

♦ ♦ ♦ 1 C A P Í T U L O 1

Introducción a la gestión de recursos

La funcionalidad de gestión de recursos de Oracle Solaris permite controlar el modo en que las aplicaciones utilizan los recursos del sistema disponibles. Puede realizar lo siguiente:

- Asignar recursos informáticos, como tiempo del procesador
- Supervisar el modo en que se utilizan las asignaciones y luego ajustarlas según sea preciso
- Generar información de contabilidad ampliada para análisis, facturación y planificación de capacidades

En este capítulo se tratan los temas siguientes.

- [“Descripción general de la gestión de recursos” \[13\]](#)
- [“Cuándo utilizar la administración de recursos” \[16\]](#)
- [“Configuración del mapa de tareas de gestión de recursos” \[18\]](#)

Descripción general de la gestión de recursos

Los entornos informáticos modernos deben proporcionar una respuesta flexible a las diferentes cargas de trabajo que generan las distintas aplicaciones de un sistema. Una *carga de trabajo* es una adición de todos los procesos de una aplicación o grupo de aplicaciones. Si no se utilizan las funciones de gestión de recursos, el sistema operativo Oracle Solaris responde a las demandas de carga de trabajo adaptándose a las nuevas solicitudes de aplicaciones de forma dinámica. Esta respuesta predeterminada general significa que toda la actividad del sistema tiene el mismo acceso a los recursos. Las funciones de gestión de recursos le permiten tratar las cargas de trabajo individualmente. Puede realizar lo siguiente:

- Restringir el acceso a un recurso específico
- Ofrecer recursos a las cargas de trabajo de modo preferencial
- Aislar cargas de trabajo entre sí

La capacidad de minimizar los compromisos de rendimiento de varias cargas de trabajo, junto con las funciones que supervisan el uso de los recursos, se denomina *administración de recursos*. La administración de recursos se implementa a través de un conjunto de algoritmos. Los algoritmos controlan las series de solicitudes de capacidades que presenta una aplicación durante su ejecución.

Las funciones de administración de recursos permiten modificar el comportamiento predeterminado del sistema operativo con respecto a las diferentes cargas de trabajo. El *comportamiento* hace referencia principalmente al conjunto de decisiones que toman los algoritmos del sistema operativo cuando una aplicación presenta una o más solicitudes de recursos para el sistema. Puede utilizar las funciones de administración de recursos para:

- Denegar recursos o preferir una aplicación para un conjunto de asignaciones mayor que no se permitiría con otra aplicación
- Tratar determinadas asignaciones de forma colectiva en lugar de utilizar mecanismos aislados

La implementación de una configuración del sistema que utilice las funciones de administración de recursos puede tener varias finalidades. Puede realizar lo siguiente:

- Impedir a una aplicación el consumo indiscriminado de recursos
- Cambiar una prioridad de aplicación basándose en eventos externos
- Equilibrar las garantías de recursos para un conjunto de aplicaciones con el fin de maximizar el uso del sistema

Al planificar una configuración administrada por recursos, algunos de los requisitos clave son:

- Identificar las cargas de trabajo implicadas en el sistema
- Distinguir las cargas de trabajo que no están en conflicto de las que tienen requisitos de rendimiento que afectan a las cargas de trabajo principales

Una vez identificadas las cargas de trabajo conflictivas y las de cooperación, puede crear una configuración de recursos que presente el menor compromiso para los objetivos de servicio del negocio, dentro de las limitaciones de las funciones del sistema.

El sistema Oracle Solaris cuenta con una gestión de recursos eficaz que ofrece mecanismos de control, de notificación y de supervisión. Muchas de estas funciones se proporcionan a través de mejoras en los mecanismos existentes, como el sistema de archivos [proc\(4\)](#), los juegos de procesadores y las clases de planificación. Otras funciones son específicas de la administración de recursos. Estas funciones se describen en los capítulos siguientes.

Clasificación de los recursos

Un recurso es cualquier aspecto del sistema informático que pueda manipularse con la finalidad de cambiar el comportamiento de la aplicación. En consecuencia, un recurso es una capacidad que solicita una aplicación implícita o explícitamente. Si se deniega o limita la capacidad, la ejecución de una aplicación potente es más lenta.

La clasificación de recursos, a diferencia de la identificación de recursos, puede realizarse para una serie de ejes. Los ejes se pueden solicitar de forma implícita (en contraposición a su solicitud explícita), pueden basarse en el tiempo, como el tiempo de la CPU (en contraposición a las solicitudes independientes del tiempo, como los recursos compartidos de la CPU), etc.

Generalmente, la administración de recursos basada en el planificador se aplica a los recursos que puede solicitar la aplicación de forma implícita. Por ejemplo, para continuar la ejecución, una aplicación solicita de forma implícita tiempo de CPU adicional. Para grabar datos en un socket de red, una aplicación solicita ancho de banda implícitamente. Pueden colocarse limitaciones en el uso total de un recurso solicitado de forma implícita.

Pueden presentarse interfaces adicionales para que los niveles de servicio de CPU o el ancho de banda se negocien de forma explícita. Los recursos que se solicitan de forma explícita, como una solicitud de subprocesso adicional, se pueden administrar por limitación.

Mecanismos de control de administración de recursos

Los tres tipos de mecanismos de control disponibles en el sistema operativo Oracle Solaris son las limitaciones, la programación y las particiones.

Mecanismos de limitación

Las limitaciones permiten al administrador o al desarrollador de aplicaciones definir los límites de consumo de recursos específicos para una carga de trabajo. Con unos límites establecidos, el consumo de recursos de modelación pasa a ser un proceso más sencillo. Asimismo, pueden utilizarse los límites para controlar las aplicaciones cuyo comportamiento incorrecto podría afectar al rendimiento o la disponibilidad del sistema a través de solicitudes de recursos no reguladas.

Las limitaciones suponen complicaciones para la aplicación. La relación entre la aplicación y el sistema puede modificarse hasta que la aplicación deje de funcionar. Para atenuar este riesgo, puede reducir gradualmente las limitaciones en las aplicaciones cuyos recursos tienen un comportamiento desconocido. Los controles de recursos que se describen en [Capítulo 6, Acerca de los controles de recursos](#) ofrecen un mecanismo de limitación. Las aplicaciones más nuevas pueden programarse para que tengan en cuenta sus limitaciones de recursos, pero no todos los programadores deciden hacerlo.

Mecanismos de planificación

La planificación hace referencia a la toma de una serie de decisiones de asignación a intervalos específicos. La decisión que se toma se basa en un algoritmo predecible. Una aplicación que no necesita su asignación actual deja el recurso disponible para el uso de otra aplicación. La administración de recursos basada en la planificación permite el uso completo de una configuración no asignada, a la vez que proporciona asignaciones controladas en una situación de asignaciones excesivas. El algoritmo subyacente define el modo en que se interpreta el término "controlado". En algunas instancias, es posible que el algoritmo de planificación,

garantice que todas las aplicaciones tengan algún tipo de acceso al recurso. El planificador por reparto equitativo (FSS) descrito en [Capítulo 8, Acerca del planificador por reparto equitativo](#) gestiona el acceso de la aplicación a los recursos de la CPU de un modo controlado.

Mecanismos de partición

La partición se utiliza para vincular una carga de trabajo con un subjuego de los recursos disponibles del sistema. Este vínculo garantiza que siempre haya disponible una cantidad determinada de recursos para la carga de trabajo. Las funciones de agrupaciones de recursos que se describen en [Capítulo 12, Acerca de las agrupaciones de recursos](#) permiten limitar las cargas de trabajo de subjuegos específicos del equipo.

En las configuraciones que utilizan la partición se puede evitar la asignación excesiva en el sistema. Sin embargo, al evitar esta asignación excesiva, la capacidad de obtener usos elevados puede verse reducida. Un grupo reservado de recursos, como procesadores, no está disponible para ser utilizado por otra carga de trabajo cuando la carga vinculada a ellos está inactiva.

Configuración de la gestión de recursos

Partes de la configuración de administración de recursos puede colocarse en un servicio de nombres de la red. Esta función permite al administrador aplicar limitaciones de gestión de recursos a un grupo de equipos, en lugar de basarse en los equipos individuales. Los trabajos relacionados pueden compartir un identificador común, y el uso adicional de dichos trabajos se puede tabular desde los datos de cuentas.

La configuración de administración de recursos y los identificadores orientados a la carga de trabajo se describen con mayor detalle en [Capítulo 2, Acerca de los proyectos y las tareas](#). La función de contabilidad ampliada que vincula estos identificadores con el uso del recurso de la aplicación se describe en [Capítulo 4, Acerca de la contabilidad ampliada](#).

Interacción con zonas no globales

Las funciones de gestión de recursos pueden utilizarse con zonas para detallar más el entorno de aplicación. Las interacciones entre estas funciones y las zonas se describen en las secciones aplicables de esta guía.

Cuándo utilizar la administración de recursos

Utilice la administración de recursos para asegurarse de que las aplicaciones cumplan los tiempos de respuesta requeridos.

La administración de recursos también puede aumentar el uso de los recursos. Al categorizar y priorizar el uso, puede utilizar de forma eficaz la capacidad de reserva durante los períodos no pico, lo que a menudo acaba con la necesidad de potencia de procesamiento adicional. También puede asegurarse de que los recursos no se malgasten debido a la variabilidad de la carga.

Consolidación de servidores

La administración de recursos es ideal para los entornos que consolidan una serie de aplicaciones en un único servidor.

El coste y la complejidad de administrar varios equipos fomenta la consolidación de varias aplicaciones en servidores más grandes y más escalables. En lugar de ejecutar cada carga de trabajo en un sistema distinto, con acceso completo a los recursos de dicho sistema, puede utilizar el software de administración de recursos para segregar las cargas de trabajo en el sistema. La gestión de recursos permite reducir el costo total de propiedad al ejecutar y controlar varias aplicaciones en un único sistema Oracle Solaris.

Si proporciona servicios de aplicaciones e Internet, puede utilizar la administración de recursos para:

- Alojarse varios servidores Web en un único equipo. Puede controlar el consumo de recursos para cada sitio web y proteger cada sitio de los posibles excesos de otros sitios.
- Evitar que una secuencia de interfaz Common Gateway Interface (CGI) agote los recursos de la CPU.
- Detener una aplicación cuyo comportamiento sea anómalo para que no afecte a la memoria virtual disponible.
- Asegurarse de que las aplicaciones de un cliente no se vean afectadas por las aplicaciones de otro cliente que se ejecutan en el mismo sitio.
- Proporcionar clases o niveles de servicios diferenciados en el mismo equipo.
- Obtener información de cuentas para fines de facturación.

Bases de usuarios diversas y de gran tamaño

Utilice las funciones de administración de recursos de cualquier sistema que tenga una base de usuarios diversa y de gran tamaño, como una institución educativa. Si tiene una combinación de cargas de trabajo, el software se puede configurar para otorgar prioridad a determinados proyectos.

Por ejemplo, en grandes agencias de corredores, los agentes necesitan acceso de forma intermitente y rápida para ejecutar una consulta o realizar un cálculo. En cambio, otros usuarios del sistema tienen cargas de trabajo más coherentes. Si asigna una cantidad proporcionalmente mayor de potencia de procesamiento a los proyectos de los agentes, éstos obtendrán la respuesta que necesitan.

La administración de recursos también resulta ideal para los sistemas e clientes sencillos. Estas plataformas proporcionan consolas sin estado con búferes de trama y dispositivos de entrada, como tarjetas inteligentes. Los cálculos reales se llevan a cabo en un servidor compartido, lo que tiene como resultado un tipo de entorno en el que se comparte el tiempo. Utilice las funciones de administración de recursos para aislar los usuarios del servidor. Un usuario que genere una carga excesiva no monopolizará los recursos de hardware ni afectará a los demás usuarios del sistema.

Configuración del mapa de tareas de gestión de recursos

El siguiente mapa de tareas proporciona una descripción general de alto nivel de los pasos para configurar la gestión de recursos en el sistema.

Tarea	Descripción	Instrucciones
Identificar las cargas de trabajo del sistema y categorizar cada una de ellas por proyecto.	Crea entradas de proyecto en el archivo <code>/etc/project</code> , el mapa NIS o el servicio de directorios LDAP.	“Base de datos project” [23]
Priorizar las cargas de trabajo en el sistema.	Determina qué aplicaciones son críticas. Estas cargas de trabajo pueden requerir acceso preferencial a los recursos.	Consulte los objetivos del servicio corporativo.
Supervisar la actividad del sistema en tiempo real.	Utiliza las herramientas del sistema para ver el consumo de recursos de las cargas de trabajo que se ejecutan en el sistema. Puede evaluar si debe limitar el acceso a un recurso específico o aislar cargas de trabajo concretas de otras cargas de trabajo.	Páginas del comando <code>man cpustat(1M)</code> , <code>iostat(1M)</code> , <code>mpstat(1M)</code> , <code>prstat(1M)</code> , <code>sar(1)</code> y <code>vmstat(1M)</code> .
Realizar modificaciones temporales en las cargas de trabajo que se ejecutan en el sistema.	Para determinar los valores que se pueden modificar, consulte los controles de recursos que están disponibles en el sistema Oracle Solaris. Puede actualizar los valores desde la línea de comandos mientras se ejecuta la tarea o el proceso.	Páginas del comando <code>man “Controles de recursos disponibles” [62]</code> , <code>“Acciones locales y globales en valores de controles de recursos” [68]</code> , <code>“Actualización temporal de los valores de controles de recursos en un sistema en ejecución” [73]</code> , <code>rctladm(1M)</code> y <code>prctl(1)</code> .
Definir controles de recursos y atributos de proyecto para cada entrada de proyecto en la base de datos <code>project</code> o la base de datos del proyecto de servicio de nombres.	Cada entrada de proyecto del archivo <code>/etc/project</code> o la base de datos del proyecto del servicio de nombres puede contener uno o más atributos o controles de recursos. Los controles de recursos limitan las tareas y los procesos vinculados al proyecto. Para cada valor umbral que se coloque en un control de recursos, puede asociar una o más acciones que se deben realizar cuando se alcance dicho valor. Puede configurar los controles de recursos utilizando la interfaz de la línea de comandos.	“Base de datos project” [23] , “Formato de archivo <code>/etc/project</code> local” [25] , “Controles de recursos disponibles” [62] , “Acciones locales y globales en valores de controles de recursos” [68] y Capítulo 8, Acerca del planificador por reparto equitativo

Tarea	Descripción	Instrucciones
Coloque un vínculo superior sobre el consumo de recursos de la memoria física por parte de los grupos de procesos vinculados a un proyecto.	El daemon de aplicación límite (cap) de recursos aplicará el límite de recursos de memoria física definido para el atributo <code>rcap.max-rss</code> del proyecto en el archivo <code>/etc/project</code> .	“Base de datos project” [23] y Capítulo 10, Acerca del control de memoria física mediante el daemon de limitación de recursos
Crear configuraciones de agrupaciones de recursos.	Las agrupaciones de recursos permiten particionar los recursos del sistema, como los procesadores, y mantener las particiones después de rearrancar el sistema. Puede agregar un atributo <code>project.pool</code> a cada entrada en el archivo <code>/etc/project</code> .	“Base de datos project” [23] y Capítulo 12, Acerca de las agrupaciones de recursos
Convertir el planificador de reparto justo (FSS) en el planificador predeterminado del sistema.	Asegura que todos los procesos del usuario de un sistema CPU único o un conjunto de procesadores pertenezcan a la misma clase de planificación.	“Configuración de FSS” [101] y página del comando <code>man dispadm(1M)</code>
Activar la función de contabilidad ampliada para supervisar y registrar el consumo de recursos por tareas o procesos.	Utiliza los datos de contabilidad ampliada para evaluar los controles de recursos actuales y para planificar los requisitos de capacidades para futuras cargas de trabajo. Puede realizarse un seguimiento del uso adicional en todo el sistema. Para obtener estadísticas completas sobre el uso de cargas de trabajo relacionadas que implican más de un sistema, varios equipos pueden compartir el nombre del proyecto.	Cómo activar la contabilidad ampliada para los flujos, los procesos, las tareas y los componentes de red [52] y página del comando <code>man acctadm(1M)</code>
(Opcional) Si necesita realizar ajustes adicionales en la configuración, puede seguir modificando los valores desde la línea de comandos. Puede modificar los valores mientras se ejecuta la tarea o el proceso.	Las modificaciones de las tareas existentes se pueden aplicar temporalmente sin necesidad de reiniciar el proyecto. Ajuste los valores hasta obtener un rendimiento satisfactorio. A continuación, actualice los valores actuales del archivo <code>/etc/project</code> o de la base de datos del proyecto del servicio de nombres.	Páginas del comando <code>man</code> “Actualización temporal de los valores de controles de recursos en un sistema en ejecución” [73] , <code>rctladm(1M)</code> y <code>prctl(1)</code> .
(Opcional) Capturar datos de cuentas extendidas.	Crea registros de contabilidad ampliada para las tareas y los procesos activos. Los archivos que se generan pueden utilizarse para fines de planificación, contracargos y facturación. También hay una interfaz de Lenguaje Práctico de Extracción e Informes (Practical Extraction and Report Language o Perl) para <code>libexacct</code> que permite desarrollar secuencias de extracción e informes personalizados.	Página del comando <code>man wracct(1M)</code> y “Interfaz Perl para libexacct” [48] .

BETA
DRAFT

♦ ♦ ♦ 2 C A P Í T U L O 2

Acerca de los proyectos y las tareas

En este capítulo, se describen las funciones de *proyectos* y *tareas* de la gestión de recursos de Oracle Solaris. Los proyectos y tareas se utilizan para etiquetar cargas de trabajo y separarlas entre sí.

En este capítulo, se describen los siguientes temas:

- “Funciones de proyectos y tareas” [21]
- “Identificadores de proyecto” [22]
- “Identificadores de tareas” [27]
- “Comandos utilizados con proyectos y tareas” [29]

Para utilizar las funciones de proyectos y tareas, consulte el [Capítulo 3, Administración de proyectos y tareas](#).

Funciones de proyectos y tareas

Para optimizar la respuesta de la carga de trabajo, primero debe ser capaz de identificar las cargas de trabajo que se ejecutan en el sistema que está analizando. Esta información puede ser difícil de obtener utilizando un método orientado únicamente a procesos o usuarios. En el sistema Oracle Solaris, pueden utilizarse dos funciones adicionales para separar e identificar las cargas de trabajo: el proyecto y la tarea. El *proyecto* proporciona un identificador administrativo de red para el trabajo relacionado. La *tarea* recopila un grupo de procesos en una entidad con posibilidades de administración que representa un componente de la carga de trabajo.

Los controles especificados en la base de datos del servicio de nombres de `project` se configuran en el proceso, la tarea y el proyecto. Dado que los controles del proceso y la tarea se heredan de las llamadas del sistema `fork` y `settaskid`, todos los procesos y tareas que se crean en el proyecto heredan dichos controles. Para obtener información sobre estas llamadas del sistema, consulte las páginas de comando `man fork(2)` y `settaskid(2)`.

Según su pertenencia a un proyecto o una tarea, los procesos en ejecución se pueden manipular con los comandos estándar de Oracle Solaris. La función de contabilidad ampliada puede informar del uso de procesos y tareas, y etiquetar cada registro con el identificador del proyecto en cuestión. Este proceso permite correlacionar el análisis de carga de trabajo fuera de línea

con la supervisión en línea. El identificador de proyecto puede compartirse en varios equipos mediante la base de datos de servicio de nombres de `project`. De este modo, el consumo de recursos de las cargas de trabajo relacionadas que se ejecutan o abarcan varios equipos se puede analizar en última instancia en todos los equipos.

Identificadores de proyecto

El identificador de proyecto es un identificador administrativo que se utiliza para identificar el trabajo relacionado. El identificador de proyecto se puede considerar una etiqueta de carga de trabajo equivalente a los identificadores de usuario y grupo. Un usuario o grupo puede pertenecer a un proyecto o más. Estos proyectos pueden utilizarse para representar las cargas de trabajo en las que el usuario (o el grupo de usuarios) tiene permiso para participar. Esta pertenencia al grupo puede constituir la base del contracargo que se basa, por ejemplo, en el uso o las asignaciones de recursos iniciales. Aunque debe asignarse un usuario a un proyecto predeterminado, el proceso que inicia el usuario se puede asociar con cualquier proyecto del que es miembro el usuario.

Determinación de un proyecto predeterminado del usuario

Para iniciar sesión en el sistema, un usuario debe tener asignado un proyecto predeterminado. Un usuario es automáticamente miembro del proyecto predeterminado, aunque no se encuentre en la lista de usuarios o grupos del proyecto.

Dado que cada proceso del sistema procesa la pertenencia a un grupo del proyecto, se necesita un algoritmo para asignar un proyecto predeterminado al proceso de conexión u otro proceso inicial. El algoritmo se documenta en la página del comando `man getproject(3C)`. El sistema sigue una serie de pasos para determinar el proyecto predeterminado. Si no se encuentra ningún proyecto predeterminado, se deniega la conexión del usuario o su solicitud para iniciar un proceso.

El sistema sigue estos pasos de forma secuencial para determinar un proyecto predeterminado del usuario:

1. Si el usuario tiene una entrada con un atributo `project` definido en la base de datos de atributos de usuario extendida `/etc/user_attr`, el valor del atributo `project` es el proyecto predeterminado. Consulte la página del comando `man user_attr(4)`.
2. Si hay un proyecto con el nombre `user.user-id` en la base de datos `project`, dicho proyecto es el predeterminado. Para obtener más información, consulte la página del comando `man project(4)`.
3. Si hay un proyecto con el nombre `group.group-name` en la base de datos `project`, donde `group-name` es el nombre del grupo predeterminado para el usuario, como se especifica en

el archivo `passwd`, ese proyecto es el predeterminado. Para obtener información sobre el archivo `passwd`, consulte la página del comando `man passwd(4)`.

4. Si el proyecto especial `default` está presente en la base de datos `project`, dicho proyecto es el predeterminado.

Esta lógica la proporciona la función de biblioteca `getdefaultproj()`. Para obtener más información, consulte la página del comando `man getproject(3PROJECT)`.

Definición de atributos de usuario con los comandos `useradd` y `usermod`

Puede utilizar los siguientes comandos con la opción `-K` y un par *clave=valor* para definir los atributos de usuario en los archivos locales:

<code>useradd</code>	Definir proyecto predeterminado para el usuario
<code>usermod</code>	Modificar información del usuario

Los archivos locales pueden incluir:

- `/etc/group`
- `/etc/passwd`
- `/etc/project`
- `/etc/shadow`
- `/etc/user_attr`

Si se utiliza un servicio de nombres de red como NIS para complementar el archivo local con entradas adicionales, estos comandos no pueden cambiar la información proporcionada por el servicio de nombres de red. Sin embargo, los comandos comprueban lo siguiente en la *base de datos de servicio de nombres* externa:

- Exclusividad del nombre de usuario (o su función)
- Exclusividad del ID de usuario
- Existencia de cualquier nombre de grupo especificado

Para obtener más información, consulte las páginas del comando `man useradd(1M)`, `usermod(1M)` y `user_attr(4)`.

Base de datos `project`

Puede almacenar datos del proyecto en un archivo local, en el sistema de nombre de dominio (DNS), en el mapa de proyecto del servicio de información de la red (NIS) o en un servicio de

directorio de protocolo ligero de acceso a directorios (LDAP). El archivo `/etc/project` o el servicio de nombres lo utilizan durante la conexión todas las solicitudes de administración de cuentas del módulo de autenticación enchufable (PAM) para vincular un usuario a un proyecto predeterminado.

Nota - Las actualizaciones de las entradas de la base de datos del proyecto, tanto si son para el archivo `/etc/project` como para una representación de la base de datos en un servicio de nombres de red, no se aplican a los proyectos activos. Las actualizaciones se aplican a las nuevas tareas que se unen al proyecto cuando se utilizan los comandos `login` o `newtask`. Para obtener más información, consulte las páginas del comando `man login(1)` y `newtask(1)`.

Subsistema PAM

Las operaciones que cambian o definen identidades incluyen el registro en el sistema, la invocación de un comando `rcp` o `rsh`, utilizando `ftp` o `su`. Cuando una operación implica cambiar o definir una identidad, se utiliza un conjunto de módulos configurables para proporcionar autenticación y administración de cuentas, credenciales y sesiones.

Para ver una descripción general de PAM, consulte [Capítulo 1, “Uso de módulos de autenticación conectables”](#) de “[Gestión de Kerberos y otros servicios de autenticación en Oracle Solaris 11.2](#)”.

Configuración de servicios de nombres

La administración de recursos admite las bases de datos `project` del servicio de nombres. La ubicación donde se almacena la base de datos `project` se define en el archivo `/etc/nsswitch.conf`. De forma predeterminada, `files` se muestra primero, pero los orígenes se pueden mostrar en cualquier orden.

```
project: files [nis] [ldap]
```

Si se enumera más de un origen para la información del proyecto, el archivo `nsswitch.conf` dirige la rutina para empezar a buscar la información en el primer origen enumerado y, luego, buscar los orígenes subsiguientes.

Para obtener más información sobre el archivo `/etc/nsswitch.conf`, consulte [Capítulo 2, “Acerca del cambio de servicio de nombres”](#) de “[Trabajo con servicios de nombres y de directorio en Oracle Solaris 11.2: DNS y NIS](#)” and `nsswitch.conf(4)`.

Formato de archivo `/etc/project local`

Si selecciona `files` como su origen de base de datos `project` en el archivo `nsswitch.conf`, el proceso de inicio de sesión busca el archivo `/etc/project` para obtener información del proyecto. Consulte las páginas del comando `man projects(1)` y `project(4)` para obtener información adicional.

El archivo `project` contiene una entrada de una línea con el siguiente formato para cada proyecto reconocido por el sistema:

```
projname:projid:comment:user-list:group-list:attributes
```

Los campos se definen del modo siguiente:

<i>projname</i>	Nombre del proyecto. El nombre debe ser una cadena que esté compuesta por caracteres alfanuméricos, guiones bajos (<code>_</code>), guiones (<code>-</code>) y puntos (<code>.</code>). El punto, que queda reservado para los proyectos que tengan un significado especial para el sistema operativo, sólo se puede utilizar en los nombres de proyectos predeterminados para usuarios. <i>projname</i> no puede contener dos puntos (<code>:</code>) o caracteres de línea nueva.
<i>projid</i>	El ID numérico exclusivo del proyecto (PROJID) en el sistema. El valor máximo del campo <i>projid</i> es <code>UID_MAX</code> (2147483647).
<i>comment</i>	Descripción del proyecto.
<i>user-list</i>	Lista separada por comas de los usuarios que se permiten en el proyecto. En este campo pueden utilizarse comodines. El símbolo de asterisco (*) permite a los usuarios unirse al proyecto. Un signo de exclamación de cierre seguido de un asterisco (!*) excluye del proyecto a todos los usuarios. Un signo de exclamación (!) seguido de un nombre de usuario excluye del proyecto al usuario especificado.
<i>group-list</i>	Lista separada por comas de los grupos de usuarios que se permiten en el proyecto. En este campo pueden utilizarse comodines. El símbolo de asterisco (*) permite a todos los grupos unirse al proyecto. Un signo de exclamación de cierre seguido de un asterisco (!*) excluye del proyecto a todos los grupos. Un signo de exclamación (!) seguido de un nombre de grupo excluye del proyecto al grupo especificado.
<i>attributes</i>	Una lista de pares de nombre-valor separados por puntos y coma, como los controles de recursos (consulte Capítulo 6, Acerca de los controles de recursos). <i>name</i> es una cadena arbitraria que especifica el atributo relacionado con el objeto y <i>value</i> es el valor opcional de dicho atributo.

name[=value]

En el par de nombre-valor, los nombres se restringen a letras, dígitos, subrayados y puntos. Un punto suele utilizarse como separador entre las categorías y subcategorías del control de recursos (rctl). El primer carácter de un nombre de atributo debe ser una letra. El nombre distingue entre mayúsculas y minúsculas.

Los valores pueden estructurarse utilizando comas y paréntesis para establecer el orden de precedencia.

El símbolo de punto y coma se utiliza para separar pares de nombre-valor. No se puede utilizar un punto y coma en una definición de valor. El símbolo de dos puntos se utiliza para separar campos de proyecto. No se puede utilizar el símbolo de dos puntos en una definición de valor.

Nota - Las rutinas que leen este archivo se detienen si detectan una entrada mal creada. Cualquier proyecto que se especifique después de la entrada incorrecta no podrá asignarse.

Este ejemplo muestra el archivo /etc/project predeterminado:

```
system:0:::  
user.root:1:::  
noproject:2:::  
default:3:::  
group.staff:10:::
```

Este ejemplo muestra el archivo /etc/project predeterminado con entradas de proyecto agregadas al final:

```
system:0:::  
user.root:1:::  
noproject:2:::  
default:3:::  
group.staff:10:::  
user.ml:2424:Lyle Personal::  
booksite:4113:Book Auction Project:ml,mp,jtd,kjh::
```

También puede agregar controles de recursos y atributos al archivo /etc/project:

- Para agregar controles de recursos para un proyecto, consulte [“Configuración de controles de recursos” \[76\]](#).
- Para definir un límite de recursos de memoria física para un proyecto mediante el daemon de limitación de recursos que se describe en [rcapd\(1M\)](#), consulte [“Atributo para limitar el uso de memoria física para proyectos” \[106\]](#).
- Para agregar un atributo `project.pool` a la entrada de un proyecto, consulte [“Creación de la configuración” \[174\]](#).

Configuración del proyecto para NIS

Si utiliza NIS, puede especificar en el archivo `/etc/nsswitch.conf` la búsqueda de las asignaciones de proyecto NIS para los proyectos:

```
project: nis files
```

Las asignaciones NIS, tanto `project.byname` como `project.bynumber`, tienen la misma forma que el archivo `/etc/project`:

```
projname:projid:comment:user-list:group-list:attributes
```

Para obtener más información, consulte [Capítulo 5, “Acerca del servicio de información de red” de “Trabajo con servicios de nombres y de directorio en Oracle Solaris 11.2: DNS y NIS”](#).

Configuración del proyecto para LDAP

Si utiliza LDAP, puede especificar en el archivo `/etc/nsswitch.conf` la búsqueda de bases de datos LDAP `project` para los proyectos:

```
project: ldap files
```

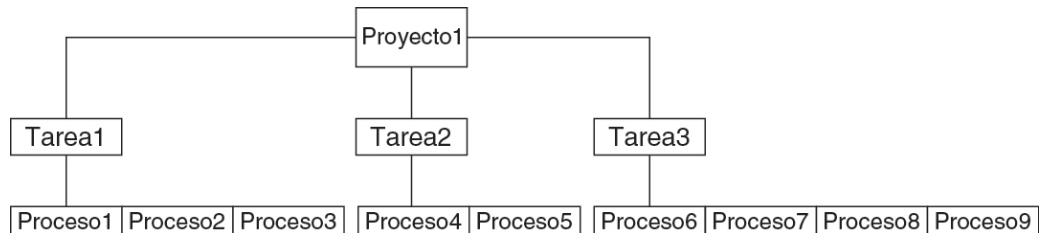
Para obtener más información acerca de LDAP, consulte [Capítulo 1, “Introducción al servicio de nombres LDAP” de “Trabajo con servicios de nombres y de directorio en Oracle Solaris 11.2: LDAP”](#). Para obtener más información sobre el esquema para las entradas de proyecto en una base de datos LDAP, consulte [“Esquemas de Oracle Solaris” de “Trabajo con servicios de nombres y de directorio en Oracle Solaris 11.2: LDAP”](#).

Identificadores de tareas

Cada inicio de sesión correcto en un proyecto crea una nueva *tarea* que contiene el proceso de inicio de sesión. La tarea es un proceso colectivo que representa un conjunto de trabajos en el tiempo. Una tarea también puede visualizarse como *componente de carga de trabajo*. A cada tarea se asigna automáticamente un ID.

Cada proceso es miembro de una tarea, y cada tarea se asocia con un proyecto.

FIGURA 2-1 Árbol de proyectos y tareas



Todas las operaciones de los grupos de procesos, como la entrega de señales, también se admiten para las tareas. Asimismo, puede vincular una tarea a un *conjunto de procesadores* y definir una prioridad de planificación y una clase para una tarea, lo que modifica todos los procesos actuales y subsiguientes de la tarea.

Se crea una tarea siempre que se incorpora un proyecto. Los siguientes comandos, acciones y funciones crean tareas:

- `login`
- `cron`
- `newtask`
- `setproject`
- `su`

Puede crear una tarea finalizada siguiendo uno de estos métodos. Cualquier intento adicional de crear tareas fallará.

- Puede utilizar el comando `newtask` con la opción `-F`.
- Puede definir el atributo `task.final` en un proyecto en la base de datos del servicio de nombres `project`. Todas las tareas creadas en dicho proyecto por `setproject` tienen la etiqueta `TASK_FINAL`.

Para obtener más información, consulte las páginas del comando `man login(1)`, `newtask(1)`, `cron(1M)`, `su(1M)` y `setproject(3PROJECT)`.

La función de contabilidad ampliada puede proporcionar datos de cuentas para los procesos. Los datos se agregan en el nivel de la tarea.

Comandos utilizados con proyectos y tareas

Los comandos que se incluyen en la tabla siguiente proporcionan la interfaz administrativa principal para las funciones de proyectos y tareas.

Referencia de página del comando man	Descripción
projects(1)	Muestra la pertenencia de los usuarios a un grupo del proyecto. Enumera los proyectos de la base de datos <code>project</code> . Imprime información sobre determinados proyectos. Si no se proporciona ningún nombre de proyecto, se muestra la información para todos los proyectos. Utilice el comando <code>projects</code> con la opción <code>-l</code> para imprimir un listado detallado.
newtask(1)	Ejecuta el comando especificado o el shell predeterminado del usuario, y coloca el comando de ejecución en una nueva tarea que pertenece al proyecto especificado. <code>newtask</code> también puede utilizarse para cambiar la vinculación de la tarea y del proyecto para un proceso en ejecución. Utilícela con la opción <code>-F</code> para crear una tarea finalizada.
projadd(1M)	Agrega una entrada de proyecto nueva al archivo <code>/etc/project</code>. El comando <code>projadd</code> crea una entrada de proyecto sólo en el sistema local. <code>projadd</code> no puede cambiar la información que proporciona el servicio de nombres de la red. Se puede utilizar para editar archivos de proyecto que no sean el predeterminado, <code>/etc/project</code>. Proporciona comprobación de sintaxis para el archivo <code>project</code>. Valida y edita los atributos del proyecto. Admite valores a escala.
projmod(1M)	Modifica la información para un proyecto en el sistema local. <code>projmod</code> no puede cambiar la información que proporciona el servicio de nombres de la red. Sin embargo, el comando sí que verifica la exclusividad del nombre y el ID del proyecto con el servicio de nombres externo. Se puede utilizar para editar archivos de proyecto que no sean el predeterminado, <code>/etc/project</code>. Proporciona comprobación de sintaxis para el archivo <code>project</code>. Valida y edita los atributos del proyecto. Se puede utilizar para agregar un atributo nuevo, agregar valores a un atributo o eliminar un atributo. Admite valores a escala. Se puede utilizar con la opción <code>-A</code> para aplicar los valores de control de recursos encontrados en la base de datos de proyectos en el proyecto activo. Los valores existentes que no coincidan con los valores definidos en el archivo <code>project</code> se eliminan.
projdel(1M)	Suprime un proyecto del sistema local. <code>projdel</code> no puede cambiar la información que proporciona el servicio de nombres de la red.
useradd(1M)	Agrega definiciones de proyecto predeterminadas a los archivos locales. Utilícela con la opción <code>-K key=value</code> para agregar o sustituir atributos de usuario.
userdel(1M)	Suprime una cuenta de usuario del archivo local.
usermod(1M)	Modifica la información de inicio de sesión de un usuario en el sistema. Utilícela con la opción <code>-K key=value</code> para agregar o sustituir atributos de usuario.

BETA
DRAFT

3

♦ ♦ ♦ C A P Í T U L O 3

Administración de proyectos y tareas

En este capítulo, se describe cómo utilizar las funciones de proyectos y tareas de la gestión de recursos de Oracle Solaris.

Se tratan los temas siguientes.

- [“Ejemplos y opciones de comandos” \[32\]](#)
- [“Administración de proyectos” \[35\]](#)

Para obtener una descripción general de las funciones de proyectos y tareas, consulte [Capítulo 2, Acerca de los proyectos y las tareas](#).

Nota - Si utiliza estas funciones en un sistema Oracle Solaris con zonas instaladas, únicamente los procesos de la misma zona serán visibles mediante interfaces de llamada del sistema que obtienen ID de proceso cuando estos comandos se ejecutan en una zona no global.

Administración del mapa de tareas de proyectos y tareas

Tarea	Descripción	Instrucciones
Ver ejemplos de comandos y opciones que se utilizan con proyectos y tareas.	Muestra ID de proyectos y tareas, varias estadísticas para los procesos y los proyectos que se ejecutan en el sistema.	“Ejemplos y opciones de comandos” [32]
Definir un proyecto.	Agrega una entrada de proyecto al archivo <code>/etc/project</code> y modifica los valores para dicha entrada.	Cómo definir un proyecto y ver el proyecto actual [35]
Suprimir un proyecto.	Elimina una entrada de proyecto del archivo <code>/etc/project</code> .	Cómo suprimir un proyecto del archivo <code>/etc/project</code> [37]
Validar el archivo <code>project</code> o la base de datos del proyecto.	Comprueba la sintaxis del archivo <code>/etc/project</code> o verifica la exclusividad del nombre y el ID del proyecto con el servicio de nombres externo.	“Cómo validar el contenido del archivo <code>/etc/project</code>” [38]
Obtener información sobre la pertenencia del proyecto a un grupo.	Muestra la pertenencia del proyecto actual a un grupo del proceso que ejecuta el comando.	“Cómo obtener información sobre la pertenencia de un proyecto a un grupo” [39]

Tarea	Descripción	Instrucciones
Crear una tarea.	Crea una tarea en un proyecto específico utilizando el comando <code>newtask</code> .	Cómo crear una tarea [39]
Asociar un proceso en ejecución con un proyecto y una tarea diferentes.	Asocia un número de proceso con un ID de tarea nuevo en un proyecto específico.	Cómo mover un proceso en ejecución a una nueva tarea [39]
Agregar y trabajar con atributos de proyecto.	Utiliza los comandos de administración de la base de datos del proyecto para agregar, editar, validar y eliminar atributos del proyecto.	“Edición y validación de atributos de proyecto” [40]

Ejemplos y opciones de comandos

En esta sección se incluyen ejemplos de comandos y opciones que se utilizan con proyectos y tareas.

Opciones de comandos utilizadas con proyectos y tareas

Comando `ps`

Utilice el comando `ps` con la opción `-o` para ver los ID de proyectos y tareas. Por ejemplo, para ver el ID de proyecto, escriba:

```
# ps -o user,pid,uid,projid
USER PID  UID  PROJID
jtd  89430 124  4113
```

Comando `id`

Utilice el comando `id` con la opción `-p` para imprimir el ID de proyecto actual además de los ID de grupo y usuario. Si se proporciona el operando `user`, se imprime el proyecto asociado con el inicio de sesión normal del usuario:

```
# id -p
uid=124(jtd) gid=10(staff) projid=4113(booksite)
```


Comandos pgrep y pkill

Para hacer coincidir sólo los procesos con un ID de proyecto en una lista específica, utilice los comandos `pgrep` y `pkill` con la opción `-J`:

```
# pgrep -J projidlist
# pkill -J projidlist
```

Para hacer coincidir sólo los procesos con un ID de tarea en una lista específica, utilice los comandos `pgrep` y `pkill` con la opción `-T`:

```
# pgrep -T taskidlist
# pkill -T taskidlist
```

Comando prstat

Para mostrar diversas estadísticas de procesos y proyectos que se están ejecutando actualmente en el sistema, utilice el comando `prstat` con la opción `-J`:

```
% prstat -J
  PID USERNAME  SIZE  RSS STATE PRI NICE   TIME   CPU PROCESS/NLWP
12905 root      4472K 3640K cpu0  59   0 0:00:01 0.4% prstat/1
 829 root         43M   33M sleep 59   0 0:36:23 0.1% Xorg/1
 890 gdm          88M   26M sleep 59   0 0:22:22 0.0% gdm-simple-gree/1
 686 root      3584K 2756K sleep 59   0 0:00:34 0.0% automountd/4
   5 root         0K    0K sleep 99  -20 0:02:43 0.0% zpool-rpool/138
9869 root         44M   17M sleep 59   0 0:02:06 0.0% pooldd/9
 804 root      7104K 5968K sleep 59   0 0:01:28 0.0% intrd/1
 445 root      7204K 4680K sleep 59   0 0:00:38 0.0% nsd/33
 881 gdm      7140K 5912K sleep 59   0 0:00:06 0.0% gconfd-2/1
 164 root      2572K 1648K sleep 59   0 0:00:00 0.0% pfexecd/3
 886 gdm      7092K 4920K sleep 59   0 0:00:00 0.0% bonobo-activati/2
  45 netcfg     2252K 1308K sleep 59   0 0:00:00 0.0% netcfgd/2
 142 daemon     7736K 5224K sleep 59   0 0:00:00 0.0% kcfcd/3
  43 root      3036K 2020K sleep 59   0 0:00:00 0.0% dlmgrtd/5
 405 root      6824K 5400K sleep 59   0 0:00:18 0.0% halld/5
PROJID  NPROC  SWAP  RSS MEMORY   TIME   CPU PROJECT
   1      4 4728K   19M   0.9% 0:00:01 0.4% user.root
   0     111 278M  344M   17% 1:15:02 0.1% system
  10      2 1884K  9132K   0.4% 0:00:00 0.0% group.staff
   3      3 1668K  6680K   0.3% 0:00:00 0.0% default
```

Total: 120 processes, 733 lwps, load averages: 0.01, 0.00, 0.00

Para mostrar varias estadísticas para procesos y tareas que se ejecutan en el sistema, utilice el comando `prstat` con la opción `-T`:

```
% prstat -T
  PID USERNAME  SIZE  RSS STATE PRI NICE   TIME   CPU PROCESS/NLWP
```

```
12907 root      4488K 3588K cpu0    59    0    0:00:00 0.3% prstat/1
829  root      43M   33M sleep    59    0    0:36:24 0.1% Xorg/1
890  gdm       88M   26M sleep    59    0    0:22:22 0.0% gdm-simple-gree/1
9869 root      44M   17M sleep    59    0    0:02:06 0.0% poold/9
5   root      0K    0K sleep    99   -20    0:02:43 0.0% zpool-rpool/138
445  root     7204K 4680K sleep    59    0    0:00:38 0.0% nscd/33
881  gdm     7140K 5912K sleep    59    0    0:00:06 0.0% gconfd-2/1
164  root     2572K 1648K sleep    59    0    0:00:00 0.0% pfexecd/3
886  gdm     7092K 4920K sleep    59    0    0:00:00 0.0% bonobo-activati/2
45  netcfg    2252K 1308K sleep    59    0    0:00:00 0.0% netcfgd/2
142 daemon    7736K 5224K sleep    59    0    0:00:00 0.0% kcfcd/3
43   root     3036K 2020K sleep    59    0    0:00:00 0.0% dlmgmt/5
405  root     6824K 5400K sleep    59    0    0:00:18 0.0% hald/5
311  root     3488K 2512K sleep    59    0    0:00:00 0.0% picld/4
409  root     4356K 2768K sleep    59    0    0:00:00 0.0% hald-addon-cpuf/1

TASKID  NPROC  SWAP  RSS  MEMORY  TIME  CPU  PROJECT
1401    2  2540K 8120K  0.4%   0:00:00 0.3% user.root
94      15   84M  162M  7.9%   0:59:37 0.1% system
561     1   37M   24M  1.2%   0:02:06 0.0% system
0       2    0K    0K  0.0%   0:02:47 0.0% system
46      1 4224K 5524K  0.3%   0:00:38 0.0% system

Total: 120 processes, 733 lwps, load averages: 0.01, 0.00, 0.00
```

Nota - Las opciones -J y -T no se pueden utilizar de forma simultánea.

Uso de cron y su con proyectos y tareas

Comando cron

El comando cron emite settaskid para asegurarse de que cada tarea cron, at y batch se ejecute en una tarea distinta, con el proyecto predeterminado adecuado para el usuario que realiza la ejecución. Los comandos at y batch también capturan el ID de proyecto actual, lo que garantiza que el ID de proyecto se restablezca al ejecutar una tarea at.

Comando su

El comando su se une al proyecto predeterminado del usuario de destino al crear una tarea nueva, como parte de la simulación de un inicio de sesión.

Para alternar el proyecto predeterminado del usuario con el comando su, escriba:

```
# su - user
```

Administración de proyectos

▼ Cómo definir un proyecto y ver el proyecto actual

En este ejemplo se describe cómo utilizar el comando `projadd` para agregar una entrada de proyecto y el comando `projmod` para modificar dicha entrada.

1. **Conviértase en root o asuma un rol similar.**
2. **Visualice el archivo `/etc/project` predeterminado en el sistema utilizando `projects -l`.**

```
# projects -l
system
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
user.root
    projid : 1
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
noproject
    projid : 2
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
default
    projid : 3
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
group.staff
    projid : 10
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
```

3. **Agregue un proyecto con el nombre *booksite*. Asigne el proyecto a un usuario con el nombre *mark* con el número de ID de proyecto *4113*.**

```
# projadd -U mark -p 4113 booksite
```

4. Vuelva a visualizar el archivo /etc/project.

```
# projects -l
system
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
user.root
    projid : 1
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
noproject
    projid : 2
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
default
    projid : 3
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
group.staff
    projid : 10
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
booksite
    projid : 4113
    comment: ""
    users  : mark
    groups : (none)
    attribs:
```

5. Agregue un comentario que describa el proyecto en el campo de comentarios.

```
# projmod -c 'Book Auction Project' booksite
```

6. Visualice los cambios en el archivo /etc/project.

```
# projects -l
system
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
user.root
```

```
projid : 1
comment: ""
users  : (none)
groups : (none)
attribs:
noproject
projid : 2
comment: ""
users  : (none)
groups : (none)
attribs:
default
projid : 3
comment: ""
users  : (none)
groups : (none)
attribs:
group.staff
projid : 10
comment: ""
users  : (none)
groups : (none)
attribs:
booksite
projid : 4113
comment: "Book Auction Project"
users  : mark
groups : (none)
attribs:
```

Véase también Para vincular proyectos, tareas y procesos con una agrupación, consulte [“Configuración de atributos de agrupaciones y vinculación a una agrupación” \[167\]](#).

▼ Cómo suprimir un proyecto del archivo `/etc/project`

En este ejemplo se muestra cómo utilizar el comando `projdel` para suprimir un proyecto.

1. **Conviértase en root o asuma un rol similar.**
2. **Elimine el proyecto `booksite` utilizando el comando `projdel`.**

```
# projdel booksite
```

3. **Visualice el archivo `/etc/project`.**

```
# projects -l
system
```

```
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
user.root
    projid : 1
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
noproject
    projid : 2
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
default
    projid : 3
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
group.staff
    projid : 10
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
```

4. **Inicie sesión como usuario *mark* y escriba `projects` para ver los proyectos asignados al usuario.**

```
# su - mark
# projects
default
```

Cómo validar el contenido del archivo /etc/project

Si no se facilitan opciones de edición, el comando `projmod` valida el contenido del archivo `project`.

Para validar una asignación NIS, escriba lo siguiente:

```
# ypcat project | projmod -f -
```

Para comprobar la sintaxis del archivo `/etc/project`, escriba:

```
# projmod -n
```

Cómo obtener información sobre la pertenencia de un proyecto a un grupo

Utilice el comando `id` con la etiqueta `-p` para mostrar la pertenencia del proyecto actual a un grupo del proceso que ejecuta el comando.

```
$ id -p
uid=100(mark) gid=1(other) projid=3(default)
```

▼ Cómo crear una tarea

1. Inicie sesión como miembro del proyecto de destino, *booksite*, en este ejemplo.
2. Cree una nueva tarea en el proyecto *booksite* utilizando el comando `newtask` con la opción `-v` (detallado) para obtener el ID de tarea del sistema.

```
machine% newtask -v -p booksite
16
```

La ejecución de `newtask` crea una tarea en el proyecto especificado, y coloca el shell predeterminado del usuario en esta tarea.

3. Visualice la pertenencia del proyecto actual a un grupo del proceso que ejecuta el comando.

```
machine% id -p
uid=100(mark) gid=1(other) projid=4113(booksite)
```

El proceso ahora es miembro del nuevo proyecto.

▼ Cómo mover un proceso en ejecución a una nueva tarea

Este ejemplo muestra cómo asociar un proceso en ejecución con una tarea diferente y un proyecto nuevo. Para realizar esta acción, debe ser el usuario `root`, tener el perfil de derechos necesarios o ser el propietario del proceso y ser un miembro del proyecto nuevo.

1. Conviértase en `root` o asuma un rol similar.

Nota - Si es el propietario del proceso o un miembro del proyecto nuevo, puede omitir este paso.

2. **Obtenga el ID de proceso del proceso *book_catalog*.**

```
# pgrep book_catalog
8100
```

3. **Asocie el proceso *8100* con un nuevo ID de tarea en el proyecto *booksite*.**

```
# newtask -v -p booksite -c 8100
17
```

La opción -c especifica que newtask opera en el proceso con nombre existente.

4. **Confirme la tarea para procesar la asignación de ID.**

```
# pgrep -T 17
8100
```

Edición y validación de atributos de proyecto

Puede utilizar los comandos de administración de bases de datos de proyecto *projadd* y *projmod* para editar los atributos del proyecto.

La opción -K especifica una lista de sustitución de los atributos. Los atributos se delimitan con puntos y coma (;). Si se utiliza la opción -K con la opción -a, se agrega el atributo o el valor de atributo. Si se utiliza la opción -K con la opción -r, se elimina el atributo o el valor de atributo. Si se utiliza la opción -K con la opción -s, se sustituye el atributo o el valor de atributo.

▼ **Cómo agregar atributos y valores de atributos a los proyectos**

Utilice el comando *projmod* con las opciones -a y -K para añadir valores a un atributo de proyecto. Si el atributo no existe, se crea.

1. **Conviértase en root o asuma un rol similar.**
2. **Agregue el atributo de control de recursos *task.max-lwps* sin valores en el proyecto *myproject*. Una tarea que se introduce en el proyecto sólo tiene el valor de sistema para el atributo.**

```
# projmod -a -K task.max-lwps myproject
```

3. **A continuación, puede agregar un valor para *task.max-lwps* en el proyecto *myproject*. El valor se compone de un nivel de privilegio, un valor umbral y una acción asociada con el umbral.**


```
# projmod -a -K "task.max-lwps=(priv,100,deny)" myproject
```

4. Dado que los controles de recursos pueden tener varios valores, puede agregar otro valor a la lista utilizando las mismas opciones.

```
# projmod -a -K "task.max-lwps=(priv,1000,signal=KILL)" myproject
```

Los valores múltiples se separan con comas. La entrada `task.max-lwps` ahora es:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

▼ Cómo eliminar valores de atributo de los proyectos

Este procedimiento utiliza los valores:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

1. Conviértase en root o asuma un rol similar.
2. Para eliminar un valor de atributo del control de recursos `task.max-lwps` del proyecto *myproject*, utilice el comando `projmod` con las opciones `-r` y `-K`.

```
# projmod -r -K "task.max-lwps=(priv,100,deny)" myproject
```

Si `task.max-lwps` tiene múltiples valores, como:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

se eliminará el primer valor coincidente. El resultado sería:

```
task.max-lwps=(priv,1000,signal=KILL)
```

▼ Cómo eliminar un atributo de control de recursos de un proyecto

Para eliminar el control de recursos `task.max-lwps` del proyecto *myproject*, utilice el comando `projmod` con las opciones `-r` y `-K`.

1. Conviértase en root o asuma un rol similar.
2. Elimine el atributo `task.max-lwps` y todos sus valores del proyecto *myproject*:

```
# projmod -r -K task.max-lwps myproject
```

▼ Cómo sustituir atributos y valores de atributos para los proyectos

Para sustituir un valor diferente para el atributo `task.max-lwps` en el proyecto *myproject*, utilice el comando `projmod` con las opciones `-s` y `-K`. Si el atributo no existe, se crea.

1. **Conviértase en root o asuma un rol similar.**
2. **Sustituya los valores `task.max-lwps` actuales con los nuevos valores:**

```
# projmod -s -K "task.max-lwps=(priv,100,none),(priv,120,deny)" myproject
```

El resultado sería:

```
task.max-lwps=(priv,100,none),(priv,120,deny)
```

▼ Cómo eliminar los valores existentes para un atributo de control de recursos

1. **Conviértase en root o asuma un rol similar.**
2. **Para eliminar los valores actuales para `task.max-lwps` del proyecto *myproject*, escriba:**

```
# projmod -s -K task.max-lwps myproject
```

4

♦ ♦ ♦ C A P Í T U L O 4

Acerca de la contabilidad ampliada

Al utilizar las funciones de proyectos y tareas que se describen en [Capítulo 2, Acerca de los proyectos y las tareas](#) para etiquetar y separar cargas de trabajo, puede supervisar el consumo de recursos por cada carga de trabajo. Puede utilizar el subsistema de *contabilidad ampliada* para capturar un conjunto detallado de estadísticas de consumo de recursos en los proyectos y tareas.

En este capítulo se cubren los temas siguientes.

- “Introducción a las cuentas extendidas” [43]
- “Funcionamiento de la contabilidad ampliada” [44]
- “Configuración de contabilidad ampliada” [46]
- “Comandos utilizados con contabilidad ampliada” [47]
- “Interfaz Perl para libexacct” [48]

Para comenzar a utilizar la contabilidad ampliada, pase a [Cómo activar la contabilidad ampliada para los flujos, los procesos, las tareas y los componentes de red](#) [52].

Introducción a las cuentas extendidas

El subsistema de contabilidad ampliada etiqueta los registros de uso con el proyecto para el que se ha realizado el trabajo. También puede utilizar la contabilidad ampliada, junto con el módulo de contabilidad de flujos de IPQoS (Internet Protocol Quality of Service, calidad de servicio de protocolo de Internet), a fin de capturar información de flujo de red en un sistema.

Antes de aplicar los mecanismos de administración de recursos, debe caracterizar las demandas de consumo de recursos de las distintas cargas de trabajo en un sistema. La función de contabilidad ampliada del sistema operativo Oracle Solaris proporciona un método flexible para registrar el consumo de recursos de red y del sistema para lo siguiente:

- Tareas.
- Procesos.
- Los selectores proporcionados por el módulo `flowacct` de IPQoS. Para obtener más información, consulte `ipqos(7IPP)`.

- Gestión de red. Consulte [dladm\(1M\)](#) y [flowadm\(1M\)](#).

A diferencia de otras herramientas de supervisión en línea, que permiten medir el uso del sistema en tiempo real, la contabilidad ampliada permite examinar el uso histórico. Puede realizar evaluaciones de los requisitos de capacidades para las futuras cargas de trabajo.

Gracias a los datos de contabilidad ampliada, puede desarrollar o adquirir software para el contracargo de recursos, la supervisión de la carga de trabajo o la planificación de las capacidades.

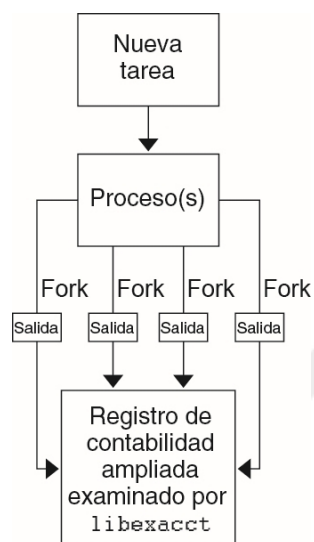
Funcionamiento de la contabilidad ampliada

La función de contabilidad ampliada en el sistema operativo Oracle Solaris utiliza un formato de archivo ampliable con versiones que contiene datos contables. Se puede acceder a los archivos que utilizan este formato de datos o éstos se pueden crear utilizando la API que se proporciona con la biblioteca incluida, `libexacct` (consulte [libexacct\(3LIB\)](#)). Estos archivos pueden analizarse en cualquier plataforma con la función de contabilidad ampliada activada, y sus datos se pueden utilizar para el contracargo y la planificación de capacidades.

Si la función de contabilidad ampliada está activa, se recopilan estadísticas que pueden examinarse con la API `libexacct`. `libexacct` permite examinar los archivos `exacct` hacia delante o hacia atrás. La API admite archivos de terceros que genera `libexacct`, así como los archivos que crea el núcleo. Hay una interfaz de Lenguaje Práctico de Extracción e Informes (Practical Extraction and Report Language o Perl) para `libexacct` que permite desarrollar secuencias de comandos de extracción e informes personalizados. Consulte [“Interfaz Perl para libexacct” \[48\]](#).

Por ejemplo, con la contabilidad ampliada activada, la tarea supervisa el uso de los recursos adicionales de sus procesos miembro. Al finalizar la tarea se guarda un registro de cuentas de las tareas. También pueden guardarse registros provisionales sobre los procesos y tareas en ejecución. Para obtener más información sobre las tareas, consulte [Capítulo 2, Acerca de los proyectos y las tareas](#).

FIGURA 4-1 Supervisión de tareas con la contabilidad ampliada activada



Formato ampliable

El formato de contabilidad ampliada es significativamente más ampliable que el formato del software contable del sistema heredado. La contabilidad ampliada permite agregar y eliminar métrica de cuentas en las distintas versiones del sistema, e incluso durante el funcionamiento del sistema.

Nota - La contabilidad ampliada y el software de contabilidad del sistema heredado pueden estar activos en el sistema de forma simultánea.

Registros y formato exacct

Las rutinas que permiten crear registros exacct tienen dos finalidades.

- Permitir la creación de archivos exacct de terceros.
- Para activar la creación de registros de etiquetas para integrarlos en el archivo de cuentas núcleo utilizando la llamada del sistema putacct (consulte [getacct\(2\)](#)).

Nota - La llamada del sistema putacct también está disponible en la interfaz de Perl.

El formato permite capturar diferentes formas de registros de cuentas sin necesidad de que cada cambio sea un cambio de versión explícito. Las aplicaciones potentes que consumen datos de cuentas deben omitir los registros que no comprenden.

La biblioteca libexacct convierte y produce archivos en el formato exacct. Esta biblioteca es la *única* interfaz compatible con los archivos de formato exacct.

Nota - Las llamadas de sistema getacct, putacct y wracct no se aplican a los flujos. El núcleo crea registros de flujo y los guarda en el archivo cuando se configuran las cuentas de flujo IPQoS.

Uso de contabilidad ampliada en un sistema Oracle Solaris con Zones instalado

El subsistema de contabilidad ampliada recopila y registra la información para todo el sistema (incluidas las zonas no globales) cuando se ejecuta en la zona global. El administrador global o un usuario que ha recibido autorizaciones adecuadas mediante la utilidad zonecfg también pueden determinar el consumo de recursos por zonas. Consulte [Capítulo 2, “Descripción general de la configuración de zonas no globales”](#) de “Introducción a Zonas de Oracle Solaris ” para obtener más información.

Configuración de contabilidad ampliada

El directorio /var/adm/exacct es la ubicación estándar para colocar datos de contabilidad ampliada. Puede utilizar el comando acctadm para especificar una ubicación distinta para los archivos de datos y cuentas de procesos y tareas. Para obtener más información, consulte [acctadm\(1M\)](#).

Inicio y activación persistente de contabilidad ampliada

El comando acctadm descrito en [acctadm\(1M\)](#) inicia la contabilidad ampliada por medio del servicio de utilidad de gestión de servicios (SMF) de Oracle Solaris descrito en [smf\(5\)](#).

La configuración de contabilidad ampliada se almacena en el repositorio de SMF. La configuración es restaurada en el inicio mediante una instancia de servicio (una para cada tipo de contabilidad). Cada uno de los tipos de contabilidad ampliada está representado por una instancia separada del servicio SMF:

<code>svc:/system/ extended- accounting:flow</code>	Contabilidad de flujos
<code>svc:/system/ extended- accounting:process</code>	Contabilidad del proceso
<code>svc:/system/ extended- accounting:task</code>	Contabilidad de tareas
<code>svc:/system/ extended- accounting:net</code>	Contabilidad de redes

La activación de contabilidad ampliada mediante [acctadm\(1M\)](#) hace que la instancia de servicio correspondiente se active si no está activada, de modo que la configuración de contabilidad ampliada se restaurará en el siguiente inicio. Del mismo modo, si la configuración provoca la desactivación de la contabilidad para un servicio, la instancia de servicio se desactivará. Las instancias se activan o desactivan mediante el comando `acctadm` según sea necesario.

Para activar de forma permanente active la contabilidad ampliada para un recurso, ejecute:

```
# acctadm -e resource_list
```

lista_recursos es una lista de recursos o grupos de recursos separada por comas.

Registros

El comando `acctadm` agrega nuevos registros a un archivo existente en `/var/adm/exacct`.

Comandos utilizados con contabilidad ampliada

Referencia de comando	Descripción
acctadm(1M)	Modifica diferentes atributos de la función de contabilidad ampliada, detiene e inicia la contabilidad ampliada y se utiliza para seleccionar atributos de contabilidad para supervisar procesos, tareas, flujos y redes.

Referencia de comando	Descripción
wracct(1M)	Guarda registros de contabilidad ampliada para las tareas y los procesos activos.
lastcomm(1)	Muestra los comandos invocados previamente. lastcomm puede consumir datos de procesos y cuentas estándar o datos de procesos de contabilidad ampliada.

Para obtener información sobre los comandos que están asociados con las tareas y los proyectos, consulte “Ejemplos y opciones de comandos” [32]. Para obtener información sobre la contabilidad de flujos de IPQoS, consulte la página del comando man [ipqosconf\(1M\)](#).

Interfaz Perl para libexacct

La interfaz Perl permite crear secuencias de comandos Perl que lean los archivos de cuentas producidos por la estructura exacct. También puede crear secuencias de comandos Perl que escriban archivos exacct.

La interfaz es funcionalmente equivalente a la API C subyacente. Cuando sea posible, los datos obtenidos de la API C subyacente se presentan como tipos de datos Perl. Esta interfaz permite acceder fácilmente a los datos, y elimina la necesidad de realizar operaciones de empaquetado y desempaqueado de memoria intermedia. Asimismo, todas las funciones de administración de la memoria las lleva a cabo la biblioteca Perl.

Las diferentes funciones relacionadas con proyectos, tareas y exacct se separan en grupos. Cada grupo de funciones se encuentra en un módulo Perl distinto. Cada módulo empieza por el prefijo de paquete Perl `Sun::Solaris::` estándar de Oracle Solaris. Todas las clases proporcionadas por la biblioteca Perl exacct se encuentran en el módulo `Sun::Solaris::Exacct`.

La biblioteca [libexacct\(3LIB\)](#) subyacente proporciona operaciones en los archivos de formato exacct, las etiquetas de catálogo y los objetos exacct. Los objetos exacct se subdividen en dos tipos:

- Elementos, que son valores de datos únicos (escalares)
- Grupos, que son listas de elementos

La tabla siguiente resume cada uno de los módulos.

Módulo (no debe contener espacios)	Descripción	Para obtener más información
<code>Sun::Solaris::Project</code>	Este módulo proporciona funciones para acceder a las funciones de manipulación del proyecto getprojid(2) , endproject(3PROJECT) , fgetproject(3PROJECT) , getdefaultproj(3PROJECT) ,	Project(3PERL)

Módulo (no debe contener espacios)	Descripción	Para obtener más información
	getprojbyid(3PROJECT) , getprojbyname(3PROJECT) , getprojent(3PROJECT) , getprojidbyname(3PROJECT) , inproj(3PROJECT) , project_ walk(3PROJECT) , setproject(3PROJECT) y setprojent(3PROJECT) .	
Sun::Solaris::Task	Este módulo proporciona funciones para acceder a las funciones de manipulación de tareas gettaskid(2) y settaskid(2) .	Task(3PERL)
Sun::Solaris::Exacct	Este módulo es el módulo exacct de nivel superior. Este módulo proporciona funciones para acceder a las llamadas del sistema relacionadas con exacct getacct(2) , putacct(2) y wracct(2) . Este módulo también proporciona funciones para acceder a la función de biblioteca libexacct(3LIB) ea_error(3EXACCT) . En este módulo también se proporcionan las constantes para todas las macros exacct EO_*, EW_*, EXR_*, P_* y TASK_*.	Exacct(3PERL)
Sun::Solaris::Exacct::Catalog	Este módulo proporciona los métodos orientados a objetos para acceder a los campos de bits en una etiqueta de catálogo exacct. Este módulo también proporciona acceso a las constantes de las macros EXC_*, EXD_* y EXD_*.	Exacct::Catalog(3PERL)
Sun::Solaris::Exacct::File	Este módulo proporciona métodos orientados a objetos para acceder a las funciones de archivos de cuentas libexacct ea_open(3EXACCT) , ea_close(3EXACCT) , ea_get_creator(3EXACCT) , ea_get_hostname(3EXACCT) , ea_next_object(3EXACCT) , ea_previous_object(3EXACCT) y ea_write_object(3EXACCT) .	Exacct::File(3PERL)
Sun::Solaris::Exacct::Object	Este módulo proporciona métodos orientados a objetos para acceder a un objeto de archivo de cuentas exacct individual. Un objeto exacct se representa como referencia opaca designada en la subclase Sun::Solaris::Exacct::Object apropiada. Este módulo se subdivide aún más en los tipos de objetos Elemento y Grupo. En este nivel, hay dos métodos para acceder a las funciones ea_match_object_catalog(3EXACCT) y ea_attach_to_object(3EXACCT) .	Exacct::Object(3PERL)
Sun::Solaris::Exacct::Object::Item	Este módulo proporciona métodos orientados a objetos para acceder a un elemento de archivo de cuentas exacct individual. Los objetos de este tipo heredan de Sun::Solaris::Exacct::Object.	Exacct::Object::Item(3PERL)

Módulo (no debe contener espacios)	Descripción	Para obtener más información
Sun::Solaris::Exacct::Object::Group	Este módulo proporciona métodos orientados a objetos para acceder a un grupo de archivos de cuentas exacct individual. Los objetos de este tipo heredan de Sun::Solaris::Exacct::Object. Estos objetos proporcionan acceso a la función ea_attach_to_group(3EXACCT) . Los elementos que contiene el grupo se presentan como matriz Perl.	Exacct::Object::Group(3PERL)
Sun::Solaris::Kstat	Este módulo proporciona una interfaz hash vinculada a Perl para la función kstat. Puede encontrar un ejemplo de uso de este módulo en /bin/kstat, que se escribe en Perl.	Kstat(3PERL)

Para ver ejemplos sobre cómo utilizar los módulos descritos en la tabla anterior, consulte [“Uso de la interfaz de Perl para libexacct” \[55\]](#).

♦ ♦ ♦ CAPÍTULO 5

Administración de tareas de contabilidad ampliada

Este capítulo describe cómo administrar el subsistema de contabilidad ampliada.

Para ver una descripción general del subsistema de contabilidad ampliada, consulte [Capítulo 4, Acerca de la contabilidad ampliada](#).

Administración del mapa de tareas de la función de contabilidad ampliada

Tarea	Descripción	Instrucciones
Activar la función de contabilidad ampliada.	Utiliza la contabilidad ampliada para supervisar el consumo de recursos de cada proyecto que se ejecuta en el sistema. Puede utilizar el subsistema de <i>contabilidad ampliada</i> para capturar los datos históricos para los procesos, tareas y flujos.	Cómo activar la contabilidad ampliada para los flujos, los procesos, las tareas y los componentes de red [52]
Mostrar el estado de la contabilidad ampliada.	Determina el estado de la función de contabilidad ampliada.	“Cómo mostrar el estado de la contabilidad ampliada” [53]
Ver los recursos de cuentas disponibles.	Visualiza los recursos de cuentas disponibles en el sistema.	“Cómo ver los recursos de cuentas disponibles” [53]
Desactivar las instancias de contabilidad de flujos, procesos, tareas y red.	Desactiva la función de contabilidad ampliada.	Cómo desactivar la contabilidad de procesos, tareas, flujos y gestión de redes [54]
Utilizar la interfaz de Perl para la función de contabilidad ampliada.	Utiliza la interfaz de Perl para desarrollar secuencias de comandos de informes y extracción personalizadas.	“Uso de la interfaz de Perl para libexacct” [55]

Uso de funciones de cuentas extendidas

Los usuarios pueden gestionar la contabilidad ampliada (iniciar contabilidad, detener contabilidad y cambiar parámetros de configuración de contabilidad) si tienen el perfil de derechos adecuado para el tipo de contabilidad que se debe gestionar:

- Gestión de flujo de contabilidad ampliada
- Gestión de procesos
- Administración de tareas
- Gestión de redes

▼ Cómo activar la contabilidad ampliada para los flujos, los procesos, las tareas y los componentes de red

Para activar la función de contabilidad ampliada para tareas, procesos, flujos y componentes de red, utilice el comando `acctadm`. El parámetro final opcional para `acctadm` indica si el comando se aplica a los componentes de contabilidad de flujos, procesos, tareas del sistema o redes de la función de contabilidad ampliada.

Nota - Los roles incluyen autorizaciones y comandos con privilegios. Para obtener información sobre cómo crear el rol y asignarlo a un usuario mediante la función de control de acceso basado en roles (RBAC) de Oracle Solaris, consulte [“Protección de los usuarios y los procesos en Oracle Solaris 11.2”](#).

1. **Conviértase en root o asuma un rol similar.**
2. **Active la contabilidad ampliada para los procesos.**

```
# acctadm -e extended -f /var/adm/exacct/proc process
```

3. **Active la contabilidad ampliada para las tareas.**

```
# acctadm -e extended,mstate -f /var/adm/exacct/task task
```

4. **Active la contabilidad ampliada para los flujos.**

```
# acctadm -e extended -f /var/adm/exacct/flow flow
```

5. **Active la contabilidad ampliada para la red.**

```
# acctadm -e extended -f /var/adm/exacct/net net
```

Ejecute `acctadm` en los enlaces y flujos administrados por los comandos `dladm` y `flowadm`.

Véase también Para obtener más información, consulte [acctadm\(1M\)](#).

Cómo mostrar el estado de la contabilidad ampliada

Escriba `acctadm` sin argumentos para mostrar el estado actual de la función de cuentas extendidas.

```
machine% acctadm
Task accounting: active
Task accounting file: /var/adm/exacct/task
Tracked task resources: extended
Untracked task resources: none
Process accounting: active
Process accounting file: /var/adm/exacct/proc
Tracked process resources: extended
Untracked process resources: host
Flow accounting: active
Flow accounting file: /var/adm/exacct/flow
Tracked flow resources: extended
Untracked flow resources: none
```

En el ejemplo anterior, la cuenta de tareas del sistema está activa en el modo extendido y el modo `mstate`. Las cuentas de proceso y flujo están activas en modo extendido.

Nota - En el contexto de la contabilidad ampliada, el microestado (`mstate`) hace referencia a los datos ampliados, asociados con las transiciones de procesos de microestado, disponibles en el archivo de uso del proceso (consulte [proc\(4\)](#)). Estos datos proporcionan mucha más cantidad de detalles sobre las actividades del proceso que los registros básicos o extendidos.

Cómo ver los recursos de cuentas disponibles

Los recursos disponibles pueden variar según el sistema y la plataforma. Utilice el comando `acctadm` con la opción `-r` para ver los grupos de recursos de contabilidad disponibles en el sistema.

```
machine% acctadm -r
process:
extended pid,uid,gid,cpu,time,command,tty,projid,taskid,ancpid,wait-status,zone,flag,
memory,mstate displays as one line
basic pid,uid,gid,cpu,time,command,tty,flag
task:
extended taskid,projid,cpu,time,host,mstate,anctaskid,zone
basic taskid,projid,cpu,time
flow:
```

```

extended
saddr,daddr,sport,dport,proto,dsfield,nbytes,npkts,action,ctime,lseen,projid,uid
basic      saddr,daddr,sport,dport,proto,nbytes,npkts,action
net:
  extended name,devname,edest,vlan_tpid,vlan_tci,sap,cpuid, \
  priority,bwlimit,curtime,ibytes,obytes,ipkts,opks,ierrpks \
  oerrpks,saddr,daddr,sport,dport,protocol,dsfield
  basic   name,devname,edest,vlan_tpid,vlan_tci,sap,cpuid, \
  priority,bwlimit,curtime,ibytes,obytes,ipkts,opks,ierrpks \
  oerrpks

```

▼ Cómo desactivar la contabilidad de procesos, tareas, flujos y gestión de redes

Para desactivar la contabilidad de procesos, tareas, flujos y redes, desactive cada uno de ellos de forma individual utilizando el comando `acctadm` con la opción `-x`.

1. **Conviértase en root o asuma un rol similar.**

2. **Desactive la contabilidad de procesos.**

```
# acctadm -x process
```

3. **Desactive la contabilidad de tareas.**

```
# acctadm -x task
```

4. **Desactive la contabilidad de flujos.**

```
# acctadm -x flow
```

5. **Desactive la contabilidad de gestión de redes.**

```
# acctadm -x net
```

6. **Compruebe que se haya desactivado la contabilidad de tareas, procesos, flujos y redes.**

```

# acctadm
Task accounting: inactive
Task accounting file: none
Tracked task resources: none
Untracked task resources: extended
Process accounting: inactive
Process accounting file: none
Tracked process resources: none
Untracked process resources: extended
Flow accounting: inactive
Flow accounting file: none
Tracked flow resources: none

```

```
Untracked flow resources: extended
    Net accounting: inactive
    Net accounting file: none
Tracked Net resources: none
Untracked Net resources: extended
```

Uso de la interfaz de Perl para libexacct

Cómo imprimir de forma recurrente el contenido de un objeto exacct

Utilice el código siguiente para imprimir de forma recurrente el contenido de un objeto exacct. Tenga en cuenta que esta función la proporciona la biblioteca como función de `Sun::Solaris::Exacct::Object::dump()`. Esta función también está disponible mediante la función de conveniencia `ea_dump_object()`.

```
sub dump_object
{
    my ($obj, $indent) = @_;
    my $istr = ' ' x $indent;

    #
    # Retrieve the catalog tag. Because we are
    # doing this in an array context, the
    # catalog tag will be returned as a (type, catalog, id)
    # triplet, where each member of the triplet will behave as
    # an integer or a string, depending on context.
    # If instead this next line provided a scalar context, e.g.
    #   my $cat = $obj->catalog()->value();
    # then $cat would be set to the integer value of the
    # catalog tag.
    #
    my @cat = $obj->catalog()->value();

    #
    # If the object is a plain item
    #
    if ($obj->type() == &EO_ITEM) {
        #
        # Note: The '%s' formats provide s string context, so
        # the components of the catalog tag will be displayed
        # as the symbolic values. If we changed the '%s'
        # formats to '%d', the numeric value of the components
        # would be displayed.
        #
        printf("%sITEM\n%s Catalog = %s|%s|%s\n",
            $istr, $istr, @cat);
        $indent++;
    }
}
```

```
#
# Retrieve the value of the item. If the item contains
# in turn a nested exacct object (i.e., an item or
# group), then the value method will return a reference
# to the appropriate sort of perl object
# (Exacct::Object::Item or Exacct::Object::Group).
# We could of course figure out that the item contained
# a nested item orgroup by examining the catalog tag in
# @cat and looking for a type of EXT_EXACCT_OBJECT or
# EXT_GROUP.
#
my $val = $obj->value();
if (ref($val)) {
    # If it is a nested object, recurse to dump it.
    dump_object($val, $indent);
} else {
    # Otherwise it is just a 'plain' value, so
    # display it.
    printf("%s  Value = %s\n", $istr, $val);
}

#
# Otherwise we know we are dealing with a group. Groups
# represent contents as a perl list or array (depending on
# context), so we can process the contents of the group
# with a 'foreach' loop, which provides a list context.
# In a list context the value method returns the content
# of the group as a perl list, which is the quickest
# mechanism, but doesn't allow the group to be modified.
# If we wanted to modify the contents of the group we could
# do so like this:
#   my $grp = $obj->value(); # Returns an array reference
#   $grp->[0] = $newitem;
# but accessing the group elements this way is much slower.
#
} else {
    printf("%sGROUP\n%s  Catalog = %s|%s|\n",
        $istr, $istr, @cat);
    $indent++;
    # 'foreach' provides a list context.
    foreach my $val ($obj->value()) {
        dump_object($val, $indent);
    }
    printf("%sENDGROUP\n", $istr);
}
}
```


Cómo crear un registro de grupo y guardarlo en un archivo

Utilice esta secuencia de comandos para crear un registro de grupos y guardarlo en un archivo denominado /tmp/exacct.

```
#!/usr/bin/perl

use strict;
use warnings;
use Sun::Solaris::Exacct qw(:EXACCT_ALL);
# Prototype list of catalog tags and values.
my @items = (
    [ &EXT_STRING | &EXC_DEFAULT | &EXD_CREATOR    => "me"      ],
    [ &EXT_UINT32  | &EXC_DEFAULT | &EXD_PROC_PID   => $$        ],
    [ &EXT_UINT32  | &EXC_DEFAULT | &EXD_PROC_UID   => $<        ],
    [ &EXT_UINT32  | &EXC_DEFAULT | &EXD_PROC_GID   => $(         ],
    [ &EXT_STRING  | &EXC_DEFAULT | &EXD_PROC_COMMAND => "/bin/rec" ],
);

# Create a new group catalog object.
my $cat = ea_new_catalog(&EXT_GROUP | &EXC_DEFAULT | &EXD_NONE)

# Create a new Group object and retrieve its data array.
my $group = ea_new_group($cat);
my $ary = $group->value();

# Push the new Items onto the Group array.
foreach my $v (@items) {
    push(@$ary, ea_new_item(ea_new_catalog($v->[0]), $v->[1]));
}

# Open the exacct file, write the record & close.
my $f = ea_new_file('/tmp/exacct', &O_RDWR | &O_CREAT | &O_TRUNC)
    || die("create /tmp/exacct failed: ", ea_error_str(), "\n");
$f->write($group);
$f = undef;
```

Cómo imprimir el contenido de un archivo exacct

Utilice la siguiente secuencia de comandos Perl para imprimir el contenido de un archivo exacct.

```
#!/usr/bin/perl

use strict;
use warnings;
use Sun::Solaris::Exacct qw(:EXACCT_ALL);
```

```
die("Usage is dumpexacct <exacct file>\n") unless (@ARGV == 1);

# Open the exacct file and display the header information.
my $ef = ea_new_file($ARGV[0], &O_RDONLY) || die(error_str());
printf("Creator:  %s\n", $ef->creator());
printf("Hostname: %s\n\n", $ef->hostname());

# Dump the file contents
while (my $obj = $ef->get()) {
    ea_dump_object($obj);
}

# Report any errors
if (ea_error() != EXR_OK && ea_error() != EXR_EOF) {
    printf("\nERROR: %s\n", ea_error_str());
    exit(1);
}
exit(0);
```

Ejemplo de salida de **Sun::Solaris::Exacct::Object->dump()**

A continuación se incluye un ejemplo de salida al ejecutar `Sun::Solaris::Exacct::Object->dump()` en el archivo creado en [“Cómo crear un registro de grupo y guardarlo en un archivo” \[57\]](#).

```
Creator:  root
Hostname: localhost
GROUP
  Catalog = EXT_GROUP|EXC_DEFAULT|EXD_NONE
  ITEM
    Catalog = EXT_STRING|EXC_DEFAULT|EXD_CREATOR
    Value = me
  ITEM
    Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_PID
    Value = 845523
  ITEM
    Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_UID
    Value = 37845
  ITEM
    Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_GID
    Value = 10
  ITEM
    Catalog = EXT_STRING|EXC_DEFAULT|EXD_PROC_COMMAND
    Value = /bin/rec
ENDGROUP
```

♦ ♦ ♦ C A P Í T U L O 6

Acerca de los controles de recursos

Una vez determinado el consumo de recursos de las cargas de trabajo del sistema tal como se describe en [Capítulo 4, Acerca de la contabilidad ampliada](#), puede establecer límites para el uso de recursos. Los límites evitan que las cargas de trabajo consuman recursos en exceso. La función de *controles de recursos* es el mecanismo de limitación que se utiliza para tal fin.

En este capítulo se tratan los temas siguientes.

- “Conceptos de controles de recursos” [59]
- “Configuración de controles de recursos y atributos” [61]
- “Aplicación de controles de recursos” [72]
- “Actualización temporal de los valores de controles de recursos en un sistema en ejecución” [73]
- “Comandos utilizados con controles de recursos” [74]

Para obtener información sobre cómo administrar los controles de recursos, consulte [Capítulo 7, Administración de las tareas de controles de recursos](#).

Conceptos de controles de recursos

En el sistema operativo Oracle Solaris, el concepto de límite de recursos por proceso se ha extendido a las entidades de tareas y proyectos descritas en [Capítulo 2, Acerca de los proyectos y las tareas](#). Estas mejoras se incorporan a través de la función de controles de recursos (rctl). Asimismo, las asignaciones que se configuraban mediante opciones de `/etc/system` ahora son automáticas o se configuran también mediante el mecanismo de controles de recursos.

Un control de recurso puede identificarse por el prefijo `zone`, `project`, `task` o `process`. Los controles de recursos pueden observarse en todo el sistema. Es posible actualizar los valores de controles de recursos en un sistema en ejecución.

Para ver una lista de los controles de recursos estándar disponibles en esta versión, consulte “[Controles de recursos disponibles](#)” [62]. Consulte “[Controles de recursos disponibles](#)” [62] para obtener información sobre los controles de recursos disponibles de toda la zona.

Límites y controles de recursos

Los sistemas UNIX proporcionan tradicionalmente una función de límite de recursos (*rlimit*). La función *rlimit* permite a los administradores configurar uno o más límites numéricos para la cantidad de recursos que puede consumir un proceso. Estos límites incluyen el tiempo de CPU utilizado por proceso, el tamaño de archivo núcleo por proceso y el tamaño de pila máximo por proceso. El *tamaño de pila* es la cantidad de memoria de trabajo asignada para el segmento de datos del proceso.

La función de controles de recursos ofrece interfaces de compatibilidad para la función de límites de recursos. Las aplicaciones existentes que utilizan límites de recursos siguen ejecutándose sin cambios. Estas aplicaciones pueden observarse del mismo modo que las aplicaciones modificadas para aprovechar la función de controles de recursos.

Comunicación entre procesos y controles de recursos

Los procesos pueden comunicarse entre sí utilizando uno de los distintos tipos de comunicación entre procesos (IPC). IPC permite transferir o sincronizar información entre procesos. La función de controles de recursos proporciona controles de recursos que definen el comportamiento de las funciones IPC del núcleo. Estos controles de recursos sustituyen las opciones configurables de `/etc/system`.

Los parámetros obsoletos que se utilizan para inicializar los valores de control de recursos predeterminados pueden incluirse en el archivo `/etc/system` de este sistema Oracle Solaris. Sin embargo, no se recomienda el uso de los parámetros obsoletos.

Para observar qué objetos IPC contribuyen al uso de un proyecto, utilice el comando `ipcs` con la opción `-J`. Consulte [“Cómo utilizar ipcs” \[84\]](#) para ver un ejemplo. Para obtener más información acerca del comando `ipcs`, consulte `ipcs(1)`.

Para obtener información sobre el ajuste del sistema Oracle Solaris, consulte el [“Manual de referencia de parámetros ajustables de Oracle Solaris 11.2”](#).

Mecanismos de limitación del control de recursos

Los controles de recursos ofrecen un mecanismo para la limitación de los recursos del sistema. Es posible evitar que los procesos, tareas, proyectos y zonas consuman determinadas cantidades de recursos del sistema. Este mecanismo conduce a un sistema más eficaz, puesto que evita el consumo excesivo de recursos.

Los mecanismos de limitación pueden utilizarse para complementar los procesos de planificación de capacidades. Una limitación puede proporcionar información sobre las necesidades de recursos de una aplicación sin denegar necesariamente el recurso a la aplicación.

Mecanismos de atributos de proyecto

Los controles de recursos también pueden servir de mecanismo de atributo simple para las funciones de administración de recursos. Por ejemplo, el número de recursos compartidos de CPU disponibles para un proyecto en la clase de planificación del programador de reparto justo (FSS) se define mediante el control de recurso `project.cpu-shares`. Dado que el control asigna al proyecto un número fijo de recursos compartidos, las diferentes acciones que excedan el control no son relevantes. En este contexto, el valor actual del control `project.cpu-shares` se considera un atributo del proyecto especificado.

Otro tipo de atributo de proyecto se utiliza para regular el consumo de recursos de la memoria física por parte de los grupos de procesos asociados a un proyecto. Estos atributos tienen el prefijo `rcap`, por ejemplo, `rcap.max-rss`. Al igual que los controles de recursos, este tipo de atributo se configura en la base de datos `project`. Sin embargo, mientras que los controles de recursos los aplica de forma sincronizada el núcleo, los límites de recursos los aplica el daemon de aplicación de límites de recursos, `rcapd`, en el nivel del usuario y sin sincronización. Para obtener información sobre `rcapd`, consulte [Capítulo 10, Acerca del control de memoria física mediante el daemon de limitación de recursos](#) y `rcapd(1M)`.

El atributo `project.pool` se utiliza para especificar la vinculación de agrupaciones para un proyecto. Para obtener información sobre agrupaciones de recursos, consulte [Capítulo 12, Acerca de las agrupaciones de recursos](#).

Configuración de controles de recursos y atributos

La función de controles de recursos se configura mediante la base de datos de `project`. Consulte [Capítulo 2, Acerca de los proyectos y las tareas](#). Los controles de recursos y otros atributos se configuran en el campo final de la entrada de la base de datos `project`. Los valores asociados con cada control de recursos se incluyen entre paréntesis, y aparecen como texto sin formato separado por comas. Los valores entre paréntesis comprenden una “cláusula de acción”. Cada cláusula de acción consta de un nivel de privilegio, un valor umbral y una acción asociada con el umbral específico. Cada control de recurso puede tener varias cláusulas de acción, que también se separan con comas. La entrada siguiente define un límite de proceso ligero por tarea y un límite de tiempo de CPU máximo por proceso en una entidad de proyecto. `process.max-cpu-time` enviaría `SIGTERM` a un proceso después de que el proceso se ejecute durante una hora, y `SIGKILL` si el proceso siguiera ejecutándose durante una hora y un minuto. Consulte [Tabla 6-3, “Señales disponibles para los valores de controles de recursos”](#).

```
development:101:Developers:::task.max-lwps=(privileged,10,deny);
```

```
process.max-cpu-time=(basic,3600,signal=TERM),(priv,3660,signal=KILL)
typed as one line
```

Nota - En los sistemas con zonas activadas, los controles de recursos de la zona se especifican en la configuración de zona con un formato ligeramente diferente. Consulte [“Configuración de controles de recursos de zonas”](#) de [“Introducción a Zonas de Oracle Solaris”](#) para obtener más información.

El comando `rctladm` permite realizar interrogaciones de tiempo de ejecución y modificaciones en la función de controles de recursos con *ámbito global*. El comando `prctl` permite realizar interrogaciones de tiempo de ejecución y modificaciones en la función de controles de recursos, con *ámbito local*.

Para obtener más información, consulte [“Acciones locales y globales en valores de controles de recursos”](#) [68], `rctladm(1M)` y `prctl(1)`.

Nota - En un sistema con zonas instaladas, no puede utilizar `rctladm` en una zona no global para modificar la configuración. Puede utilizar `rctladm` en una zona no global para ver el estado de registro global de cada control de recurso.

Controles de recursos disponibles

En la tabla siguiente se incluye una lista de los controles de recursos estándar disponibles en esta versión.

La tabla describe el recurso que limita cada control. Asimismo, identifica las unidades predeterminadas que utiliza la base de datos `project` para dicho recurso. Las unidades predeterminadas pueden ser de dos tipos:

- Las cantidades representan una cantidad limitada.
- Los índices representan un identificador válido máximo.

De este modo, `project.cpu-shares` especifica el número de recursos compartidos a los que tiene derecho el proyecto. `process.max-file-descriptor` especifica el número de archivo más elevado que se puede asignar a un proceso mediante la llamada de sistema `open(2)`.

TABLA 6-1 Controles de recursos de proyectos, tareas y procesos estándar

Nombre de control	Descripción	Unidad predeterminada
<code>project.cpu-cap</code>	Límite absoluto sobre la cantidad de recursos de la CPU que puede consumir un proyecto. Un valor de 100 significa el 100% de una CPU como opción de <code>project.cpu-cap</code> . Un valor de 125 es 125%, ya que el	Cantidad (número de CPU)

Nombre de control	Descripción	Unidad predeterminada
	100% corresponde a una CPU completa del sistema al utilizar el recurso <code>cpu-cap</code> .	
<code>project.cpu-shares</code>	Número de recursos compartidos de CPU concedidos a este proyecto para utilizar con el programador de reparto justo (consulte FSS(7)).	Cantidad (recursos compartidos)
<code>project.max-crypto-memory</code>	Cantidad total de memoria de núcleo que puede utilizar <code>libpkcs11</code> para la aceleración criptográfica de hardware. Las asignaciones de búferes de núcleo y estructuras relativas a las sesiones se realizan para este control de recurso.	Tamaño (bytes)
<code>project.max-locked-memory</code>	Cantidad total de memoria física bloqueada permitida. Si se asigna <code>priv_proc_lock_memory</code> a un usuario, procure configurar también este control de recurso para evitar que el usuario bloquee toda la memoria. Tenga en cuenta que este control de recursos sustituye a <code>project.max-device-locked-memory</code> , que se ha eliminado.	Tamaño (bytes)
<code>project.max-msg-ids</code>	Número máximo de ID de cola de mensajes permitidos para este proyecto.	Cantidad (ID de cola de mensajes)
<code>project.max-port-ids</code>	Número máximo de puertos de eventos permitidos.	Cantidad (número de puertos de eventos)
<code>project.max-processes</code>	Número máximo de ranuras de tablas de procesos disponibles de forma simultánea para este proyecto. Tenga en cuenta que, debido a que los procesos normales y los procesos zombie ocupan ranuras de tabla de procesos, el control <code>max-processes</code> protege contra los zombies que agotan la tabla de procesos. Debido a que los procesos zombie no tienen procesos ligeros por definición, el control <code>max-lwps</code> no puede proteger contra esta posibilidad.	Cantidad (ranuras de tabla de procesos)
<code>project.max-sem-ids</code>	Número máximo de ID de semáforo permitidos para este proyecto.	Cantidad (ID de semáforo)
<code>project.max-shm-ids</code>	Número máximo de ID de memoria compartida permitidos para este proyecto.	Cantidad (ID de memoria compartida)
<code>project.max-shm-memory</code>	Cantidad total de memoria compartida System V permitida para este proyecto.	Tamaño (bytes)
<code>project.max-lwps</code>	Número máximo de procesos ligeros disponibles de forma simultánea para este proyecto.	Cantidad (LWP)

Nombre de control	Descripción	Unidad predeterminada
project.max-tasks	Número máximo de tareas permitidas en este proyecto.	Cantidad (número de tareas)
project.max-contracts	Número máximo de contratos permitidos en este proyecto.	Cantidad (contratos)
task.max-cpu-time	Tiempo máximo de CPU disponible para los procesos de esta tarea.	Tiempo (segundos)
task.max-lwps	Número máximo de procesos ligeros disponibles de forma simultánea para los procesos de esta tarea.	Cantidad (LWP)
task.max-processes	Número máximo de ranuras de tablas de procesos disponibles de forma simultánea para los procesos de esta tarea.	Cantidad (ranuras de tabla de procesos)
process.max-cpu-time	Tiempo máximo de CPU disponible para este proceso.	Tiempo (segundos)
process.max-file-descriptor	Índice descriptor de archivos máximo disponible para este proceso.	Índice (descriptor de archivo máximo)
process.max-file-size	Desfase de archivo máximo disponible para escritura de este proceso.	Tamaño (bytes)
process.max-core-size	Tamaño máximo de archivo principal creado por este proceso.	Tamaño (bytes)
process.max-data-size	Memoria de pila máxima disponible para este proceso.	Tamaño (bytes)
process.max-stack-size	Segmento de memoria de pila máxima disponible para este proceso.	Tamaño (bytes)
process.max-address-space	Cantidad máxima de espacio de dirección, sumando los tamaños de segmentos, disponible para este proceso.	Tamaño (bytes)
process.max-port-events	Número de eventos máximo permitido por puerto de eventos.	Cantidad (número de eventos)
process.max-sem-nsems	Número máximo de semáforos permitidos por conjunto de semáforos.	Cantidad (semáforos por conjunto)
process.max-sem-ops	Número máximo de operaciones de semáforos permitidas por llamada semop (valor copiado del control de recursos en la hora de semget()).	Cantidad (número de operaciones)
process.max-msg-qbytes	Número máximo de bytes de mensajes en una cola de mensajes (valor copiado del control de recurso en la hora de msgget()).	Tamaño (bytes)
process.max-msg-messages	Número máximo de mensajes en una cola de mensajes (valor copiado del control de recurso en la hora de msgget()).	Cantidad (número de mensajes)

Puede ver los valores predeterminados de los controles de recursos de un sistema que no tenga controles de recursos definidos ni cambiados. Dicho sistema no contiene entradas que no sean

predeterminadas en `/etc/system` o la base de datos `project`. Para ver los valores, utilice el comando `prctl`.

Controles de recursos de toda la zona

Los controles de recursos de la zona limitan el uso total de los recursos de todas las entidades de procesos de una zona. Los controles de recursos de toda la zona también se pueden definir utilizando los nombres de propiedades globales descritos en [“Componentes de zonas” de “Introducción a Zonas de Oracle Solaris”](#).

TABLA 6-2 Controles de recursos de zonas

Nombre de control	Descripción	Unidad predeterminada
<code>zone.cpu-cap</code>	Límite absoluto sobre la cantidad de recursos de la CPU que puede consumir una zona no global. Un valor de 100 significa el 100% de una CPU como opción de <code>project.cpu-cap</code> . Un valor de 125 es 125%, ya que el 100% corresponde a una CPU completa del sistema al utilizar el recurso <code>cpu-cap</code> .	Cantidad (número de CPU)
<code>zone.cpu-shares</code>	Número de recursos compartidos de CPU del planificador de reparto justo (FSS) para esta zona	Cantidad (recursos compartidos)
<code>zone.max-lofi</code>	Número máximo de dispositivos <code>lofi</code> que una zona puede crear. El valor limita el uso de espacio de nombres de nodo menor de cada zona.	Cantidad (número de dispositivos <code>lofi</code>)
<code>zone.max-locked-memory</code>	Cantidad total de memoria física bloqueada disponible para una zona. Si se asigna <code>priv_proc_lock_memory</code> a una zona, procure configurar también este control de recurso para evitar que la zona bloquee toda la memoria.	Tamaño (bytes)
<code>zone.max-lwps</code>	Número máximo de procesos ligeros disponibles de forma simultánea para esta zona	Cantidad (LWP)
<code>zone.max-msg-ids</code>	Número máximo de ID de cola de mensajes permitidos para esta zona	Cantidad (ID de cola de mensajes)
<code>zone.max-processes</code>	Número máximo de ranuras de tablas de procesos disponibles de forma simultánea para esta zona. Debido a que los procesos normales y los procesos zombie ocupan ranuras de tablas de procesos, el control <code>max-processes</code>	Cantidad (ranuras de tabla de procesos)

Nombre de control	Descripción	Unidad predeterminada
	protege contra los zombies que agotan la tabla de procesos. Debido a que los procesos zombie no tienen procesos ligeros por definición, el control max-lwps no puede proteger contra esta posibilidad.	
zone.max-sem-ids	Número máximo de ID de semáforo permitidos para esta zona	Cantidad (ID de semáforo)
zone.max-shm-ids	Número máximo de ID de memoria compartida permitidos para esta zona	Cantidad (ID de memoria compartida)
zone.max-shm-memory	Cantidad total de memoria compartida System V permitida para esta zona	Tamaño (bytes)
zone.max-swap	Cantidad total de intercambio que pueden consumir las asignaciones de espacio de direcciones del proceso del usuario y los montajes tmpfs para esta zona.	Tamaño (bytes)

Para obtener información sobre la configuración de los controles de recursos de toda la zona, consulte [“Configuración de controles de recursos y atributos” \[61\]](#) y [“Configuración de controles de recursos de zonas” de “Introducción a Zonas de Oracle Solaris”](#).

Tenga en cuenta que es posible aplicar un control de recurso de la zona para la zona global.

Compatibilidad con unidades

Los indicadores globales que identifican los tipos de controles de recursos se definen para todos los controles de recursos. El sistema utiliza los indicadores para comunicar la información de tipo básico a las aplicaciones como el comando `prctl`. Las aplicaciones utilizan la información para determinar lo siguiente:

- Las cadenas de unidades apropiadas para cada control de recurso
- La escala correcta que utilizar al interpretar valores a escala

Hay disponibles los siguientes indicadores globales:

Indicador global	Cadena de tipo de control de recurso	Modificador	Escala
RCTL_GLOBAL_BYTES	bytes	B	1
		KB	2^{10}
		MB	2^{20}
		GB	2^{30}
		TB	2^{40}
		PB	2^{50}

Indicador global	Cadena de tipo de control de recurso	Modificador	Escala
RCTL_GLOBAL_SECONDS	segundos	EB	2^{60}
		s	1
		Ks	10^3
		Sra.	10^6
		Gs	10^9
		Ts	10^{12}
		Ps	10^{15}
		Es	10^{18}
RCTL_GLOBAL_COUNT	número	none	1
		K	10^3
		M	10^6
		G	10^9
		T	10^{12}
		P	10^{15}
		E	10^{18}

Pueden utilizarse valores a escala con los controles de recursos. El ejemplo siguiente muestra un valor umbral a escala:

```
task.max-lwps=(priv,1K,deny)
```

Nota - Los comandos `prctl`, `projadd` y `projmod` aceptan modificadores de unidades. No puede utilizar modificadores de unidades en la base de datos `project`.

Valores de controles de recursos y niveles de privilegio

Un valor umbral en un control de recursos constituye un punto de aplicación en el que se pueden desencadenar acciones locales o globales, como registros.

Cada valor umbral de un control de recursos debe asociarse con un nivel de privilegio. El nivel de privilegio debe ser de uno de estos tres tipos.

- Básico, que puede modificar el propietario del proceso que realiza la llamada
- Con privilegios, que sólo pueden modificar los autores de la llamada con privilegios (root)
- Sistema, que es fijo para la duración de la instancia del sistema operativo

Se garantiza que un control de recurso tiene un valor de sistema, definido por el sistema, o un proveedor de recursos. El valor del sistema representa qué cantidad del recurso es capaz de proporcionar la implementación actual del sistema operativo.

Puede definirse cualquier número de valores con privilegio, y sólo se permite un valor básico. A las operaciones que se llevan a cabo sin especificar un valor de privilegio se les asigna de forma predeterminada un privilegio básico.

El nivel de privilegio para un valor de control de recurso se define en el campo de privilegio del bloque de control de recurso como `RCTL_BASIC`, `RCTL_PRIVILEGED` o `RCTL_SYSTEM`.

Para obtener más información, consulte [setrctl\(2\)](#). Puede utilizar el comando `prctl` para modificar los valores asociados con los niveles básico y con privilegios.

Acciones locales y globales en valores de controles de recursos

Hay dos categorías de acciones en los valores de controles de recursos: globales y locales.

Acciones globales en valores de controles de recursos

Las acciones globales se aplican a los valores de controles de recursos para cada control de recurso del sistema. Puede utilizar el comando `rctladm` que se describe en la página del comando `man rctladm(1M)` para llevar a cabo las acciones siguientes:

- Mostrar el estado global de los controles de recursos activos del sistema
- Definir acciones de registro globales

Puede desactivar o activar la acción de registro global en los controles de recursos. Es posible definir la acción `syslog` hasta un grado específico asignando un nivel de gravedad, `syslog=nivel`. Las posibles configuraciones de *nivel* son:

- `debug`
- `info`
- `notice`
- `warning`
- `err`
- `crit`
- `alert`
- `emerg`

De modo predeterminado, no hay ningún registro global de los conflictos de controles de recursos. El nivel `n/a` indica los controles de recursos en los que no se puede configurar ninguna acción global.

Acciones locales en valores de controles de recursos

Las acciones locales se llevan a cabo en un proceso que intenta exceder el valor de control. Para cada valor umbral que se coloca en un control de recurso, puede asociar una o más acciones. Hay tres tipos de acciones locales: `none`, `deny` y `signal=`. Estas tres acciones se utilizan del modo siguiente:

<code>none</code>	No se emprende ninguna acción en las solicitudes de recursos para una cantidad que supere el umbral. Esta acción resulta útil para supervisar el uso de los recursos sin que ello afecte al progreso de las aplicaciones. También puede activar un mensaje global que se muestre al superar el control de recurso, aunque el proceso que supera el umbral no se vea afectado.
<code>deny</code>	Puede denegar las solicitudes de recursos para una cantidad que supere el umbral. Por ejemplo, un control de recurso <code>task.max-lwps</code> con la acción <code>deny</code> hace que la llamada de sistema <code>fork</code> falle si el nuevo proceso supera el valor de control. Consulte la página del comando <code>man fork(2)</code> .
<code>signal=</code>	Puede activar una acción de mensaje de señal global cuando se supera el control de recurso. Se envía una señal al proceso cuando se supera el valor umbral. No se envían señales adicionales si el proceso consume recursos adicionales. En Tabla 6-3, “Señales disponibles para los valores de controles de recursos” , se enumeran las señales disponibles.

No se pueden aplicar todas las acciones a cada control de recurso. Por ejemplo, un proceso no puede superar el número de recursos compartidos de CPU asignados al proyecto del cual es miembro. Por tanto, no se permite una acción de denegación en el control de recurso `project.cpu-shares`.

Debido a la restricción de la implementación, las propiedades globales de cada control pueden restringir el intervalo de acciones disponibles que se pueden configurar en el valor umbral. (Consulte la página del comando `man rctladm(1M)`). En la tabla siguiente se presenta una lista de las acciones de señales disponibles. Para obtener información adicional sobre las señales, consulte la página del comando `man signal(3HEAD)`.

TABLA 6-3 Señales disponibles para los valores de controles de recursos

Señal	Descripción	Notas
SIGABRT	Finaliza el proceso.	
SIGHUP	Envía una señal de colgar. Tiene lugar cuando el portador se encuentra en una línea abierta. La señal se envía al grupo de proceso que controla el terminal.	
SIGTERM	Finaliza el proceso. Señal de finalización que envía el software.	
SIGKILL	Finaliza el proceso y cierra el programa.	

Señal	Descripción	Notas
SIGSTOP	Detiene el proceso. Señal de control del trabajo.	
SIGXRES	Límite de control de recurso superado. Lo genera la función de control de recurso.	
SIGXFSZ	Finaliza el proceso. Límite de tamaño de archivo superado.	Disponibles sólo para los controles de recursos con la propiedad RCTL_GLOBAL_FILE_SIZE (process.max-file-size). Para obtener más información, consulte rctlblk_set_value(3C) .
SIGXCPU	Finaliza el proceso. Límite de tiempo de CPU superado.	Disponibles sólo para los controles de recursos con la propiedad RCTL_GLOBAL_CPU_TIME (process.max-cpu-time). Para obtener más información, consulte rctlblk_set_value(3C) .

Propiedades e indicadores de controles de recursos

Cada control de recurso del sistema tiene un grupo determinado de propiedades asociadas. Este conjunto de propiedades se define como un conjunto de indicadores, asociados con todas las instancias controladas de dicho recurso. Los indicadores globales no se pueden modificar, pero se pueden recuperar utilizando las llamadas de sistema `rctladm` o `getrctl`.

Los indicadores globales definen el comportamiento predeterminado y la configuración de un valor umbral específico de dicho control de recurso en un proceso específico o colectivo. Los indicadores locales de un valor umbral no afectan al comportamiento de otros valores umbral definidos para el mismo control de recurso. Sin embargo, los indicadores globales afectan al comportamiento de cada valor asociado con un control particular. Los indicadores locales pueden modificarse, dentro de los límites establecidos por sus indicadores globales correspondientes, mediante el comando `prctl` o la llamada de sistema `setrctl`. Consulte [setrctl\(2\)](#).

Para ver una lista completa de los indicadores locales y globales, y sus definiciones, consulte [rctlblk_set_value\(3C\)](#).

Para determinar el comportamiento del sistema cuando se alcanza un valor umbral para un control de recurso concreto, utilice `rctladm` para ver los indicadores globales para el control de recurso. Por ejemplo, para ver los valores de `process.max-cpu-time`, escriba lo siguiente:

```
$ rctladm process.max-cpu-time
process.max-cpu-time syslog=off [ lowerable no-deny cpu-time inf seconds ]
```

Los indicadores globales significan lo siguiente.

lowerable	No se requieren privilegios de superusuario para reducir los valores con privilegios para este control.
no-deny	Nunca se deniega el acceso al recurso, ni siquiera cuando se superan los valores umbral.
cpu-time	SIGXCPU está disponible para su envío cuando se alcanzan los valores umbral de este recurso.
seconds	Valor de tiempo para el control de recurso.
no-basic	Los valores de control de recurso con el tipo de privilegio basic no se pueden establecer. Sólo se permiten valores de control de recurso con privilegios.
no-signal	Una acción de señal local no se puede establecer en valores de control de recurso.
no-syslog	La acción de mensaje syslog global no se puede establecer para este control de recurso.
deny	Siempre que se sobrepasen los valores de umbral, deniegue la solicitud de recursos.
count	Valor (entero) de cantidad para el control de recurso.
bytes	Unidad de tamaño de control de recurso.

Utilice el comando `prctl` para ver las acciones y los valores locales para el control de recurso.

```
$ prctl -n process.max-cpu-time $$
process 353939: -ksh
NAME      PRIVILEGE  VALUE   FLAG   ACTION      RECIPIENT
process.max-cpu-time
  privileged 18.4Es    inf    signal=XCPU  -
  system    18.4Es    inf    none
```

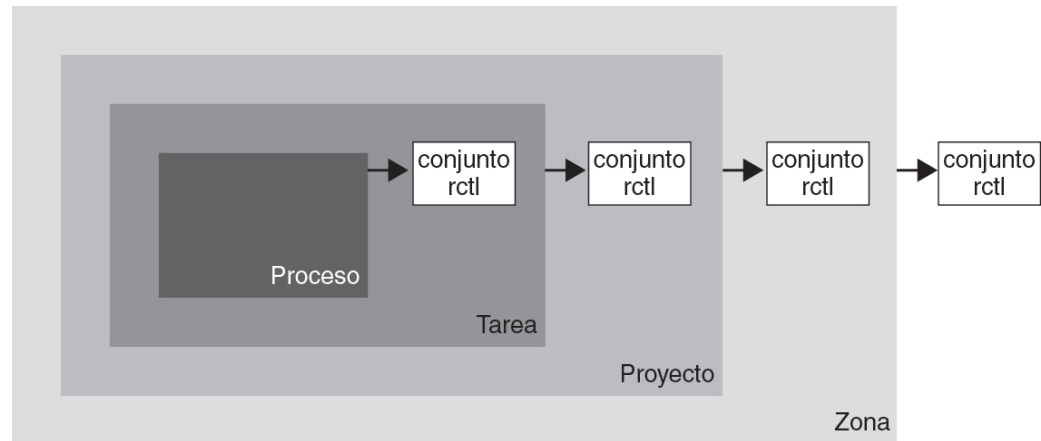
El indicador `max` (`RCTL_LOCAL_MAXIMAL`) se configura para ambos valores umbral, y el indicador `inf` (`RCTL_GLOBAL_INFINITE`) se define para este control de recurso. Un valor `inf` tiene una cantidad infinita. El valor nunca se aplica. Por ello, ambas cantidades umbral representan valores infinitos que nunca se superan.

Aplicación de controles de recursos

Un recurso puede tener más de un control de recurso. Puede haber un control de recurso en cada nivel de contenido en el modelo de proceso. Si hay controles de recursos activos en el mismo

recurso en diferentes niveles de contenido, se aplicará en primer lugar el control del contenedor más pequeño. De este modo, la acción se lleva a cabo en `process.max-cpu-time` antes que en `task.max-cpu-time` si ambos controles se encuentran de forma simultánea.

FIGURA 6-1 Procesos colectivos, relaciones de contenedores y sus conjuntos de controles de recursos



Supervisión global de los eventos de controles de recursos

A menudo, se desconoce el consumo de recursos de los procesos. Para obtener más información, utilice las acciones de control de recursos globales disponibles con el comando `rctladm`. Utilice `rctladm` para establecer una acción `syslog` en un control de recursos. A continuación, si una entidad administrada por dicho control de recurso se encuentra con un valor umbral, se registra un mensaje del sistema en el nivel de registro configurado. Si desea más información, consulte [Capítulo 7, Administración de las tareas de controles de recursos](#) y la página del comando `man rctladm(1M)`.

Aplicación de controles de recursos

Cada control de recurso incluido en [Tabla 6-1, “Controles de recursos de proyectos, tareas y procesos estándar”](#) puede asignarse a un proyecto en el inicio de sesión o cuando se invocan `newtask`, su o los demás programas de ejecución basados en proyectos `at`, `batch` o `cron`. Cada comando que se inicia se abre en una tarea separada con el proyecto predeterminado del usuario

que ejecuta el comando. Para obtener más información, consulte las páginas del comando `man login(1)`, `newtask(1)`, `at(1)`, `cron(1M)` y `su(1M)`.

Las actualizaciones de las entradas de la base de datos `project`, tanto si son para el archivo `/etc/project` o una representación de la base de datos en un servicio de nombres de red, no se aplican a los proyectos activos. Las actualizaciones se aplican cuando una tarea nueva se une al proyecto mediante el inicio de sesión o `newtask`.

Actualización temporal de los valores de controles de recursos en un sistema en ejecución

Los valores modificados en la base de datos `project` sólo pasan a ser efectivos para las nuevas tareas que se inician en un proyecto. Sin embargo, puede utilizar los comandos `rctladm` y `prctl` para actualizar los controles de recursos en un sistema en ejecución.

Actualización de estados de registro

El comando `rctladm` afecta al estado de registro global de cada control de recurso en todo el sistema. Este comando puede utilizarse para ver el estado global y configurar el nivel de registro de `syslog` cuando se superan los controles.

Actualización de controles de recursos

Puede ver y modificar temporalmente los valores de controles de recursos y acciones por proceso, tarea o proyecto utilizando el comando `prctl`. Se especifica un ID de proceso, tarea o proyecto y el comando se aplica al control de recurso en el nivel en el que esté definido el control.

Cualquier modificación en los valores y acciones surtirá efecto de inmediato. No obstante, estas modificaciones se aplican únicamente al proceso, la tarea o el proyecto actuales. Los cambios no se registran en la base de datos `project`. Si se reinicia el sistema, se pierden las modificaciones. Los cambios permanentes en los controles de recursos deben realizarse en la base de datos `project`.

Todos los parámetros de los controles de recursos que se pueden modificar en la base de datos `project` también pueden modificarse con el comando `prctl`. Pueden agregarse o suprimirse valores básicos o con privilegios. Sus acciones también pueden modificarse. De modo predeterminado, se da por sentado el tipo básico para todas las operaciones, pero los procesos y usuarios con privilegios `root` también pueden modificar los controles de recursos con privilegios. No es posible modificar los controles de recursos del sistema.

Comandos utilizados con controles de recursos

En la tabla siguiente se muestran los comandos que se utilizan con controles de recursos.

Referencia de comando	Descripción
ipcs(1)	Permite observar qué objetos IPC contribuyen al uso de un proyecto.
prctl(1)	Permite realizar interrogaciones de tiempo de ejecución y modificaciones en la función de controles de recursos, con ámbito local.
rctladm(1M)	Permite realizar interrogaciones de tiempo de ejecución y modificaciones en la función de controles de recursos, con ámbito global.

La página del comando man [resource_controls\(5\)](#) describe los controles de recursos disponibles mediante la base de datos del proyecto, incluidos los factores de escala y las unidades.

♦ ♦ ♦ CAPÍTULO 7

Administración de las tareas de controles de recursos

En este capítulo se describe cómo administrar la función de controles de recursos.

Para obtener una descripción general sobre la función de controles de recursos, consulte [Capítulo 6, Acerca de los controles de recursos](#).

Administración del mapa de tareas de controles de recursos

Tarea	Descripción	Instrucciones
Definir controles de recursos.	Define los controles de recursos para un proyecto en el archivo <code>/etc/project</code> .	“Configuración de controles de recursos” [76]
Configurar o modificar los valores de controles de recursos para los procesos, tareas o proyectos activos, en el ámbito local.	Realiza interrogaciones de tiempo de ejecución y modificaciones de los controles de recursos asociados con un proceso, tarea o proyecto activos del sistema.	“Uso del comando <code>prctl</code>” [78]
En un sistema en ejecución, visualizar o actualizar el estado global de los controles de recursos.	Visualiza el estado de registro global de cada control de recurso en el ámbito del sistema. Asimismo, establece el nivel de registro de <code>syslog</code> cuando se superan los controles.	“Uso de <code>rctladm</code>” [83]
Estado de informe de las funciones de comunicación entre procesos (IPC) activas.	Muestra información sobre las funciones de comunicación entre procesos (IPC) activas. Observe qué objetos IPC contribuyen al uso de un proyecto.	“Uso de <code>ipcs</code>” [84]
Determinar si un servidor Web tiene asignada suficiente capacidad de CPU.	Define una acción global en un control de recurso. Esta acción permite recibir advertencias de cualquier entidad que tenga un valor de control de recurso configurado demasiado bajo.	Cómo determinar si un servidor Web tiene asignada suficiente capacidad de CPU [85]

Configuración de controles de recursos

▼ Cómo definir el número máximo de procesos ligeros para cada tarea de un proyecto

Este procedimiento agrega un proyecto denominado `x-files` al archivo `/etc/project` y establece un número máximo de procesos ligeros para una tarea creada en el proyecto.

1. **Conviértase en root o asuma un rol similar.**
2. **Utilice el comando `projadd` con la opción `-K` para crear un proyecto denominado `x-files`. Defina el número máximo de procesos ligeros para cada tarea creada en el proyecto como 3.**

```
# projadd -K 'task.max-lwps=(privileged,3,deny)' x-files
```

3. **Visualice la entrada en el archivo `/etc/project` siguiendo uno de estos métodos:**

- Escriba:

```
# projects -l
system
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
.
.
.
x-files
    projid : 100
    comment: ""
    users  : (none)
    groups : (none)
    attribs: task.max-lwps=(privileged,3,deny)
```

- Escriba:

```
# cat /etc/project
system:0:System:::
.
.
.
x-files:100:::task.max-lwps=(privileged,3,deny)
```

ejemplo 7-1 Ejemplo de sesión

Una vez implementados los pasos de este procedimiento, cuando el usuario root cree una tarea nueva en el proyecto x-files uniendo el proyecto con newtask, el usuario no podrá crear más de tres procesos ligeros mientras se ejecute esta tarea. Esto puede verse en la siguiente sesión de ejemplo.

```
# newtask -p x-files csh

# prctl -n task.max-lwps $$
process: 111107: csh
NAME    PRIVILEGE    VALUE    FLAG    ACTION    RECIPIENT
task.max-lwps
        usage          3
        privileged    3        -    deny    -
        system      2.15G    max    deny    -

# id -p
uid=0(root) gid=1(other) projid=100(x-files)

# ps -o project,taskid -p $$
PROJECT TASKID
x-files   73

# csh          /* creates second LWP */

# csh          /* creates third LWP */

# csh          /* cannot create more LWPs */
Vfork failed
#
```

▼ Cómo definir múltiples controles en un proyecto

El archivo /etc/project puede contener parámetros de varios controles de recursos para cada proyecto, así como diferentes valores umbral para cada control. Los valores umbral se definen en las cláusulas de acción, que se separan con comas en caso de haber múltiples valores.

1. **Conviértase en root o asuma un rol similar.**
2. **Utilice el comando projmod con las opciones -s y -K para definir los controles de recursos en el proyecto x-files:**

```
# projmod -s -K 'task.max-lwps=(basic,10,none),(privileged,500,deny);
process.max-file-descriptor=(basic,128,deny)' x-files    one line in file
```

Se definen los controles siguientes:

- Un control basic sin acción en el máximo de procesos ligeros por tarea.

- Un control `deny` con privilegios en el máximo de procesos ligeros por tarea. Este control hace que falle la creación de cualquier proceso ligero que supere el máximo, tal como se muestra en el ejemplo de [Cómo definir el número máximo de procesos ligeros para cada tarea de un proyecto \[76\]](#).
- Un límite para los descriptores de archivo por proceso en el nivel `basic`, que hace que falle cualquier llamada `open` que supere el máximo.

3. Visualice la entrada en el archivo utilizando uno de estos métodos:

- Escriba:

```
# projects -l
.
.
.
x-files
    projid : 100
    comment: ""
    users  : (none)
    groups : (none)
    attribs: process.max-file-descriptor=(basic,128,deny)
            task.max-lwps=(basic,10,none),(privileged,500,deny)    one line in file
```

- Escriba:

```
# cat /etc/project
.
.
.
x-files:100:::process.max-file-descriptor=(basic,128,deny);
task.max-lwps=(basic,10,none),(privileged,500,deny)    one line in file
```

Uso del comando `prctl`

Utilice el comando `prctl` para realizar interrogaciones de tiempo de ejecución y modificaciones de los controles de recursos asociados con un proceso, una tarea o un proyecto activos en el sistema. Para obtener más información, consulte la página del comando `man prctl(1)`.

▼ Cómo utilizar el comando `prctl` para ver los valores de control de recursos predeterminados

Debe utilizarse este procedimiento en un sistema en el que no se hayan configurado ni modificado controles de recursos. En el archivo `/etc/system` o la base de datos `project` sólo puede haber entradas no predeterminadas.

- Utilice el comando `prctl` en cualquier proceso, como el shell en ejecución.

```
# prctl $$
process: 3320: bash
NAME      PRIVILEGE      VALUE      FLAG      ACTION      RECIPIENT
process.max-port-events
  privileged 65.5K          -          deny      -
  system    2.15G          max        deny      -
process.max-msg-messages
  privileged 8.19K          -          deny      -
  system    4.29G          max        deny      -
process.max-msg-qbytes
  privileged 64.0KB         -          deny      -
  system    16.0EB         max        deny      -
process.max-sem-ops
  privileged 512            -          deny      -
  system    2.15G          max        deny      -
process.max-sem-nsems
  privileged 512            -          deny      -
  system    32.8K          max        deny      -
process.max-address-space
  privileged 16.0EB         max        deny      -
  system    16.0EB         max        deny      -
process.max-file-descriptor
  basic      256            -          deny      3320
  privileged 65.5K          -          deny      -
  system    2.15G          max        deny      -
process.max-core-size
  privileged 8.00EB         max        deny      -
  system    8.00EB         max        deny      -
process.max-stack-size
  basic      10.0MB         -          deny      3320
  privileged 32.0TB         -          deny      -
  system    32.0TB         max        deny      -
process.max-data-size
  privileged 16.0EB         max        deny      -
  system    16.0EB         max        deny      -
process.max-file-size
  privileged 8.00EB         max        deny,sig=XFSZ -
  system    8.00EB         max        deny      -
process.max-cpu-time
  privileged 18.4Es         inf        signal=XCPU -
  system    18.4Es         inf        none      -
task.max-cpu-time
```

usage	0s			
system	18.4Es	inf	none	-
task.max-processes				
usage	2			
system	2.15G	max	deny	-
task.max-lwps				
usage	3			
system	2.15G	max	deny	-
project.max-contracts				
privileged	10.0K	-	deny	-
system	2.15G	max	deny	-
project.max-locked-memory				
usage	0B			
system	16.0EB	max	deny	-
project.max-port-ids				
privileged	8.19K	-	deny	-
system	65.5K	max	deny	-
project.max-shm-memory				
privileged	510MB	-	deny	-
system	16.0EB	max	deny	-
project.max-shm-ids				
privileged	128	-	deny	-
system	16.8M	max	deny	-
project.max-msg-ids				
privileged	128	-	deny	-
system	16.8M	max	deny	-
project.max-sem-ids				
privileged	128	-	deny	-
system	16.8M	max	deny	-
project.max-crypto-memory				
usage	0B			
privileged	510MB	-	deny	-
system	16.0EB	max	deny	-
project.max-tasks				
usage	2			
system	2.15G	max	deny	-
project.max-processes				
usage	4			
system	2.15G	max	deny	-
project.max-lwps				
usage	11			
system	2.15G	max	deny	-
project.cpu-cap				
usage	0			
system	4.29G	inf	deny	-
project.cpu-shares				
usage	1			
privileged	1	-	none	-
system	65.5K	max	none	-
zone.max-lofi				
usage	0			
system	18.4E	max	deny	-
zone.max-swap				
usage	180MB			

system	16.0EB	max	deny	-
zone.max-locked-memory				
usage	0B			
system	16.0EB	max	deny	-
zone.max-shm-memory				
system	16.0EB	max	deny	-
zone.max-shm-ids				
system	16.8M	max	deny	-
zone.max-sem-ids				
system	16.8M	max	deny	-
zone.max-msg-ids				
system	16.8M	max	deny	-
zone.max-processes				
usage	73			
system	2.15G	max	deny	-
zone.max-lwps				
usage	384			
system	2.15G	max	deny	-
zone.cpu-cap				
usage	0			
system	4.29G	inf	deny	-
zone.cpu-shares				
usage	1			
privileged	1	-	none	-
system	65.5K	max	none	-

▼ Cómo utilizar el comando `prctl` para ver información de un control de recurso

- Visualice el máximo de descriptores de archivo para el shell que está en ejecución.

```
# prctl -n process.max-file-descriptor $$
process: 110453: -sh
NAME    PRIVILEGE    VALUE    FLAG    ACTION    RECIPIENT
process.max-file-descriptor
basic        256        -    deny        11731
privileged  65.5K      -    deny        -
system      2.15G     max    deny
```

▼ Cómo utilizar `prctl` para cambiar un valor temporalmente

Este procedimiento de ejemplo utiliza el comando `prctl` para agregar temporalmente un nuevo valor con privilegios con el fin de denegar el uso de más de tres procesos ligeros por proyecto

para el proyecto `x-files`. El resultado se puede comparar con el resultado de [Cómo definir el número máximo de procesos ligeros para cada tarea de un proyecto \[76\]](#).

1. **Conviértase en root o asuma un rol similar.**
2. **Utilice `newtask` para unir el proyecto `x-files`.**
3. **Utilice el comando `id` con la opción `-p` para verificar que se ha unido el proyecto correcto.**

```
# newtask -p x-files
```

```
# id -p
uid=0(root) gid=1(other) projid=101(x-files)
```

4. **Agregue un nuevo valor con privilegios para `project.max-lwps` que limite el número de procesos ligeros a tres.**

```
# prctl -n project.max-lwps -t privileged -v 3 -e deny -i project x-files
```

5. **Compruebe el resultado.**

```
# prctl -n project.max-lwps -i project x-files
process: 111108: csh
NAME      PRIVILEGE  VALUE  FLAG   ACTION    RECIPIENT
project.max-lwps
  usage           203
  privileged      1000      -    deny
  system          2.15G    max   deny
```

▼ Cómo utilizar `prctl` para reducir el valor de control de recurso

1. **Conviértase en root o asuma un rol similar.**
2. **Utilice el comando `prctl` con la opción `-r` para cambiar el valor inferior del control de recurso `process.max-file-descriptor`.**

```
# prctl -n process.max-file-descriptor -r -v 128 $$
```

▼ Cómo utilizar `prctl` para ver, sustituir y verificar el valor de un control en un proyecto

1. **Conviértase en root o asuma un rol similar.**

2. Visualice el valor de `project.cpu-shares` en el proyecto `group.staff`.

```
# prctl -n project.cpu-shares -i project group.staff
project: 2: group.staff
NAME    PRIVILEGE    VALUE    FLAG    ACTION    RECIPIENT
project.cpu-shares
  usage          1
  privileged      1      -    none
  system        65.5K    max    none
```

3. Sustituya el valor actual `project.cpu-shares1` con el valor 10.

```
# prctl -n project.cpu-shares -v 10 -r -i project group.staff
```

4. Visualice el valor de `project.cpu-shares` en el proyecto `group.staff`.

```
# prctl -n project.cpu-shares -i project group.staff
project: 2: group.staff
NAME    PRIVILEGE    VALUE    FLAG    ACTION    RECIPIENT
project.cpu-shares
  usage          1
  privileged      1      -    none
  system        65.5K    max    none
```

Uso de rctladm

Cómo utilizar rctladm

Utilice el comando `rctladm` para realizar interrogaciones de tiempo de ejecución y modificaciones en el estado global de las funciones de controles de recursos. Para obtener más información, consulte la página del comando `man rctladm(1M)`.

Por ejemplo, puede utilizar el comando `rctladm` con la opción `-e` para permitir el atributo `syslog` global de un control de recurso. Cuando se supera el control, la notificación se registra en el nivel `syslog` especificado. Para activar el atributo `syslog` global de `process.max-file-descriptor`, escriba:

```
# rctladm -e syslog process.max-file-descriptor
```

Cuando se utiliza sin argumentos, el comando `rctladm` muestra los indicadores globales, incluido el indicador de tipo global, para cada control de recurso.

```
# rctladm
process.max-port-events    syslog=off [ deny count ]
process.max-msg-messages   syslog=off [ deny count ]
process.max-msg-qbytes     syslog=off [ deny bytes ]
```

```
process.max-sem-ops      syslog=off [ deny count ]
process.max-sem-nsems    syslog=off [ deny count ]
process.max-address-space syslog=off [ lowerable deny no-signal bytes ]
process.max-file-descriptor syslog=off [ lowerable deny count ]
process.max-core-size    syslog=off [ lowerable deny no-signal bytes ]
process.max-stack-size   syslog=off [ lowerable deny no-signal bytes ]
.
.
.
```

Uso de ipcs

Cómo utilizar ipcs

La utilidad `ipcs` permite mostrar información sobre las funciones de comunicación entre procesos (IPC) activas. Para obtener más información, consulte la página del comando `man ipcs(1)`.

Puede utilizar `ipcs` con la opción `-J` para ver qué límite de proyecto tiene asignado un objeto IPC.

```
# ipcs -J
IPC status from <running system> as of Wed Mar 26 18:53:15 PDT 2003
T      ID      KEY      MODE      OWNER      GROUP      PROJECT
Message Queues:
Shared Memory:
m      3600     0        --rw-rw-rw-  uname     staff      x-files
m      201      0        --rw-rw-rw-  uname     staff      x-files
m      1802     0        --rw-rw-rw-  uname     staff      x-files
m      503      0        --rw-rw-rw-  uname     staff      x-files
m      304      0        --rw-rw-rw-  uname     staff      x-files
m      605      0        --rw-rw-rw-  uname     staff      x-files
m      6        0        --rw-rw-rw-  uname     staff      x-files
m      107     0        --rw-rw-rw-  uname     staff      x-files
Semaphores:
s      0        0        --rw-rw-rw-  uname     staff      x-files
```

Advertencias de capacidad

Una acción global en un control de recurso permite recibir una advertencia de cualquier entidad que se encuentre con un valor de control de recurso configurado con un valor demasiado bajo.

Por ejemplo, supongamos que desea determinar si un servidor Web tiene suficiente CPU para su carga de trabajo típica. Puede analizar los datos `sar` para el tiempo de inactividad de la CPU y

el promedio de carga. También puede examinar los datos de contabilidad ampliada con el fin de determinar el número de procesos simultáneos que se ejecutan para el proceso del servidor Web.

Sin embargo, un modo más sencillo de averiguarlo es colocar el servidor Web en una tarea. Puede definir una acción global, utilizando `syslog`, para recibir una notificación cada vez que una tarea supere el número planificado de procesos ligeros apropiados para la capacidad del equipo.

Para obtener más información, consulte la página del comando `man sar(1)`.

▼ Cómo determinar si un servidor Web tiene asignada suficiente capacidad de CPU

1. Utilice el comando `prctl` para colocar un control de recursos con privilegios (propiedad de root) en las tareas que contienen un proceso `httpd`. Limite el número total de procesos ligeros de cada tarea a 40, y desactive todas las acciones locales.

```
# prctl -n task.max-lwps -v 40 -t privileged -d all `pgrep httpd`
```

2. Active una acción global de registro del sistema en el control de recurso `task.max-lwps`.

```
# rctladm -e syslog task.max-lwps
```

3. Observe si la carga de trabajo se encuentra con el control de recurso.

Si lo hace, verá `/var/adm/messages` del modo siguiente:

```
Jan  8 10:15:15 testmachine unix: [ID 859581 kern.notice]  
NOTICE: privileged rctl task.max-lwps exceeded by task 19
```

BETA
DRAFT

♦ ♦ ♦ C A P Í T U L O 8

Acerca del planificador por reparto equitativo

El análisis de los datos de carga de trabajo puede indicar que una carga de trabajo particular o un grupo de cargas de trabajo está monopolizando los recursos de la CPU. Si estas cargas de trabajo no infringen las restricciones de recursos sobre el uso de la CPU, puede modificar la política de asignación de tiempo de la CPU en el sistema. La clase de planificación de reparto justo descrita en este capítulo permite asignar tiempo de CPU basándose en los recursos compartidos en lugar del esquema de prioridades de la clase de planificación de tiempo compartido (TS).

En este capítulo se tratan los temas siguientes.

- “Introducción al programador” [87]
- “Definición de recurso compartido de CPU” [88]
- “Recursos compartidos de la CPU y estado del proceso” [89]
- “Recursos compartidos de la CPU frente al uso” [89]
- “Ejemplos de recursos compartidos de la CPU” [89]
- “Configuración de FSS” [92]
- “FSS y conjuntos de procesadores” [93]
- “Combinación de FSS con otras clases de programación” [96]
- “Configuración de la clase de programación para el sistema” [97]
- “Clase de programación en un sistema con zonas instaladas” [97]
- “Comandos utilizados con FSS” [97]

Para empezar a utilizar el planificador por reparto equitativo, consulte [Capítulo 9, Administración de las tareas del planificador por reparto equitativo](#).

Introducción al programador

Un trabajo fundamental del sistema operativo es determinar qué procesos obtienen acceso a los recursos del sistema. El programador del proceso, también denominado distribuidor, es la parte del núcleo que controla la asignación de CPU a los procesos. El programador respalda el concepto de las clases de planificación. Cada clase define una política de planificación que se utiliza para planificar procesos en la clase. El programador predeterminado del sistema operativo Oracle Solaris (programador TS) intenta asignar a cada proceso un

acceso relativamente equitativo a las CPU disponibles. Sin embargo, puede especificar que determinados procesos tengan asignados más recursos que otros.

Puede utilizar el *planificador por reparto equitativo* (FSS) para controlar la asignación de recursos disponibles de CPU entre cargas de trabajo, en función de la importancia. Esta importancia se expresa con el número de *recursos compartidos* de la CPU que asigna a cada carga de trabajo.

A cada proyecto se asignan recursos compartidos de CPU para controlar el derecho que tiene el proyecto a los recursos de la CPU. El programador FSS garantiza un reparto justo de los recursos de la CPU entre los proyectos. Se basa en los recursos compartidos asignados, independientemente del número de procesos asociados al proyecto. El programador FSS logra realizar un reparto justo al reducir el derecho de un proyecto a un uso elevado de la CPU y aumentar su derecho a un uso más reducido, de acuerdo con la existencia de otros proyectos.

El FSS consta de un módulo de clases de programación de núcleo y de versiones específicas de las clases de los comandos [dispadm\(1M\)](#) y [prioctl\(1\)](#). Los recursos compartidos del proyecto que utiliza FSS se especifican mediante la propiedad `project.cpu-shares` en la base de datos [project\(4\)](#).

Nota - Si utiliza el control de recurso `project.cpu-shares` en un sistema Oracle Solaris con zonas instaladas, consulte [“Configuración de controles de recursos de zonas”](#) de [“Introducción a Zonas de Oracle Solaris”](#) y [“Uso del programador de reparto justo en un sistema Oracle Solaris con zonas instaladas”](#) de [“Creación y uso de zonas de Oracle Solaris”](#).

Definición de recurso compartido de CPU

El término "recurso compartido" se utiliza para definir una parte de los recursos de la CPU del sistema asignada a un proyecto. Si asigna un número mayor de recursos compartidos de la CPU a un proyecto, en relación con otros proyectos, el proyecto recibirá más recursos de CPU del programador de reparto justo.

Los recursos compartidos de la CPU no son equivalentes a los porcentajes de recursos de la CPU. Los recursos compartidos se utilizan para definir la importancia relativa de las cargas de trabajo con respecto a otras cargas de trabajo. Al asignar recursos compartidos de la CPU a un proyecto, lo más importante no es el número de recursos compartidos que tiene el proyecto. Es más importante conocer cuántos recursos compartidos tiene el proyecto en comparación con otros proyectos. También debe tomar en consideración cuántos de los proyectos competirán con él por los recursos de la CPU.

Nota - Los procesos de proyectos con ningún recurso compartido siempre se ejecutan con la prioridad de sistema inferior (0). Estos procesos sólo se ejecutan cuando los proyectos que no tienen recursos compartidos no utilizan recursos de la CPU.

Recursos compartidos de la CPU y estado del proceso

En el sistema Oracle Solaris, una carga de trabajo del proyecto suele estar compuesta por más de un proceso. Desde el punto de vista del planificador por reparto equitativo, la carga de trabajo de cada proyecto puede encontrarse en un estado *inactivo* o *activo*. Un proyecto se considera inactivo si ninguno de sus procesos utiliza recursos de la CPU. Esto significa que dichos procesos están *en reposo* (a la espera de que finalice la E/S) o detenidos. Un proyecto se considera activo si al menos uno de sus procesos utiliza recursos de la CPU. La suma de recursos compartidos de todos los proyectos activos se utiliza para calcular la parte de recursos de la CPU que se asignará a los proyectos.

Cuando más proyectos pasan a estar activos, se reduce la asignación de CPU de cada proyecto, pero la proporción entre las asignaciones de los diferentes proyectos no cambia.

Recursos compartidos de la CPU frente al uso

La asignación de recursos compartidos no es lo mismo que el uso. Un proyecto que tiene asignado el 50 por ciento de los recursos de la CPU puede tener un promedio de sólo un 20 por ciento de uso de la CPU. Además, los recursos compartidos permiten limitar el uso de la CPU sólo cuando compiten con otros proyectos. Independientemente de lo baja que sea la asignación de un proyecto, siempre recibe el 100 por cien de la capacidad de procesamiento si se ejecuta solo en el sistema. Los ciclos de la CPU disponibles nunca se agotan. Se distribuyen entre proyectos.

La asignación de un recurso compartido reducido a una gran carga de trabajo puede disminuir el rendimiento. No obstante, la carga de trabajo completará la tarea si no se sobrecarga el sistema.

Ejemplos de recursos compartidos de la CPU

Supongamos que tiene un sistema con dos CPU que ejecutan dos cargas de trabajo paralelas denominadas *A* y *B*. Cada carga de trabajo se ejecuta como un proyecto independiente. Los proyectos se han configurado de modo que el proyecto *A* tenga asignados recursos compartidos S_A , y el proyecto *B* tenga asignados recursos compartidos S_B .

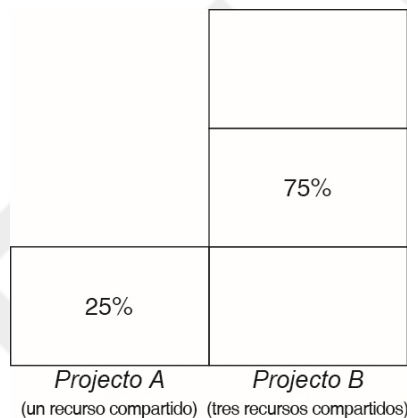
Como media, con el programador TS tradicional, cada carga de trabajo que se ejecuta en el sistema operativo tiene asignada la misma cantidad de recursos de la CPU. Cada carga de trabajo recibiría el 50 por ciento de la capacidad del sistema.

Cuando los proyectos se ejecutan bajo el control del planificador FSS con $S_A=S_B$, estos proyectos también reciben aproximadamente la misma cantidad de recursos de la CPU. Sin embargo, si los proyectos tienen asignada una cantidad diferente de recursos compartidos, sus asignaciones de recursos de la CPU también serán diferentes.

Los tres ejemplos siguientes muestran el funcionamiento de los recursos compartidos con diferentes configuraciones. Estos ejemplos muestran que los recursos compartidos sólo tienen precisión matemática para representar el uso si la demanda cumple o supera los recursos disponibles.

Ejemplo 1: Dos procesos vinculados a la CPU en cada proyecto

Si *A* y *B* tienen dos procesos vinculados a la CPU y $S_A = 1$ y $S_B = 3$, el número total de recursos compartidos es $1 + 3 = 4$. En esta configuración, si hay suficiente demanda de CPU, a los proyectos *A* y *B* se les asigna el 25 por ciento y el 75 por ciento de los recursos de la CPU, respectivamente.



Ejemplo 2: Proyectos que no compiten

Si *A* y *B* sólo tienen *un* proceso vinculado a la CPU y $S_A = 1$ y $S_B = 100$, el número total de recursos compartidos es de 101. Cada proyecto no puede utilizar más de una CPU porque cada uno sólo tiene un proceso en ejecución. Dado que en esta configuración no existe competencia entre los proyectos por los recursos de la CPU, los proyectos *A* y *B* tienen asignado cada uno un 50 por ciento de todos los recursos de la CPU. En esta configuración, los valores de recursos compartidos de la CPU no son relevantes. Las asignaciones de los proyectos serían las mismas (50/50), aunque ambos proyectos no tengan asignado ningún recurso compartido.

50%	50%
(Primera CPU)	(Segunda CPU)
<i>Proyecto A</i>	<i>Proyecto B</i>
(un recurso compartido)	(cien recursos compartidos)

Ejemplo 3: No se puede ejecutar un proyecto

Si *A* y *B* tienen dos procesos vinculados a la CPU, y al proyecto *A* se le asigna 1 recurso compartido y al *B* ninguno, el proyecto *B* no tendrá asignado ningún recurso de la CPU y el proyecto *A* tendrá asignados todos los recursos de la CPU. Los procesos de *B* siempre se ejecutan con una prioridad del sistema de 0, de modo que nunca podrán ejecutarse porque los procesos del proyecto *A* siempre tienen prioridades mayores.

100%	0%
<i>Proyecto A</i>	<i>Proyecto B</i>
(un recurso compartido)	(cero recursos compartidos)

Configuración de FSS

Proyectos y usuarios

Los proyectos son los contenedores de la carga de trabajo en el programador FSS. Los grupos de usuarios asignados a un proyecto se consideran bloques únicos. Tenga en cuenta que puede crear un proyecto con su propio número de recursos compartidos para un usuario individual.

Los usuarios pueden ser miembros de varios proyectos que tengan diferentes números de recursos compartidos asignados. Al mover procesos de un proyecto a otro, pueden asignarse diferentes cantidades de recursos de la CPU a los proyectos.

Para obtener más información sobre la base de datos `project(4)` y los servicios de nombres, consulte “Base de datos `project`” [23].

Configuración de recursos compartidos de la CPU

La configuración de los recursos compartidos de la CPU la lleva a cabo el servicio de nombres como una propiedad de la base de datos `project`.

Cuando la primera tarea (o proceso) que se asocia con un proyecto se crea mediante la función de biblioteca `setproject(3PROJECT)`, el número de recursos compartidos de la CPU definidos como control de recurso `project.cpu-shares` en la base de datos `project` se pasa al núcleo. A los proyectos que no tengan definido el control de recurso `project.cpu-shares` se les asigna un recurso compartido.

En el ejemplo siguiente, esta entrada del archivo `/etc/project` define el número de recursos compartidos para el proyecto `x-files` como 5:

```
x-files:100:::project.cpu-shares=(privileged,5,none)
```

Si modifica el número de recursos compartidos de la CPU asignados a un proyecto en la base de datos cuando hay procesos en ejecución, el número de recursos compartidos para dicho proyecto no se modificará en ese punto. Es preciso reiniciar el proyecto para que el cambio surta efecto.

Si desea cambiar temporalmente el número de recursos compartidos asignados a un proyecto sin modificar los atributos del proyecto en la base de datos `project`, utilice el comando `prctl`. Por ejemplo, para cambiar el valor del control de recurso `project.cpu-shares` del proyecto `x-files` a 3 mientras se ejecutan los procesos asociados con dicho proyecto, escriba:

```
# prctl -r -n project.cpu-shares -v 3 -i project x-files
```

Para obtener más información, consulte la página del comando `man prctl(1)`.

<code>-r</code>	Sustituye el valor actual del control de recurso nombrado.
<code>-n name</code>	Especifica el nombre del control de recurso.
<code>-v val</code>	Especifica el valor del control de recurso.
<code>-i idtype</code>	Especifica el tipo de ID del siguiente argumento.
<code>x-files</code>	Especifica el objeto del cambio. En esta instancia, el proyecto <code>x-files</code> es el objeto.

El proyecto `system` con ID 0 incluye todos los daemons del sistema que se inician mediante secuencias de inicio de tiempo de inicio. `system` puede visualizarse como proyecto en un número ilimitado de recursos compartidos. Esto significa que `system` siempre se planifica en primer lugar, independientemente de cuántos recursos compartidos se asignaron a otros proyectos. Si no desea que el proyecto `system` tenga recursos compartidos ilimitados, puede especificar un número de recursos compartidos para este proyecto en la base de datos `project`.

Como se ha mencionado anteriormente, los procesos que pertenecen a proyectos con ningún recurso compartido siempre tienen la prioridad de sistema cero. Los proyectos con uno o más recursos compartidos se ejecutan con prioridades de uno o más. Por tanto, los proyectos que no tienen ningún recurso compartido sólo se programan cuando hay disponibles recursos de la CPU que no solicite ningún proyecto de recurso compartido que no sea cero.

El número máximo de recursos compartidos que se puede asignar a un proyecto es 65535.

FSS y conjuntos de procesadores

El programador FSS puede utilizarse junto con los conjuntos de procesadores para ofrecer controles más precisos para la asignación de recursos de la CPU a los proyectos que se ejecutan en cada conjunto de procesadores que estarían disponibles únicamente con los conjuntos de procesadores. El programador FSS trata los conjuntos de procesadores como particiones completamente independientes, y cada conjunto de procesadores se controla de modo independiente con respecto a las asignaciones de la CPU.

Las asignaciones de la CPU de los proyectos que se ejecutan en un conjunto de procesadores no se ven afectadas por los recursos de la CPU o la actividad de los proyectos que se ejecutan en otro conjunto de procesadores porque los proyectos no compiten por los mismos recursos. Los proyectos sólo compiten entre sí si se ejecutan en el mismo conjunto de procesadores.

El número de recursos compartidos asignados a un proyecto se aplica a todo el sistema. Al margen del conjunto de procesadores en el que se ejecute, cada parte de un proyecto tiene asignada la misma cantidad de recursos compartidos.

Cuando se utilizan conjuntos de procesadores, las asignaciones de la CPU del proyecto se calculan para los proyectos activos que se ejecutan en cada conjunto de procesadores.

Las particiones de proyectos que se ejecutan en diferentes conjuntos de procesadores pueden tener diferentes asignaciones de la CPU. La asignación de CPU para cada partición de proyecto de un conjunto de procesadores depende únicamente de las asignaciones de otros proyectos que se ejecutan en el mismo conjunto de procesadores.

El rendimiento y la disponibilidad de las aplicaciones que se ejecutan dentro de los límites de sus conjuntos de procesadores no se ven afectados por la introducción de nuevos conjuntos de procesadores. Las aplicaciones tampoco se ven afectadas por los cambios realizados en las asignaciones de recursos compartidos de los proyectos que se ejecutan en otros conjuntos de procesadores.

Los conjuntos de procesadores vacíos (conjuntos que no contienen procesadores) o los conjuntos de procesadores que no tienen procesos vinculados no tienen ninguna repercusión en el comportamiento del programador FSS.

FSS y ejemplos de conjuntos de procesadores

Supongamos que un servidor con ocho CPU ejecuta varias aplicaciones vinculadas a CPU en los proyectos A, B y C. El proyecto A tiene asignado un recurso compartido, el proyecto B tiene asignados dos recursos compartidos y el proyecto C tiene asignados tres recursos compartidos.

El proyecto A sólo se está ejecutando en el conjunto de procesadores 1. El proyecto B se está ejecutando en el conjunto de procesadores 1 y 2. El proyecto C se está ejecutando en el conjunto de procesadores 1, 2 y 3. Se da por supuesto que cada proyecto tiene suficientes procesos para utilizar toda la energía de la CPU disponible. De este modo, cada conjunto de procesadores siempre compite por los recursos de la CPU.

Proyecto A 16,66% (1/6)	Proyecto B 40% (2/5)	Proyecto C 100% (3/3)
Proyecto B 33,33% (2/6)		
Proyecto C 50% (3/6)	Proyecto C 60% (3/5)	
Conjunto de procesadores n.º 1 2 CPU 25% del sistema	Conjunto de procesadores n.º 2 4 CPU 50% del sistema	Conjunto de procesadores n.º 3 2 CPU 25% del sistema

En la tabla siguiente se muestran las asignaciones de CPU totales del proyecto para un sistema de ese tipo.

Proyecto	Asignación
Proyecto A	$4\% = (1/6 \times 2/8)_{\text{pset1}}$
Proyecto B	$28\% = (2/6 \times 2/8)_{\text{pset1}} + (2/5 \times 4/8)_{\text{pset2}}$
Proyecto C	$67\% = (3/6 \times 2/8)_{\text{pset1}} + (3/5 \times 4/8)_{\text{pset2}} + (3/3 \times 2/8)_{\text{pset3}}$

Estos porcentajes no coinciden con las cantidades correspondientes de recursos compartidos de la CPU que se asignan a los proyectos. Sin embargo, en cada conjunto de procesadores, la asignación de CPU por proyecto es proporcional a sus respectivos recursos compartidos.

En el mismo sistema *sin* conjuntos de procesadores, la distribución de los recursos compartidos de la CPU sería diferente, tal como se muestra en la tabla siguiente.

Proyecto	Asignación
Proyecto A	16,66% = (1/6)
Proyecto B	33,33% = (2/6)
Proyecto C	50% = (3/6)

Combinación de FSS con otras clases de programación

De manera predeterminada, la clase de programación FSS utiliza el mismo rango de prioridades (de 0 a 59) que las clases de programación de tiempo compartido (TS), interactivas (IA) y de prioridad fija (FX). Por tanto, debe evitar tener procesos de estas clases de planificación que compartan *el mismo* conjunto de procesadores. Una combinación de los procesos en FSS y las clases TS, IA y FX podría ocasionar un comportamiento de programación inesperado.

Gracias al uso de conjuntos de procesadores, puede combinar TS, IA y FX con FSS en un sistema. Sin embargo, todos los procesos que se ejecutan en cada conjunto de procesadores deben estar en *una* clase de planificación, de modo que no compitan por las mismas CPU. En concreto, no debe utilizarse el programador FX junto con la clase de programación de FSS a menos que se utilicen conjuntos de procesadores. Esta acción evita que las aplicaciones de la clase FX utilicen prioridades lo suficientemente altas para quedarse sin aplicaciones en la clase FSS.

Puede combinar procesos en las clases TS e IA en el mismo conjunto de procesadores, o en el mismo sistema sin conjuntos de procesadores.

El sistema Oracle Solaris también ofrece un programador en tiempo real (RT) a los usuarios con privilegios root. De modo predeterminado, la clase de programación RT utiliza prioridades del sistema en un intervalo diferente (normalmente de 100 a 159) que FSS. Dado que RT y FSS utilizan intervalos de prioridades *inconexos* o no superpuestos, FSS puede coexistir con la clase de programación RT en el mismo conjunto de procesadores. Sin embargo, la clase de programación FSS no tiene ningún control sobre los procesos que se ejecutan en la clase RT.

Por ejemplo, en un sistema de cuatro procesadores, un proceso RT de un único subproceso puede consumir un procesador completo si el proceso está vinculado a la CPU. Si el sistema también ejecuta FSS, los procesos regulares del usuario compiten por las tres CPU restantes que no utiliza el proceso RT. Tenga en cuenta que es posible que el proceso RT no utilice la CPU de forma continuada. Cuando el proceso RT está inactivo, FSS utiliza los cuatro procesadores.

Puede escribir el comando siguiente para averiguar en qué clases de programación se ejecutan los conjuntos de procesadores y asegurarse de que cada conjunto de procesadores esté configurado para ejecutarse en procesos TS, IA, FX o FSS.

```
$ ps -ef -o pset,class | grep -v CLS | sort | uniq
1 FSS
1 SYS
2 TS
```


2 RT
3 FX

Configuración de la clase de programación para el sistema

Para definir la clase de programación predeterminada para el sistema, consulte [Cómo convertir FSS en la clase de programador predeterminada \[102\]](#), “Uso del programador de reparto justo en un sistema Oracle Solaris con zonas instaladas” de “Creación y uso de zonas de Oracle Solaris ” y `dispadmin(1M)`. Para mover los procesos en ejecución a una clase de programación diferente, consulte “Configuración de FSS” [101] y `priocntl(1)`.

Clase de programación en un sistema con zonas instaladas

Las zonas no globales utilizan la clase de programación predeterminada para el sistema. Si el sistema se actualiza con una nueva configuración de clase de programación predeterminada, las zonas no globales obtienen la nueva configuración al iniciarse o reiniciarse.

En este caso, se recomienda configurar FSS como la clase de programación predeterminada del sistema con el comando `dispadmin`. De este modo, todas las zonas se beneficiarán de un reparto justo de los recursos de la CPU del sistema. Para obtener más información sobre la clase de programación cuando las zonas están en uso, consulte “Uso del programador de reparto justo en un sistema Oracle Solaris con zonas instaladas” de “Creación y uso de zonas de Oracle Solaris ”.

Para obtener información sobre cómo mover procesos en ejecución a una clase de programación distinta sin cambiar la clase de programación predeterminada ni reiniciar, consulte la página del comando `man priocntl(1)`.

Comandos utilizados con FSS

Los comandos que se muestran en la tabla siguiente proporcionan la interfaz administrativa principal para el programador de reparto justo.

Referencia de comando	Descripción
<code>priocntl(1)</code>	Muestra o configura los parámetros de programación de los procesos especificados, y mueve los procesos en ejecución a una clase de programación diferente.
<code>ps(1)</code>	Enumera la información sobre los procesos en ejecución, e identifica en qué clases de programación se ejecutan los conjuntos de procesadores.

Referencia de comando	Descripción
dispadmin(1M)	Enumera los programadores disponibles en el sistema. Define el programador predeterminado para el sistema. También se utiliza para examinar y ajustar el valor de tiempo del programador FSS.
FSS(7)	Describe el programador de reparto justo (FSS).

♦ ♦ ♦ C A P Í T U L O 9

Administración de las tareas del planificador por reparto equitativo

En este capítulo se describe cómo utilizar el programador de reparto justo (FSS).

Para ver una descripción general de FSS, consulte [Capítulo 8, Acerca del planificador por reparto equitativo](#). Para obtener información sobre la clase de programación cuando hay zonas en uso, consulte [“Programador de reparto justo en un sistema con zonas instaladas”](#) de [“Creación y uso de zonas de Oracle Solaris”](#).

Administración del mapa de tareas del planificador de reparto equitativo

Tarea	Descripción	Para obtener información
Supervisar uso de CPU.	Supervisa el uso de la CPU de los proyectos, y los proyectos en los conjuntos de procesadores.	“Supervisión de FSS” [100]
Definir la clase predeterminada del programador.	Convierte un programador como FSS en el programador predeterminado para el sistema.	Cómo convertir FSS en la clase de programador predeterminada [102]
Mover procesos en ejecución de una clase de programador a otra, como la clase FSS.	Mueve manualmente los procesos de una clase de planificación a otra sin cambiar la clase de planificación predeterminada ni reiniciar.	Cómo mover manualmente los procesos de la clase TS a la clase FSS [102]
Mover todos los procesos en ejecución de todas las clases de planificación a una clase de planificación diferente, como la clase FSS.	Mueve manualmente los procesos de todas las clases de planificación a otra clase de planificación sin cambiar la clase de planificación predeterminada ni reiniciar.	Cómo mover manualmente los procesos de todas las clases de usuario a la clase FSS [103]
Mover los procesos de un proyecto a una clase de planificación diferente, como la clase FSS.	Mueve manualmente los procesos de un proyecto de su clase de planificación actual a una clase de planificación diferente.	Cómo mover manualmente los procesos de un proyecto a la clase FSS [103]
Examinar y ajustar parámetros FSS.	Configura el valor de tiempo del programador. El <i>valor de tiempo</i> es la cantidad de tiempo durante la que	“Cómo ajustar los parámetros del programador” [103]

Tarea	Descripción	Para obtener información
	puede ejecutarse un subproceso antes de renunciar al procesador.	

Supervisión de FSS

Puede utilizar el comando `prstat` descrito en la página del comando `man prstat(1M)` para supervisar el uso de la CPU que hacen los proyectos activos.

Puede utilizar los datos de contabilidad ampliada para las tareas con el fin de obtener estadísticas por proyecto sobre la cantidad de recursos de la CPU que se consumen durante períodos más prolongados. Para obtener más información, consulte [Capítulo 4, Acerca de la contabilidad ampliada](#).

▼ Cómo supervisar el uso que hacen los proyectos de la CPU del sistema

- Para supervisar el uso de la CPU que hacen los proyectos que se ejecutan en el sistema, utilice el comando `prstat` con la opción `-J`.

```
# prstat -J
  PID USERNAME  SIZE  RSS STATE PRI NICE   TIME CPU PROCESS/NLWP
  5107 root      4556K 3268K cpu0  59   0   0:00:00 0.0% prstat/1
  4570 root         83M   47M sleep 59   0   0:00:25 0.0% java/13
  5105 bobbyc    3280K 2364K sleep 59   0   0:00:00 0.0% su/1
  5106 root     3328K 2580K sleep 59   0   0:00:00 0.0% bash/1
    5 root         0K    0K sleep 99  -20   0:00:14 0.0% zpool-rpool/138
  333 daemon    7196K 2896K sleep 59   0   0:00:07 0.0% rcapd/1
   51 netcfg    4436K 3460K sleep 59   0   0:00:01 0.0% netcfgd/5
 2685 root     3328K 2664K sleep 59   0   0:00:00 0.0% bash/1
  101 netadm    4164K 2824K sleep 59   0   0:00:01 0.0% ipmgmt/6
  139 root     6940K 3016K sleep 59   0   0:00:00 0.0% syseventd/18
 5082 bobbyc   2236K 1700K sleep 59   0   0:00:00 0.0% csh/1
   45 root      15M  7360K sleep 59   0   0:00:01 0.0% dlmgmt/7
   12 root       23M   22M sleep 59   0   0:00:45 0.0% svc.configd/22
   10 root       15M   13M sleep 59   0   0:00:05 0.0% svc.startd/19
  337 netadm   6768K 5620K sleep 59   0   0:00:01 0.0% nwamd/9

 PROJID  NPROC  SWAP  RSS MEMORY   TIME CPU PROJECT
    1      6   25M   18M   0.9%   0:00:00 0.0% user.root
    0     73  479M  284M   14%   0:02:31 0.0% system
    3      4   28M   24M   1.1%   0:00:26 0.0% default
   10      2   14M  7288K   0.3%   0:00:00 0.0% group.staff
```

Total: 85 processes, 553 lwps, load averages: 0.00, 0.00, 0.00

▼ Cómo supervisar el uso de la CPU que hacen los proyectos en los conjuntos de procesadores

- Para supervisar el uso que hacen de la CPU los proyectos que se encuentran en una lista de conjuntos de procesadores, escriba:

```
% prstat -J -C pset-list
```

donde *pset-list* es una lista de los ID de conjuntos de procesadores, separada por comas.

Configuración de FSS

Los mismos comandos que se utilizan con otras clases de programación en el sistema Oracle Solaris se pueden utilizar con FSS. Puede definir la clase de programador, configurar los parámetros ajustables del programador y establecer las propiedades de los procesos individuales.

Tenga en cuenta que puede utilizar `svcadm restart` para reiniciar el servicio del programador. Para obtener más información, consulte [svcadm\(1M\)](#).

Enumeración de clases de programador en el sistema

Para mostrar las clases de programador en el sistema, utilice el comando `dispadmin` con la opción `-l`.

```
$ dispadmin -l
CONFIGURED CLASSES
=====

SYS      (System Class)
TS        (Time Sharing)
SDC       (System Duty-Cycle Class)
FSS       (Fair Share)
FX        (Fixed Priority)
IA        (Interactive)
```

▼ **Cómo convertir FSS en la clase de programador predeterminada**

Para que la asignación de recursos compartidos de la CPU surta efecto, FSS debe ser el programador predeterminado del sistema.

El uso de una combinación de los comandos `priocntl` y `dispadm` garantiza que FSS pasará a ser de inmediato el programador predeterminado y también tras el reinicio.

1. **Conviértase en root o asuma un rol similar.**
2. **Configure FSS como el programador predeterminado del sistema.**

```
# dispadm -d FSS
```

Este cambio surtirá efecto la próxima vez que reinicie. Tras reiniciar, todos los procesos del sistema se ejecutarán en la clase de planificación FSS.

3. **Active esta configuración inmediatamente, sin reiniciar.**

```
# priocntl -s -c FSS -i all
```

▼ **Cómo mover manualmente los procesos de la clase TS a la clase FSS**

Puede mover manualmente los procesos de una clase de planificación a otra sin cambiar la clase de planificación predeterminada ni reiniciar. Este procedimiento muestra cómo mover manualmente los procesos de la clase de planificación TS a la clase de planificación FSS.

1. **Conviértase en root o asuma un rol similar.**
2. **Mueva el proceso `init` (pid 1) a la clase de planificación FSS.**

```
# priocntl -s -c FSS -i pid 1
```

3. **Mueva todos los procesos de la clase de planificación TS a la clase de planificación FSS.**

```
# priocntl -s -c FSS -i class TS
```

Nota - Tras el reinicio, todos los procesos volverán a ejecutarse en la clase de planificación TS.

▼ Cómo mover manualmente los procesos de todas las clases de usuario a la clase FSS

Puede utilizar una clase predeterminada que no sea TS. Por ejemplo, el sistema puede ejecutar un entorno de ventanas que utilice una clase IA de modo predeterminado. Puede mover todos los procesos a la clase de planificación FSS sin cambiar la clase de planificación predeterminada ni reiniciar.

1. **Conviértase en root o asuma un rol similar.**
2. **Mueva el proceso `init` (pid 1) a la clase de planificación FSS.**
3. **Mueva todos los procesos de sus clases de planificación actuales a la clase de planificación FSS.**

```
# priocntl -s -c FSS -i pid 1
```

```
# priocntl -s -c FSS -i all
```

Nota - De nuevo, todos los procesos se ejecutarán en la clase de planificación predeterminada después de reiniciar.

▼ Cómo mover manualmente los procesos de un proyecto a la clase FSS

Puede mover manualmente los procesos de un proyecto de su clase de planificación actual a la clase de planificación FSS.

1. **Conviértase en root o asuma un rol similar.**
2. **Mueva los procesos que se ejecutan en el ID de proyecto `10` a la clase de planificación FSS.**

```
# priocntl -s -c FSS -i projid 10
```

De nuevo, los procesos del proyecto se ejecutarán en la clase de planificación predeterminada después de reiniciar.

Cómo ajustar los parámetros del programador

Puede utilizar el comando `dispadmin` para mostrar o cambiar los parámetros del programador de procesos mientras el sistema está en ejecución. Por ejemplo, puede utilizar `dispadmin` para

examinar y ajustar el valor de tiempo del programador FSS. El *valor de tiempo* es la cantidad de tiempo durante la que puede ejecutarse un subproceso antes de renunciar al procesador.

Para ver el valor de tiempo actual para el programador FSS mientras se ejecuta el sistema, escriba:

```
$ dispadmin -c FSS -g
#
# Fair Share Scheduler Configuration
#
RES=1000
#
# Time Quantum
#
QUANTUM=110
```

Si utiliza la opción -g, también puede utilizar la opción -r para especificar la resolución que se utiliza para imprimir valores de tiempo. Si no se especifica ninguna resolución, los valores de tiempo se muestran de forma predeterminada en milisegundos.

```
$ dispadmin -c FSS -g -r 100
#
# Fair Share Scheduler Configuration
#
RES=100
#
# Time Quantum
#
QUANTUM=11
```

Para definir los parámetros de planificación para la clase de planificación FSS, utilice `dispadmin -s`. Los valores de *archivo* deben tener el formato de la opción -g. Estos valores sobrescriben los valores actuales del núcleo. Escriba lo siguiente:

```
$ dispadmin -c FSS -s file
```


♦ ♦ ♦ 10 C A P Í T U L O 1 0

Acerca del control de memoria física mediante el daemon de limitación de recursos

El daemon de límite de recursos `rcapd` permite regular el consumo de memoria física por parte de los procesos que se ejecutan en los proyectos que tienen definidos límites de recursos. Si está ejecutando zonas en el sistema, puede utilizar `rcapd` desde la zona global para regular el consumo de memoria física en las zonas no globales. Consulte [“Control de memoria física y recurso capped-memory”](#) de [“Introducción a Zonas de Oracle Solaris”](#).

En este capítulo se cubren los temas siguientes.

- [“Introducción al daemon de límite de recursos”](#) [105]
- [“Funcionamiento de los límites de recursos”](#) [106]
- [“Atributo para limitar el uso de memoria física para proyectos”](#) [106]
- [“Configuración de `rcapd`”](#) [107]
- [“Supervisión del uso de recursos con `rcapstat`”](#) [111]
- [“Comandos utilizados con `rcapd`”](#) [112]

Para conocer los procedimientos que utilizan la utilidad `rcapd`, consulte [Capítulo 11, Administración de las tareas del daemon de limitación de recursos](#).

Introducción al daemon de límite de recursos

Un *límite* de recursos es un límite superior colocado para el consumo de un recurso, como la memoria física. Se admite el uso de límites de memoria física por proyecto.

El daemon de limitación de recursos y sus utilidades asociadas proporcionan mecanismos para la administración y la aplicación de límites de memoria física.

Al igual que el control de recursos, el límite de recursos se puede definir utilizando los atributos de entradas de proyecto en la base de datos `project`. Sin embargo, mientras que los controles de recursos los aplica de forma sincronizada el núcleo, los límites de recursos los aplica el daemon de límite de recursos en el nivel del usuario y sin sincronización. En el caso de la aplicación asíncrona, tiene lugar un breve retardo como consecuencia del intervalo de muestreo que utiliza el daemon.

Para obtener más información sobre `rcapd`, consulte la página del comando `man rcapd(1M)`.
Para obtener información sobre los proyectos y la base de datos `project`, consulte la página del comando `man Capítulo 2, Acerca de los proyectos y las tareas` and the `project(4)`.
Para obtener información sobre los controles de recursos, consulte `Capítulo 6, Acerca de los controles de recursos`.

Funcionamiento de los límites de recursos

El daemon muestrea de forma repetida el uso de los recursos de los proyectos que tienen límites de memoria física. El administrador especifica el intervalo de muestreo que utiliza el daemon. Consulte [“Cómo determinar los intervalos de muestra” \[111\]](#) para obtener información adicional. Cuando el uso de la memoria física del sistema supera el umbral para la aplicación del límite, y además se dan otras condiciones, el daemon emprende las acciones necesarias para reducir el consumo de recursos de los proyectos con límites de memoria a niveles iguales o inferiores a los establecidos por los límites.

El sistema de memoria virtual divide la memoria física en segmentos conocidos como páginas. Las páginas son la unidad fundamental de memoria física del subsistema de gestión de memoria de Oracle Solaris. Para leer datos de un archivo en la memoria, el sistema de memoria virtual lee una página cada vez, o *carga la página* en un archivo. Para reducir el consumo de recursos, el daemon puede *extraer*, o reasignar, las páginas poco utilizadas a un dispositivo de intercambio, que es un área fuera de la memoria física.

El daemon administra la memoria física regulando el tamaño del conjunto residente de la carga de trabajo de un proyecto para adaptarlo al tamaño de su conjunto de trabajo. El conjunto residente es el conjunto de páginas que residen en la memoria física. El conjunto de trabajo es el conjunto de páginas que utiliza la carga de trabajo de forma activa durante su ciclo de procesamiento. El conjunto de trabajo cambia con el tiempo, en función del modo de funcionamiento del proceso y el tipo de datos que se procesan. Se recomienda que cada carga de trabajo tenga acceso a la suficiente memoria física para permitir que su conjunto de trabajo permanezca como residente. Sin embargo, el conjunto de trabajo también puede incluir el uso de almacenamiento de disco secundario para contener la memoria que no quepa en la memoria física.

Sólo puede ejecutarse una instancia de `rcapd` en un momento preciso.

Atributo para limitar el uso de memoria física para proyectos

Para definir un límite de recurso de memoria física para un proyecto, establezca un límite en el tamaño de juego residente (RSS) mediante la agregación de este atributo a la entrada de la base de datos `project`:

`rcap.max-rss` La cantidad total de memoria física, en bytes, disponible para los procesos del proyecto.

Por ejemplo, la siguiente línea del archivo `/etc/project` define un límite RSS de 10 gigabytes para un proyecto denominado `db`.

```
db:100::db,root::rcap.max-rss=10737418240
```

Nota - El sistema puede redondear el valor límite especificado a un tamaño de página.

Además, puede utilizar el comando `projmod` para configurar el atributo `rcap.max-rss` en el archivo `/etc/project`.

Para obtener más información, consulte la sección de configuración de límite en el tamaño de conjunto residente.

Configuración de rcapd

El comando `rcapadm` se utiliza para configurar el daemon de limitación de recursos. Puede realizar las siguientes acciones:

- Definir el valor umbral para la aplicación del límite
- Establecer intervalos para las operaciones realizadas por `rcapd`
- Activar o desactivar los límites de recursos
- Mostrar el estado actual del daemon de límite de recursos configurado

Para configurar el daemon, debe ser el usuario `root` o tener los derechos administrativos necesarios.

Los cambios de configuración se pueden incorporar en `rcapd` de acuerdo con el intervalo de configuración (consulte [“Intervalos de funcionamiento de rcapd” \[110\]](#)) o según se solicite enviando `SIGHUP` (consulte la página del comando `man kill(1)`).

Si se utiliza sin argumentos, `rcapadm` muestra el estado actual del daemon de límite de recursos si se ha configurado.

En las subsecciones siguientes, se describe la aplicación del límite, los valores de límite y los intervalos de funcionamiento de `rcapd`.

Uso del daemon de límite de recursos en un sistema con zonas instaladas

Puede controlar el uso del tamaño del juego residente (RSS) de una zona configurando el recurso capped-memory cuando configura la zona. Para obtener más información, consulte [“Control de memoria física y recurso capped-memory”](#) de [“Introducción a Zonas de Oracle Solaris”](#). Para utilizar el recurso capped-memory, el paquete resource-cap debe estar instalado en la zona global. Puede ejecutar el comando rcapd en una zona, incluida la zona global, para aplicar límites de memoria a los proyectos de dicha zona.

Puede establecer un límite temporal para la cantidad máxima de memoria que puede consumir una zona especificada, hasta la próxima vez que se reinicie el sistema. Consulte [Cómo especificar un límite de recursos temporal para una zona \[119\]](#).

Si utiliza rcapd en una zona para regular el consumo de memoria física por parte de los procesos que se ejecutan en proyectos que tienen límites de recursos definidos, debe configurar el daemon en dichas zonas.

Al seleccionar límites de memoria para las aplicaciones en diferentes zonas, normalmente no es necesario que las aplicaciones residan en zonas distintas. Una excepción a esta norma son los servicios por zona. Dichos servicios consumen memoria. Este consumo de memoria debe tenerse en cuenta a la hora de determinar la cantidad de memoria física para un sistema, así como los límites de memoria.

Umbral de aplicación de límite de memoria

El *umbral de aplicación de límite de memoria* es el porcentaje de utilización de la memoria física del sistema que activa la aplicación del límite. Cuando el sistema supera esta utilización, se aplican los límites. La memoria física que utilizan las aplicaciones y el núcleo se incluye en este porcentaje. El porcentaje de utilización determina el modo en que se aplican los límites de memoria.

Para aplicar los límites, puede extraer memoria de las cargas de trabajo del proyecto.

- La memoria puede extraerse para reducir el tamaño de la parte de memoria que supera el límite para una carga de trabajo determinada.
- La memoria puede extraerse para reducir la proporción de memoria física utilizada que supera el umbral de aplicación del límite de memoria en el sistema.

Se permite una carga de trabajo para utilizar la memoria física hasta su límite. Una carga de trabajo puede utilizar memoria adicional siempre que el uso de memoria del sistema esté por debajo del umbral de aplicación del límite de la memoria.

Para definir el valor para la aplicación del límite, consulte [Cómo establecer el umbral de aplicación del límite de memoria \[117\]](#).

Determinación de valores límite

Si un límite de proyecto está configurado con un valor demasiado bajo, es posible que no haya suficiente memoria para que la carga de trabajo siga funcionando de forma eficaz en condiciones normales. La paginación que se produce porque la carga de trabajo requiere más memoria afecta negativamente al rendimiento del sistema.

Los proyectos que tienen límites demasiado elevados pueden consumir la memoria física disponible antes de alcanzar los límites. En ese caso, el núcleo administra de forma eficaz la memoria física, no el comando rcapd.

Tenga en cuenta los siguientes aspectos a la hora de determinar los límites de los proyectos.

Impacto en el sistema de E/S

El daemon puede intentar reducir el uso de memoria física de la carga de trabajo de un proyecto siempre que el uso muestreado supere el límite del proyecto. Durante la aplicación del límite, se utilizan los dispositivos de intercambio y otros dispositivos que contienen archivos que ha asignado la carga de trabajo. El rendimiento de los dispositivos de intercambio es un factor crítico a la hora de determinar el rendimiento de una carga de trabajo que excede su límite de forma rutinaria. La ejecución de la carga de trabajo es similar a su ejecución en un equipo con la misma cantidad de memoria física que el límite de la carga de trabajo.

Impacto en el uso de CPU

El uso de CPU del daemon varía según el número de procesos de las cargas de trabajo del proyecto que limita y el tamaño de los espacios de direccionamiento de las cargas de trabajo.

Una pequeña parte del tiempo de CPU del daemon se dedica a muestrear el uso de cada carga de trabajo. Agregar procesos a las cargas de trabajo aumenta el tiempo que se dedica a muestrear el uso.

Otra parte del tiempo de CPU del daemon se dedica a aplicar los límites cuando se superan. El tiempo dedicado es proporcional a la cantidad de memoria virtual asignada. El tiempo de CPU dedicado aumenta o disminuye según los cambios correspondientes en el tamaño total del espacio de direccionamiento de la carga de trabajo. Esta información se registra en la columna `vm` de la salida de `rcapstat`. Para obtener más información, consulte [“Supervisión del uso de recursos con rcapstat” \[111\]](#) y la página del comando `man rcapstat(1)`.

Informes de memoria compartida

El daemon rcapd informa al RSS sobre las páginas de memoria que se comparten con otros procesos o que se asignan varias veces en un mismo proceso, a modo de promedio bastante preciso. Si hay procesos de distintos proyectos que comparten la misma memoria, dicha memoria se cuenta con el total del RSS para todos los proyectos que comparten la memoria.

El promedio se utiliza con cargas de trabajo como bases de datos, que utilizan una gran cantidad de memoria compartida. Para las cargas de trabajo de bases de datos, también puede probar con un uso habitual del proyecto para definir un valor límite inicial apropiado utilizando el resultado de las opciones -J o -Z del comando `prstat`. Para obtener más información, consulte la página del comando `man prstat(1M)`.

Intervalos de funcionamiento de rcapd

Puede configurar los intervalos para las operaciones periódicas que lleve a cabo `rcapd`.

Todos los intervalos se especifican en segundos. En la tabla siguiente se describen las operaciones `rcapd` y sus valores de intervalo predeterminados.

Operación	Valor de intervalo predeterminado en segundos	Descripción
scan	15	Número de segundos entre exploraciones para los procesos que se han incorporado o extraído de una carga de trabajo del proyecto. El valor mínimo es de 1 segundo.
sample	5	Número de segundos entre muestreos del tamaño del conjunto residente y las aplicaciones de límites subsiguientes. El valor mínimo es de 1 segundo.
report	5	Número de segundos entre actualizaciones de las estadísticas de paginación. Si se configura como 0, las estadísticas no se actualizan y la salida de <code>rcapstat</code> no es actual.
config	60	Número de segundos entre reconfiguraciones. En un evento de reconfiguración, <code>rcapadm</code> lee el archivo de configuración para detectar las actualizaciones y busca en la base de datos <code>project</code> los límites de proyectos nuevos o revisados. El envío de <code>SIGHUP</code> a <code>rcapd</code> provoca una reconfiguración inmediata.

Para configurar los intervalos, consulte [Cómo configurar intervalos de funcionamiento \[117\]](#).

Cómo determinar los intervalos de exploración de rcapd

El intervalo de análisis controla la frecuencia con la que rcapd busca procesos nuevos. En los sistemas con múltiples procesos en ejecución, la exploración de la lista requiere más tiempo, de modo que quizá sea recomendable alargar el intervalo con el fin de reducir el tiempo de CPU global dedicado. Sin embargo, el intervalo de exploración también representa la cantidad de tiempo mínima durante la que debe existir un proceso para asignarse a una carga de trabajo limitada. Si hay cargas de trabajo que ejecutan múltiples procesos cortos, rcapd podría no atribuir los procesos a una carga de trabajo si se prolonga el intervalo de exploración.

Cómo determinar los intervalos de muestra

El intervalo de muestra configurado con rcapadm es la menor cantidad de tiempo que rcapd espera entre una muestra del uso de una carga de trabajo y la aplicación del límite si se excede. Si reduce este intervalo, de los casos rcapd aplicará los límites con mayor frecuencia, lo que posiblemente generará una mayor E/S debido a la paginación. Sin embargo, un intervalo de muestra más breve también puede disminuir el impacto que podría tener un aumento repentino del uso de memoria física de una carga de trabajo específica en otras cargas de trabajo. Se estrecha la ventana entre muestreos, en la que la carga de trabajo puede consumir memoria no afectada y posiblemente obtener memoria de otras cargas de trabajo limitadas.

Si el intervalo de muestra especificado como rcapstat es más breve que el intervalo especificado como rcapd con rcapadm, la salida de algunos intervalos puede ser cero. Esta situación tiene lugar porque rcapd no actualiza las estadísticas con mayor frecuencia que el intervalo especificado con rcapadm. El intervalo especificado con rcapadm es independiente del intervalo de muestreo que utiliza rcapstat.

Supervisión del uso de recursos con rcapstat

Utilice rcapstat para supervisar el uso de recursos de proyectos limitados. Para ver un ejemplo del informe rcapstat, consulte [“Creación de informes con rcapstat” \[119\]](#).

Puede configurar el intervalo de muestreo para el informe y especificar el número de veces que se repiten las estadísticas.

<i>interval</i>	Especifica el intervalo de muestreo en segundos. El intervalo predeterminado es de 5 segundos.
<i>count</i>	Especifica el número de veces que se repiten las estadísticas. De modo predeterminado, rcapstat registra las estadísticas hasta que se recibe una señal de finalización o hasta que termina el proceso de rcapd.

Las estadísticas de paginación del primer informe que emite `rcapstat` muestran la actividad desde el inicio del daemon. Los informes subsiguientes reflejan la actividad desde que se emitió el último informe.

La tabla siguiente define las cabeceras de columna de un informe `rcapstat`.

Cabeceras de columna <code>rcapstat</code>	Descripción
<code>id</code>	El ID de proyecto del proyecto limitado.
<code>project</code>	El nombre de proyecto.
<code>nproc</code>	El número de procesos del proyecto.
<code>vm</code>	Cantidad total de memoria virtual que utilizan los procesos del proyecto, incluidos todos los archivos y dispositivos asignados, en kilobytes (K), megabytes (M) o gigabytes (G).
<code>rss</code>	Cantidad estimada de tamaño del conjunto residente total (RSS) de los procesos del proyecto, en kilobytes (K), megabytes (M) o gigabytes (G), que no representan páginas compartidas.
<code>cap</code>	El límite de RSS definido para el proyecto. Consulte “Atributo para limitar el uso de memoria física para proyectos” [106] o la página del comando <code>man rcapd(1M)</code> para obtener información sobre cómo especificar límites de memoria.
<code>at</code>	Cantidad total de memoria que <code>rcapd</code> intenta extraer desde la última muestra de <code>rcapstat</code> .
<code>avgat</code>	Promedio de memoria que <code>rcapd</code> intenta extraer durante cada ciclo de muestra desde la última muestra de <code>rcapstat</code> . La frecuencia a la que el RSS de obtención de muestras <code>rcapd</code> puede establecerse con <code>rcapadm</code> . Consulte “Intervalos de funcionamiento de rcapd” [110] .
<code>pg</code>	Cantidad total de memoria que <code>rcapd</code> extrae correctamente desde la última muestra de <code>rcapstat</code> .
<code>avgpg</code>	Promedio estimado de la cantidad de memoria que <code>rcapd</code> extrae correctamente durante cada ciclo de muestra desde la última muestra de <code>rcapstat</code> . La frecuencia a la que los tamaños RSS de procesos de muestras <code>rcapd</code> pueden establecerse con <code>rcapadm</code> . Consulte “Intervalos de funcionamiento de rcapd” [110] .

Comandos utilizados con rcapd

Referencia de comando	Descripción
rcapstat(1)	Supervisa el uso de recursos de los proyectos limitados.

Referencia de comando	Descripción
rcapadm(1M)	Configura el daemon de límite de recursos, muestra el estado actual del daemon de límite de recursos si se ha configurado y activa o desactiva el límite de recursos. También se utiliza para definir un límite de memoria temporal.
rcapd(1M)	El daemon de límite de recursos.

BETA
DRAFT

BETA
DRAFT

11

♦ ♦ ♦ C A P Í T U L O 1 1

Administración de las tareas del daemon de limitación de recursos

Este capítulo contiene procedimientos para configurar y utilizar el daemon de límite de recursos rcapd.

Para ver una descripción general de rcapd, consulte [Capítulo 10, Acerca del control de memoria física mediante el daemon de limitación de recursos](#).

Configuración de límite del tamaño del conjunto residente

Defina un límite de tamaño del juego residente (RSS) del recurso de memoria física para un proyecto mediante la agregación de un atributo `rcap.max-rss` a la entrada de la base de datos `project`.

▼ Cómo agregar un atributo `rcap.max-rss` para un proyecto

1. **Conviértase en root o asuma un rol similar.**
2. **Agregue este atributo al archivo `/etc/project`:**

```
rcap.max-rss=value
```

ejemplo 11-1 Límite de proyecto de RSS

Por ejemplo, la siguiente línea del archivo `/etc/project` define un límite de RSS de 10 gigabytes para un proyecto denominado `db`.

```
db:100::db,root::rcap.max-rss=10737418240
```

El sistema puede redondear el valor de límite especificado a un tamaño de página.

▼ Cómo utilizar el comando `projmod` para agregar un atributo `rcap.max-rss` para un proyecto

1. Conviértase en `root` o asuma un rol similar.
2. Defina un atributo `rcap.max-rss` de 10 gigabytes en el archivo `/etc/project`, en este caso, para un proyecto denominado `db`.

```
# projmod -a -K rcap.max-rss=10GB db
```

El archivo `/etc/project` contiene la línea:

```
db:100::db,root::rcap.max-rss=10737418240
```

Configuración y uso del mapa de tareas del daemon de limitación de recursos

Tarea	Descripción	Instrucciones
Definir el umbral de aplicación del límite de memoria.	Configura un límite que se aplicará cuando haya demasiado poca memoria física disponible para los procesos.	Cómo establecer el umbral de aplicación del límite de memoria [117]
Definir el intervalo de operación.	El intervalo se aplica a las operaciones periódicas que lleva a cabo el daemon de límite de recursos.	Cómo configurar intervalos de funcionamiento [117]
Activar limitación de recursos.	Activa el límite de recursos en el sistema.	Cómo activar los límites de recursos [118]
Desactivar limitación de recursos.	Desactiva el límite de recursos del sistema.	Cómo desactivar los límites de recursos [118]
Límite de informe e información del proyecto.	Visualiza comandos de ejemplo para producir informes.	“Límite de informe e información del proyecto” [120]
Supervisar el tamaño del conjunto residente de un proyecto.	Produce un informe del tamaño del conjunto residente de un proyecto.	“Supervisión del RSS de un proyecto” [120]
Determinar el tamaño del conjunto de trabajo de un proyecto.	Produce un informe del tamaño del conjunto de trabajo de un proyecto.	“Cómo determinar el tamaño del conjunto de trabajo de un proyecto” [121]
Informe sobre el uso y los límites de la memoria.	Imprime una línea de aplicación del límite y el uso de la memoria al final del informe para cada intervalo.	“Informes del uso de la memoria y el umbral de aplicación del límite de la memoria” [122]

Administración del daemon de límite de recursos con `rcapadm`

Esta sección contiene los procedimientos para configurar el daemon de límite de recursos con `rcapadm`. Para obtener más información, consulte [“Configuración de `rcapd`” \[107\]](#) y la página del comando `man rcapadm(1M)`. El uso de `rcapadm` para especificar un límite de recursos temporal para una zona también queda cubierto.

Si se utiliza sin argumentos, `rcapadm` muestra el estado actual del daemon de límite de recursos si se ha configurado.

▼ Cómo establecer el umbral de aplicación del límite de memoria

Pueden configurarse los límites para que no se apliquen hasta que quede poca memoria física disponible para los procesos. Consulte [“Umbral de aplicación de límite de memoria” \[108\]](#) para obtener más información.

El valor mínimo (y predeterminado) es 0, lo que significa que los límites de memoria siempre se aplican. Para definir un mínimo distinto, siga este procedimiento.

1. **Conviértase en root o asuma un rol similar.**
2. **Utilice la opción `-c` de `rcapadm` para definir un valor de uso de memoria física distinto para la aplicación del límite de memoria.**

```
# rcapadm -c percent
```

por ciento va de 0 a 100. Los valores más altos son menos restrictivos. Un valor más elevado significa que las cargas de trabajo del proyecto limitado pueden ejecutarse sin tener los límites aplicados hasta que el uso de memoria del sistema supere este umbral.

Véase también Para ver el uso actual de la memoria física y el umbral de aplicación del límite, consulte [“Informes del uso de la memoria y el umbral de aplicación del límite de la memoria” \[122\]](#).

▼ Cómo configurar intervalos de funcionamiento

[“Intervalos de funcionamiento de `rcapd`” \[110\]](#) contiene información sobre los intervalos para las operaciones periódicas que lleva a cabo `rcapd`. Siga este procedimiento para configurar los intervalos de funcionamiento utilizando `rcapadm`.

1. **Conviértase en root o asuma un rol similar.**
2. **Utilice la opción `-i` para configurar los valores del intervalo.**

```
# rcapadm -i interval=value,...,interval=value
```

Nota - Todos los valores de intervalo se especifican en segundos.

▼ **Cómo activar los límites de recursos**

Hay tres maneras de activar la limitación de recursos en su sistema. La activación de los límites de recursos también configura el archivo `/etc/rcap.conf` con los valores predeterminados.

1. **Conviértase en root o asuma un rol similar.**
2. **Active el daemon de límite de recursos mediante uno de estos procedimientos:**

- Active los límites de recursos utilizando el comando `svcadm`.

```
# svcadm enable rcap
```

- Active el daemon de límite de recursos para que se inicie ahora y también cada vez que se inicie el sistema:

```
# rcapadm -E
```

- Active el daemon de límite de recursos al iniciar sin iniciarlo ahora especificando también la opción `-n`:

```
# rcapadm -n -E
```

▼ **Cómo desactivar los límites de recursos**

Hay tres modos de desactivar los límites de recursos del sistema.

1. **Conviértase en root o asuma un rol similar.**
2. **Desactive el daemon de límite de recursos mediante uno de estos procedimientos:**

- **Desactive el límite de recursos utilizando el comando `svcadm`.**

```
# svcadm disable rcap
```

- Para desactivar el daemon de límite de recursos para que se detenga ahora y que no se inicie al iniciar el sistema, escriba:

```
# rcapadm -D
```

- Para desactivar el daemon de límite de recursos sin detenerlo, especifique también la opción `-n`:

```
# rcapadm -n -D
```

Sugerencia - Use `rcapadm -D` para desactivar `rcapd` de forma segura. Si se cierra el daemon (consulte la página del comando `man kill(1)`), los procesos podrían quedar detenidos y deberían reiniciarse manualmente. Para reanudar un proceso en ejecución, utilice el comando `prun`. Para obtener más información, consulte la página del comando `man prun(1)`.

▼ Cómo especificar un límite de recursos temporal para una zona

Este procedimiento se utiliza para asignar la cantidad máxima de memoria que puede consumir una zona especificada. Este valor sólo dura hasta la próxima vez que se reinicie. Para establecer un límite persistente, utilice el comando `zonecfg`.

1. Conviértase en `root` o asuma un rol similar.
2. Establezca un valor máximo de memoria de 512 megabytes para la zona `my-zone`.

```
# rcapadm -z testzone -m 512M
```

Creación de informes con `rcapstat`

Utilice `rcapstat` para registrar las estadísticas de límites de recursos. “[Supervisión del uso de recursos con `rcapstat`](#)” [111] explica cómo utilizar el comando `rcapstat` para generar informes. En dicha sección también se describen las cabeceras de columna en el informe. La página del comando `man rcapstat(1)` también contiene esta información.

Las siguientes subsecciones utilizan ejemplos para ilustrar el modo en que se producen informes para fines específicos.

Límite de informe e información del proyecto

En este ejemplo, se definen límites para dos proyectos asociados con dos usuarios. user1 tiene un límite de 50 megabytes y user2 un límite de 10 megabytes.

El comando siguiente produce cinco informes a intervalos de muestreo de 5 segundos.

```
user1machine% rcapstat 5 5
  id project nproc  vm  rss  cap  at avgat  pg avgpg
112270  user1    24 123M  35M  50M  50M  0K 3312K  0K
78194  user2     1 2368K 1856K 10M  0K  0K  0K  0K
  id project nproc  vm  rss  cap  at avgat  pg avgpg
112270  user1    24 123M  35M  50M  0K  0K  0K  0K
78194  user2     1 2368K 1856K 10M  0K  0K  0K  0K
  id project nproc  vm  rss  cap  at avgat  pg avgpg
112270  user1    24 123M  35M  50M  0K  0K  0K  0K
78194  user2     1 2368K 1928K 10M  0K  0K  0K  0K
  id project nproc  vm  rss  cap  at avgat  pg avgpg
112270  user1    24 123M  35M  50M  0K  0K  0K  0K
78194  user2     1 2368K 1928K 10M  0K  0K  0K  0K
  id project nproc  vm  rss  cap  at avgat  pg avgpg
112270  user1    24 123M  35M  50M  0K  0K  0K  0K
78194  user2     1 2368K 1928K 10M  0K  0K  0K  0K
```

Las tres primeras líneas de la salida constituyen el primer informe, que contiene el límite y la información del proyecto para los dos proyectos y las estadísticas de paginación desde el inicio de rcapd. Las columnas at y pg tienen un número mayor que cero para user1 y cero para user2, lo que indica que en algún punto del historial del daemon user1 superó su límite pero user2 no lo hizo.

Los informes subsiguientes no muestran ninguna actividad significativa.

Supervisión del RSS de un proyecto

En el ejemplo siguiente, se utiliza el user1 del proyecto, que fue un RSS que superó su límite.

El comando siguiente produce cinco informes a intervalos de muestreo de 5 segundos.

```
user1machine% rcapstat 5 5
  id project nproc  vm  rss  cap  at avgat  pg avgpg
376565  user1     3 6249M 6144M 6144M 690M 220M 5528K 2764K
376565  user1     3 6249M 6144M 6144M  0M 131M 4912K 1637K
376565  user1     3 6249M 6171M 6144M 27M 147M 6048K 2016K
376565  user1     3 6249M 6146M 6144M 4872M 174M 4368K 1456K
376565  user1     3 6249M 6156M 6144M 12M 161M 3376K 1125K
```

El proyecto user1 tiene tres procesos que utilizan activamente la memoria física. Los valores positivos de la columna pg indican que rcapd está extrayendo de forma coherente la memoria

al intentar alcanzar el límite bajando el uso de la memoria física de los procesos del proyecto. Sin embargo, `rcapd` no consigue mantener el RSS por debajo del valor límite. Esto se indica mediante los valores `rss` variables que no muestran una disminución correspondiente. En cuanto se extrae la memoria, la carga de trabajo vuelve a utilizarla y el número de RSS vuelve a subir. Esto significa que toda la memoria residente del proyecto se está utilizando de forma activa y el tamaño del conjunto de trabajo (`WSS`) es superior al límite. De este modo, `rcapd` fuerza la extracción de parte del conjunto de trabajo para cumplir el límite. Con esta condición, el sistema seguirá experimentando altos índices de fallos de paginación y la E/S asociada, hasta que tenga lugar una de las condiciones siguientes:

- Se reduzca el `WSS`.
- Se aumente el límite.
- La aplicación cambie su patrón de acceso a la memoria.

En esta situación, la reducción del intervalo de muestreo podría disminuir la discrepancia entre el valor de RSS y el límite al hacer que `rcapd` muestree la carga de trabajo y aplique los límites con mayor frecuencia.

Nota - Se produce un fallo de página cuando debe crearse una página o el sistema debe copiar una página desde un dispositivo de intercambio.

Cómo determinar el tamaño del conjunto de trabajo de un proyecto

El ejemplo siguiente es una continuación del anterior, y utiliza el mismo proyecto.

El ejemplo anterior muestra que el proyecto `user1` está utilizando más memoria física de lo que permite su límite. Este ejemplo muestra cuánta memoria requiere la carga de trabajo del proyecto.

```
user1machine% rcapstat 5 5
  id project  nproc   vm   rss   cap   at avgat   pg  avgpg
376565  user1      3 6249M 6144M 6144M 690M   0K   689M   0K
376565  user1      3 6249M 6144M 6144M   0K   0K    0K   0K
376565  user1      3 6249M 6171M 6144M  27M   0K   27M   0K
376565  user1      3 6249M 6146M 6144M 4872K   0K  4816K   0K
376565  user1      3 6249M 6156M 6144M  12M   0K   12M   0K
376565  user1      3 6249M 6150M 6144M 5848K   0K  5816K   0K
376565  user1      3 6249M 6155M 6144M  11M   0K   11M   0K
376565  user1      3 6249M 6150M  10G  32K   0K   32K   0K
376565  user1      3 6249M 6214M  10G   0K   0K    0K   0K
376565  user1      3 6249M 6247M  10G   0K   0K    0K   0K
376565  user1      3 6249M 6247M  10G   0K   0K    0K   0K
376565  user1      3 6249M 6247M  10G   0K   0K    0K   0K
376565  user1      3 6249M 6247M  10G   0K   0K    0K   0K
376565  user1      3 6249M 6247M  10G   0K   0K    0K   0K
```

```
376565 user1 3 6249M 6247M 10G 0K 0K 0K 0K
```

A mitad del ciclo, el límite del proyecto `user1` se aumenta de 6 a 10 gigabytes. Este aumento detiene la aplicación del límite y permite el aumento del tamaño del conjunto residente, limitado sólo por los demás procesos y la cantidad de memoria del equipo. La columna `rss` podría estabilizarse para reflejar el tamaño del conjunto de trabajo del proyecto (WSS), en este caso 6247M. Se trata del valor de límite mínimo que permite que los procesos del proyecto funcionen sin incurrir en fallos de página continuos.

Mientras que el límite de `user1` es de 6 gigabytes, en cada intervalo de 5 segundos el RSS disminuye y la E/S aumenta mientras `rcapd` extrae parte de la memoria de la carga de trabajo. Un poco después de completar la extracción, la carga de trabajo, que necesita dichas páginas, las vuelve a insertar mientras sigue ejecutándose. Este ciclo se repite hasta que se eleva el límite a 10 gigabytes, aproximadamente por la mitad del ejemplo. El RSS se estabiliza en 6,1 gigabytes. Dado que el RSS de la carga de trabajo está ahora por debajo del límite, no se produce ninguna paginación adicional. También se detiene la E/S asociada con la paginación. De este modo, el proyecto necesita 6,1 gigabytes para llevar a cabo el trabajo que estaba realizando en el momento en que se observó.

Consulte también las páginas del comando `man vmstat(1M)` and `iostat(1M)`.

Informes del uso de la memoria y el umbral de aplicación del límite de la memoria

La opción `-g` de `rcapstat` permite registrar lo siguiente:

- Uso actual de la memoria física como porcentaje de la memoria física instalada en el sistema
- Umbral de aplicación del límite de memoria del sistema definido por `rcapadm`

La opción `-g` imprime una línea de aplicación del límite y el uso de la memoria al final del informe para cada intervalo.

```
# rcapstat -g
  id project  nproc  vm  rss  cap  at avgat  pg  avgpg
376565  rcap      0   0K   0K  10G   0K   0K   0K   0K
physical memory utilization: 55%  cap enforcement threshold: 0%
  id project  nproc  vm  rss  cap  at avgat  pg  avgpg
376565  rcap      0   0K   0K  10G   0K   0K   0K   0K
physical memory utilization: 55%  cap enforcement threshold: 0%
```

12

♦ ♦ ♦ C A P Í T U L O 1 2

Acerca de las agrupaciones de recursos

Este capítulo analiza las siguientes tecnologías:

- Agrupaciones de recursos, que se utilizan para particionar los recursos del equipo
- Agrupaciones de recursos dinámicos (DRP), que ajustan de forma dinámica la asignación de recursos de cada agrupación de recursos para cumplir los objetivos del sistema

Las agrupaciones de recursos y las agrupaciones de recursos dinámicos son servicios de la utilidad de gestión de servicios (SMF) de Oracle Solaris. Cada uno de estos servicios se activa por separado.

En este capítulo, se describen los siguientes temas:

- “Introducción a las agrupaciones de recursos” [124]
- “Introducción a las agrupaciones de recursos dinámicos” [125]
- “Activación y desactivación de agrupaciones de recursos y agrupaciones de recursos dinámicos” [125]
- “Agrupaciones de recursos utilizadas en zonas” [125]
- “Cuándo se utilizan las agrupaciones” [126]
- “Estructura de agrupaciones de recursos” [127]
- “Implementación de agrupaciones en un sistema” [129]
- “Atributo `project.pool`” [130]
- “Agrupaciones de recursos y operaciones de reconfiguración dinámica” [130]
- “Creación de configuraciones de agrupaciones” [131]
- “Manipulación directa de la configuración dinámica” [132]
- “Descripción general de `pool`” [132]
- “Administración de agrupaciones de recursos dinámicos” [133]
- “Objetivos y restricciones de configuración” [133]
- “Funcionalidad `pool` que se puede configurar” [138]
- “Cómo funciona la asignación de recursos dinámicos” [141]
- “Uso de `poolstat` para supervisar la función de agrupaciones y el uso de los recursos” [144]
- “Comandos utilizados con la función de agrupaciones de recursos” [146]

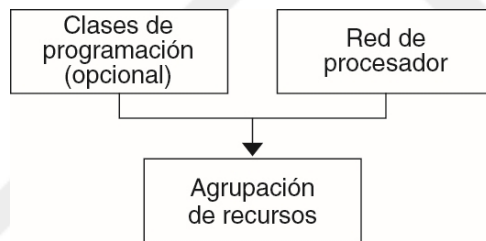
Para conocer los procedimientos para usar esta función, consulte [Capítulo 13, Creación y administración de las tareas de agrupaciones de recursos](#).

Introducción a las agrupaciones de recursos

Las *agrupaciones de recursos* permiten separar cargas de trabajo para que el consumo de carga de trabajo de determinados recursos no se superponga. Esta reserva de los recursos permite obtener un rendimiento predecible en los sistemas con cargas de trabajo mixtas.

Las agrupaciones de recursos proporcionan un mecanismo de configuración persistente para la configuración del conjunto de procesadores (pset) y, opcionalmente, la planificación de la asignación de clases.

FIGURA 12-1 Estructura de la agrupación de recursos



Una agrupación puede considerarse una vinculación específica de los diferentes conjuntos de recursos que están disponibles en el sistema. Puede crear agrupaciones que representen diferentes tipos de combinaciones de recursos posibles:

```
pool1: pset_default  
pool2: pset1  
pool3: pset1, pool.scheduler="FSS"
```

Al agrupar varias particiones, las agrupaciones pueden asociarse con las cargas de trabajo etiquetadas. Cada entrada de proyecto del archivo `/etc/project` puede tener asociada una única agrupación, que se especifica mediante el atributo `project.pool`.

Si las agrupaciones están activas, la configuración básica se establece mediante una *agrupación predeterminada* y un *juego de procesadores predeterminado*. Pueden crearse y agregarse a la configuración agrupaciones definidas por el usuario y conjuntos de procesadores adicionales. Una CPU sólo puede pertenecer a un conjunto de procesadores. Los conjuntos de procesadores y agrupaciones definidas por el usuario se pueden destruir. El conjunto de procesadores y la agrupación que se han configurado de forma predeterminada no se pueden destruir.

La agrupación predeterminada tiene la propiedad `pool.default` configurada como `true`. El juego de procesadores tiene la propiedad `pset.default` configurada como `true`. De este modo, pueden identificarse tanto la agrupación como el conjunto de procesadores predeterminados aunque sus nombres hayan cambiado.

El mecanismo de agrupaciones definidas por el usuario se utiliza principalmente en grandes equipos con más de una CPU. Sin embargo, los equipos pequeños también pueden aprovechar esta función. En el caso de los equipos pequeños, puede crear agrupaciones que compartan particiones de recursos no críticos. Las agrupaciones sólo se separan basándose en los recursos críticos.

Introducción a las agrupaciones de recursos dinámicos

Las agrupaciones de recursos dinámicos proporcionan un mecanismo para ajustar dinámicamente la asignación de recursos de cada agrupación como respuesta a los eventos del sistema y los cambios de carga de la aplicación. Las DRP simplifican y reducen el número de decisiones que debe tomar un administrador. Se realizan ajustes automáticamente para mantener los objetivos de rendimiento del sistema que especifica un administrador. Los cambios realizados en la configuración se registran. Estas funciones se realizan principalmente a través del controlador de recursos `pool`, un daemon de sistema que siempre debería estar activo si se requiere la asignación de recursos dinámicos. De forma periódica, `pool` examina la carga en el sistema y determina si se requiere una intervención para permitir que el sistema mantenga un rendimiento óptimo en lo que se refiere al consumo de recursos. La configuración de `pool` se realiza en la configuración de `libpool`. Para obtener más información sobre `pool`, consulte la página del comando `man pool(1M)`.

Activación y desactivación de agrupaciones de recursos y agrupaciones de recursos dinámicos

Para activar y desactivar las agrupaciones de recursos y agrupaciones de recursos dinámicos, consulte [“Activación y desactivación de la función de agrupaciones” \[150\]](#).

Agrupaciones de recursos utilizadas en zonas

Como alternativa a la asociación de una zona con una agrupación de recursos del sistema, puede utilizar el comando `zonecfg` para crear una agrupación temporal vigente mientras se ejecuta la zona. Para obtener más información, consulte [“Creación y uso de zonas de Oracle Solaris ”](#).

En un sistema con zonas activas, puede asociarse una zona no global con una agrupación de recursos, aunque no es necesario que la agrupación esté asignada exclusivamente a una zona concreta. Asimismo, no puede vincular procesos individuales en zonas no globales a una agrupación diferente utilizando el comando `poolbind` desde la zona global. Para asociar una zona no global con una agrupación, consulte [“Creación y uso de zonas de Oracle Solaris ”](#).

Observe que si configura una clase de programación para una agrupación y asocia una zona no global con dicha agrupación, la zona utiliza dicha clase de planificación de forma predeterminada.

Si utiliza agrupaciones de recursos dinámicos, el ámbito de una instancia de `pool` en ejecución se limita a la zona global.

Si la utilidad `poolstat` se ejecuta en una zona no global muestra sólo información sobre la agrupación asociada con la zona. El comando `pooladm` ejecutado sin argumentos en una zona no global muestra sólo información sobre la agrupación asociada con la zona.

Para obtener información sobre los comandos de agrupaciones de recursos, consulte [“Comandos utilizados con la función de agrupaciones de recursos” \[146\]](#).

Cuándo se utilizan las agrupaciones

Las agrupaciones de recursos ofrecen un mecanismo versátil que se puede aplicar a múltiples situaciones administrativas.

Servidor de cálculo
por lotes

Utilice la función de agrupaciones para dividir un servidor en dos agrupaciones. Se utiliza una agrupación para las sesiones de inicio de sesión y el trabajo interactivo de los usuarios que comparten el tiempo. La otra agrupación se utiliza para los trabajos que se envían a través del sistema por lotes.

Servidor de base de
datos o aplicación

Particione los recursos de las aplicaciones interactivas de acuerdo con los requisitos de las aplicaciones.

Activación de las
aplicaciones por
fases

Defina las expectativas del usuario.

Puede implementar inicialmente un equipo que ejecute sólo una fracción de los servicios que se espera que la máquina ofrezca en último término. El usuario puede tener dificultades si no se establecen mecanismos de administración de los recursos basados en las reservas cuando el equipo está en línea.

Por ejemplo, el programador de reparto justo optimiza el uso de la CPU. Los tiempos de respuesta de un equipo en el que sólo se ejecuta una aplicación pueden ser rápidos de manera equívoca. Los usuarios no verán estos tiempos de respuesta con varias aplicaciones cargadas. Al utilizar diferentes agrupaciones para cada aplicación, puede colocar un

	máximo para el número de CPU disponibles para cada aplicación antes de implementar todas las aplicaciones.
Servidor de tiempo compartido complejo	Particione un servidor que admite grandes cantidades de usuarios. La partición del servidor proporciona un mecanismo de aislamiento que conduce a una respuesta por usuario más predecible. Al dividir a los usuarios en grupos que se vinculan a agrupaciones diferentes, y utilizar el programador de reparto justo (FSS), puede configurar las asignaciones de CPU para favorecer a los conjuntos de usuarios que tengan prioridad. Esta asignación puede basarse en el rol del usuario, el contracargo, etc.
Cargas de trabajo que cambian temporalmente	Utilice agrupaciones de recursos para adaptarse a las variaciones en la demanda. El sitio puede experimentar cambios predecibles en la demanda de carga de trabajo durante largos períodos de tiempo, por ejemplo cada mes, cada trimestre o cada año. Si su sitio experimenta dichos cambios, puede alternar entre varias configuraciones de agrupaciones invocando <code>pooladm</code> desde un trabajo cron. (Consulte “Estructura de agrupaciones de recursos” [127].)
Aplicaciones en tiempo real	Cree una agrupación en tiempo real utilizando el programador RT y recursos de procesador designados.
Uso del sistema	Aplique los objetivos del sistema que establezca. Utilice la función de daemon de agrupaciones automatizadas para identificar los recursos disponibles y luego supervisar las cargas de trabajo con el fin de detectar el momento en que dejan de cumplirse los objetivos especificados. El daemon puede emprender la acción correctiva, si es posible, o puede registrarse la condición.

Estructura de agrupaciones de recursos

El archivo de configuración `/etc/pooladm.conf` describe la configuración de las agrupaciones estáticas. Una configuración estática representa el modo en que un administrador desea configurar un sistema con respecto a la funcionalidad de las agrupaciones de recursos. Puede especificarse un nombre de archivo alternativo.

Cuando se utiliza el comando `pooladm -e` o la utilidad de gestión de servicios (SMF) para activar la estructura de agrupaciones de recursos, si existe un archivo `/etc/pooladm.conf`, la configuración que contiene el archivo se aplica al sistema.

El kernel contiene información sobre la disposición de los recursos en la estructura de las agrupaciones de recursos. Esto se conoce como configuración dinámica, y representa la

función de agrupaciones de recursos para un sistema concreto en un determinado momento. La configuración dinámica se puede visualizar mediante el comando `pooladm`. Tenga en cuenta que el orden en el que se muestran las propiedades para las agrupaciones y conjuntos de recursos puede variar. Las modificaciones en la configuración dinámica se realizan de los siguientes modos:

- Indirectamente, aplicando un archivo de configuración estática
- Directamente, utilizando el comando `poolcfg` con la opción `-d`

Puede haber más de un archivo de configuración de agrupaciones estáticas, para su activación en diferentes momentos. Puede alternar entre múltiples configuraciones de agrupaciones invocando `pooladm` desde un trabajo `cron`. Consulte la página del comando [man cron\(1M\)](#) para obtener más información sobre la utilidad `cron`.

De modo predeterminado, la estructura de las agrupaciones de recursos no está activa. Las agrupaciones de recursos deben activarse para crear o modificar la configuración dinámica. Los archivos de configuración estática pueden manipularse con los comandos `poolcfg` o `libpool` aunque la estructura de las agrupaciones de recursos esté desactivada. Los archivos de configuración estática no se pueden crear si no está activa la función de agrupaciones. Para obtener más información sobre el archivo de configuración, consulte [“Creación de configuraciones de agrupaciones” \[131\]](#).

Los comandos utilizados con las agrupaciones de recursos y el daemon de sistema `poold` se describen en las siguientes páginas del comando `man`:

- [pooladm\(1M\)](#)
- [poolbind\(1M\)](#)
- [poolcfg\(1M\)](#)
- [poold\(1M\)](#)
- [poolstat\(1M\)](#)
- [libpool\(3LIB\)](#)

Contenido de `/etc/pooladm.conf`

Todas las configuraciones de agrupaciones de recursos, incluida la configuración dinámica, puede incluir los siguientes elementos.

sistema	Propiedades que afectan al comportamiento global del sistema
agrupación	Definición de una agrupación de recursos
pset	Definición de un conjunto de procesadores

cpu

Definición de un procesador

Todos estos elementos tienen propiedades que se pueden modificar para cambiar el comportamiento de la estructura de las agrupaciones de recursos. Por ejemplo, la propiedad de agrupación `pool.importance` indica la importancia relativa de una agrupación concreta. Esta propiedad se utiliza para la posible resolución de conflictos relativos a los recursos. Para obtener más información, consulte [libpool\(3LIB\)](#).

Propiedades de agrupaciones

La utilidad de agrupaciones admite propiedades con nombre escritas que se pueden colocar en una agrupación, un recurso o un componente. Los administradores pueden almacenar propiedades adicionales en los distintos elementos de la agrupación. Se utiliza un espacio de nombre de propiedad similar al atributo del proyecto.

Por ejemplo, el siguiente comentario indica que se asocia un pset concreto con una base de datos `Datatree` específica.

```
Datatree, pset.dbname=warehouse
```

Para obtener más información sobre los tipos de propiedades, consulte “[Propiedades de pool](#)” [137].

Nota - Se ha reservado una serie de propiedades especiales para uso interno y no se pueden configurar ni eliminar. Para obtener más información, consulte la página del comando `man libpool(3LIB)`.

Implementación de agrupaciones en un sistema

Las agrupaciones definidas por el usuario se pueden implementar en un sistema mediante uno de estos métodos.

- Cuando se inicia el software de Oracle Solaris, una secuencia de comandos `init` comprueba si existe el archivo `/etc/pooladm.conf`. Si se encuentra el archivo y las agrupaciones están activas, se invoca `pooladm` para activar esta configuración de agrupaciones. El sistema crea una configuración dinámica para reflejar la organización que se solicita en `/etc/pooladm.conf`, y los recursos del equipo se particionan en consonancia.
- Cuando el sistema Oracle Solaris está en ejecución, puede activar una configuración de agrupaciones si no está presente todavía, o modificarla utilizando el comando `pooladm`. De modo predeterminado, el comando `pooladm` funciona en `/etc/pooladm.conf`. Sin embargo, de forma opcional puede especificar una ubicación y un nombre de archivo alternativos, y utilizar dicho archivo para actualizar la configuración de las agrupaciones.

Para obtener información sobre la activación y desactivación de agrupaciones de recursos, consulte [“Activación y desactivación de la función de agrupaciones” \[150\]](#). La función de agrupaciones no puede desactivarse cuando se están utilizando agrupaciones o recursos definidos por el usuario.

Para configurar las agrupaciones de recursos, debe tener privilegios de root o un perfil con los derechos necesarios.

El controlador de recursos `pool` se inicia con la función de agrupación de recursos dinámicos.

Atributo `project.pool`

Se puede agregar el atributo `project.pool` a una entrada de proyecto del archivo `/etc/project` para asociar una única agrupación con dicha entrada. Los nuevos trabajos que se inicien en un proyecto estarán vinculados a la agrupación adecuada. Para obtener más información, consulte [Capítulo 2, Acerca de los proyectos y las tareas](#).

Por ejemplo, puede utilizar el comando `projmod` para definir el atributo `project.pool` para el proyecto `sales` en el archivo `/etc/project`:

```
# projmod -a -K project.pool=mypool sales
```

SPARC: Agrupaciones de recursos y operaciones de reconfiguración dinámica

La reconfiguración dinámica (DR) le permite reconfigurar el hardware mientras el sistema está en ejecución. Una operación de DR puede aumentar, reducir o no tener ningún efecto sobre un tipo de recurso específico. Dado que la DR puede afectar a las cantidades de recursos disponibles, la función de agrupaciones debe incluirse en estas operaciones. Cuando se inicia una operación de DR, la estructura de las agrupaciones actúa para validar la configuración.

Si el funcionamiento de DR puede continuar sin que la configuración de las agrupaciones actuales deje de ser válida, se actualiza el archivo de configuración privado. Un archivo de configuración no válido es aquél que no admiten los recursos disponibles.

Si el funcionamiento de DR hace que la configuración de las agrupaciones deje de ser válida, se producirá un error y se notificará mediante un mensaje en el registro de mensajes. Si desea forzar la finalización de la configuración, debe utilizar la opción de forzado de DR. La configuración de las agrupaciones se modifica para cumplir la nueva configuración del recurso. Para obtener información sobre el proceso de DR y la opción de forzado, consulte la guía del usuario sobre reconfiguración dinámica para el hardware de Sun.

Si utiliza agrupaciones de recursos dinámicas, tenga en cuenta que es posible que una partición quede fuera del control de `pool`d mientras el daemon está activo. Para más información, consulte “[Identificación de recursos insuficientes](#)” [143].

Creación de configuraciones de agrupaciones

El archivo de configuración contiene una descripción de las agrupaciones que se deben crear en el sistema. El archivo describe los elementos que se pueden manipular.

- sistema
- agrupación
- pset
- cpu

Consulte [poolcfg\(1M\)](#) para obtener más información sobre los elementos que se manipulan.

Si las agrupaciones están activadas, puede crear un archivo `/etc/pooladm.conf` estructurado de dos modos.

- Puede utilizar el comando `pooladm` con la opción `-s` para conocer los recursos del sistema actual y colocar los resultados en un archivo de configuración.
Se recomienda utilizar este método. Se registran todos los recursos y componentes del sistema que pueden manipular las agrupaciones. Los recursos incluyen las configuraciones de conjuntos de procesadores existentes. Puede modificar la configuración para cambiar el nombre de los conjuntos de procesadores o crear agrupaciones adicionales si es preciso.
- Puede utilizar el comando `poolcfg` con la opción `-c` y los subcomandos `discover` o `create system name` para crear una nueva configuración de las agrupaciones.

Estas opciones se mantienen para la compatibilidad con versiones anteriores.

Utilice `poolcfg` o `libpool` para modificar el archivo `/etc/pooladm.conf`. No edite directamente este archivo.

Asignación específica de CPU, núcleos centrales y sockets

Utilice los subcomandos `assign` y `unassign` asigne CPU, núcleos centrales y sockets específicos.

Los subcomandos `assign` y `unassign` son aplicables a las configuraciones persistente y de tiempo de ejecución de las agrupaciones. El uso de `assign` y la configuración de `pset.min` y `pset.max` directamente se excluyen mutuamente. Cada método sobrescribe la configuración definida por el otro. Las CPU configuradas como psets con las propiedades `pset.min` y `pset.max` se consideran *distribuidas* en lugar de asignadas. `assign` y `unassign` agregan y

eliminan CPU específicas desde un pset o a un pset. La primera assign borrará cualquier configuración de pset configurada por una distribución anterior. Utilice unassign sólo después de una assign correcta. unassign no se puede utilizar para manipular CPU de psets distribuidos.

Consulte también [“Recurso dedicated-cpu”](#) de [“Introducción a Zonas de Oracle Solaris ”](#).

Manipulación directa de la configuración dinámica

Los tipos de recursos de la CPU de la configuración dinámica se pueden manipular directamente utilizando el comando `poolcfg` con la opción `-d`. Existen dos métodos para transferir los recursos.

- Puede realizar una solicitud general para transferir cualquier recurso identificado disponible de un conjunto a otro.
- Puede transferir recursos con ID específicos a un conjunto de destino. Observe que los ID de sistema asociados con los recursos pueden cambiar cuando se modifica la configuración de los recursos o después de reiniciar el sistema.

Para ver un ejemplo, consulte [“Transferencia de recursos”](#) [165].

Si DRP está en uso, tenga en cuenta que la transferencia de recursos podría desencadenar una acción de `pool`. Consulte [“Descripción general de pool”](#) [132] para obtener más información.

Descripción general de pool

El controlador de recursos de agrupaciones, `pool`, utiliza objetivos del sistema y estadísticas observables para mantener los objetivos de rendimiento del sistema que especifique. El daemon del sistema siempre debe estar activo cuando se requiere la asignación de recursos dinámica.

El controlador de recursos `pool` identifica los recursos disponibles y luego supervisa las cargas de trabajo con el fin de determinar cuándo dejan de cumplirse los objetivos de uso del sistema. `pool` considera configuraciones alternativas en términos de objetivos y emprende las acciones correctivas oportunas. Si es posible, los recursos se vuelven a configurar para poder cumplir los objetivos. Si no se pueden emprender las acciones correctivas, el daemon registra que no se pueden seguir cumpliendo los objetivos especificados por el usuario. Tras una reconfiguración, el daemon reanuda la supervisión de los objetivos de la carga de trabajo.

`pool` mantiene un historial de las decisiones que puede consultar. El historial de decisiones permite eliminar las reconfiguraciones que no conllevan mejoras.

Tenga en cuenta que también se puede desencadenar una reconfiguración de forma asíncrona si se modifican los objetivos de la carga de trabajo o los recursos disponibles para el sistema.

Administración de agrupaciones de recursos dinámicos

El servicio de agrupaciones de recursos dinámicos lo administra la utilidad de gestión de servicios (SMF) con el identificador de servicio `svc:/system/pools/dynamic`.

Las acciones administrativas de este servicio, como la activación, la desactivación o la solicitud de reinicio, pueden llevarse a cabo con el comando `svcadm`. El comando `svcs` permite consultar el estado del servicio. Para obtener más información, consulte las páginas del comando `man svcs(1)` y `svcadm(1M)`.

La interfaz de SMF es el método recomendado para controlar las agrupaciones de recursos dinámicos, pero también se siguen otros métodos para conseguir compatibilidad con versiones anteriores.

- Si no se requiere la asignación de recursos dinámicos, `poold` puede detenerse con las señales `SIGQUIT` o `SIGTERM`. Estas dos señales hacen que `poold` termine correctamente.
- Aunque `poold` detecte automáticamente los cambios en la configuración de recursos o agrupaciones, también puede forzar una reconfiguración utilizando la señal `SIGHUP`.

Objetivos y restricciones de configuración

Al realizar cambios en una configuración, `poold` actúa en las indicaciones que proporciona. Estas indicaciones se especifican como una serie de objetivos y restricciones. `poold` utiliza las especificaciones para determinar el valor relativo de las diferentes posibilidades de configuración en relación con la configuración existente. `poold` luego cambia las asignaciones de recursos de la configuración actual para generar nuevas configuraciones candidatas.

Restricciones de la configuración

Las restricciones afectan el rango de posibles configuraciones al eliminar algunos de los posibles cambios que pueden realizarse en una configuración. Están disponibles las siguientes restricciones, que se especifican en la configuración de `libpool`.

- Las asignaciones de CPU mínima y máxima
- Los componentes asociados que no están disponibles para moverlos desde un conjunto
- Factor de importancia de la agrupación

Consulte la página del comando `man libpool(3LIB)` y “Propiedades de agrupaciones” [129] para obtener más información sobre las propiedades de las agrupaciones.

Consulte [Cómo establecer restricciones de configuración \[161\]](#) para obtener instrucciones de uso.

Propiedad `pset.min` y restricciones de la propiedad `pset.max`

Estas dos propiedades establecen los límites para la cantidad de procesadores que se pueden asignar a un conjunto, tanto el máximo como el mínimo. Consulte la [Tabla 12-1, “Nombres de propiedad definidos”](#) para obtener más información sobre estas propiedades.

Dentro de estas restricciones, los recursos de la partición están disponibles para su asignación a otras particiones de recursos de la misma instancia de Oracle Solaris. El acceso al recurso se obtiene vinculando una agrupación asociada con el conjunto de recursos. La vinculación se lleva a cabo durante el inicio de sesión o la realiza manualmente un administrador que tenga el privilegio `PRIV_SYS_RES_CONFIG`.

Restricción de la propiedad `cpu.pinned`

La propiedad `cpu-pinned` indica que las agrupaciones de recursos dinámicos no deben mover una CPU específica del juego de procesadores en el que se encuentra. Puede configurar esta propiedad `libpool` para maximizar el uso de caché para una aplicación específica que se ejecuta dentro de un conjunto de procesadores.

Consulte la [Tabla 12-1, “Nombres de propiedad definidos”](#) para obtener más información sobre esta propiedad.

Restricción de la propiedad `pool.importance`

La propiedad `pool.importance` describe la importancia relativa de una agrupación según lo establecido por el administrador.

Objetivos de la configuración

Los objetivos se especifican de forma similar a las restricciones. El juego completo de objetivos se describe en [Tabla 12-1, “Nombres de propiedad definidos”](#).

Existen dos categorías de objetivos.

Dependientes de la carga de trabajo	Un objetivo dependiente de la carga de trabajo es un objetivo que varía según la naturaleza de la carga de trabajo que se ejecuta en el sistema. Un ejemplo es el objetivo de <code>utilization</code> . La cifra de utilización de un
-------------------------------------	--

conjunto de recursos varía según la naturaleza de la carga de trabajo que hay activa en el conjunto.

Independientes de la carga de trabajo

Un objetivo independiente de la carga de trabajo es un objetivo que no varía según la naturaleza de la carga de trabajo que se ejecuta en el sistema. Un ejemplo es el objetivo de `locality` de la CPU. La medida de localidad evaluada para un conjunto de recursos no varía según la naturaleza de la carga de trabajo que está activa en el conjunto.

Puede definir tres tipos de objetivos.

Nombre	Elementos válidos	operadores	Valores
<code>wt-load</code>	<code>system</code>	N/A	N/A
<code>locality</code>	<code>pset</code>	N/A	<code>loose</code> <code>tight</code> <code>none</code>
<code>utilization</code>	<code>pset</code>	<code><</code> <code>></code> <code>~</code>	0–100%

Los objetivos se almacenan en las cadenas de propiedad de la configuración de `libpool`. Los nombres de propiedad son los siguientes:

- `system.pool objectives`
- `pset.pool objectives`

Los objetivos tienen la siguiente sintaxis:

- `objectives = objective [; objective]*`
- `objective = [n:] keyword [op] [value]`

Todos los objetivos adoptan un prefijo de importancia opcional. La importancia actúa como multiplicador del objetivo y, por tanto, aumenta el significado de su contribución a la evaluación de la función de objetivos. El valor puede estar entre 0 y `INT64_MAX` (9223372036854775807). Si no se especifica, el valor predeterminado de importancia es de 1.

Algunos tipos de elementos admiten más de un tipo de objetivo. Un ejemplo es `pset`. Puede especificar varios tipos de objetivos para estos elementos. También puede especificar varios objetivos de utilización en un único elemento `pset`.

Consulte [Cómo definir los objetivos de configuración \[161\]](#) para ver algunos ejemplos de uso.

Objetivo de `wt-load`

El objetivo de `wt-load` favorece configuraciones que asocian asignaciones de recursos con utilizaciones de recursos. A un juego de recursos que utiliza más recursos se le asignan más recursos cuando este objetivo está activo. `wt-load` significa *carga ponderada*.

Este objetivo se utiliza cuando se está conforme con las restricciones que se han establecido utilizando las propiedades mínima y máxima, y se desea que el daemon manipule los recursos libremente dentro de dichas restricciones.

El objetivo `locality`

El objetivo `locality` influye en el impacto que tiene la localidad, medida por los datos de grupo de localidad (`lgroup`), en la configuración seleccionada. La latencia es otro modo de definir la localidad. `lgroup` describe los recursos de memoria y CPU. El sistema Oracle Solaris emplea `lgroup` para determinar la distancia entre recursos y usa el tiempo como unidad de medición.

Este objetivo puede adoptar uno de los tres valores siguientes:

<code>tight</code>	Si se configura, se favorecen las configuraciones que maximizan la localidad de los recursos.
<code>loose</code>	Si se configura, se favorecen las configuraciones que minimizan la localidad de los recursos.
<code>none</code>	Si se configura, no se favorece ninguna configuración basándose en la localidad de los recursos. Se trata del valor predeterminado para el objetivo <code>locality</code> .

En general, el objetivo `locality` debe configurarse como `tight`. Sin embargo, para maximizar el ancho de banda de la memoria o minimizar el impacto de las operaciones de recursos dinámicos en un conjunto de recursos, puede configurar este objetivo como `loose` o dejar la configuración predeterminada de `none`.

Objetivo `utilization`

El objetivo `utilization` favorece configuraciones que asignan recursos a particiones que no cumplen el objetivo de utilización especificado.

Este objetivo se especifica utilizando operadores y valores. Los operadores son los siguientes:

<code><</code>	El operador "menor que" indica que el valor especificado representa un valor de destino máximo.
<code>></code>	El operador "mayor que" indica que el valor especificado representa un valor de destino mínimo.
<code>~</code>	El operador "similar" indica que el valor especificado es un valor de destino para el que se acepta cierta fluctuación.

Pset sólo puede tener configurado un objetivo de utilización para cada tipo de operador.

- Si se configura el operador $\sim$, no es posible configurar los operadores $< y >$.
- Si se configuran los operadores $< y >$, no es posible configurar el operador $\sim$. Tenga en cuenta que la configuración de los operadores $< y >$ no puede contradecirse.

Puede definir los operadores $< y >$ a la vez para crear un rango. Los valores se validarán para asegurarse de que no se solapen.

Ejemplo de objetivos de configuración

En el ejemplo siguiente, poold evaluará estos objetivos para pset:

- El valor de utilization debe estar entre 30 y 80 por ciento.
- El valor de locality debe maximizarse para el conjunto de procesadores.
- Los objetivos deben adoptar la importancia predeterminada de 1.

EJEMPLO 12-1 Ejemplo de objetivos de poold

```
pset.poold.objectives "utilization > 30; utilization < 80; locality tight"
```

Consulte [Cómo definir los objetivos de configuración \[161\]](#) para ver algunos ejemplos de uso adicionales.

Propiedades de poold

Hay cuatro categorías de propiedades:

- Configuración
- Restricción
- Objetivo
- Parámetro de objetivo

TABLA 12-1 Nombres de propiedad definidos

Nombre de propiedad	Tipo	Category	Descripción
system.poold.log-level	cadena	Configuración	Nivel de registro
system.poold.log-location	cadena	Configuración	Ubicación de registro
system.poold.monitor-interval	uint64	Configuración	Intervalo de muestra de supervisión

Nombre de propiedad	Tipo	Category	Descripción
<code>system.pool.history-file</code>	cadena	Configuración	Ubicación de historial de decisiones
<code>pset.max</code>	uint64	Restricción	Número máximo de CPU para este conjunto de procesadores
<code>pset.min</code>	uint64	Restricción	Número mínimo de CPU para este conjunto de procesadores
<code>cpu.pinned</code>	bool	Restricción	CPU asociadas a este conjunto de procesadores
<code>system.pool.objectives</code>	cadena	Objetivo	Cadena con formato de acuerdo con la sintaxis de expresión de objetivos de <code>pool</code>
<code>pset.pool.objectives</code>	cadena	Objetivo	Cadena con formato de acuerdo con la sintaxis de expresión de <code>pool</code>
<code>pool.importance</code>	int64	Parámetro de objetivo	Importancia asignada por el usuario

Funcionalidad `pool` que se puede configurar

Puede configurar estos aspectos del comportamiento del daemon.

- Intervalo de supervisión
- Nivel de registro
- Ubicación de registro

Estas opciones se especifican en la configuración de las agrupaciones. También puede controlar el nivel de registro de la línea de comandos invocando `pool`.

Intervalo de supervisión de `pool`

Utilice el nombre de propiedad `system.pool.monitor-interval` para especificar un valor en milisegundos.

Información de registro de `pool`

En el registro se proporcionan tres categorías de información. Estas categorías se identifican en los registros:

- Configuración
- Supervisión
- Optimización

Utilice el nombre de propiedad `system.pool.log-level` para especificar el parámetro de registro. Si no se especifica esta propiedad, el nivel de registro predeterminado es `NOTICE`. Los niveles de parámetros son jerárquicos. La configuración de un nivel de registro de `DEBUG` causará que `pool` registre todos los mensajes definidos. El nivel `INFO` proporciona un equilibrio útil de la información para la mayoría de los administradores.

En la línea de comandos, puede utilizar el comando `pool` con la opción `-l` y un parámetro para especificar el nivel de información de registro generada.

Hay disponibles los siguientes parámetros:

- `ALERT`
- `CRIT`
- `ERR`
- `WARNING`
- `NOTICE`
- `INFO`
- `DEBUG`

Los niveles de parámetros se asignan directamente a sus equivalentes de `syslog`. Consulte [“Ubicación de registro” \[141\]](#) para más información sobre el uso de `syslog`.

Para obtener más información sobre cómo configurar el registro de `pool`, consulte [Cómo establecer el nivel de registro de `pool` \[164\]](#).

Registro de información de configuración

Pueden generarse los siguientes tipos de mensajes:

ALERT	Indica problemas para acceder a la configuración de <code>libpool</code> u otro error importante no previsto de la función <code>libpool</code> . Cierra el daemon y requiere una acción inmediata por parte del administrador.
CRIT	Problemas debidos a errores no previstos. Cierra el daemon y requiere una acción inmediata por parte del administrador.
ERR	Indica problemas con los parámetros especificados por el usuario que controlan el funcionamiento, como objetivos de utilización conflictivos

que no se pueden resolver para un conjunto de recursos. Requiere la intervención del administrador para corregir los objetivos. pool'd intenta emprender las acciones correctivas omitiendo los objetivos conflictivos, pero algunos errores hacen que se cierre el daemon.

WARNING	Advertencias relativas a la configuración de parámetros que, aunque son técnicamente correctos, podrían no ser adecuados para el entorno de ejecución específico. Un ejemplo es marcar todos los recursos de la CPU como asociados, lo que significa que pool'd no puede mover los recursos de la CPU entre conjuntos de procesadores.
DEBUG	Mensajes que contienen la información detallada que se necesita al depurar el procesamiento de la configuración. Normalmente esta información no la utilizan los administradores.

Supervisión del registro de información

Pueden generarse los siguientes tipos de mensajes:

CRIT	Problemas debidos a fallos de supervisión no previstos. Cierra el daemon y requiere una acción inmediata por parte del administrador.
ERR	Problemas debidos a un error de supervisión no previsto. Podrían requerir una acción correctiva por parte del administrador.
NOTICE	Mensajes sobre las transiciones de regiones de control de recursos.
INFO	Mensajes sobre las estadísticas de uso de los recursos.
DEBUG	Mensajes que contienen la información detallada que se necesita al depurar el procesamiento de supervisión. Normalmente esta información no la utilizan los administradores.

Registro de información de optimización

Pueden generarse los siguientes tipos de mensajes:

WARNING	Podrían aparecer mensajes sobre problemas a la hora de tomar las decisiones correctas. Algunos ejemplos podrían incluir conjuntos de recursos demasiado restringidos por sus valores mínimo y máximo o por el número de componentes asociados. Podrían mostrarse mensajes sobre problemas a la hora de realizar una reasignación óptima a causa de limitaciones imprevistas. Algunos
---------	--

ejemplos podrían conllevar la eliminación del último procesador del conjunto de procesadores que contiene un consumidor de recursos vinculado.

NOTICE	Mensajes sobre configuraciones utilizables o configuraciones que no se implementarán porque podrían solaparse los historiales de decisiones.
INFO	Mensajes sobre configuraciones alternativas que se toman en consideración.
DEBUG	Mensajes que contienen la información detallada que se necesita al depurar el procesamiento de optimización. Normalmente esta información no la utilizan los administradores.

Ubicación de registro

La propiedad `system.pool.d.log-location` se utiliza para especificar la ubicación de la salida registrada de `pool.d`. Puede especificar una ubicación de SYSLOG para la salida de `pool.d` (consulte `syslog(3C)`).

Si no se especifica esta propiedad, la ubicación predeterminada de la salida registrada de `pool.d` es `/var/log/pool/pool.d`.

Esta propiedad no se utiliza cuando se invoca `pool.d` desde la línea de comandos. Se escriben entradas de registro en `stderr` en el terminal que ejecuta el comando.

Administración de registros con `logadm`

Si `pool.d` está activo, el archivo `logadm.conf` incluye una entrada para administrar el archivo predeterminado `/var/log/pool/pool.d`. La entrada es:

```
/var/log/pool/pool.d -N -s 512k
```

Consulte las páginas del comando `man logadm(1M)` y `logadm.conf(4)`.

Cómo funciona la asignación de recursos dinámicos

En esta sección se describe el proceso y los factores que utiliza `pool.d` para asignar recursos dinámicamente.

Acerca de los recursos disponibles

Los recursos disponibles se consideran todos los recursos disponibles para utilizar en el ámbito del proceso `pool`. El ámbito de control es, como máximo, una única instancia de Oracle Solaris.

En un sistema con zonas activas, el ámbito de una instancia de ejecución de `pool` está limitado a la zona global.

Determinación de recursos disponibles

Las agrupaciones de recursos abarcan todos los recursos del sistema que están disponibles para que las aplicaciones los consuman.

Para una única instancia de Oracle Solaris en ejecución, debe asignarse un recurso de un único tipo, como una CPU, a una sola partición. Puede haber una o más particiones para cada tipo de recurso. Cada partición contiene un conjunto de recursos único.

Por ejemplo, un equipo con cuatro CPU y dos conjuntos de procesadores puede tener la siguiente configuración:

pset 0: 0 1

pset 1: 2 3

donde 0, 1, 2 y 3 después de los dos puntos representan los ID de CPU. Tenga en cuenta que los dos conjuntos de procesadores representan las cuatro CPU.

El mismo equipo no puede tener la siguiente configuración:

pset 0: 0 1

pset 1: 1 2 3

No puede tener esta configuración porque la CPU 1 no puede aparecer en más de un pset a la vez.

No se puede acceder a los recursos desde una partición que no sea la partición a la que pertenecen.

Para conocer los recursos disponibles, `pool` interroga la configuración de las agrupaciones activas para buscar las particiones. Se resumen todos los recursos de todas las particiones para determinar la cantidad total de recursos disponibles para cada tipo de recurso que se controla.

Esta cantidad de recursos es la cifra básica que utiliza `pool` en sus operaciones. Sin embargo, esta cifra tiene restricciones que limitan la flexibilidad que tiene `pool` para

realizar asignaciones. Para obtener información sobre las restricciones disponibles, consulte [“Restricciones de la configuración” \[133\]](#).

Identificación de recursos insuficientes

El ámbito de control para `pool` se define como el juego de recursos disponibles para los que `pool` tiene la responsabilidad principal de la administración y partición eficaces. No obstante, otros mecanismos que tienen permiso para manipular los recursos de este ámbito de control pueden seguir afectando a una configuración. Si debe colocar la partición fuera del control mientras `pool` está activo, `pool` intenta restaurar el control mediante la manipulación sensata de los recursos disponibles. Si `pool` no puede localizar los recursos adicionales dentro de su ámbito, el daemon registra información sobre los recursos insuficientes.

Determinación del uso de recursos

`pool` dedica la mayor parte del tiempo a observar el uso de los recursos en su ámbito de control. Esta supervisión se lleva a cabo para verificar que se cumplan los objetivos que dependen de la carga de trabajo.

Por ejemplo, para los conjuntos de procesadores, todas las mediciones se realizan en todos los procesadores de un conjunto. El uso de recursos muestra la proporción de tiempo que el recurso está en uso durante el intervalo de ejemplo. El uso del recurso se muestra como un porcentaje de 0 a 100.

Identificación de infracciones del control

Se utilizan las directivas que se describen en [“Objetivos y restricciones de configuración” \[133\]](#) para detectar el fallo de enfoque de un sistema a la hora de cumplir sus objetivos. Estos objetivos están relacionados directamente con la carga de trabajo.

Una partición que no cumple los objetivos especificados por el usuario es una infracción de control. Los dos tipos de infracciones de control son la infracción síncrona y la infracción asíncrona.

- Una infracción síncrona de un objetivo la detecta el daemon durante la supervisión de su carga de trabajo.
- Una infracción asíncrona de un objetivo tiene lugar independientemente de la acción de supervisión del daemon.

Los siguientes eventos ocasionan infracciones de objetivos asíncronas:

- Se añaden o eliminan recursos de un ámbito de control.

- Se reconfigura el ámbito de control.
- Se reinicia el controlador de recursos poold.

Se da por sentado que las contribuciones de los objetivos que no están relacionados con la carga de trabajo permanecen constantes entre las evaluaciones de la función de objetivos. Los objetivos que no están relacionados con la carga de trabajo sólo se vuelven a evaluar cuando se desencadena una reevaluación mediante una de las infracciones asíncronas.

Determinación de la acción correctiva apropiada

Cuando el controlador de recursos determina que un consumidor de recursos tiene recursos insuficientes, la respuesta inicial es que el aumento de los recursos mejorará el rendimiento.

Se examinan y evalúan configuraciones alternativas que cumplan los objetivos especificados en la configuración para el ámbito de control.

Este proceso se perfecciona con el tiempo a medida que se supervisa el desplazamiento de los recursos y se evalúa la capacidad de respuesta de cada partición de recurso. Se consulta el historial de decisiones para eliminar las reconfiguraciones que no aportaron mejoras para conseguir la función del objetivo. Se utiliza otra información, como nombres de procesos y cantidades, para evaluar todavía más la importancia de los datos históricos.

Si el daemon no puede emprender las acciones correctivas, la condición se registra. Para obtener más información, consulte [“Información de registro de poold” \[138\]](#).

Uso de poolstat para supervisar la función de agrupaciones y el uso de los recursos

La utilidad poolstat se utiliza para supervisar el uso de recursos cuando las agrupaciones están activadas en su sistema. Esta utilidad examina de forma iterativa todas las agrupaciones activas en un sistema y muestra estadísticas basándose en el modo de salida seleccionado. Las estadísticas de poolstat permiten determinar qué particiones de recursos se utilizan de forma intensiva. Puede analizar estas estadísticas para tomar decisiones sobre la reasignación de recursos cuando el sistema está bajo presión a causa de los recursos.

La utilidad poolstat incluye opciones que se pueden utilizar para examinar agrupaciones específicas y mostrar estadísticas relativas al conjunto de recursos.

Si se implementan zonas en el sistema y utiliza poolstat en una zona no global, se muestra información sobre los recursos asociados con la agrupación de la zona.

Para obtener más información sobre la utilidad poolstat, consulte la página del comando [man poolstat\(1M\)](#). Para obtener información sobre la tarea poolstat y su uso, consulte

“Uso de `poolstat` para registrar estadísticas para los recursos relacionados con las agrupaciones” [169].

Salida de `poolstat`

En el formato de salida predeterminado, `poolstat` crea una línea de cabecera y luego muestra una línea para cada agrupación. Una línea de agrupación empieza con el ID y el nombre de la agrupación, seguido de una columna de datos estadísticos sobre el conjunto de procesadores asociado a la agrupación. Los conjuntos de recursos asociados a más de una agrupación se enumeran varias veces, una para cada agrupación.

Las cabeceras de columnas son:

<code>id</code>	ID de la agrupación.
<code>pool</code>	Nombre de la agrupación.
<code>rid</code>	ID del conjunto de recursos.
<code>rset</code>	Nombre del conjunto de recursos.
<code>type</code>	Tipo del conjunto de recursos.
<code>min</code>	Tamaño mínimo del conjunto de recursos.
<code>max</code>	Tamaño máximo del conjunto de recursos.
<code>size</code>	Tamaño actual del conjunto de recursos.
<code>used</code>	Cantidad del conjunto de recursos que está en uso. Este uso se calcula como un porcentaje de la utilización del conjunto de recursos multiplicado por el tamaño del conjunto de recursos. Si se ha reconfigurado un recurso durante el último intervalo de muestra, es posible que no se muestre este valor. Un valor no registrado aparece como un guión (-).
<code>load</code>	Representación absoluta de la carga que se coloca en el conjunto de recursos. Para obtener más información sobre esta propiedad, consulte la página del comando <code>man libpool(3LIB)</code> .

Puede especificar lo siguiente en la salida de `poolstat`:

- El orden de las columnas
- Las cabeceras que aparecen

Ajuste de los intervalos de funcionamiento de `poolstat`

Puede personalizar las operaciones que lleva a cabo `poolstat`. Puede configurar el intervalo de muestreo para el informe y especificar el número de veces que se repiten las estadísticas:

<i>interval</i>	Ajuste los intervalos para las operaciones periódicas que lleva a cabo <code>poolstat</code> . Todos los intervalos se especifican en segundos.
<i>count</i>	Especifique el número de veces que se repiten las estadísticas. De modo predeterminado, <code>poolstat</code> sólo muestra las estadísticas una vez.

Si no se especifican *intervalo* ni *número*, las estadísticas sólo se muestran una vez. Si se especifica *intervalo* pero no *número*, las estadísticas se muestran de modo indefinido.

Comandos utilizados con la función de agrupaciones de recursos

Los comandos descritos en la siguiente tabla proporcionan la principal interfaz administrativa para la utilidad de agrupaciones. Para obtener información sobre el uso de estos comandos en un sistema con zonas activadas, consulte [“Agrupaciones de recursos utilizadas en zonas” \[125\]](#).

Referencia de página del comando <code>man</code>	Descripción
pooladm(1M)	Activa o desactiva la función de agrupaciones en el sistema. Activa una configuración específica o elimina la configuración actual y devuelve el estado predeterminado a los recursos asociados. Si se ejecuta sin opciones, <code>pooladm</code> imprime la configuración actual de las agrupaciones dinámicas.
poolbind(1M)	Permite la vinculación manual de proyectos, tareas y procesos a una agrupación de recursos.
poolcfg(1M)	Proporciona operaciones de configuración para las agrupaciones y los conjuntos. Las configuraciones creadas con esta herramienta se instancian en un sistema de destino utilizando <code>pooladm</code> . Si se ejecuta con el argumento de subcomando <code>info</code> para la opción <code>-c</code> , <code>poolcfg</code> muestra información sobre la configuración estática en <code>/etc/pooladm.conf</code> . Si se agrega un argumento de nombre de archivo, este comando muestra información sobre la configuración estática del archivo con nombre. Por ejemplo, <code>poolcfg -c info /tmp/newconfig</code> muestra información sobre la configuración estática que contiene el archivo <code>/tmp/newconfig</code> .
poolld(1M)	El daemon del sistema de agrupaciones. El daemon utiliza los destinos del sistema y las estadísticas observables para mantener los objetivos de rendimiento

Referencia de página del comando man	Descripción
	del sistema que especifique el administrador. Si no puede emprender la acción correctiva necesaria cuando no se cumplan los objetivos, <code>pool</code> registra la condición.
<code>poolstat(1M)</code>	Muestra estadísticas para los recursos relacionados con las agrupaciones. Simplifica el análisis de rendimiento y proporciona información de gran utilidad para los administradores del sistema en cuanto a la partición de los recursos y las tareas. Se han incorporado opciones para examinar las agrupaciones especificadas y mostrar las estadísticas específicas de los conjuntos de recursos.

`libpool` proporciona una API de biblioteca (consulte la página del comando man `libpool(3LIB)`). Los programas pueden utilizar la biblioteca para manipular las configuraciones de agrupaciones.

BETA
DRAFT

13

♦ ♦ ♦ C A P Í T U L O 13

Creación y administración de las tareas de agrupaciones de recursos

En este capítulo se describe cómo configurar y administrar agrupaciones de recursos en el sistema.

Para obtener información sobre agrupaciones de recursos, consulte [Capítulo 12, Acerca de las agrupaciones de recursos](#).

Administración del mapa de tareas de agrupaciones de recursos

Tarea	Descripción	Instrucciones
Activar o desactivar las agrupaciones de recursos.	Activa o desactiva las agrupaciones de recursos del sistema.	“Activación y desactivación de la función de agrupaciones” [150]
Activar o desactivar las agrupaciones de recursos dinámicos.	Activa o desactiva las funciones de agrupaciones de recursos dinámicos del sistema.	“Activación y desactivación de la función de agrupaciones” [150]
Crear una configuración de agrupaciones de recursos estáticos.	Crea un archivo de configuración estática que coincide con la configuración dinámica actual. Para obtener más información, consulte “Estructura de agrupaciones de recursos” [127] .	Cómo crear una configuración estática [155]
Modificar una configuración de agrupaciones de recursos.	Revisa una configuración de agrupaciones del sistema, por ejemplo, creando agrupaciones adicionales.	Cómo modificar una configuración [157]
Asociar una agrupación de recursos con una clase de planificación.	Asocia una agrupación con una clase de planificación para que todos los procesos vinculados a la agrupación utilicen el planificador especificado.	Cómo asociar una agrupación con una clase de planificación [159]
Definir restricciones y objetivos de configuración.	Especifica los objetivos que debe tener en cuenta el sistema a la hora de emprender una acción correctiva. Para obtener más información sobre	Cómo establecer restricciones de configuración [161] and Cómo definir los objetivos de configuración [161]

Tarea	Descripción	Instrucciones
	los objetivos de configuración, consulte “Descripción general de pool” [132].	
Definir el nivel de registro.	Especifica el nivel de información de registro que genera pool.	Cómo establecer el nivel de registro de pool [164]
Usar un archivo de texto con el comando poolcfg.	El comando poolcfg puede obtener información de un archivo de texto.	Cómo utilizar los archivos de comando con poolcfg [164]
Transferir recursos en el núcleo.	Transferir recursos en el núcleo. Por ejemplo, transfiere recursos con ID específicos a un conjunto de destino.	“Transferencia de recursos” [165]
Activar una configuración de agrupaciones.	Activa la configuración en el archivo de configuración predeterminado.	Cómo activar una configuración de agrupaciones [166]
Validar una configuración de agrupaciones antes de confirmar la configuración.	Valida una configuración de agrupaciones para probar qué ocurrirá cuando tenga lugar la validación.	Cómo validar una configuración antes de confirmarla [166]
Eliminar una configuración de agrupaciones del sistema.	Todos los recursos asociados, como los conjuntos de procesadores, se devuelven a su estado predeterminado.	Cómo eliminar una configuración de agrupaciones [167]
Vincular procesos a una agrupación.	Asociar manualmente un proceso en ejecución en el sistema con una agrupación de recursos.	Cómo vincular procesos a una agrupación [168]
Vincular tareas o proyectos a una agrupación.	Asocia tareas o proyectos con una agrupación de recursos.	Cómo vincular tareas o proyectos a una agrupación [168]
Vincular procesos nuevos a una agrupación de recursos.	Para vincular automáticamente los procesos nuevos de un proyecto a una agrupación específica, agregue un atributo a cada entrada de la base de datos project.	Cómo configurar el atributo project.pool para un proyecto [168]
Utilizar el atributo project para vincular un proceso a una agrupación diferente.	Modifica la vinculación de agrupaciones para los nuevos procesos que se inician.	Cómo utilizar los atributos project para vincular un proceso a una agrupación diferente [169]
Utilizar la utilidad poolstat para crear informes.	Crea múltiples informes en los intervalos especificados.	“Creación de varios informes en los intervalos especificados” [170]
Mostrar estadísticas del conjunto de recursos.	Utiliza la utilidad poolstat para mostrar estadísticas para un conjunto de recursos pset.	“Registro de estadísticas del conjunto de recursos” [170]

Activación y desactivación de la función de agrupaciones

Puede activar y desactivar los servicios de agrupaciones de recursos y de agrupaciones de recursos dinámicos en su sistema mediante el uso del comando `svcadm` descrito en la página del comando `man svcadm(1M)`.

También puede utilizar el comando `pooladm` descrito en la página del comando `man pooladm(1M)` para realizar las tareas siguientes:

- Activar la función de agrupaciones para poder manipular las agrupaciones
- Desactivar la función de agrupaciones para que no se puedan manipular las agrupaciones

Nota - Cuando se actualiza un sistema, si la estructura de las agrupaciones de recursos está activada y existe el archivo `/etc/pooladm.conf`, se activa el servicio de agrupaciones y la configuración que contiene el archivo se aplica al sistema.

▼ Cómo activar el servicio de agrupaciones de recursos utilizando svcadm

1. Conviértase en root o asuma un rol similar.
2. Active el servicio de agrupaciones de recursos.

```
# svcadm enable system/pools:default
```

▼ Cómo desactivar el servicio de agrupaciones de recursos utilizando svcadm

1. Conviértase en root o asuma un rol similar.
2. Desactive el servicio de agrupaciones de recursos.

```
# svcadm disable system/pools:default
```

▼ Cómo activar el servicio de agrupaciones de recursos dinámicos utilizando svcadm

1. Conviértase en root o asuma un rol similar.
2. Active el servicio de agrupaciones de recursos dinámicos.

```
# svcadm enable system/pools/dynamic:default
```

ejemplo 13-1 Dependencia del servicio de agrupaciones de recursos dinámicos del servicio de agrupaciones de recursos

Este ejemplo muestra que primero debe activar las agrupaciones de recursos si desea ejecutar las asignaciones de recursos dinámicos.

Existe una dependencia entre las agrupaciones de recursos y las agrupaciones de recursos dinámicos. La asignación de recursos dinámicos es ahora un servicio dependiente de las agrupaciones de recursos. La asignación de recursos dinámicos puede activarse y desactivarse de forma independiente de las agrupaciones de recursos.

A continuación se muestra que tanto las agrupaciones de recursos como las agrupaciones de recursos dinámicos están desactivadas:

```
# svcs "*pool*"
STATE      STIME      FMRI
disabled   2011      svc:/system/pools:default
disabled   2011      svc:/system/pools/dynamic:default
```

Active las agrupaciones de recursos dinámicos:

```
# svcadm enable svc:/system/pools/dynamic:default
# svcs -a | grep pool
STATE      STIME      FMRI
disabled   2011      svc:/system/pools:default
offline    2011      svc:/system/pools/dynamic:default
```

Observe que el servicio de asignaciones de recursos dinámicos sigue estando desactivado.

Utilice la opción -x del comando svcs para determinar por qué está desconectado el servicio de asignaciones de recursos dinámicos:

```
# svcs -x "*pool*"
svc:/system/pools:default (resource pools framework)
  State: disabled since Sat Feb 12 02:36:15 2011
  Reason: Disabled by an administrator.
    See: http://support.oracle.com/msg/SMF-8000-05
    See: libpool(3LIB)
    See: pooladm(1M)
    See: poolbind(1M)
    See: poolcfg(1M)
    See: poolstat(1M)
  Impact: This service is not running.

svc:/system/pools/dynamic:default (dynamic resource pools)
  State: disabled since Sat Feb 12 02:36:16 2011
  Reason: Disabled by an administrator.
    See: http://support.oracle.com/msg/SMF-8000-05
    See: poold(1M)
  Impact: This service is not running.
```

Active el servicio de agrupaciones de recursos para que se pueda ejecutar el servicio de asignaciones de recursos dinámicos:


```
# svcadm enable svc:/system/pools:default
```

Cuando se utiliza el comando `svcs "*pool*"`, el sistema muestra:

```
# svcs "*pool*"
STATE      STIME      FMRI
online     2011      svc:/system/pools/dynamic:default
online     2011      svc:/system/pools:default
```

ejemplo 13-2 Efecto en las agrupaciones de recursos dinámicos cuando el servicio de agrupaciones de recursos está desactivado

Si ambos servicios están en línea y desactiva el servicio de agrupaciones de recursos:

```
# svcadm disable svc:/system/pools:default
```

Cuando se utiliza el comando `svcs "*pool*"`, el sistema muestra:

```
# svcs "*pool*"
STATE      STIME      FMRI
disabled   2011      svc:/system/pools:default
online     2011      svc:/system/pools/dynamic:default
```

Sin embargo, el servicio DRP pasa a `offline` porque se desactivó el servicio de agrupaciones de recursos:

```
# svcs "*pool*"
STATE      STIME      FMRI
disabled   2011      svc:/system/pools:default
offline    2011      svc:/system/pools/dynamic:default
```

Determine por qué está desconectado el servicio de asignaciones de recursos dinámicos:

```
# svcs -x "*pool*"
svc:/system/pools:default (resource pools framework)
State: disabled since Sat Feb 12 02:36:15 2011
Reason: Disabled by an administrator.
  See: http://support.oracle.com/msg/SMF-8000-05
  See: libpool(3LIB)
  See: pooladm(1M)
  See: poolbind(1M)
  See: poolcfg(1M)
  See: poolstat(1M)
Impact: 1 dependent service is not running. (Use -v for list.)

svc:/system/pools/dynamic:default (dynamic resource pools)
State: offline since Sat Feb 12 02:36:15 2011
Reason: Service svc:/system/pools:default is disabled.
  See: http://support.oracle.com/msg/SMF-8000-GE
  See: poold(1M)
  See: /var/svc/log/system-pools-dynamic:default.log
Impact: This service is not running.
```

Para que funcionen las asignaciones de recursos dinámicos, es necesario haber iniciado la agrupaciones de recursos. Por ejemplo, para iniciar las agrupaciones de recursos podría utilizarse el comando `pooladm` con la opción `-e`:

```
# pooladm -e
```

A continuación, el comando `svcs` `"*pool*"` muestra:

```
# svcs "*pool*"
STATE      STIME    FMRI
online     2011    svc:/system/pools:default
online     2011    svc:/system/pools/dynamic:default
```

▼ Cómo desactivar el servicio de agrupaciones de recursos dinámicos utilizando svcadm

1. Conviértase en root o asuma un rol similar.
2. Desactive el servicio de agrupaciones de recursos dinámicos.

```
# svcadm disable system/pools/dynamic:default
```

▼ Cómo activar las agrupaciones de recursos utilizando pooladm

1. Conviértase en root o asuma un rol similar.
2. Active la función de agrupaciones.

```
# pooladm -e
```

▼ Cómo desactivar las agrupaciones de recursos utilizando pooladm

1. Conviértase en root o asuma un rol similar.
2. Desactive la función de agrupaciones.

```
# pooladm -d
```

Asignación de CPU específica

Puede assign y unassign CPU, núcleos centrales y sockets.

`cpus=` Lista de CPU asignadas a la zona.
`cores=` Lista de núcleos centrales asignados a la zona.
`sockets=` Lista de sockets asignados a la zona.

EJEMPLO 13-3 Asignar núcleos centrales

Asignar los núcleos centrales a pset *new*.

```
# poolcfg -dc 'assign to pset new (core 0 ; core 1)'
```

EJEMPLO 13-4 Actualizar la agrupación en ejecución para que coincida con la configuración estática persistente

Utilice el comando `zonecfg` para asignar núcleos centrales. Utilice el comando `pooladm` con la opción `-c` para que las agrupaciones en ejecución coincidan con la configuración estática.

```
# poolcfg -c 'assign to pset new (core 0 ; core 1)'  
# pooladm -c
```

Configuración de agrupaciones

▼ **Cómo crear una configuración estática**

Utilice la opción `-s` para `/usr/sbin/pooladm` con el fin de crear un archivo de configuración estática que coincida con la configuración dinámica actual, y preserve los cambios después de los reinicios. A menos que se especifique un nombre de archivo diferente, se utiliza la ubicación predeterminada `/etc/pooladm.conf`.

Confirme su configuración utilizando el comando `pooladm` con la opción `-c`. A continuación, utilice el comando `pooladm` con la opción `-s` para actualizar la configuración estática para que coincida con el estado de la configuración dinámica.

Nota - La última funcionalidad `pooladm -s` tiene prioridad sobre la funcionalidad anterior `poolcfg -c discover` en el momento de crear una nueva configuración que coincida con la configuración dinámica.

Antes de empezar Active las agrupaciones en el sistema.

1. **Conviértase en root o asuma un rol similar.**
2. **Actualice el archivo de configuración estática para que coincida con la configuración dinámica actual.**

```
# pooladm -s
```

3. **Visualice el contenido del archivo de configuración en un formato legible.**

Tenga en cuenta que la configuración contiene elementos predeterminados creados por el sistema.

```
# poolcfg -c info
system tester
  string  system.comment
  int     system.version 1
  boolean system.bind-default true
  int     system.poolid.pid 177916

  pool pool_default
    int     pool.sys_id 0
    boolean pool.active true
    boolean pool.default true
    int     pool.importance 1
    string  pool.comment
    pset    pset_default

  pset pset_default
    int     pset.sys_id -1
    boolean pset.default true
    uint    pset.min 1
    uint    pset.max 65536
    string  pset.units population
    uint    pset.load 10
    uint    pset.size 4
    string  pset.comment
    boolean testnullchanged true

  cpu
    int     cpu.sys_id 3
    string  cpu.comment
    string  cpu.status on-line

  cpu
    int     cpu.sys_id 2
    string  cpu.comment
    string  cpu.status on-line

  cpu
    int     cpu.sys_id 1
    string  cpu.comment
    string  cpu.status on-line
```

```
cpu
    int    cpu.sys_id 0
    string cpu.comment
    string cpu.status on-line
```

4. **Confirme la configuración en `/etc/pooladm.conf`.**

```
# pooladm -c
```

5. **(Opcional) Para copiar la configuración dinámica en un archivo de configuración estática denominado `/tmp/backup`, escriba:**

```
# pooladm -s /tmp/backup
```

▼ Cómo modificar una configuración

Para mejorar la configuración, cree un juego de procesadores denominado `pset_batch` y una agrupación denominada `pool_batch`. A continuación, asocie la agrupación con el conjunto de procesadores.

Tenga en cuenta que debe incluir entre comillas los argumentos de subcomandos que contengan espacios.

1. **Conviértase en root o asuma un rol similar.**

2. **Cree el conjunto de procesadores `pset_batch`.**

```
# poolcfg -c 'create pset pset_batch (uint pset.min = 2; uint pset.max = 10)'
```

3. **Cree la agrupación `pool_batch`.**

```
# poolcfg -c 'create pool pool_batch'
```

4. **Asocie la agrupación con el conjunto de procesadores.**

```
# poolcfg -c 'associate pool pool_batch (pset pset_batch)'
```

5. **Visualice la configuración editada.**

```
# poolcfg -c info
system tester
    string system.comment kernel state
    int    system.version 1
    boolean system.bind-default true
    int    system.poold.pid 177916

    pool pool_default
        int    pool.sys_id 0
```

```
boolean pool.active true
boolean pool.default true
int    pool.importance 1
string pool.comment
pset   pset_default

pset pset_default
int    pset.sys_id -1
boolean pset.default true
uint   pset.min 1
uint   pset.max 65536
string pset.units population
uint   pset.load 10
uint   pset.size 4
string pset.comment
boolean testnullchanged true

cpu
    int    cpu.sys_id 3
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 2
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 1
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 0
    string cpu.comment
    string cpu.status on-line

pool pool_batch
boolean pool.default false
boolean pool.active true
int    pool.importance 1
string pool.comment
pset   pset_batch

pset pset_batch
int    pset.sys_id -2
string pset.units population
boolean pset.default true
uint   pset.max 10
uint   pset.min 2
string pset.comment
boolean pset.escapable false
uint   pset.load 0
uint   pset.size 0
```

```
cpu
    int    cpu.sys_id 5
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 4
    string cpu.comment
    string cpu.status on-line
```

6. **Confirme la configuración en `/etc/pooladm.conf`.**

```
# pooladm -c
```

7. **(Opcional) Para copiar la configuración dinámica en un archivo de configuración estática denominado `/tmp/backup`, escriba:**

```
# pooladm -s /tmp/backup
```

▼ Cómo asociar una agrupación con una clase de planificación

Puede asociar una agrupación con una clase de planificación para que todos los procesos vinculados a la agrupación utilicen este programador. Para ello, configure la propiedad `pool.scheduler` con el nombre del programador. Este ejemplo asocia la agrupación `pool_batch` con el programador de reparto justo (FSS).

1. **Conviértase en root o asuma un rol similar.**
2. **Modifique la agrupación `pool_batch` que se asociará con FSS.**

```
# poolcfg -c 'modify pool pool_batch (string pool.scheduler="FSS")'
```

3. **Visualice la configuración editada.**

```
# poolcfg -c info
system tester
    string system.comment
    int    system.version 1
    boolean system.bind-default true
    int    system.poold.pid 177916

pool pool_default
    int    pool.sys_id 0
    boolean pool.active true
    boolean pool.default true
    int    pool.importance 1
    string pool.comment
```

```
pset    pset_default

pset pset_default
  int    pset.sys_id -1
  boolean pset.default true
  uint   pset.min 1
  uint   pset.max 65536
  string pset.units population
  uint   pset.load 10
  uint   pset.size 4
  string pset.comment
  boolean testnullchanged true

  cpu
    int    cpu.sys_id 3
    string cpu.comment
    string cpu.status on-line

  cpu
    int    cpu.sys_id 2
    string cpu.comment
    string cpu.status on-line

  cpu
    int    cpu.sys_id 1
    string cpu.comment
    string cpu.status on-line

  cpu
    int    cpu.sys_id 0
    string cpu.comment
    string cpu.status on-line

pool pool_batch
  boolean pool.default false
  boolean pool.active true
  int pool.importance 1
  string pool.comment
  string pool.scheduler FFS
  pset batch

pset pset_batch
  int pset.sys_id -2
  string pset.units population
  boolean pset.default true
  uint pset.max 10
  uint pset.min 2
  string pset.comment
  boolean pset.escapable false
  uint pset.load 0
  uint pset.size 0

  cpu
    int    cpu.sys_id 5
```



```
string  cpu.comment
string  cpu.status on-line

cpu
int     cpu.sys_id 4
string  cpu.comment
string  cpu.status on-line
```

4. **Confirme la configuración en `/etc/pooladm.conf`:**

```
# pooladm -c
```

5. **(Opcional) Para copiar la configuración dinámica en un archivo de configuración estática denominado `/tmp/backup`, escriba:**

```
# pooladm -s /tmp/backup
```

▼ Cómo establecer restricciones de configuración

Las restricciones afectan a las posibles configuraciones al eliminar algunos de los posibles cambios que podrían realizarse en una configuración. Este procedimiento muestra cómo definir la propiedad `cpu.pinned`.

En los ejemplos siguientes, `cpuid` es un entero.

1. **Conviértase en root o asuma un rol similar.**

2. **Modifique la propiedad `cpu.pinned` en la configuración estática o dinámica:**

■ **Modifique la configuración de tiempo de inicio (estática):**

```
# poolcfg -c 'modify cpu <cpuid> (boolean cpu.pinned = true)'
```

■ **Modifique la configuración en ejecución (dinámica) sin modificar la configuración del tiempo de inicio:**

```
# poolcfg -dc 'modify cpu <cpuid> (boolean cpu.pinned = true)'
```

▼ Cómo definir los objetivos de configuración

Puede especificar objetivos que debe tener en cuenta `poold` al llevar a cabo la acción correctiva.

En el procedimiento siguiente, se configura el objetivo `wt-load` para que `poold` intente asociar la asignación de recursos con el uso de recursos. El objetivo `locality` está desactivado para poder alcanzar este objetivo de configuración.

1. Conviértase en root o asuma un rol similar.

2. Modifique el sistema tester para favorecer el objetivo wt-load.

```
# poolcfg -c 'modify system tester (string system.pool objectives="wt-load")'
```

3. Desactive el objetivo locality para el conjunto de procesadores predeterminado.

```
# poolcfg -c 'modify pset pset_default (string pset.pool objectives="locality none")' one line
```

4. Desactive el objetivo locality para el conjunto de procesadores pset_batch.

```
# poolcfg -c 'modify pset pset_batch (string pset.pool objectives="locality none")' one line
```

5. Visualice la configuración editada.

```
# poolcfg -c info
system tester
  string system.comment
  int system.version 1
  boolean system.bind-default true
  int system.pool.pid 177916
  string system.pool objectives wt-load

pool pool_default
  int pool.sys_id 0
  boolean pool.active true
  boolean pool.default true
  int pool.importance 1
  string pool.comment
  pset pset_default

pset pset_default
  int pset.sys_id -1
  boolean pset.default true
  uint pset.min 1
  uint pset.max 65536
  string pset.units population
  uint pset.load 10
  uint pset.size 4
  string pset.comment
  boolean testnullchanged true
  string pset.pool objectives locality none

cpu
  int cpu.sys_id 3
  string cpu.comment
  string cpu.status on-line

cpu
  int cpu.sys_id 2
```

```
        string  cpu.comment
        string  cpu.status on-line

    cpu
        int     cpu.sys_id 1
        string  cpu.comment
        string  cpu.status on-line

    cpu
        int     cpu.sys_id 0
        string  cpu.comment
        string  cpu.status on-line

    pool pool_batch
        boolean pool.default false
        boolean pool.active true
        int     pool.importance 1
        string  pool.comment
        string  pool.scheduler FSS
        pset batch

    pset pset_batch
        int     pset.sys_id -2
        string  pset.units population
        boolean pset.default true
        uint    pset.max 10
        uint    pset.min 2
        string  pset.comment
        boolean pset.escapable false
        uint    pset.load 0
        uint    pset.size 0
        string  pset.poolid.objectives locality none

    cpu
        int     cpu.sys_id 5
        string  cpu.comment
        string  cpu.status on-line

    cpu
        int     cpu.sys_id 4
        string  cpu.comment
        string  cpu.status on-line
```

6. Confirme la configuración en `/etc/pooladm.conf`.

```
# pooladm -c
```

7. (Opcional) Para copiar la configuración dinámica en un archivo de configuración estática denominado `/tmp/backup`, escriba:

```
# pooladm -s /tmp/backup
```

▼ Cómo establecer el nivel de registro de poold

Para especificar el nivel de información de registro que genera poold, defina la propiedad `system.pold.log-level` en la configuración de poold. La configuración de poold se realiza en la configuración de libpool. Para obtener información, consulte [“Información de registro de poold” \[138\]](#) y las páginas del comando `man poolcfg(1M)` y `libpool(3LIB)`.

También puede utilizar el comando poold en la línea de comandos para especificar el nivel de información de registro que genera poold.

1. **Conviértase en root o asuma un rol similar.**
2. **Defina el nivel de registro utilizando el comando poold con la opción `-l` y un parámetro, por ejemplo, `INFO`.**

```
# /usr/lib/pool/pold -l INFO
```

Para obtener información sobre los parámetros disponibles, consulte [“Información de registro de poold” \[138\]](#). El nivel de registro predeterminado es NOTICE.

▼ Cómo utilizar los archivos de comando con poolcfg

El comando poolcfg con la opción `-f` puede obtener información de un archivo de texto que contenga argumentos de subcomando poolcfg para la opción `-c`. Este método resulta adecuado si desea establecer las operaciones que se llevarán a cabo. Cuando se procesan varios comandos, la configuración sólo se actualiza si todos los comandos son correctos. En el caso de las configuraciones complejas o de mayor tamaño, esta técnica puede ser más útil que las invocaciones por subcomandos.

Tenga en cuenta que los archivos de comando, el carácter `#` actúa como marca de comentario para el resto de la línea.

1. **Cree el archivo de entrada poolcmds.txt.**

```
$ cat > poolcmds.txt
create system tester
create pset pset_batch (uint pset.min = 2; uint pset.max = 10)
create pool pool_batch
associate pool pool_batch (pset pset_batch)
```

2. **Conviértase en root o asuma un rol similar.**
3. **Ejecute el comando:**

```
# /usr/sbin/poolcfg -f poolcmds.txt
```

Transferencia de recursos

Utilice el argumento de subcomando `transfer` para la opción `-c` de `poolcfg` con la opción `-d` para transferir recursos en el núcleo. La opción `-d` especifica que el comando actúe directamente en el núcleo y no obtenga información de un archivo.

El siguiente procedimiento mueve dos CPU del conjunto de procesadores `pset1` al conjunto de procesadores `pset2` del núcleo.

▼ Cómo mover CPU entre conjuntos de procesadores

1. **Conviértase en root o asuma un rol similar.**
2. **Mueva dos CPU de `pset1` a `pset2`.**

Las subcláusulas `from` y `to` pueden utilizarse en cualquier orden. Sólo se admite una subcláusula `to` y `from` por comando.

```
# poolcfg -dc 'transfer 2 from pset pset1 to pset2'
```

ejemplo 13-5 Método alternativo para mover CPU entre conjuntos de procesadores

Si deben transferirse ID conocidos específicos de un tipo de recurso, se proporciona una sintaxis alternativa. Por ejemplo, el siguiente comando asigna dos CPU con los ID 0 y 2 al conjunto de procesadores `pset_large`:

```
# poolcfg -dc 'transfer to pset pset_large (cpu 0; cpu 2)'
```

Solución de problemas

Si falla una transferencia porque no hay suficientes recursos para la solicitud o porque no se puede encontrar los ID especificados, el sistema emitirá un mensaje de error.

Activación y eliminación de configuraciones de agrupaciones

Utilice el comando `pooladm` para activar una configuración de agrupación concreta o para eliminar la configuración de agrupación activa. Consulte la página del comando [man pooladm\(1M\)](#) para obtener más información sobre este comando.

▼ Cómo activar una configuración de agrupaciones

Para activar la configuración en el archivo de configuración predeterminado, `/etc/pooladm.conf`, invoque `pooladm` con la opción `-c` de confirmación de la configuración.

1. **Conviértase en root o asuma un rol similar.**
2. **Confirme la configuración en `/etc/pooladm.conf`.**
3. **(Opcional) Copie la configuración dinámica en un archivo de configuración estática, por ejemplo, `/tmp/backup`.**

```
# pooladm -c
```

```
# pooladm -s /tmp/backup
```

▼ Cómo validar una configuración antes de confirmarla

Puede utilizar la opción `-n` con la opción `-c` para probar qué ocurrirá cuando tenga lugar la validación. La configuración no se confirmará realmente.

El siguiente comando intenta validar la configuración de `/home/admin/newconfig`. Se muestra cualquier condición de error, pero no se modifica la configuración.

1. **Conviértase en root o asuma un rol similar.**
2. **Pruebe la validez de la configuración antes de confirmarla.**

```
# pooladm -n -c /home/admin/newconfig
```

▼ Cómo eliminar una configuración de agrupaciones

Para eliminar la configuración activa y devolver su estado predeterminado a todos los recursos asociados, como los juegos de procesadores, utilice la opción `-x` para eliminar la configuración.

1. **Conviértase en root o asuma un rol similar.**
2. **Elimine la configuración activa.**

```
# pooladm -x
```

La opción `-x` para `pooladm` elimina todos los elementos definidos por el usuario de la configuración dinámica. Todos los recursos vuelven a su estado predeterminado, y todos los vínculos de agrupaciones se sustituyen con un vínculo a la agrupación predeterminada.

Mezcla de clases de planificación en un conjunto de procesadores

Puede mezclar procesos en las clases TS e IA en el mismo conjunto de procesadores de forma segura. Si mezcla otras clases de planificación en un conjunto de procesadores, puede obtener resultados inesperados. Si el uso de `pooladm -x` crea clases de planificación mezcladas en un conjunto de procesadores, utilice el comando `priocntl` para mover los procesos en ejecución a una clase de planificación diferente. Consulte [Cómo mover manualmente los procesos de la clase TS a la clase FSS \[102\]](#). También puede consultar la página del comando `man priocntl(1)`.

Configuración de atributos de agrupaciones y vinculación a una agrupación

Puede configurar un atributo `project.pool` para asociar una agrupación de recursos con un proyecto.

Puede vincular un proceso en ejecución a una agrupación de dos maneras:

- Puede utilizar el comando `poolbind` que se describe en el comando [poolbind\(1M\)](#) para vincular un proceso específico a una agrupación de recursos concreta.
- Puede utilizar el atributo `project.pool` en la base de datos `project` para identificar la vinculación de agrupaciones para una nueva tarea o sesión de inicio que se abre con el comando `newtask`. Consulte las páginas del comando `man newtask(1)`, [projmod\(1M\)](#) y [project\(4\)](#).

▼ Cómo vincular procesos a una agrupación

El procedimiento siguiente utiliza `poolbind` con la opción `-p` para vincular manualmente un proceso (en este caso, el shell activo) a una agrupación denominada `ohare`.

1. **Conviértase en root o asuma un rol similar.**
2. **Vincule manualmente un proceso a una agrupación:**

```
# poolbind -p ohare $$
```
3. **Compruebe la vinculación de la agrupación para el proceso utilizando `poolbind` con la opción `-q`.**

```
$ poolbind -q $$  
155509 ohare
```

El sistema muestra el ID de proceso y la vinculación de la agrupación.

▼ Cómo vincular tareas o proyectos a una agrupación

Para vincular tareas o proyectos a una agrupación, utilice el comando `poolbind` con la opción `-i`. El ejemplo siguiente vincula todos los procesos del proyecto `airmiles` a la agrupación `laguardia`.

1. **Conviértase en root o asuma un rol similar.**
2. **Vincule todos los procesos del proyecto `airmiles` a la agrupación `laguardia`.**

```
# poolbind -i project -p laguardia airmiles
```

▼ Cómo configurar el atributo `project.pool` para un proyecto

Puede definir el atributo `project.pool` para vincular los procesos de un proyecto a una agrupación de recursos.

1. **Conviértase en root o asuma un rol similar.**
2. **Agregue un atributo `project.pool` a cada entrada de la base de datos `project`.**


```
# projmod -a -K project.pool=poolname project
```

▼ Cómo utilizar los atributos `project` para vincular un proceso a una agrupación diferente

Supongamos que tiene una configuración con dos agrupaciones denominadas `studio` y `backstage`. El archivo `/etc/project` incluye lo siguiente:

```
user.paul:1024:::project.pool=studio
user.george:1024:::project.pool=studio
user.ringo:1024:::project.pool=backstage
passes:1027::paul::project.pool=backstage
```

Con esta configuración, los procesos que inicia el usuario `paul` están vinculados de forma predeterminada a la agrupación `studio`.

El usuario `paul` puede modificar la vinculación de la agrupación para los procesos que inicia. `paul` también puede utilizar `newtask` para vincular trabajos a la agrupación `backstage`, iniciando el proyecto `passes`.

1. Inicie un proceso en el proyecto `passes`.

```
$ newtask -l -p passes
```

2. Utilice el comando `poolbind` con la opción `-q` para verificar la vinculación de la agrupación para el proceso. Utilice también un símbolo de doble dólar (`$$`) para pasar el número de proceso del shell principal al comando.

```
$ poolbind -q $$
6384 pool backstage
```

El sistema muestra el ID de proceso y la vinculación de la agrupación.

Uso de `poolstat` para registrar estadísticas para los recursos relacionados con las agrupaciones

Se utiliza el comando `poolstat` para mostrar las estadísticas para los recursos relacionados con las agrupaciones. Para obtener más información, consulte [“Uso de `poolstat` para supervisar la función de agrupaciones y el uso de los recursos” \[144\]](#) y la página del comando `manpoolstat(1M)`.

Las siguientes subsecciones utilizan ejemplos para ilustrar el modo en que se producen informes para fines específicos.

Visualización de la salida predeterminada de `poolstat`

Si escribe `poolstat` sin argumentos se crea una línea de cabecera y una línea de información para cada agrupación. La línea de información muestra el ID de agrupación, el nombre de la agrupación y las estadísticas de recursos para el conjunto de procesadores asociado con la agrupación.

```
machine% poolstat
```

		pset		
id	pool	size	used	load
0	pool_default	4	3.6	6.2
1	pool_sales	4	3.3	8.4

Creación de varios informes en los intervalos especificados

El comando siguiente crea tres informes a intervalos de muestreo de 5 segundos.

```
machine% poolstat 5 3
```

		pset		
id	pool	size	used	load
46	pool_sales	2	1.2	8.3
0	pool_default	2	0.4	5.2

		pset		
id	pool	size	used	load
46	pool_sales	2	1.4	8.4
0	pool_default	2	1.9	2.0

		pset		
id	pool	size	used	load
46	pool_sales	2	1.1	8.0
0	pool_default	2	0.3	5.0

Registro de estadísticas del conjunto de recursos

El ejemplo siguiente utiliza el comando `poolstat` con la opción `-r` para registrar las estadísticas para el conjunto de recursos del conjunto de procesadores. Tenga en cuenta que el conjunto de recursos `pset_default` está asociado a más de una agrupación, de modo que este procesador se enumera una vez para cada agrupación.

```
machine% poolstat -r pset
```

id	pool	type	rid	rset	min	max	size	used	load
0	pool_default	pset	-1	pset_default	1	65K	2	1.2	8.3
6	pool_sales	pset	1	pset_sales	1	65K	2	1.2	8.3
2	pool_other	pset	-1	pset_default	1	10K	2	0.4	5.2

BETA
DRAFT

BETA
DRAFT

14

♦ ♦ ♦ C A P Í T U L O 1 4

Ejemplo de configuración de administración de recursos

En este capítulo se revisa la estructura de administración de los recursos y describe un proyecto de consolidación de servidores hipotético.

En este capítulo, se describen los siguientes temas:

- “Configuración que consolidar” [173]
- “Configuración de consolidación” [174]
- “Creación de la configuración” [174]
- “Visualización de la configuración” [176]

Configuración que consolidar

En este ejemplo se están consolidando cinco aplicaciones en un único sistema. Las aplicaciones de destino tienen diferentes requisitos de recursos, poblaciones de usuarios y arquitecturas. Actualmente, cada aplicación se encuentra en un servidor dedicado diseñado para satisfacer los requisitos de la aplicación. En la tabla siguiente se identifican las aplicaciones y sus características.

Descripción de la aplicación	Características
Application Server	Muestra una escalabilidad negativa más allá de 2 CPU
Instancia de base de datos para servidor de aplicaciones	Procesamiento de transacciones densas
Servidor de aplicaciones en entorno de prueba y desarrollo	Basado en GUI, con ejecución de código no probado
Servidor de procesamiento de transacciones	Se encarga principalmente del tiempo de respuesta
Instancia de base de datos autónoma	Procesa una gran cantidad de transacciones y sirve a varias zonas horarias

Configuración de consolidación

La siguiente configuración se utiliza para consolidar las aplicaciones en un único sistema que tiene activadas las utilidades de agrupaciones de recursos y agrupaciones de recursos dinámicos.

- El servidor de aplicaciones tiene un conjunto de procesadores de dos CPU.
- La instancia de base de datos del servidor de aplicaciones y la instancia de base de datos autónoma se consolidan en un único conjunto de procesadores de como mínimo cuatro CPU. La instancia de base de datos autónoma tiene garantizado un 75 por ciento de dicho recurso.
- El servidor de aplicaciones de prueba y desarrollo requiere la clase de planificación IA para garantizar la respuesta de la IU. Se imponen limitaciones de memoria para atenuar los efectos del código mal generado.
- Al servidor de proceso de transacciones se le asigna un conjunto de procesadores dedicados de como mínimo dos CPU, con el fin de minimizar la latencia de respuesta.

Esta configuración se asigna a aplicaciones conocidas que ejecutan y consumen ciclos de procesador en cada conjunto de recursos. De este modo, pueden establecerse limitaciones que permiten transferir el recurso del procesador a conjuntos en los que se requiere el recurso.

- El objetivo de `wt-load` es permitir a los conjuntos de recursos que se utilizan con gran frecuencia recibir mayores asignaciones de recursos que los conjuntos que se utilizan menos.
- El objetivo de `locality` se configura como `tight`, que se utiliza para maximizar la localidad del procesador.

También se aplica una limitación adicional para evitar que el uso supere el 80 por ciento de cualquier conjunto de recursos. Esta limitación garantiza que todas las aplicaciones tengan acceso a los recursos que necesitan. Asimismo, para el conjunto de procesadores de transacciones, el objetivo de mantener el uso por debajo del 80 por ciento es el doble de importante que para cualquier otro objetivo especificado. Esta importancia se establece en la configuración.

Creación de la configuración

Edite el archivo de base de datos `/etc/project`. Añada entradas para implementar los controles de recursos necesarios y para asignar usuarios a los agrupaciones de recursos y, a continuación, visualizar el archivo.

```
# cat /etc/project
.
```

```
.
user.app_server:2001:Production Application Server::project.pool=appserver_pool
user.app_db:2002:App Server DB::project.pool=db_pool;project.cpu-shares=(privileged,1,deny)
development:2003:Test and development::staff:project.pool=dev_pool;
process.max-address-space=(privileged,536870912,deny)    keep with previous line
user.tp_engine:2004:Transaction Engine::project.pool=tp_pool
user.geo_db:2005:EDI DB::project.pool=db_pool;project.cpu-shares=(privileged,3,deny)
.
.
.
```

Nota - El equipo de desarrollo debe ejecutar tareas en el proyecto de desarrollo porque el acceso a este proyecto se basa en un ID de grupo de usuario (GID).

Cree un archivo de entrada llamado `pool.host`, que se utilizará para configurar las agrupaciones de recursos necesarias. Visualice el archivo.

```
# cat pool.host
create system host
create pset dev_pset (uint pset.min = 0; uint pset.max = 2)
create pset tp_pset (uint pset.min = 2; uint pset.max=8)
create pset db_pset (uint pset.min = 4; uint pset.max = 6)
create pset app_pset (uint pset.min = 1; uint pset.max = 2)
create pool dev_pool (string pool.scheduler="IA")
create pool appserver_pool (string pool.scheduler="TS")
create pool db_pool (string pool.scheduler="FSS")
create pool tp_pool (string pool.scheduler="TS")
associate pool dev_pool (pset dev_pset)
associate pool appserver_pool (pset app_pset)
associate pool db_pool (pset db_pset)
associate pool tp_pool (pset tp_pset)
modify system tester (string system.poolid.objectives="wt-load")
modify pset dev_pset (string pset.poolid.objectives="locality tight; utilization < 80")
modify pset tp_pset (string pset.poolid.objectives="locality tight; 2: utilization < 80")
modify pset db_pset (string pset.poolid.objectives="locality tight;utilization < 80")
modify pset app_pset (string pset.poolid.objectives="locality tight; utilization < 80")
```

Actualice la configuración utilizando el archivo de entrada `pool.host`.

```
# poolcfg -f pool.host
```

Active la configuración.

```
# pooladm -c
```

La estructura pasará a estar operativa en el sistema.

Habilite DRP.

```
# svcadm enable pools/dynamic:default
```

Visualización de la configuración

Para ver la configuración de la estructura, que también contiene elementos predeterminados creados por el sistema, escriba:

```
# pooladm
system host
    string  system.comment
    int     system.version 1
    boolean system.bind-default true
    int     system.poold.pid 177916
    string  system.poold.objectives wt-load

pool dev_pool
    int     pool.sys_id 125
    boolean pool.default false
    boolean pool.active true
    int     pool.importance 1
    string  pool.comment
    string  pool.scheduler IA
    pset    dev_pset

pool appserver_pool
    int     pool.sys_id 124
    boolean pool.default false
    boolean pool.active true
    int     pool.importance 1
    string  pool.comment
    string  pool.scheduler TS
    pset    app_pset

pool db_pool
    int     pool.sys_id 123
    boolean pool.default false
    boolean pool.active true
    int     pool.importance 1
    string  pool.comment
    string  pool.scheduler FSS
    pset    db_pset

pool tp_pool
    int     pool.sys_id 122
    boolean pool.default false
    boolean pool.active true
    int     pool.importance 1
    string  pool.comment
    string  pool.scheduler TS
    pset    tp_pset

pool pool_default
    int     pool.sys_id 0
    boolean pool.default true
    boolean pool.active true
```



```
    int    pool.importance 1
    string pool.comment
    string pool.scheduler TS
    pset   pset_default

pset dev_pset
    int    pset.sys_id 4
    string pset.units population
    boolean pset.default false
    uint   pset.min 0
    uint   pset.max 2
    string pset.comment
    boolean pset.escapable false
    uint   pset.load 0
    uint   pset.size 0
    string pset.pool objectives locality tight; utilization < 80

pset tp_pset
    int    pset.sys_id 3
    string pset.units population
    boolean pset.default false
    uint   pset.min 2
    uint   pset.max 8
    string pset.comment
    boolean pset.escapable false
    uint   pset.load 0
    uint   pset.size 0
    string pset.pool objectives locality tight; 2: utilization < 80

cpu
    int    cpu.sys_id 1
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 2
    string cpu.comment
    string cpu.status on-line

pset db_pset
    int    pset.sys_id 2
    string pset.units population
    boolean pset.default false
    uint   pset.min 4
    uint   pset.max 6
    string pset.comment
    boolean pset.escapable false
    uint   pset.load 0
    uint   pset.size 0
    string pset.pool objectives locality tight; utilization < 80

cpu
    int    cpu.sys_id 3
    string cpu.comment
```

```
        string  cpu.status on-line

cpu
    int      cpu.sys_id 4
    string   cpu.comment
    string   cpu.status on-line

cpu
    int      cpu.sys_id 5
    string   cpu.comment
    string   cpu.status on-line

cpu
    int      cpu.sys_id 6
    string   cpu.comment
    string   cpu.status on-line

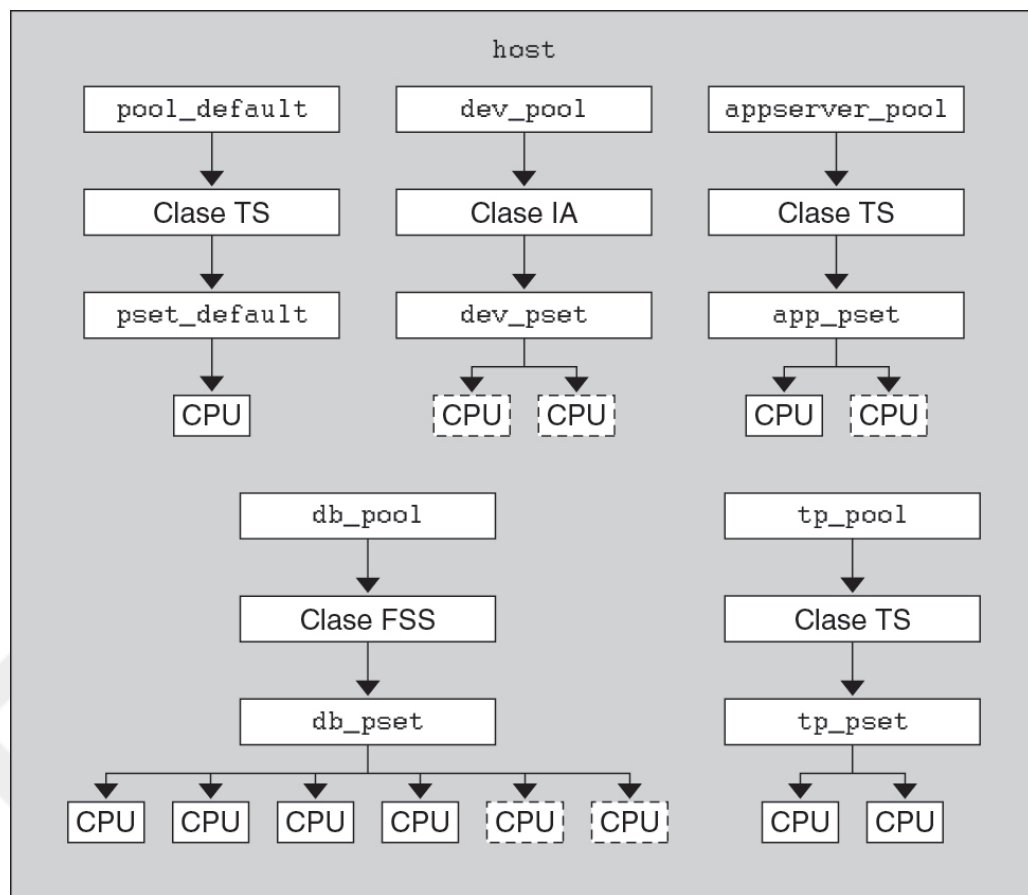
pset app_pset
    int      pset.sys_id 1
    string   pset.units population
    boolean  pset.default false
    uint     pset.min 1
    uint     pset.max 2
    string   pset.comment
    boolean  pset.escapable false
    uint     pset.load 0
    uint     pset.size 0
    string   pset.pool objectives locality tight; utilization < 80
    cpu
        int      cpu.sys_id 7
        string   cpu.comment
        string   cpu.status on-line

pset pset_default
    int      pset.sys_id -1
    string   pset.units population
    boolean  pset.default true
    uint     pset.min 1
    uint     pset.max 4294967295
    string   pset.comment
    boolean  pset.escapable false
    uint     pset.load 0
    uint     pset.size 0

    cpu
        int      cpu.sys_id 0
        string   cpu.comment
        string   cpu.status on-line
```

A continuación se incluye una representación gráfica de la estructura.

FIGURA 14-1 Configuración de la consolidación del servidor



Nota - En la agrupación db_pool, la instancia de base de datos autónoma tiene garantizado el 75 por ciento de los recursos de la CPU.

BETA
DRAFT

Índice

Números y símbolos

/etc/project
 archivo, 23
 formato de entrada, 25

A

activación de agrupaciones de recursos, 150
 activación de agrupaciones de recursos dinámicos, 150
 activación de la contabilidad ampliada, 52
 activación de limitación de recursos, 118
 actualización temporal de controles de recursos, 73
 administración de agrupaciones de recursos, 146
 administración de recursos
 definición, 13
 limitaciones, 15
 partición, 16
 planificación, 15
 agrupación de recursos predeterminada, 124
 agrupaciones, 124
 agrupaciones de recursos, 124
 /etc/pooladm.conf, 127
 activación, 150
 activación de configuración, 166
 administración, 146
 configuración de agrupaciones estáticas, 127
 creación, 131
 desactivación, 150
 elementos de configuración, 128
 eliminación, 167
 eliminación de configuración, 167
 implementación, 129
 propiedades, 129
 reconfiguración dinámica, 130
 vinculación a, 167
 agrupaciones de recursos dinámicos

activación, 150
 desactivación, 150
 archivo exacct, 44
 archivo/etc/user_attr, 22
 atributo
 project.pool, 130
 atributo project.pool, 130
 atributo rcap.max-rss, 106

B

base de datos project, 23
 biblioteca libexacct, 44

C

cambio temporal de controles de recursos, 73
 clases de programación, 96
 comando acctadm, 53
 comando rcapadm, 107
 comando rcapstat, 111
 comandos
 contabilidad ampliada, 48
 controles de recursos, 74
 programador de reparto justo (FSS), 97
 proyectos y tareas, 29
 comunicación entre procesos (IPC) *Ver* controles de recursos
 configuración
 rcapd, 107
 configuración de agrupaciones dinámicas, 127
 configuración de controles de recursos, 61
 configuración de recursos compartidos de la CPU, 92
 configurar atributos de agrupaciones de recursos, 167
 consolidación de servidores, 17

contabilidad ampliada

- activación, 52
- comandos, 48
- contracargo, 44
- descripción general, 43
- estado, visualización, 53
- formato de archivo, 44
- SMF, 46

controles de recursos

- acciones globales, 68
- acciones locales, 61, 69
- actualización temporal, 73
- cambio temporal, 73
- comunicación entre procesos (IPC), 60
- configuración, 61
- definición, 59
- descripción general, 59
- lista de, 62
- valor inf , 71
- valores umbral, 61, 68, 69

creación de agrupaciones de recursos, 131

D

- daemon de limitación de recursos, 105
- desactivación de agrupaciones de recursos, 150
- desactivación de agrupaciones de recursos dinámicos, 150
- desactivación de limitación de recursos, 119
- directorio /var/adm/exacct , 46
- DRP, 125

E

- eliminación de agrupaciones de recursos, 167

F

- formato de entrada
 - archivo /etc/project, 25
- FSS Ver planificador por reparto equitativo (FSS)

G

- gestión de recursos

- tareas, 27

I

- implementación de agrupaciones de recursos, 129
- interfaz Perl, 48

J

- juego de procesadores predeterminado, 124

L

- limitación de recursos
 - activación, 118
 - desactivación, 119
- límite de recursos, 105
- límites de recursos, 60
- llamada del sistema putacct, 45

M

- módulo de autenticación conectable Ver PAM

N

- niveles de privilegio, valores umbral, 67

P

- PAM (módulo de autenticación conectable)
 - gestión de identidades, 24
- planificador por reparto equitativo (FSS), 88
 - configuración , 101
 - project.cpu-shares, 88
- poold
 - ámbito de control, 143
 - asignación de recursos dinámicos, 125
 - componentes configurables, 138
 - descripción, 132
 - información de registro, 138
 - infracción del control asíncrono, 143, 143

- objetivos, 134
- propiedad `cpu-pinned`, 134
- restricciones, 133
- `poolstat`
 - descripción, 144
 - ejemplos de uso, 169
 - formato de salida, 145
- programador de reparto justo
 - y conjuntos de procesadores, 93
- programador de reparto justo (FSS)
 - definición de recurso compartido, 88
- `project.cpu-shares`, 92
- proyecto
 - con ningún recurso compartido, 88
 - definición, 22
 - estado activo, 89
 - estado inactivo, 89
- proyecto 0, 93
- proyecto predeterminado, 22
- `projectssystem` *Ver* proyecto 0

R

- `rcapd`
 - configuración, 107
 - intervalos de análisis, 111
 - intervalos de muestra, 111
- `rcapd daemon`, 105
- `rctl`, 59 *Ver* controles de recursos
- `rlimits` *Ver* límites de recursos

T

- tareas
 - gestión de recursos, 27

U

- umbral de aplicación de límite de memoria, 108

V

- valores umbral, controles de recursos, 67
- vinculación a la agrupación de recursos, 167

- visualización del estado de contabilidad ampliada, 53

BETA
DRAFT