

Oracle® Solaris 11.2 カーネルのチューンアップ・リファレンスマニュアル

ORACLE®

Part No: E53991-02
2014 年 12 月

Copyright © 2000, 2014, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

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.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ, AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

このドキュメントの使用方法	11
1 Oracle Solaris システムのチューニングの概要	13
Oracle Solaris 11.2 システムチューニングの新機能	13
Oracle Solaris システムのチューニング	13
チューニング可能パラメータの説明形式	14
Oracle Solaris カーネルのチューニング	16
/etc/system ファイルおよび /etc/system.d ディレクトリ	16
kldb コマンド	18
mdb コマンド	18
Oracle Solaris の特殊な tune および var 構造体	19
Oracle Solaris システム構成情報の表示	19
sysdef コマンド	20
kstat ユーティリティ	20
2 Oracle Solaris カーネルチューニング可能パラメータ	21
カーネルとメモリの一般的なパラメータ	22
physmem	22
default_stksize	23
lwp_default_stksize	24
logevent_max_q_sz	25
segkpsize	25
noexec_user_stack	26
fsflush とそれに関連するパラメータ	27
fsflush	27
tune_t_fsflushr	28
autoup	29
dopageflush	30
doiflush	30

プロセス規模調整パラメータ	31
maxusers	31
reserved_procs	33
pidmax	33
max_nprocs	34
maxuprc	35
ngroups_max	35
ページング関連パラメータ	36
lotsfree	38
desfree	39
minfree	40
throttlefree	41
pageout_reserve	41
pages_pp_maximum	42
tune_t_minarmem	43
fastscan	44
slowscan	45
min_percent_cpu	45
handsreadpages	46
pages_before_pager	46
maxpgio	47
スワッピング関連パラメータ	48
swapfs_reserve	48
swapfs_minfree	49
カーネルメモリアロケータ	50
kmem_flags	50
kmem_stackinfo	51
一般的なドライバパラメータ	52
moddebug	52
ddi_msix_alloc_limit	54
ネットワークドライバパラメータ	54
カーネル内の IP プロトコルパラメータ	54
igb パラメータ	56
ixgbe Parameters	57
一般的な入出力パラメータ	61
maxphys	61
rlim_fd_max	62
rlim_fd_cur	63

一般的なファイルシステムパラメータ	64
ncsize	64
dnlc_dir_enable	65
dnlc_dir_min_size	65
dnlc_dir_max_size	66
dnlc_dircache_percent	66
TMPFS パラメータ	67
tmpfs:tmpfs_maxkmem	67
tmpfs:tmpfs_minfree	68
仮想端末	68
pt_cnt	69
pt_pctofmem	70
pt_max_pty	71
STREAMS パラメータ	71
nstrpush	71
strmsgsz	72
strctlsz	72
System V メッセージキュー	73
System V セマフォ	73
System V 共有メモリー	73
segspt_minfree	74
pr_segp_disable	74
スケジューリング	76
disp_rechoose_interval	76
タイマー	77
hires_tick	77
timer_max	78
SPARC: プラットフォーム固有のパラメータ	78
tsb_alloc_hiwater_factor	78
default_tsb_size	79
enable_tsb_rss_sizing	80
tsb_rss_factor	80
近傍性グループのパラメータ	81
lpg_alloc_prefer	81
lgrp_mem_pset_aware	82
3 Oracle Solaris ZFS チューニング可能パラメータ	85
ZFS のチューニングの考慮事項	85

ZFS ARC パラメータ	86
zfs_arc_min	86
zfs_arc_max	87
ZFS ファイルレベルプリフェッチ	87
zfs_prefetch_disable	87
ZFS デバイスの入出力キューの深さ	88
zfs_vdev_max_pending	88
フラッシュストレージ使用時の ZFS のチューニング	90
ZFS ログデバイスまたはキャッシュデバイスとしてのフラッシュデバイスの追加	91
フラッシュおよび NVRAM ストレージデバイスの適切なキャッシュフラッシュ動作の保証	92
データベース製品に対する ZFS のチューニング	95
Oracle データベース用の ZFS のチューニング	95
ZFS を MySQL と一緒に使用するときの考慮事項	100
4 NFS チューニング可能パラメータ	101
NFS 環境のチューニング	101
NFS モジュールのパラメータ	102
nfs:nfs3_pathconf_disable_cache	102
nfs:nfs_allow_preepoch_time	102
nfs:nfs_cots_timeo	103
nfs:nfs3_cots_timeo	104
nfs:nfs4_cots_timeo	104
nfs:nfs_do_symlink_cache	105
nfs:nfs3_do_symlink_cache	106
nfs:nfs_dynamic	106
nfs:nfs3_dynamic	107
nfs:nfs_lookup_neg_cache	108
nfs:nfs3_lookup_neg_cache	108
nfs:nfs4_lookup_neg_cache	109
nfs:nfs_max_threads	110
nfs:nfs3_max_threads	111
nfs:nfs4_max_threads	112
nfs:nfs_nra	113
nfs:nfs3_nra	113
nfs:nrnode	114
nfs:nfs_shrinkreaddir	115

nfs:nfs3_shrinkreaddir	116
nfs:nfs_write_error_interval	117
nfs:nfs_write_error_to_cons_only	117
nfs:nfs_disable_rddir_cache	118
nfs:nfs3_bsize	119
nfs:nfs4_bsize	120
nfs:nfs_async_clusters	120
nfs:nfs3_async_clusters	121
nfs:nfs4_async_clusters	122
nfs:nfs_async_timeout	123
nfs:nacache	124
nfs:nfs3_jukebox_delay	125
nfs:nfs3_max_transfer_size	125
nfs:nfs4_max_transfer_size	126
nfs:nfs3_max_transfer_size_clts	127
nfs:nfs3_max_transfer_size_cots	128
NFS 関連の SMF 構成パラメータ	129
server_authz_cache_refresh	129
netgroup_refresh	129
rpcmod モジュールパラメータ	129
rpcmod:clnt_max_conns	130
rpcmod:clnt_idle_timeout	130
rpcmod:svc_idle_timeout	131
rpcmod:svc_default_stksize	131
rpcmod:maxdupreqs	132
rpcmod:cotsmaxdupreqs	133
5 インターネットプロトコル群のチューニング可能パラメータ	135
IP パラメータのチューニングの概要	135
IP パラメータの妥当性検証	136
RFC (Internet Request for Comments)	136
IP チューニング可能パラメータ	136
_icmp_err_interval と _icmp_err_burst	136
_respond_to_echo_broadcast と _respond_to_echo_multicast (ipv4 または ipv6)	137
send_redirects (ipv4 or ipv6)	137
forwarding (ipv4 または ipv6)	138
ttl	138

hoplimit (ipv6)	138
_addr_per_if	139
hostmodel (ipv4 または ipv6)	139
重複アドレスの検出に関連した IP チューニング可能パラメータ	140
特別な注意を要する IP チューニング可能パラメータ	146
TCP チューニング可能パラメータ	147
_deferred_ack_interval	147
_local_dack_interval	147
_deferred_acks_max	148
_local_dacks_max	148
_wscale_always	149
_tstamp_always	149
send_buf	150
recv_buf	150
max_buf	151
_cwnd_max	151
_slow_start_initial	152
_local_slow_start_initial	152
_slow_start_after_idle	152
sack	153
_rev_src_routes	153
_time_wait_interval	154
ecn	154
_conn_req_max_q	155
_conn_req_max_q0	156
_conn_req_min	156
_rst_sent_rate_enabled	157
_rst_sent_rate	157
特別な注意を要する TCP パラメータ	159
UDP チューニング可能パラメータ	163
send_buf	163
recv_buf	163
max_buf	164
smallest_anon_port	164
largest_anon_port	164
IPQoS チューニング可能パラメータ	165
_policy_mask	165
SCTP チューニング可能パラメータ	166

_max_init_retr	166
_pa_max_retr	166
_pp_max_retr	167
_cwnd_max	167
_ipv4_ttl	167
_heartbeat_interval	168
_new_secret_interval	169
_initial_mtu	169
_deferred_ack_interval	169
_ignore_path_mtu	170
_initial_ssthresh	170
send_buf	170
_xmit_lowat	171
recv_buf	171
max_buf	172
_rto_min	172
_rto_max	172
_rto_initial	173
_cookie_life	173
_max_in_streams	173
_initial_out_streams	174
_shutack_wait_bound	174
_maxburst	175
_addip_enabled	175
_prsctp_enabled	175
smallest_anon_port	176
largest_anon_port	176
ルート別のメトリック	177
6 システム機能のパラメータ	179
システムのデフォルトのパラメータ	179
autofs	179
cron	180
devfsadm	180
dhcpgent	180
fs	180
ftp	181

inetinit	181
init	181
ipsec	182
kbd	182
keyserv	182
login	183
mpathd	183
nfs	183
nfslogd	183
nss	183
passwd	184
su	184
syslog	184
tar	184
telnetd	184
utmpd	185
A システムチェックスクリプト	187
システムのフラッシュ動作の確認	187
索引	189

このドキュメントの使用方法

- **概要** - Oracle Solaris OS のカーネルおよびネットワークのチューニング可能パラメータに関する参照情報を提供します。このマニュアルは、デスクトップシステムやJava環境に関するチューニング可能パラメータの情報は提供しません。
- **対象者** - 状況によってはカーネルのチューニング可能パラメータを変更する必要があるシステム管理者。
- **前提知識** - Oracle Solaris または UNIX の基本的なシステム管理の経験、および一般的なファイルシステム管理の経験。

製品ドキュメントライブラリ

この製品の最新情報や既知の問題は、ドキュメントライブラリ (<http://www.oracle.com/pls/topic/lookup?ctx=solaris11>) に含まれています。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> (聴覚に障害をお持ちの場合は <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>) を参照してください。

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

◆◆◆ 第 1 章

Oracle Solaris システムのチューニングの概要

このセクションでは、このマニュアルで使用するチューニング情報の記載形式の概要を示します。また、このセクションでは Oracle Solaris システムの別のチューニング方法についても説明します。

- [13 ページの「Oracle Solaris 11.2 システムチューニングの新機能」](#)
- [13 ページの「Oracle Solaris システムのチューニング」](#)
- [14 ページの「チューニング可能パラメータの説明形式」](#)
- [16 ページの「Oracle Solaris カーネルのチューニング」](#)
- [19 ページの「Oracle Solaris の特殊な tune および var 構造体」](#)
- [19 ページの「Oracle Solaris システム構成情報の表示」](#)
- [20 ページの「kstat ユーティリティー」](#)

Oracle Solaris 11.2 システムチューニングの新機能

このセクションでは、Oracle Solaris 11.2 リリースで追加または変更されたパラメータについて説明します。

- NFS SMF サーバーのチューニング可能なパラメータの情報については、[129 ページの「NFS 関連の SMF 構成パラメータ」](#)で説明します。
- Oracle Solaris ZFS のフラッシュストレージのチューニング可能パラメータ情報は、[90 ページの「フラッシュストレージ使用時の ZFS のチューニング」](#)にあります。

Oracle Solaris システムのチューニング

Oracle Solaris オペレーティングシステムは、システム負荷に簡単に適応するため、最小限のチューニングで済みます。それでも、場合によってはチューニングが必要になることもあります。

このドキュメントでは、Oracle Solaris OS で利用可能な、公式にサポートされているチューニングオプションについて詳しく説明します。

Oracle Solaris カーネルは、常にロードされているコア部分と、参照された時点でロードされる多数のロード可能モジュールとで構成されています。このガイドで示すカーネルパラメータの多くはコアパラメータです。ただし、ロード可能なモジュールに属するパラメータもいくつかあります。

パフォーマンスを向上させるためにシステムパラメータをチューニングするのは、多くの場合、もっとも効率の悪い方法である点に注意してください。アプリケーションを改善およびチューニングするほうが、効果の高い方法です。また、物理メモリーを増やしたり、ディスクの入出力パターンのバランスをとることでパフォーマンスを向上させることもできます。システムパラメータを変更することでパフォーマンスが大幅に向上することはめったにありません。

あるシステムの `/etc/system` 設定値が全体として、または部分的に、別のシステムの環境に当てはまらないこともあるということを忘れないでください。したがって、使用する環境に応じて、このファイルに設定する値を慎重に検討する必要があります。このドキュメントにリストされているシステム変数を変更する場合は、システムの動作を前もって理解していなければなりません。

Oracle Solaris システムをチューニングするには、空のファイルを作成します。ファイルには会社固有の名前を付け、ファイル名のコンポーネントをコロンで区切ります。たとえば、`MyCompany:kernel:configurations` などになります。最初のステップとしては、自社製またはサードパーティー製のアプリケーションで必要とされるチューニング可能パラメータだけを追加してください。基準検査の確立後に、システムパフォーマンスを評価して、チューニング可能パラメータの追加設定が必要かどうかを決定します。



注意 - このドキュメントで説明するチューニング可能パラメータは、Oracle Solaris のリリースごとに変更される可能性があります。これらのチューニング可能パラメータを公開することによって、予告なくチューニング可能パラメータやその説明が変更されることがなくなるわけではありません。

チューニング可能パラメータの説明形式

このセクションでは、Oracle Solaris パラメータをチューニングする形式について説明します。

<i>Parameter</i>	<code>/etc/system</code> ファイルに入力されるか、または <code>/etc/default/facility</code> ファイルに指定されているとおりの名前。 一部のパラメータでは、そのパラメータがロード可能なモジュールに属していることを示すために、 <code>module:parameter</code> という命名規則を使用し
------------------	--

	ます。たとえば、 <code>tmpfs:tmpfs_maxkmem</code> は、 <code>tmpfs_maxkmem</code> が <code>tmpfs</code> モジュールのパラメータであることを意味します。
説明	パラメータが何を行うのか、何を制御するのかという簡単な説明。
データ型	符号付きまたは符号なし short 整数または long 整数を指定。long 整数のビット幅は整数の 2 倍です。例：符号なし整数 = 32 ビット、符号なし long 整数 = 64 ビット
単位	(オプション) 単位の種類を表します。
デフォルト	システムがデフォルトで使用する値を示します。
範囲	システムの検証で取り得る範囲や、データ型の上下限を表す範囲です。 <ul style="list-style-type: none"> ■ MAXINT - 符号付き整数の最大値 (2,147,483,647) を表します。 ■ MAXUINT - 符号なし整数の最大値 (4,294,967,295) を表します。
動的か	パラメータが、 <code>mdb</code> または <code>kmdb</code> デバッガを使用して稼働中のシステム上で構成可能なのか (Yes)、ブート時の初期化中のみ構成可能なのか (No) を示します。
検証	システムが、 <code>/etc/system</code> ファイルに指定されたとおりの値とデフォルトの値のいずれを変数の値に適用するか調べます。また、検証がいつ適用されるかも示します。
暗黙的制約	(オプション) パラメータに対する暗黙的な制約事項 (特に他のパラメータとの関係において) を表します。
どのような場合に 変更するか	この値を変更したくなる理由について説明します。エラーメッセージまたは戻りコードが含まれます。
ゾーン構成	パラメータを排他的 IP ゾーン内で設定できるか、または大域ゾーン内で設定する必要があるかを識別します。共有 IP ゾーン内で設定できるパラメータはありません。
コミットレベル	インタフェースの安定性を表します。このマニュアルで記述するパラメータの多くは「発展中 (Evolving)」または「変更の可能性あり (Unstable)」のいずれかに分類されます。詳細は、 attributes(5) を参照してください。

Oracle Solaris カーネルのチューニング

次の表では、パラメータに適用可能なチューニングの方法を示します。

チューニング可能パラメータの適用方法	参照先
/etc/system.d ディレクトリ内の構成ファイルにパラメータを設定します。	16 ページの「 /etc/system ファイルおよび /etc/system.d ディレクトリ 」
カーネルデバッグ (kldb) を使用します。	18 ページの「 kldb コマンド 」
モジュールデバッグ (mdb) を使用します。	18 ページの「 mdb コマンド 」
ipadm コマンドを使用して、TCP/IP パラメータを設定します。	第5章「 インターネットプロトコル群のチューニング可能パラメータ 」
/etc/default のファイルを変更します。	第6章「 システム機能のパラメータ 」

/etc/system ファイルおよび /etc/system.d ディレクトリ

/etc/system ファイルは、カーネルパラメータの値を静的に調整するメカニズムを提供します。このファイルに指定された値は、ブート時に読み込まれ適用されます。このファイルに対する変更は、システムがリブートされるまでオペレーティングシステムに適用されません。

構成パラメータが計算される前に、すべての値を設定するために 1 回のパスが行われます。

注記 - パラメータをチューニングするには、/etc/system.d ディレクトリ内の構成ファイルにパラメータ値を設定します。/etc/system ファイルを直接変更しないでください。

例 1-1 特定のシステム用の ZFS パラメータの設定

次のエントリでは、ZFS ARC の最大値 (zfs_arc_max) を 30G バイトに設定します。

```
set zfs:zfs_arc_max = 0x78000000
```

会社の名前が Widget, Inc. であるとして、このエントリを widget:zfs または、/etc/system.d ディレクトリ内の同様の名前のファイルに格納します。システムがブートすると、/etc/system.d 内のすべてのパラメータ構成が /etc/system ファイルに追加されます。これにより、システムは /etc/system の内容に従って構成されます。

適正でない値からの復元

間違った値を元に戻すには、次のいずれかの方法を実行します。

`/etc/system.d/file` 内のパラメータのリセット

`/etc/system.d` ディレクトリ内の構成ファイルから欠陥のあるパラメータ設定を削除します。ブート時に、`/etc/system` ファイルが以前の構成で更新され、それらがシステムに再適用されます。

クローニングされたブート環境を使用する

システムパラメータの変更を導入する前に、まずブート環境のクローンを作成してください。

```
# beadm create BE-cloneName
```

その後、`/etc/system` に変更を適用したあと現在の BE が使用不能になった場合は、システムをリブートします。x86 GRUB メニューまたは SPARC ブートメニューで、「BE clone」を選択します。ブートが完了したあと、以降のシステムブートで使用されるデフォルトの BE になるように、オプションで BE クローンをアクティブ化できます。

ファイルコピーを使用する

正しくない値から簡単に回復できるように、`/etc/system.d` ディレクトリ内の構成ファイルの新しいパラメータで更新する前に、`/etc/system` ファイルのコピーを作成します。例:

```
# cp /etc/system /etc/system.good
```

`/etc/system.d` 内の構成ファイルに指定した値が原因でシステムがブートできなくなった場合は、次のコマンドで回復できます。

```
ok boot -a
```

このコマンドを実行すると、ブートプロセスで使用する各ファイルの名前をシステムから要求されます。`/etc/system` ファイルの名前が要求されるまで Return キーを押して、デフォルトの値を適用します。Name of system file [/etc/system]: というプロンプトが表示されたら、正しい `/etc/system` ファイルの名前かまたは `/dev/null` を入力します。

```
Name of system file [/etc/system]: /etc/system.good
```

`/dev/null` を指定した場合は、このパスによってシステムは `/dev/null` から構成情報を読み取ろうとします。このファイルは空なので、システムはデフォルト値を使用することになります。システムがブートしたあと、`/etc/system` ファイルを修正できます。

システム回復についての詳細は、『Oracle Solaris 11.2 でのシステム管理のトラブルシューティング』を参照してください。

kldb コマンド

kldb は対話式カーネルデバッガであり、その一般的な構文は mdb と同じです。対話式カーネルデバッガの利点は、ブレークポイントを設定できることです。ブレークポイントに達すると、データを検証し、カーネルコードの手順を 1 つずつ実行できます。

kldb は必要に応じてロードしたりロード解除したりできます。対話的にカーネルをデバッグするためにシステムをリブートする必要はありません。kadb は必要でした。

詳細は、[kldb\(1\)](#) を参照してください。

mdb コマンド

モジュラーデバッガ mdb は、簡単に拡張できるため、Solaris デバッガの中では珍しいものです。このデバッガのプログラミング API を使用して、モジュールをコンパイルすることによって、デバッガのコンテキスト内で希望するタスクを実行することができます。

さらに、mdb には、コマンド行での編集、コマンド履歴、組み込み出力ページャ、構文チェック、コマンドパイプラインなどの、いくつかの便利な機能があります。カーネルに対する事後検査用のデバッガとしては、mdb をお勧めします。

詳細は、[mdb\(1\)](#) を参照してください。

例 1-2 mdb を使用した情報の表示

システムのメモリー使用量の概要を表すビューを表示します。例:

```
# mdb -k
Loading modules: [ unix genunix specfs dtrace mac cpu.generic
cpu_ms.AuthenticAMD.15 uppc pcplusmp scsi_vhci zfs mpt sd ip
hook neti arp usba sockfs kssl qlc fctl stmf stmf_sbd md lofs
random idm fcp crypto cpc smbsrv nfs fcip sPPP ufs logindmux
ptm nsmb scu mpt_sas pmcs emlxs ]
> ::memstat
Page Summary                      Pages                      MB    %Tot
-----
Kernel                              160876                      628    16%
ZFS File Data                        303401                      1185   30%
```

Anon	25335	98	2%
Exec and libs	1459	5	0%
Page cache	5083	19	1%
Free (cachelist)	6616	25	1%
Free (freelist)	510870	1995	50%
Total	1013640	3959	
Physical	1013639	3959	

> \$q

モジュラーデバッグの使用についての詳細は、『Oracle Solaris モジュラーデバッグ』を参照してください。

kldb デバッグまたは mdb デバッグを使用する場合、モジュール名の接頭辞は不要です。モジュールのロード後、そのシンボルはコアカーネルのシンボルやすでにロードされている他のモジュールのシンボルとともに共通の名前空間を形成するからです。

Oracle Solaris の特殊な tune および var 構造体

Oracle Solaris のチューニング可能パラメータはさまざまな形を取ります。/usr/include/sys/tuneable.h ファイルで定義された tune 構造体は、tune_t_fsflushr、tune_t_minarmem、および tune_t_flkrec の実行時の表現です。カーネルが初期設定されたあとは、これらの変数に対する参照はすべて、この tune 構造体の対応フィールドに入ります。

ブート時にこの構造体にパラメータを設定するには、必要なフィールド名に対応する特別なパラメータを初期設定する必要があります。そうすれば、これらの値がシステム初期設定プロセスで tune 構造体にロードされます。

複数のチューニング可能パラメータが置かれるもう 1 つの構造体に、v という名前の var 構造体があります。var 構造体の定義は /usr/include/sys/var.h ファイルにあります。autoup や bufhwm などの変数の実行時の状態はここに格納されます。

システムの動作中に tune 構造体や v 構造体を変更しないでください。システムの動作中にこれらの構造体のフィールドを変更すると、システムがパニックになることがあります。

Oracle Solaris システム構成情報の表示

システム構成情報を調べるツールはいくつかあります。ツールによっては、スーパーユーザー権限が必要です。それ以外のツールは、一般ユーザーの権限で実行できます。カーネルデバッグです

すべての構造体やデータアイテムを調べるには、動作中のシステム上で `mdb` を使うか、あるいは `kldb` でブートします。

詳細は、[mdb\(1\)](#) または [kadb\(1M\)](#) を参照してください。

sysdef コマンド

`sysdef` コマンドは、メモリとプロセスのリソース制限の値、および `tune` 構造体と `v` 構造体の一部を提供します。たとえば、500G バイトのメモリを備えた SPARC T3-4 システムの `sysdef` 「チューニング可能パラメータ」セクションは次のとおりです。

```
2206203904      maximum memory allowed in buffer cache (bufhwm)
65546           maximum number of processes (v.v_proc)
99             maximum global priority in sys class (MAXCLSYSPRI)
65541          maximum processes per user id (v.v_maxup)
30             auto update time limit in seconds (NAUTOUP)
25            page stealing low water mark (GPGSLO)
1             fsflush run rate (FSFLUSHR)
25            minimum resident memory for avoiding deadlock (MINARMEM)
25            minimum swapable memory for avoiding deadlock (MINASMEM)
```

詳細は、[sysdef\(1M\)](#) を参照してください。

kstat ユーティリティー

`kstat` データ構造体群は、さまざまなカーネルのサブシステムやドライバによって維持されています。この構造体群は、カーネル内のデータをユーザープログラムに提供するメカニズムを提供します。このメカニズムを利用する場合、プログラムはカーネルのメモリを読んだり、スーパーユーザー権限を持つ必要はありません。詳細は、[kstat\(1M\)](#) または [kstat\(3KSTAT\)](#) を参照してください。

◆◆◆ 第 2 章

Oracle Solaris カーネルチューニング可能パラメータ

この章では、ほとんどの Oracle Solaris カーネルチューニング可能パラメータについて説明します。

- 22 ページの「カーネルとメモリーの一般的なパラメータ」
- 27 ページの「`fsflush` とそれに関連するパラメータ」
- 31 ページの「プロセス規模調整パラメータ」
- 36 ページの「ページング関連パラメータ」
- 48 ページの「スワッピング関連パラメータ」
- 50 ページの「カーネルメモリアロケータ」
- 52 ページの「一般的なドライバパラメータ」
- 54 ページの「ネットワークドライバパラメータ」
- 61 ページの「一般的な入出力パラメータ」
- 64 ページの「一般的なファイルシステムパラメータ」
- 67 ページの「TMPFS パラメータ」
- 68 ページの「仮想端末」
- 71 ページの「STREAMS パラメータ」
- 73 ページの「System V メッセージキュー」
- 73 ページの「System V セマフォ」
- 73 ページの「System V 共有メモリー」
- 76 ページの「スケジューリング」
- 77 ページの「タイマー」
- 78 ページの「プラットフォーム固有のパラメータ」
- 81 ページの「近傍性グループのパラメータ」

ほかのタイプのチューニング可能パラメータについては、次を参照してください。

- Oracle Solaris ZFS チューニング可能パラメータ - [第3章「Oracle Solaris ZFS チューニング可能パラメータ」](#)
- NFS チューニング可能パラメータ - [第4章「NFS チューニング可能パラメータ」](#)
- インターネットプロトコルスイートのチューニング可能パラメータ - [第5章「インターネットプロトコル群のチューニング可能パラメータ」](#)
- システム機能のチューニング可能パラメータ - [第6章「システム機能のパラメータ」](#)

カーネルとメモリーの一般的なパラメータ

このセクションでは、物理メモリーやスタック構成に関連する一般的なカーネルパラメータについて説明します。ZFS 関連のメモリーパラメータについては、[第3章「Oracle Solaris ZFS チューニング可能パラメータ」](#)を参照してください。

phymem

説明	Oracle Solaris OS とファームウェアが把握されたあとで、メモリーの物理ページ数に関するシステム構成を変更します。
データ型	符号なし long
デフォルト	そのシステムで使用できる物理メモリーのページ数。これには、コアカーネルとそのデータが格納されているメモリーは含まれません。
範囲	1 からシステムの物理メモリーの総量まで
単位	ページ
動的か	いいえ
検証	なし
どのような場合に 変更するか	より少ない物理メモリーでシステムを実行したときの影響を調べたい場合。このパラメータに対しては、コアカーネルやそのデータ、その他のさまざまなデータ構造体 (起動処理の初期に割り当て) などに使用されるメモリーは考慮されません。したがって、phymem の値は、より小さなメモリー量を表すよう、想定したページ数より小さくすべきです。
コミットレベル	変更の可能性あり

default_stksize

説明	すべてのスレッドのデフォルトスタックサイズを指定します。default_stksize より小さいスタックサイズを指定してスレッドを作成することはできません。default_stksize が設定されている場合、それは lwp_default_stksize をオーバーライドします。24 ページの「lwp_default_stksize」も参照してください。
データ型	Integer
デフォルト	<ul style="list-style-type: none"> ■ sun4u プロセッサ搭載の SPARC システムでは PAGESIZE の 3 倍 ■ sun4v プロセッサ搭載の SPARC システムでは PAGESIZE の 4 倍 ■ x64 システムでは PAGESIZE の 5 倍。
範囲	<p>最小値はデフォルト値です。</p> <ul style="list-style-type: none"> ■ sun4u プロセッサ搭載の SPARC システムでは PAGESIZE の 3 倍 ■ sun4v プロセッサ搭載の SPARC システムでは PAGESIZE の 4 倍 ■ x64 システムでは PAGESIZE の 5 倍。 <p>最大値はデフォルト値の 32 倍です。</p>
単位	getpagesize パラメータから返された値の倍数単位で指定されたバイト数です。詳細は、 getpagesize(3C) を参照してください。
動的か	はい。変数の変更後、作成されるスレッドに影響があります。
検証	<p>8192 以上、262,144 (256 x 1024) 以下にする必要があります。また、システムページサイズの倍数でなければなりません。これらの条件が満たされないと、次のメッセージが表示されます。</p> <pre>Illegal stack size, Using N</pre> <p>N の値は、default_stksize のデフォルト値です。</p>
どのような場合に 変更するか	<p>スタック容量が足りないためにシステムがパニックになる場合。この問題を解決するもっともよい方法は、システムが容量を使い果たす原因を明らかにし、それを修正することです。</p> <p>デフォルトのスタックサイズを増やすと、ほとんどすべてのカーネルスレッドのスタックが大きくなるため、カーネルのメモリー使用量が不当に増加します。通常、そのスペースは使用されません。さらに、カーネルの使用量が増えると、同じメモリープールを使用する他のリソースの容量が少なくなるため、システムの作業を行う能力が低下するおそれがあります。副次的な影響として、カーネルが作成できるスレッドの数が少なくなります。し</p>

たがって、この方法は、根本的な原因が解消されるまでの一時的な回避策として使用すべきです。

コミットレベル 変更の可能性あり

lwp_default_stksize

説明	カーネルスレッドの作成時に呼び出しルーチンが明示的に使用サイズを提供しなかった場合に使用する、スタックの大きさのデフォルト値を指定します。指定されたスタックサイズは、1 ページのレッドゾーンだけ増分されます。
データ型	Integer
デフォルト	<ul style="list-style-type: none"> ■ デフォルトの SPARC スタックサイズは 3 ページ (3 x 8,192 = 24,576) + 8 KB (レッドゾーン) です。 ■ デフォルトの x64 スタックサイズは 5 ページ (5 x 4,096 = 20,480) + 4 KB (レッドゾーン) です。
範囲	<p>最小値はデフォルト値です。</p> <ul style="list-style-type: none"> ■ SPARCシステムでは PAGESIZE の 3 倍。 ■ x64 システムでは PAGESIZE の 5 倍。 <p>最大値はデフォルト値の 32 倍です。</p>
単位	getpagesize パラメータから返された値の倍数単位で指定されたバイト数です。詳細は、 getpagesize(3C) を参照してください。
動的か	はい。変数の変更後、作成されるスレッドに影響があります。
検証	<p>8192 以上、262,144 (256 x 1024) 以下にする必要があります。また、システムページサイズの倍数でなければなりません。これらの条件が満たされないと、次のメッセージが表示されます。</p> <pre>Illegal stack size, Using N</pre> <p>N の値は、lwp_default_stksize のデフォルト値です。</p>
どのような場合に 変更するか	<p>スタック容量が足りないためにシステムがパニックになる場合。この問題を解決するもっともよい方法は、システムが容量を使い果たす原因を明らかにし、それを修正することです。</p> <p>デフォルトのスタックサイズを増やすと、ほとんどすべてのカーネルスレッドのスタックが大きくなるため、カーネルのメモリー使用量が不当に増加します。通常、そのスペースは使用されません。さらに、カーネルの使用量</p>

が増えると、同じメモリープールを使用する他のリソースの容量が少なくなるため、システムの作業を行う能力が低下するおそれがあります。副次的な影響として、カーネルが作成できるスレッドの数が少なくなります。したがって、この方法は、根本的な原因が解消されるまでの一時的な回避策として使用すべきです。

コミットレベル 変更の可能性あり

logevent_max_q_sz

説明	キューに格納して <code>syseventd</code> デモンへの配信を待機させることのできる、システムイベントの最大数です。システムイベントキューのサイズがこの制限に達すると、ほかのシステムイベントをキューに入れることはできません。
データ型	Integer
デフォルト	5000
範囲	0 から MAXINT
単位	システムイベント
動的か	はい
検証	<code>ddi_log_sysevent</code> と <code>sysevent_post_event</code> によってシステムイベントが生成されるたびに、システムイベントフレームワークはこの値をチェックします。 詳細は、 ddi_log_sysevent(9F) および sysevent_post_event(3SYSEVENT) を参照してください。
どのような場合に 変更するか	システムイベントのログ、生成、または送信が失敗したことをエラーログメッセージが示す場合。
コミットレベル	変更の可能性あり

segkpsize

説明	利用できるページング可能なカーネルのメモリー量を指定します。このメモリーは主にカーネルスレッドのスタックに使用されます。この値を増やすと、スレッドの数を増やさないのであれば個々のスレッドでより大きなスタックが使用でき、あるいはより多くのスレッドを使用できるようになります。システムスレッドのデフォルトのスタックサイズについては、 24 ページの「lwp_default_stksize」 を参照してください。
----	--

- SPARC: このパラメータを変更するには /etc/system ファイルを編集します。
- x64: このパラメータは次の方法によってのみ変更できます。
 - カーネルデバッグを使用してブートする
 - システム起動プロセスの先頭にブレークポイントを設定する
 - 目的の値に設定する

データ型	符号なし long
デフォルト	2G バイト x nCPU/128 または物理メモリーの容量/256G バイトのどちらか小さい方
範囲	512M バイトから 64G バイト (SPARC) 200M バイトから 8G バイト (x64)
単位	ページ
動的か	いいえ
検証	<p>値は最小および最大サイズと比較されます。最小値に満たないか、または最大値を超えている場合は 2G バイトにリセットされます。その作用に関するメッセージが表示されます。</p> <p>SPARC システムでは、<code>segkpsize</code> 値は物理メモリーのサイズの 2 倍を超えることができません。x64 システムでは、値は物理メモリーのサイズを超えることができません。</p>
どのような場合に 変更するか	システム上で多数のプロセスをサポートしなければならない場合。このデフォルトサイズでは、65,535 個のカーネルスレッドに 32K バイトのスタックを作成できます。64 ビットカーネルのカーネルスタックサイズは、プロセスが 32 ビットプロセスでも 64 ビットプロセスでも同じです。
コミットレベル	変更の可能性あり

noexec_user_stack

説明	<p>スタックを実行不能として指定できるので、バッファオーバーフロー攻撃がいつそう困難になります。</p> <p>64 ビットカーネルが動作している Oracle Solaris システムでは、すべての 64 ビットアプリケーションのスタックがデフォルトで実行不能になります。このパラメータの設定は、32 ビットアプリケーションを実行不能にするために必要です。</p>
----	--

データ型	符号付き整数
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
単位	切り替え (オン/オフ)
動的か	はい。ただし、すでに実行中のプロセスに対しては無効です。値が設定されたあとに起動されたプロセスに対してのみ有効です。
検証	なし
どのような場合に 変更するか	アプリケーションが、 <code>mprotect</code> を使用してスタックを実行可能にすることなくスタックに実行可能コードを意図的に置いている場合を除き、この変数は常に有効にすべきです。詳細は、 mprotect(2) を参照してください。
コミットレベル	変更の可能性あり

fsflush とそれに関連するパラメータ

このセクションでは、`fsflush` とそれに関連するチューニング可能パラメータについて説明します。

fsflush

システムデーモン `fsflush` は定期的に行われ、主に次の 3 つのタスクを行います。

1. `fsflush` は呼び出されるたびに、一定期間が経過した汚れたファイルシステムページをディスクにフラッシュします。
2. `fsflush` は呼び出されるたびに、メモリーの一部を検証し、変更されたページをバッキングストアに書き出します。ページは、変更されており、かつ次の条件のどれにも該当しない場合に書き込まれます。
 - ページはカーネルページです
 - ページは使用されていません
 - ページがロックされています
 - ページにスワップデバイスが対応づけられています
 - ページが入出力操作に現在関与しています

この結果、書き込み権限に基づいて mmap でマッピングされ、かつ実際に変更されているファイルのページがフラッシュされます。

ページはバッキングストアにフラッシュされますが、それを使用しているプロセスとの接続は保たれます。フラッシュしておく、システムのメモリーが不足したときのページの再利用が簡単になります。これは、フラッシュ後にそのページが変更されていなければ、ページを回収する前にそのページをバッキングストアに書き出す必要がなくなり、遅延を避けられるからです。

3. fsflush はファイルシステムのメタデータをディスクに書き込みます。この書き込みは n 回目の呼び出しごとに行われます。 n はさまざまな構成変数から計算されます。詳細は、[28 ページの「tune_t_fsflushr」](#) と [29 ページの「autoup」](#) を参照してください。

次の機能を構成できます。

- 呼び出し頻度 (tune_t_fsflushr)
- メモリー走査を実行するかどうか (dopageflush)
- ファイルシステムデータのフラッシュを行うかどうか (doiflush)
- フラッシュシステムデータのフラッシュを実行する頻度 (autoup)

ほとんどのシステムでは、fsflush によって、メモリーの走査と、ファイルシステムメタデータの同期化を行うのが一般的です。システムの使用状況によっては、メモリーの走査はほとんど意味がなかったり、CPU 時間を使用しすぎることがあります。

tune_t_fsflushr

説明	fsflush の呼び出し間隔を秒数で指定します。
データ型	符号付き整数
デフォルト	1
範囲	1 から MAXINT
単位	秒
動的か	いいえ
検証	値がゼロ以下の場合、値は 1 にリセットされ、警告メッセージが出力されます。この確認が行われるのはブート時だけです。

どのような場合に
変更するか `autoup` パラメータを参照してください。

コミットレベル 変更の可能性あり

autoup

説明 個々の呼び出しでダーティーページに関して検査するメモリー量と、ファイルシステム同期操作の頻度を、`tune_t_flushr` とともに制御します。

さらに、`autoup` の値は、空リストからバッファーを書き出すかどうかの制御にも使用されます。`B_DELWRI` フラグが付いているバッファー (変更されているファイルコンテンツページを示す) は、空リストに置かれている時間が `autoup` 秒を超えると書き出されます。`autoup` の値を増やすと、バッファーがメモリーに置かれている時間が長くなります。

データ型 符号付き整数

デフォルト 30

範囲 1 から MAXINT

単位 秒

動的か いいえ

検証 `autoup` がゼロ以下の場合、30 に再設定され、警告メッセージが出力されます。この確認が行われるのはブート時だけです。

暗黙的制約 `autoup` は `tune_t_fsflushr` の整数倍でなければなりません。最小でも `autoup` は `tune_t_fsflushr` 値の 6 倍以上でなければなりません。そうでないと、`fsflush` が呼び出されるたびに余計なメモリーが走査されます。

`dopageflush` がゼロでない場合にメモリーを検査するには、全体のシステムページ数に `tune_t_fsflushr` を掛け合わせた値が `autoup` 以上でなければなりません。

どのような場合に
変更するか `autoup` または `tune_t_fsflushr` (あるいはその両方) の変更が必要になる状況をいくつか示します。

- 大きなメモリーをもつシステム - この場合には、`autoup` を増やすと、`fsflush` の個々の呼び出しで走査されるメモリー量が少なくなります。
- メモリーの要求量が最小限のシステム - `autoup` と `tune_t_fsflushr` を両方とも増やすと、走査の回数が減ります。`autoup` 対

tune_t_fsflushr の現在の比率を維持するには autoup も増やす必要があります。

- 一時ファイルの数が多いシステム (メールサーバーやソフトウェアビルドマシンなど) – 多数のファイルが作成されて削除された時、fsflush によって、これらのファイルのデータページがディスクに不必要に書き込まれるおそれがあります。

コミットレベル 変更の可能性あり

dopageflush

説明	fsflush の呼び出し時に、変更されたページの有無についてメモリーを検証するかどうかを制御します。fsflush を呼び出すたびに、システムの物理メモリーページ数が判別されます。この値は動的再構成動作によって変更されている可能性があります。呼び出しのたびに、次のアルゴリズムを使用して走査が実行されます。ページ総数 x tune_t_fsflushr / autoup ページ
データ型	符号付き整数
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
単位	切り替え (オン/オフ)
動的か	はい
検証	なし
どのような場合に 変更するか	システムページスキャナの実行がまれな場合 (vmstat 出力の sr 欄に値 0 が示される)。
コミットレベル	変更の可能性あり

doiflush

fsflush 呼び出し時にファイルシステムメタデータの同期化を行うかどうかを制御します。同期化は、fsflush の N 回目の呼び出しごとに行われます。ここで N は $(\text{autoup} / \text{tune_t_fsflushr})$ です。このアルゴリズムは整数の割り算であるため、tune_t_fsflushr が autoup より大きい

	と、同期化は <code>fsflush</code> が呼び出されるたびに行われます (反復カウンタが N 以上であるかどうかをコードがチェックするため)。 N は <code>fsflush</code> を実行するときに 1 度だけ計算されることに注意してください。そのあとで <code>tune_t_fsflushr</code> や <code>autoup</code> を変更しても、同期化操作の頻度に影響はありません。
データ型	符号付き整数
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
単位	切り替え (オン/オフ)
動的か	はい
検証	なし
どのような場合に 変更するか	<p>一定期間にファイルが頻繁に変更されるため、フラッシュによる負荷がシステムの動作に悪影響を与える場合。</p> <p>システムがリブートされる際に消えたり状態の一貫性が維持されなくても構わないファイルは、TMPFS ファイルシステム (<code>/tmp</code> など) に置いた方がいいでしょう。システム上の <code>i</code> ノードトラフィックを減らすには、<code>mount -noatime</code> オプションを使用します。このオプションを使うと、ファイルがアクセスされた時に <code>i</code> ノードの更新が行われません。</p> <p>リアルタイム処理を行うシステムでは、このオプションを無効にし、アプリケーションによってファイルの同期化を明示的に行い、一貫性を保つことを望むこともあるでしょう。</p>
コミットレベル	変更の可能性あり

プロセス規模調整パラメータ

システムで使用されるプロセスの数や個々のユーザーが作成できるプロセスの数を制御するパラメータ (または変数) がいくつかあります。基本パラメータは `maxusers` です。このパラメータによって、`max_nprocs` と `maxuprc` に値が割り当てられます。

maxusers

説明 `maxusers` は、当初、システムがサポートできるログインユーザーの数を指定するものでした。カーネルの生成時に、この設定値に基づいて各種

テーブルの大きさが決定されました。Oracle Solaris 最新リリースでは、そのサイジングの大半をシステム上のメモリー容量に基づいて行います。したがって、`maxusers` の使い方がこれまでとは大きく変わりました。引き続き、`maxusers` に基づいて決定されるサブシステムには次のものがあります。

- システムで使用できるプロセスの最大数
- システムに保持される割り当て構造体の数
- ディレクトリ名検索キャッシュ (DNLC) の大きさ

データ型	符号付き整数
デフォルト	M バイト単位のメモリーの容量と 2048 のどちらか小さい方、およびその値と <code>nCPU x 8</code> のどちらか大きい方
範囲	<code>/etc/system</code> ファイル内に設定されていない場合、物理メモリーのサイズに基づいて、1 から 2048 と <code>nCPU x 8</code> のどちらか大きい方まで <code>/etc/system</code> ファイル内に設定されている場合、1 から 4096 と <code>nCPU x 8</code> のどちらか大きい方まで
単位	ユーザー数
動的か	いいえ。このパラメータに依存する変数を計算したあとに <code>maxusers</code> がふたたび参照されることはありません。
検証	許容される最大数より値が大きい場合、最大数にリセットされます。その作用に関するメッセージが表示されます。
どのような場合に 変更するか	システムによって計算されたデフォルトのユーザープロセス数が小さすぎる場合。このような状況は、システムコンソールに表示される次のメッセージでわかります。 <code>out of processes</code> 次の状況のように、デフォルトのプロセス数が多すぎる場合に、このパラメータを変更するかもしれません。 <ul style="list-style-type: none"> ■ 多量のメモリーがあり、動作しているプロセスの数が比較的少ないデータベースサーバーでは、<code>maxusers</code> のデフォルト値を少なくすることによってシステムメモリーを節約できます。 ■ 多量のメモリーがあり、動作しているプロセスがほとんどないファイルサーバーでは、この値を減らせる場合があります。しかし、その場合、DNLCのサイズを明示的に設定する必要があります。64 ページの「ncsize」を参照してください。

コミットレベル 変更の可能性あり

reserved_procs

説明 UIDが root (0) のプロセス用に、プロセステーブルで確保するシステムプロセススロット数を指定します。たとえば、`fsflush` には root (0) の UID が与えられます。

データ型 符号付き整数

デフォルト 5

範囲 5 から MAXINT

単位 プロセス数

動的か いいえ。最初のパラメータ計算の後は使用されません。

検証 `/etc/system` のどの設定も受け入れられます。

コミットレベル 変更の可能性あり

どのような場合に
変更するか たとえば、システムの UID 0 (root) のプロセスの数を、通常の数から 10 大きくした場合を考えてみてください。この設定をしないとユーザーレベルのプロセスを作れないような状況でも、この設定を行うことによって root でシェルを起動するために必要な余裕が生まれます。

pidmax

説明 使用可能な最大プロセス ID の値を指定します。
`pidmax` は `maxpid` 変数の値を設定します。したがって、`maxpid` がいったん設定されると、`pidmax` は無視されます。`maxpid` は、カーネルの別のところで、最大のプロセス ID を判別したり、妥当性検証を行うために使用されます。
`/etc/system` ファイルに `maxpid` エントリを追加して設定しようとしても、効果はありません。

データ型 符号付き整数

デフォルト 30,000

範囲 5 から 999,999

単位	プロセス数
動的か	いいえ。pidmax の値を設定するためにブート時だけ使用されます。
検証	はい。reserved_procs の値と 999,999 に対して値を比較します。reserved_procs より小さい場合、または 999,999 より大きい場合、値は 999,999 に設定されます。
暗黙的制約	max_nprocs に対して範囲の検査が行われ、max_nprocs は常にこの値以下に保たれます。
どのような場合に 変更するか	システム上で 30,000 を超える数のプロセスをサポートできるようにするために必要です。34 ページの「max_nprocs」も参照してください。
コミットレベル	変更の可能性あり

max_nprocs

説明	<p>システム上に作成できるプロセスの最大数を指定します。システムプロセスとユーザープロセスを含みます。/etc/system に指定した任意の値が maxuprc の計算に使用されます。</p> <p>この値は、ほかのいくつかのシステムデータ構造体のサイズを決定する場合にも使用されます。このパラメータが作用する他のデータ構造体は、次のとおりです。</p> <ul style="list-style-type: none"> ■ ディレクトリ名検索キャッシュのサイズを決めるとき (ncsize が指定されていない場合) ■ 構成されたシステム V セマフォによって使用されるメモリーの総量がシステム限度を超えていないか確認するとき ■ x86 プラットフォーム向けのハードウェアアドレス変換のリソースを構成するとき
データ型	符号付き整数
デフォルト	<p>maxusers が /etc/system ファイル内で設定されている場合は、10 + (16 x maxusers)</p> <p>maxusers が /etc/system ファイル内で設定されていない場合は、30,000 と 10 + (128 x CPU の数) のどちらか大きい方</p>
範囲	26 から maxpid の値
動的か	いいえ

検証	はい。値が <code>maxpid</code> を超える場合、 <code>maxpid</code> に設定されます。
どのような場合に 変更するか	このパラメータの変更は、1 つのシステムで 30,000 を越えるプロセスを可能にするために必要となる手順の 1 つです。
コミットレベル	変更の可能性あり

maxuprc

説明	個々のユーザーがシステム上に作成できるプロセスの最大数を指定します。
データ型	符号付き整数
デフォルト	<code>max_nprocs - reserved_procs</code>
範囲	1 から <code>max_nprocs - reserved_procs</code>
単位	プロセス数
動的か	いいえ
検証	はい。この値は <code>max_nprocs - reserved_procs</code> と比較され、2 つの値のうちの小さい方に設定されます。
どのような場合に 変更するか	1 ユーザーが作成できるプロセスの数を強く制限するために、デフォルト値より小さい値を指定したい場合 (システムが作成できるプロセスの数も多くても)。この限度を超えると、次の警告メッセージがコンソールかメッセージファイルに出力されます。 <code>out of per-user processes for uid N</code>
コミットレベル	変更の可能性あり

ngroups_max

説明	プロセスごとの追加グループの最大数を指定します。
データ型	符号付き整数
デフォルト	16

範囲	0 から 1024
単位	グループ
動的か	いいえ
検証	はい。ngroups_max が無効な値に設定された場合、もっとも近い有効値に自動的にリセットされます。たとえば、ゼロより小さい値に設定された場合は、0 にリセットされます。1024 より大きい値に設定された場合は、1024 にリセットされます。
どのような場合に 変更するか	NFS AUTH_SYS 認証を使用していて、デフォルトの ngroups_max 値を増やす場合、次の考慮事項を検討してください。 <ol style="list-style-type: none"> 1. ngroups_max が 16 に設定されるか、指定されたクライアントの AUTH_SYS 資格のグループが 15 以下の場合、クライアントのグループ情報が使用されます。 2. ngroups_max が 16 より大きく設定され、かつネームサーバーからのクライアントの AUTH_SYS 資格に、許容される最大数である 16 グループが含まれている場合、NFS サーバーはネームサーバーを調べてクライアントの UID をユーザー名と照合します。その後、ネームサーバーはユーザーが属するグループのリストを計算します。
コミットレベル	変更の可能性あり

ページング関連パラメータ

Solaris OS では、必要に応じてページングされる仮想メモリーシステムを使用します。システム稼働に伴ってページが必要になると、そのページがメモリーに読み込まれます。メモリーの占有率が一定のしきい値を超え、さらにメモリーの要求が続くと、ページングが発生します。ページングには、特定のパラメータで制御されるいくつかのレベルがあります。

一般的なページングアルゴリズムは次のとおりです。

- メモリーの不足が認識されます。ページ走査スレッドが実行され、メモリーのチェックを開始します。この際、2 段階のアルゴリズムが使用されます。
 1. 使用されていないページを識別します。
 2. 一定の間隔後にもそのページが使用されていないければ、そのページを再利用の対象とみなします。

ページが変更されていれば、ページアウトスレッドに対して、ページの入出力をスケジューリングするように要求されます。さらに、ページスキャナが引き続きメモリーを調べます。ページアウトは、そのページをページのバッキングストアに書き込み、空リストに置くようにします。ページスキャナがメモリーを走査するときに、ページの内容の区別はありません。ページは、データファイルからのものもあれば、実行可能ファイルのテキスト、データ、スタックからのものもあります。

- システムのメモリーの使用が著しくなってくるに従い、このアルゴリズムは、再利用の候補とみなすページや、ページングアルゴリズムを実行する頻度に関する基準を強化します。(詳細は、44 ページの「fastscan」および45 ページの「slowscan」を参照してください)。使用可能なメモリーが `lotsfree` から `minfree` の範囲内になると、システムはページアウトスレッドが呼び出されるたびに走査するメモリー量を、`slowscan` で指定された値から `fastscan` で指定された値に直線的に増やします。システムは、`desfree` パラメータを使用して、リソースの使用や動作に関する決定回数を制御します。

システムはページアウト操作を 1 つの CPU の 4% 以内の使用に限定しようとしています。メモリーへの負荷が大きくなると、それに比例してページアウト操作をサポートするために消費される CPU 時間が増加し、最大で 1 つの CPU の 80% が消費されます。このアルゴリズムは、`slowscan` と `fastscan` の間のメモリー量を調べ、次の条件のどれかに当てはまると走査を終了します。

- メモリー不足を解消するだけのページが見つかりました。
- 予定のページ数を調べました。
- 長すぎる時間が経過しました。

ページアウトが走査を終了してもメモリー不足が解消しない場合は、後で別の走査が 1/4 秒間スケジュールされます。

ページングサブシステムの構成メカニズムが変更されました。システムは `fastscan`、`slowscan`、および `handspreadpages` の事前定義された値を使用せずに、ブート時にこれらのパラメータへ適切な値を割り当てます。`/etc/system` ファイル内のこれらのパラメータを設定すると、システムが最適でない数値を使用する場合があります。



注意 - `/etc/system` ファイルから、VM システムのチューニングをすべて削除してください。まずデフォルトの設定値で実行してから、これらのパラメータの調整が必要かどうかを判定してください。また、`cachefree` および `priority_paging` を設定しないでください。

CPU とメモリーの動的再構成 (DR) がサポートされています。システムでメモリーの追加や削除を伴う DR 操作があると、該当のパラメータが `/etc/system` に明示的に設定されていなく

れば、そのパラメータ値が再計算されます。明示的に設定されている場合は、変数の値に対する制約に反しないかぎり、`/etc/system` に指定された値が使用されます。この場合は、値がリセットされます。

lotsfree

説明	システムのページングを開始する最初のきっかけになります。ページ数がこのしきい値に達すると、ページスキャナが立ち上がり、再利用するメモリーページを探します。
データ型	符号なし long
デフォルト	物理メモリーの 1/64 または 512K バイトのどちらか大きい方
範囲	<p>最小値は、512K バイトまたは物理メモリーの 1/64 のどちらか大きい方であり、<code>getpagesize</code> によって返されるページサイズに基づくページ数で表されます。詳細は、getpagesize(3C) を参照してください。</p> <p>最大値は物理メモリーのページ数です。この最大値は、物理メモリーの 30% 以内であるべきです。システムは、「検証」のセクションで記述している場合以外は、この範囲を強制しません。</p>
単位	ページ
動的か	はい。ただしメモリーに関して DR 操作が行われると、動的な変更は失われます。
検証	<code>lotsfree</code> が物理メモリーの総量より大きい場合、値はデフォルトにリセットされます。
暗黙的制約	<code>lotsfree</code> が <code>desfree</code> よりも大きく、 <code>desfree</code> が <code>minfree</code> よりも大きいという関係が常に維持されるようにする必要があります。
どのような場合に 変更するか	<p>ページ要求が急激に増えるような場合には、メモリアルゴリズムが要求に対応できないことがあります。これを回避するには、早期にメモリーの回収を開始するのも 1 つの方法です。これは、ページングシステムにいくらか余裕を与えることとなります。</p> <p>経験則によると、このパラメータは、システムが 2 - 3 秒で割り当てる必要がある量の 2 倍にします。このパラメータの適正值は負荷によって異なります。DBMS サーバーはデフォルトの設定で支障がないはずですが、しかし、ファイルシステムの入出力負荷が非常に大きい場合は、このパラメータを調整する必要があるかもしれません。</p> <p>負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。許容される最小値は 512K バイトであり、<code>getpagesize</code> によって返されるページサイズに基づくページ数で表されます。</p>

コミットレベル 変更の可能性あり

desfree

説明	システム上で常時解放しておくべきメモリー容量を指定します。
データ型	符号なし整数
デフォルト	lotsfree / 2
範囲	<p>最小値は、256K バイトまたは物理メモリーの 1/128 のどちらか大きい方であり、getpagesize によって返されるページサイズに基づくページ数で表されます。</p> <p>最大値は物理メモリーのページ数です。この最大値は物理メモリーの 15% 以内であるべきです。システムは、「検証」のセクションで記述している場合以外は、この範囲を強制しません。</p>
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、/etc/system ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検証	desfree が lotsfree より大きい場合、desfree は lotsfree / 2 に設定されます。メッセージは表示されません。
暗黙的制約	lotsfree が desfree よりも大きく、desfree が minfree よりも大きいという関係が常に維持されるようにする必要があります。
副次的な影響	<p>このパラメータの値を増やすと、いくつかの副次的な影響が現われることがあります。新しい値がシステム上で使用できるメモリー容量に近いかそれを超えると、次の現象が生じることがあります。</p> <ul style="list-style-type: none"> ■ 使用可能なメモリーが desfree を超えないかぎり、非同期の入出力要求が処理されません。したがって、desfree の値を増やすと、増やす前なら処理されたであろう要求が拒否されることがあります。 ■ NFS の非同期書き込みが、同期書き込みとして実行されます。 ■ スワッパーが本来より早く立ち上がり、そのスワッパーの動作が、積極的な動作をする方向に傾きます。 ■ システムに前もって読み込む実行可能ページの数本来よりも少なくなることがあります。この副次的な影響の結果、アプリケーションの動作が本来よりも遅くなる可能性があります。

どのような場合に
変更するか 負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。許容される最小値は 256K バイトであり、`getpagesize` によって返されるページサイズに基づくページ数で表されます。

コミットレベル 変更の可能性あり

minfree

説明 許容される最低メモリーレベルを指定します。メモリーがこの値を下回ると、システムはページアウト動作の完了に必要な割り当て、またはプロセスのスワップ完了に必要な割り当てに重点を置いて、メモリーを割り当てます。それ以外の割り当て要求は拒否されたりブロックされたりします。

データ型 符号なし整数

デフォルト `desfree / 2`

範囲 最小値は、128K バイトまたは物理メモリーの 1/256 のどちらか大きい方であり、`getpagesize` によって返されるページサイズに基づくページ数で表されます。

最大値は物理メモリーのページ数です。この最大値は物理メモリーの 7.5% 以内であるべきです。システムは、「検証」のセクションで記述している場合以外は、この範囲を強制しません。

単位 ページ

動的か はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、`/etc/system` ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。

検証 `minfree` が `desfree` より大きい場合、`minfree` は `desfree / 2` に設定されます。メッセージは表示されません。

暗黙的制約 `lotsfree` が `desfree` よりも大きく、`desfree` が `minfree` よりも大きいという関係が常に維持されるようにする必要があります。

どのような場合に
変更するか 一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。許容される最小値は 128K バイトであり、`getpagesize` によって返されるページサイズに基づくページ数で表されます。

コミットレベル 変更の可能性あり

throttlefree

説明	要求を満たせるだけのメモリーがある場合でも、メモリー割り当て要求ブロッキングをスリープ状態にするメモリー レベルを指定します。
データ型	符号なし整数
デフォルト	minfree
範囲	最小値は、128K バイトまたは物理メモリーの 1/256 のどちらか大きい方であり、getpagesize によって返されるページサイズに基づくページ数で表されます。 最大値は物理メモリーのページ数です。この最大値は物理メモリーの 4% 以内であるべきです。システムは、「検証」のセクションで記述している場合以外は、この範囲を強制しません。
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、/etc/system ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検証	throttlefree が desfree よりも大きい場合、throttlefree は minfree に設定されます。メッセージは表示されません。
暗黙的制約	lotsfree が desfree よりも大きく、desfree が minfree よりも大きいという関係が常に維持されるようにする必要があります。
どのような場合に 変更するか	一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。許容される最小値は 128K バイトであり、getpagesize によって返されるページサイズに基づくページ数で表されます。詳細は、 getpagesize(3C) を参照してください。
コミットレベル	変更の可能性あり

pageout_reserve

説明	ページアウトスレッドまたはスケジューラスレッドが独占使用できるように確保するページ数を指定します。使用可能なメモリーがこの値を下回ると、ページアウトやスケジューラ以外のプロセスに対するブロックしない割り当ては拒否されます。ページアウトには専用の小さなメモリープールが
----	---

必要です。ページアウトは、ページをバッキングストアに書き込む入出力に必要なデータ構造体をここから割り当てます。

データ型	符号なし整数
デフォルト	throttelfree / 2
範囲	最小値は、64K バイトまたは物理メモリーの 1/512 のどちらか大きい方であり、getpagesize(3C) によって返されるページサイズに基づくページ数で表されます。 最大値は物理メモリーのページ数です。この最大値は物理メモリーの 2% 以内であるべきです。システムは、「検証」のセクションで記述している場合以外は、この範囲を強制しません。
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、/etc/system ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検証	pageout_reserve が throttelfree / 2 より大きい場合、pageout_reserve は throttelfree / 2 に設定されます。メッセージは表示されません。
暗黙的制約	lotsfree が desfree よりも大きく、desfree が minfree よりも大きいという関係が常に維持されるようにする必要があります。
どのような場合に 変更するか	一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。許容される最小値は 64K バイトであり、getpagesize によって返されるページサイズに基づくページ数で表されます。
コミットレベル	変更の可能性あり

pages_pp_maximum

説明	ロック解除されていなければならないページ数を指定します。ページのロック要求によって使用可能なメモリーがこの値を下回る場合は、その要求は拒否されます。
データ型	符号なし long
デフォルト	tune_t_minarmem + 100 と、ブート時に使用可能なメモリーの 4% + 4M バイトのどちらか大きい方

範囲	システムが強制する最小値は <code>tune_t_minarmem + 100</code> です。最大値については、システムは強制しません。
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、 <code>/etc/system</code> ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検証	<p><code>/etc/system</code> ファイルで指定された値、またはデフォルトで計算された値が <code>tune_t_minarmem + 100</code> よりも小さい場合、この値は <code>tune_t_minarmem + 100</code> へリセットされます。</p> <p><code>/etc/system</code> ファイルからの値が増やされても、メッセージは表示されません。検証は、ブート時とメモリーの追加または削除を伴う動的再構成が行われた場合に限って実行されます。</p>
どのような場合に 変更するか	メモリーのロック要求や、 <code>SHARE_MMU</code> フラグを指定した共有メモリーセグメントへの接続が失敗したが、使用可能なメモリーが十分ありそうな場合。大きすぎる値が原因で、メモリーのロック要求 (<code>mlock</code> 、 <code>mlockall</code> 、および <code>memcntl</code>) が不必要に失敗する場合があります。詳細は、 mlock(3C) 、 mlockall(3C) 、および memcntl(2) を参照してください。
コミットレベル	変更の可能性あり

tune_t_minarmem

説明	デッドロックを回避するために維持しなければならない、利用可能な最小常駐 (スワップ不能) メモリーを指定します。この値は、OS のコアによって使用されるメモリー部分を予約するために使用されます。この方法で制限されたページは、OS が使用可能なメモリーの最大量を判定するときには計算に入れられません。
データ型	符号付き整数
デフォルト	25
範囲	1 から物理メモリー
単位	ページ
動的か	いいえ

検証	ありません。値が大きいと、物理メモリーが無駄になります。
どのような場合に 変更するか	一般にはデフォルト値で十分です。システムがロックされ、使用できるメモリーがないことがデバッグ情報からわかった場合は、デフォルト値を増やすことを検討してください。
コミットレベル	変更の可能性あり

fastscan

説明	メモリー要求が大きいときにシステムが調べる、最大ページ数 / 秒を指定します。
データ型	符号付き整数
デフォルト	<p>fastscan のデフォルト値は次のいずれかの方法で設定されます。</p> <ul style="list-style-type: none"> ■ /etc/system ファイルに設定された fastscan 値が使用されます。 ■ /etc/system ファイルに設定された maxfastscan 値が使用されます。 ■ /etc/system ファイルに fastscan も maxfastscan も設定されていない場合は、システムのブート時に、fastscan が 64M バイトに設定されます。数分間のシステムのブート後、fastscan 値が、スキャナが CPU の 10% を使用して 1 秒間で走査できるページの数に設定されます。 <p>これらのすべての状況で、派生した値がシステムのメモリーの半分を超えた場合、fastscan 値はシステムのメモリーの半分の値に制限されます。</p>
範囲	64M バイトからシステムの物理メモリーの半分
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、/etc/system ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検証	最大値は、64M バイトと物理メモリーの 1/2 のどちらか小さい方です。
どのような場合に 変更するか	メモリー不足のときにメモリーの走査を優先させたい場合。特に、システムでメモリーの要求が急激に多くなる場合や、多数のファイル入出力が行われることがある場合。

コミットレベル 変更の可能性あり

slowscan

説明	メモリーの再要求時にシステムが調べる、最小ページ数 / 秒を指定します。
データ型	符号付き整数
デフォルト	物理メモリーの 1/20 (ページ数) か 100 (小さい方)
範囲	1 から fastscan / 2
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、 <code>/etc/system</code> ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検証	slowscan が fastscan / 2 より大きい場合、slowscan は fastscan / 2 にリセットされます。メッセージは表示されません。
どのような場合に 変更するか	メモリー不足のときにメモリーの走査を優先させたい場合、特に、システムでメモリーの要求が急激に多くなるときがある場合。
コミットレベル	変更の可能性あり

min_percent_cpu

説明	pageout が最低限消費できる CPU の割合を指定します。このパラメータは、ページスキャナで使用できる最大時間を判定するための開始点として使用されます。
データ型	符号付き整数
デフォルト	4
範囲	1 から 80
単位	%
動的か	はい

検証	なし
どのような場合に 変更するか	複数の CPU と多くのメモリーを備えたシステム (このようなシステムではメモリーの要求が急激に多くなるときがある) でこの値を増やすと、ページャがメモリーの検出に使用できる時間が増えます。
コミットレベル	変更の可能性あり

handspreadpages

説明	Oracle Solaris OS は双針クロックアルゴリズムを使用して、メモリー不足のときに再利用の候補となるページを探します。最初の針はメモリーに使用されていないという印を付けていきます。次の針は、最初の針の少し後から、そのページに依然として使用されていないという印が付けられているかを調べます。そうであれば、そのページが再利用の対象になります。最初の針と次の針の間隔が <code>handspreadpages</code> です。
データ型	符号なし long
デフォルト	<code>fastscan</code>
範囲	1 からシステムの物理メモリーの最大ページ数
単位	ページ
動的か	はい。このパラメータを設定する場合、カーネルパラメータ <code>reset_hands</code> もゼロ以外の値に設定する必要があります。 <code>handspreadpages</code> の新しい値がいったん認識されると、 <code>reset_hands</code> はゼロに設定されます。
検証	値は物理メモリー容量と <code>handspreadpages value</code> のどちらか小さい方に設定されます。
どのような場合に 変更するか	ページが再利用されるまで置いておく時間を長くする場合。この値を増やすと 2 つの段階の間の時間が長くなるため、ページが再利用されるまでの時間が増えます。
コミットレベル	変更の可能性あり

pages_before_pager

説明	再利用に備えてページを保管する代わりに、入出力の完了後ただちにページを解放する、システムしきい値の部分を指定します。このしきい値
----	--

	は <code>lotsfree + pages_before_pager</code> です。さらに、NFS 環境も、メモリーが不足するとこのしきい値を使用して非同期の活動を減らします。
データ型	符号付き整数
デフォルト	200
範囲	1 から物理メモリーのページ数
単位	ページ
動的か	いいえ
検証	なし
どのような場合に 変更するか	<p>入出力の大半が 1 回限りのページの読み取りまたは書き込みであり、二度と参照されない場合、このパラメータを変更することができるかもしれませんが。この変数を大きなメモリーの値に設定すると、ページは空リストに追加され続けます。</p> <p>システムが繰り返し強いメモリー要求を受ける場合も、このパラメータを変更することがあります。より大きな値は、この要求に対するより大きな緩衝剤となります。</p>
コミットレベル	変更の可能性あり

maxpgio

説明	ページングシステムがキューに入れることのできるページ入出力要求の最大数を指定します。ページングシステムは、実際に使用する最大数を計算するために、この数字を 4 で割ります。このパラメータは、要求の数を制限する他に、プロセスのスワッピングを制御するためにも使用されます。
データ型	符号付き整数
デフォルト	400
範囲	最小値は 1 です。最大値はシステムアーキテクチャーによって決まります。主に、コントローラやディスクの数、そしてディスクのスワップサイズなどの入出力サブシステムによります。
単位	入出力
動的か	いいえ

検証	なし
暗黙的制約	ページャからの入出力要求の最大数は、要求バッファのリストのサイズによって制限されます。現在のサイズは 256 です。
どのような場合に 変更するか	このパラメータはメモリのページアウトを早くするために増やします。複数のスワップデバイスが構成されているか、またはスワップデバイスがストライプ化デバイスである場合、この値を増やすとメモリ不足の解消が早くなる可能性があります。既存の入出力サブシステムは、追加される入出力の負荷に対処できる必要があります。また、スワップパーティションとアプリケーションファイルが同じディスク上にある場合、スワップ入出力の増加はアプリケーションの入出力のパフォーマンスを低下させることがあります。
コミットレベル	変更の可能性あり

スワッピング関連パラメータ

Oracle Solaris OS のスワッピングは、`swapfs` 擬似ファイルシステムによって行われます。スワップデバイスの空間と物理メモリーを合わせたものが、匿名メモリーのバッキングストアを維持するために利用可能な空間プールとして扱われます。システムは、バッキングストアとして最初にディスクデバイスから空間を割り当てようとし、その次に物理メモリーを使用します。`swapfs` がバッキングストアとしてシステムメモリーを使用しなければならない場合は、`swapfs` によるメモリーの使いすぎによってシステムがデッドロックに陥ることがないように制約が課せられます。

`swapfs_reserve`

説明	システム (UID = 0) プロセス用に予約するシステムメモリー容量を指定します。
データ型	符号なし long
デフォルト	4M バイトと物理メモリーの 1/16 のどちらか小さい方
範囲	最小値は、4M バイトまたは物理メモリーの 1/16 のどちらか小さい方であり、 <code>getpagesize</code> によって返されるページサイズに基づくページ数で表されます。 最大値は物理メモリーのページ数です。最大値は、物理メモリーの 10% 以内であるべきです。システムは、「検証」のセクションで記述している場合以外は、この範囲を強制しません。

単位	ページ
動的か	いいえ
検証	なし
どのような場合に 変更するか	変更は一般には必要ありません。ソフトウェアプロバイダからの推奨があったり、スワップ空間が取得できないためにシステムプロセスが終了してしまう場合だけ変更します。しかし、それより良い解決策は、物理メモリーかスワップデバイスをシステムに追加することです。
コミットレベル	変更の可能性あり

swapfs_minfree

説明	システムのほかの部分のために、解放しておくべき物理メモリーの容量を指定します。プロセスのスワップ空間としてメモリーを予約しようとするときに、それによって使用可能なメモリーがこの値を下回るおそれがあるとシステムが判断する場合、この要求は拒否されます。この方法で予約されたページは、カーネルやユーザーレベルプロセスによってロックダウンされた割り当てに対してのみ使用できます。
データ型	符号なし long
デフォルト	2M バイトと物理メモリーの 12.5% のどちらか大きい方
範囲	1 から物理メモリーのページ数
単位	ページ
動的か	いいえ
検証	なし
どのような場合に 変更するか	システムに使用可能なメモリーがあるのにスワップ空間が得られないためにプロセスが失敗する場合、このパラメータ値を削減することを検討してください。たとえば、システムメモリーの 6.25% を超えて使用しないようにこの値を変更しても、システムメモリーの 5% 未満に削減しないでください。 SPARC システムでは、この値は <code>tsb_alloc_hiwater_factor</code> の値の少なくとも 2 倍にするようにします。詳細は、 78 ページの「tsb_alloc_hiwater_factor」 を参照してください。
コミットレベル	変更の可能性あり

カーネルメモリアロケータ

Oracle Solaris カーネルメモリアロケータは、カーネル内の各クライアントに使用するメモリーのチャンクを配分します。アロケータは、そのクライアントが使用するさまざまなサイズのキャッシュを作成します。一方、クライアントは、特定サイズの構造体の割り当てのためなど、クライアントが使用するキャッシュの作成をアロケータに要求できます。アロケータが管理する各キャッシュに関する統計は、`kstat -c kmem_cache` コマンドで表示できます。

メモリーが壊されたために、システムがパニックになることがまれにあります。カーネルメモリアロケータは、バッファの各種整合性検査を実行するデバッグインタフェース (一連のフラグ) をサポートします。カーネルメモリアロケータは、アロケータに関する情報も収集します。整合性検査によって、発生まぎわのエラーを検出する機会が得られます。収集された情報は、サポート担当者にとって、パニックの原因追及を試みるための追加情報となります。

フラグを使用すると、システム操作で余分なオーバーヘッドと余分なメモリーの使用が発生します。したがって、フラグの使用は、メモリーの損傷が疑われるときだけに限るべきです。

kmem_flags

説明

Oracle Solaris カーネルメモリアロケータには、さまざまなデバッグオプションおよびテストオプションがあります。

次に、サポートされる 5 つのフラグの設定について説明します。

フラグ	設定	説明
AUDIT	0x1	アロケータは、自身の活動の最近の履歴が入ったログを維持します。ログされる項目の数は、 <code>CONTENTS</code> も設定されているかどうかによって異なります。このログは固定の大きさです。領域を使い果たすと、古い記録から再利用されます。
TEST	0x2	アロケータは解放されたメモリーにパターンを書き込み、そのバッファを次に割り当てるときに、そのパターンが変更されていないことをチェックします。バッファの一部が変更されている場合は、そのバッファを前に割り当て、解放したクライアントがそのメモリーを使用した可能性が強いことを意味します。上書きが検知されると、システムがパニックになります。
REDZONE	0x4	アロケータは要求されたバッファの終りに余分のメモリーを割り当て、そのメモリーに特殊なパターンを挿入します。そして、バッファが解放されたら、パターン

フラグ	設定	説明
		をチェックして、データがバッファの終りより後ろに書き込まれていないか調べます。上書きが検知されると、カーネルがパニックになります。
CONTENTS	0x8	アロケータは、バッファが解放されると、バッファの内容を 256 バイトまでログします。このフラグを使用するには、AUDIT も設定する必要があります。
		これらのフラグの数値は、論理的に合算し、/etc/system ファイルによって設定できます。
LITE	0x100	バッファを割り当てたり解放したりするときに、最小限の整合性検査を行います。このフラグが有効になっていると、アロケータは、レッドゾーンが書き込まれていないことや、解放されたバッファが再び解放されていないこと、解放されるバッファのサイズが割り当てられたものと同じであることをチェックします。このフラグは他のフラグと併用しないでください。

データ型	符号付き整数
デフォルト	0 (無効)
範囲	0 (無効)、1 - 15、256 (0x100)
動的か	はい。実行時の変更は、新しいカーネルメモリーキャッシュだけに有効です。システムの初期設定後に新しいキャッシュを作成することはまれです。
検証	なし
どのような場合に 変更するか	メモリーの損傷が疑われる場合
コミットレベル	変更の可能性あり

kmem_stackinfo

説明	カーネルスレッドの作成時に、/etc/system ファイルの kmem_stackinfo 変数が有効になっている場合、カーネルスレッドスタックが、0 ではなく、特定のパターンで埋められます。カーネルスレッドの実行時に、このカーネルスレッドスタックのパターンが徐々に上書きされます。パターンが見つからなくなるまで、スタックの最上部から単純にカウントすることで、カーネルスレッドで使用される最大のカーネルスタック空間である高位境界値が得られます。このメカニズムにより、次の機能が可能になります。
----	---

- システムの現在のカーネルスレッドで実際に使用されたカーネルスレッドスタックの割合 (高位境界値) を計算します。
- カーネルスレッドが終了すると、システムは、ほとんどのカーネルスレッドスタックを使用した最後のカーネルスレッドを、終了前に、小さい循環メモリーバッファに記録します。

データ型	符号なし整数
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
動的か	はい
検証	なし
どのような場合に 変更するか	カーネルスレッドスタックの使用状況をモニターする場合。kmem_stackinfo を有効にしていると、カーネルスレッドの作成と削除のパフォーマンスが低下することに注意してください。詳細は、『Oracle Solaris モジュールデバッグガイド』を参照してください。
ゾーン構成	このパラメータは、大域ゾーン内に設定する必要があります。
コミットレベル	変更の可能性あり

一般的なドライバパラメータ

moddebug

説明	このパラメータが有効なとき、モジュールのロードプロセスの各種ステップについてのメッセージが表示されます。
データ型	符号付き整数
デフォルト	0 (メッセージを表示しない)
範囲	もっとも有用な値は次のとおりです。 <ul style="list-style-type: none"> ■ 0x80000000 - [un] loading... メッセージを出力します。モジュールがロードされるたびに、次のようなメッセージがコンソールと /var/adm/messages ファイルに出力されます。

```
Apr 20 17:18:04 neo genunix: [ID 943528 kern.notice] load 'sched/
TS_DPTBL' id 15
loaded @ 0x7be1b2f8/0x19c8380 size 176/2096
Apr 20 17:18:04 neo genunix: [ID 131579 kern.notice] installing
TS_DPTBL,
module id 15.
```

- 0x40000000 - 詳細なエラーメッセージを出力します。モジュールがロードされるたびに、次のようなメッセージがコンソールと `/var/adm/messages` ファイルに出力されます。

```
Apr 20 18:30:00 neo unix: Errno = 2
Apr 20 18:30:00 neo unix: kobj_open: vn_open of /platform/sun4v/
kernel/exec/sparcv9/intpexec fails
Apr 20 18:30:00 neo unix: Errno = 2
Apr 20 18:30:00 neo unix: kobj_open: '/kernel/exec/sparcv9/intpexec'
Apr 20 18:30:00 neo unix: vp = 60015777600
Apr 20 18:30:00 neo unix: kobj_close: 0x60015777600
Apr 20 18:30:00 neo unix: kobj_open: vn_open of /platform/SUNW,Sun-
Fire-T200/kernel/exec/sparcv9
/intpexec fails,
Apr 20 18:30:00 neo unix: Errno = 2
Apr 20 18:30:00 neo unix: kobj_open: vn_open of /platform/sun4v/
kernel/exec/sparcv9/intpexec fails
```

- 0x20000000 - より詳細なメッセージを出力します。この値は、システムブート時には `0x40000000` フラグが出力する以上の詳細情報は出力しません。モジュールのロード解除時には、モジュールの解放に関する詳細情報を出力します。

これらの値は足し合わせて指定できます。

動的か

はい

検証

なし

どのような場合に
変更するか

期待通りにモジュールがロードされない場合や、モジュールのロード中にシステムがハングしている疑いがある場合。`0x40000000` を設定すると、多数のメッセージがコンソールに書き込まれるため、システムのブートがかなり遅くなることに留意してください。

コミットレベル 変更の可能性あり

ddi_msix_alloc_limit

説明	x86 のみ: このパラメータは、デバイスインスタンスで割り当てることのできる MSI-X (拡張メッセージ信号割り込み) の数を制御します。既存のシステムの制限により、デフォルト値は 2 です。このパラメータの値を大きくすることによって、デバイスインスタンスが割り当てることのできる MSI-X 割り込みの数を増やすことができます。このパラメータを設定するには、 <code>/etc/system</code> ファイルを編集するか、またはデバイスドライバの接続が発生する前に <code>mdb</code> を使用してパラメータを設定します。
データ型	符号付き整数
デフォルト	SPARC ベースのシステムの場合 8。 x86 ベースのシステムの場合 2。システムが x2APIC をサポートしている場合、 <code>apix</code> モジュールはデフォルト値を 8 に上げることができます。
範囲	2-8
動的か	はい
検証	なし
どのような場合に 変更するか	デバイスインスタンスが割り当てることのできる MSI-X 割り込みの数を増やすため。ただし、デバイスインスタンスが割り当てることのできる MSI-X 割り込みの数を増やすと、割り込み数が不足してすべての割り当て要求を満足できなくなる可能性があります。この状況が起きた場合、一部のデバイスが機能を停止したり、システムがブートに失敗したりする可能性があります。そのような場合は、パラメータの値を小さくするか、またはパラメータを削除してください。
コミットレベル	変更の可能性あり

ネットワークドライバパラメータ

カーネル内の IP プロトコルパラメータ

次の IP パラメータは、`/etc/system` ファイル内でのみ設定できます。ファイルを変更したあとで、システムをリブートしてください。

たとえば、次のエントリでは `ipcl_conn_hash_size` パラメータを設定します。

```
set ip:ipcl_conn_hash_size=value
```

`ipcl_conn_hash_size`

説明	IP によって使用される接続ハッシュテーブルのサイズを制御します。デフォルト値 0 の場合、システムは、使用可能なメモリー容量に基づいて、ブート時にこのパラメータの適切な値を自動的に決定します。
データ型	符号なし整数
デフォルト	0
範囲	0 から 82,500
動的か	いいえ。このパラメータはブート時にのみ変更できます。
どのような場合に 変更するか	常に非常に大量の TCP 接続が確立されているシステムでは、それに応じてこの値を増やすことができます。ハッシュテーブルサイズを大きくすると、より多くのメモリーが消費され、ユーザーアプリケーション用として提供できるメモリーの量が減ります。
コミットレベル	変更の可能性あり

`ip_queue_worker_wait`

説明	TCP/IP パケットを <code>queue</code> に格納して処理するワークスレッドが起動するまでの最大遅延時間を制御します。 <code>queue</code> は TCP/IP カーネルコードが TCP/IP パケット処理に使用する直列化キューです。
デフォルト	10 ミリ秒
範囲	0 から 50 ミリ秒
動的か	はい
どのような場合に 変更するか	遅延が発生しないほうが望ましく、かつ、ネットワークトラフィックが少ない場合。たとえば、対話型ネットワークトラフィックの大部分がマシンで処理されている場合など。 ネットワークファイルサーバー、Web サーバーなど、実際にネットワークトラフィックが発生しているすべてのサーバーでは、通常、デフォルト値が最適です。
ゾーン構成	このパラメータを設定できるのは、大域ゾーン内だけです。

コミットレベル 変更の可能性あり

ip_squeue_fanout

説明	squeue と TCP/IP 接続 を関連付けるモードを判定します。 値 0 の場合、新しい TCP/IP 接続と、この接続を作成した CPU が関連付けられます。値 1 の場合、異なる CPU に属する複数の squeue との接続が関連付けられます。
デフォルト	1
範囲	0 または 1
動的か	はい
どのような場合に 変更するか	特定の条件下で、すべての CPU に負荷を分散したい場合、このパラメータの値を 1 に設定します。たとえば、CPU 数が NIC 数を上回り、単一の NIC のネットワーク負荷を処理できない CPU ができた場合、このパラメータの値を 1 にします。
ゾーン構成	このパラメータを設定できるのは、大域ゾーン内だけです。
コミットレベル	変更の可能性あり

igb パラメータ

mr_enable

説明	このパラメータは、igb ネットワークドライバによって使用される複数の受信および送信キューを有効または無効にします。このパラメータは、igb ドライバ接続が発生する前に /etc/driver/drv/igb.conf ファイルを編集することで設定できます。
データ型	ブール型
デフォルト	1 (複数のキューを無効にする)
範囲	0 (複数のキューを有効にする) または 1 (複数のキューを無効にする)
動的か	いいえ
検証	なし

どのような場合に 変更するか	igb ネットワークドライバによって使用される複数の受信および送信 キューを有効または無効にする場合。
コミットレベル	変更の可能性あり

intr_force

説明	このパラメータは、igb ネットワークドライバによって使用される MSI、MSI-X、またはレガシーなどの割り込みの種類を強制するために 使用します。このパラメータは、igb ドライバ接続が発生する前に /etc/ driver/drv/igb.conf ファイルを編集することで設定できます。
データ型	符号なし整数
デフォルト	0 (割り込みの種類を強制しない)
範囲	0 (割り込みの種類を強制しない) 1 (MSI-X 割り込みの種類を強制する) 2 (MSI 割り込みの種類を強制する) 3 (レガシーの割り込みの種類を強制する)
動的か	いいえ
検証	なし
どのような場合に 変更するか	igb ネットワークドライバによって使用される割り込みの種類を強制する 場合。
コミットレベル	変更の可能性あり

ixgbe Parameters

tx_queue_number

説明	このパラメータは、ixgbe ネットワークドライバによって使用される送信 キューの数を制御します。このパラメータの値を大きくすることによって、 送信キューの数を増やすことができます。このパラメータは、ixgbe ドライ バ接続が発生する前に /etc/driver/drv/ixgbe.conf ファイルを編集 することで設定できます。
データ型	符号なし整数

デフォルト	8
範囲	1 から 32
動的か	いいえ
検証	なし
どのような場合に 変更するか	ixgbe ネットワークドライバによって使用される送信キューの数を変更する 場合。
コミットレベル	変更の可能性あり

rx_queue_number

説明	このパラメータは、ixgbe ネットワークドライバによって使用される受信 キューの数を制御します。このパラメータの値を大きくすることによって、 受信キューの数を増やすことができます。このパラメータは、ixgbe ドライ バ接続が発生する前に /etc/driver/drv/ixgbe.conf ファイルを編集 することで設定できます。
データ型	符号なし整数
デフォルト	8
範囲	1 から 64
動的か	いいえ
検証	なし
どのような場合に 変更するか	ixgbe ネットワークドライバによって使用される受信キューの数を変更す る場合。
コミットレベル	変更の可能性あり

intr_throttling

説明	このパラメータは、ixgbe ネットワークドライバの割り込みスロットリング率 を制御します。このパラメータの値を小さくすることによって、割り込み率 を高くすることができます。このパラメータは、ixgbe ドライバ接続が発生 する前に /etc/driver/drv/ixgbe.conf ファイルを編集することで設定 できます。
データ型	符号なし整数

デフォルト	200
範囲	0 から 65535
動的か	いいえ
検証	なし
どのような場合に 変更するか	ixgbe ネットワークドライバによって使用される割り込みスロットリング率 を変更する場合。
コミットレベル	変更の可能性あり

rx_limit_per_intr

説明	このパラメータは、ixgbe ネットワークドライバによって使用される、割り込みあたりの受信キューバッファ記述子の最大数を制御します。このパラメータの値を大きくすることによって、受信キューバッファ記述子の数を増やすことができます。このパラメータは、ixgbe ドライバ接続が発生する前に /etc/driver/drv/ixgbe.conf ファイルを編集することで設定できます。
データ型	符号なし整数
デフォルト	256
範囲	16 から 4096
動的か	いいえ
検証	なし
どのような場合に 変更するか	ixgbe ネットワークドライバによって、割り込みあたりに処理される受信キューバッファ記述子の数を変更する場合。
コミットレベル	変更の可能性あり

tx_ring_size

説明	このパラメータは、ixgbe ネットワークドライバによって使用される送信キューサイズを制御します。このパラメータの値を大きくすることによって、送信キューサイズを増やすことができます。このパラメータは、ixgbe ドライバ接続が発生する前に /etc/driver/drv/ixgbe.conf ファイルを編集することで設定できます。
データ型	符号なし整数

デフォルト	1024
範囲	64 から 4096
動的か	いいえ
検証	なし
どのような場合に 変更するか	ixgbe ネットワークドライバによって使用される送信キューサイズを変更 する場合。
コミットレベル	変更の可能性あり

rx_ring_size

説明	このパラメータは、ixgbe ネットワークドライバによって使用される受信 キューサイズを制御します。このパラメータの値を大きくすることによって、 受信キューサイズを増やすことができます。このパラメータは、ixgbe ドラ イバ接続が発生する前に /etc/driver/drv/ixgbe.conf ファイルを編 集することで設定できます。
データ型	符号なし整数
デフォルト	1024
範囲	64 から 4096
動的か	いいえ
検証	なし
どのような場合に 変更するか	ixgbe ネットワークドライバによって使用される受信キューサイズを変更 する場合。
コミットレベル	変更の可能性あり

tx_copy_threshold

説明	このパラメータは、ixgbe ネットワークドライバによって使用される送信 バッファコピーのしきい値を制御します。このパラメータの値を大きくす ることによって、送信バッファコピーのしきい値を増やすことができま す。このパラメータは、ixgbe ドライバ接続が発生する前に /etc/driver/ drv/ixgbe.conf ファイルを編集することで設定できます。
データ型	符号なし整数

デフォルト	512
範囲	0 から 9126
動的か	いいえ
検証	なし
どのような場合に 変更するか	ixgbe ネットワークドライバによって使用される送信バッファコピーのしきい値を変更する場合。
コミットレベル	変更の可能性あり

rx_copy_threshold

説明	このパラメータは、ixgbe ネットワークドライバによって使用される受信バッファコピーのしきい値を制御します。このパラメータの値を大きくすることによって、受信バッファコピーのしきい値を増やすことができます。このパラメータは、ixgbe ドライバ接続が発生する前に /etc/driver/driv/ixgbe.conf ファイルを編集することで設定できます。
データ型	符号なし整数
デフォルト	128
範囲	0 から 9126
動的か	いいえ
検証	なし
どのような場合に 変更するか	ixgbe ネットワークドライバによって使用される受信バッファコピーのしきい値を変更する場合。
コミットレベル	変更の可能性あり

一般的な入出力パラメータ

maxphys

説明	物理入出力要求の最大サイズを指定します。要求がこのサイズより大きいと、ドライバはこの要求を maxphys サイズのチャンクに分割します。
----	---

個々のファイルシステムは独立して制限値を持つことが可能で、実際に独立した制限値を持ちます。

データ型	符号付き整数
デフォルト	131,072 (sun4u または sun4v) または 57,344 (x86)。ワイド転送をサポートする sd ドライバは 1,048,576 を使用します。ssd ドライバはデフォルトで 1,048,576 を使用します。
範囲	マシン固有のページサイズから MAXINT
単位	バイト
動的か	はい。しかし、多くのファイルシステムでは、ファイルシステムがマウントされるときに、この値がマウントポイントごとのデータ構造体に設定されません。ドライバによっては、デバイスがドライバ固有のデータ構造体に設定されるときに、この値が設定されます。
検証	なし
どのような場合に 変更するか	raw デバイスに対する入出力を大きなチャンクで行う場合。OLTP 操作を伴う DBMS では小さいサイズの入出力が頻繁に行われることに留意してください。その場合、maxphys を変更してもパフォーマンスは向上しません。
コミットレベル	変更の可能性あり

rlim_fd_max

説明	1 つのプロセスが開くことのできるファイル記述子に対して、強い限度を指定します。この制限をオーバーライドするには、スーパーユーザー特権が必要です。
データ型	符号付き整数
デフォルト	65,536
範囲	1 から MAXINT
単位	ファイル記述子
動的か	いいえ
検証	なし

どのような場合に 変更するか	<p>1 プロセス当たりの最大オープンファイル数が十分でない場合。ただし、システムには他の制限もあるため、ファイル記述子の数を増やすことが必ずしも有用とは限りません。例:</p> <ul style="list-style-type: none"> ■ 標準入出力を使用する 32 ビットプログラムでは、256 ファイル記述子に制限されます。標準入出力を使用する 64 ビットプログラムでは、20 億の記述子まで使用できます。具体的には、標準入出力は libc(3LIB) の stdio(3C) 関数を指します。 ■ select はデフォルトで、fd_set につき 1024 の記述子に制限されます。詳細は、select(3C) を参照してください。32 ビットアプリケーションコードは、より大きな fd_set サイズ (65,536 以下) で再コンパイルできます。64 ビットアプリケーションの fd_set サイズは 65,536 で、変更することはできません。 <p>システム全体に対してこれを変更する別の方法として plimit(1) コマンドがあります。plimit を使用して親となるプロセスの限度を変更すると、すべての子プロセスがその限度を継承します。この方法は inetd などのデーモンに有効です。</p>
コミットレベル	変更の可能性あり

rlim_fd_cur

説明	1 つのプロセスが開くことのできるファイル記述子に対して、「ソフト」限度を指定します。プロセスは、 setrlimit() 呼び出しを使用するか、プロセスが動作しているシェルで limit コマンドを実行することで、プロセスのファイル記述子数限度を、 rlim_fd_max で指定される「強い」限度の範囲内で調整できます。「強い」限度の範囲内で値を調整する場合には、スーパーユーザー特権は必要ありません。
データ型	符号付き整数
デフォルト	256
範囲	1 から MAXINT
単位	ファイル記述子
動的か	いいえ
検証	rlim_fd_max と比較します。 rlim_fd_cur が rlim_fd_max より大きい場合、 rlim_fd_cur は rlim_fd_max にリセットされます。

どのような場合に 変更するか	1 プロセス当たりのデフォルトのオープンファイル数が十分でない場合。 この値を増やす意味は、プログラムで <code>setrlimit</code> を使用して自身で使用できる最大ファイル記述子数を増やす必要がなくなる点にあります。
コミットレベル	変更の可能性あり

一般的なファイルシステムパラメータ

ncsize

説明	ディレクトリ名検索キャッシュ (DNLC) のエントリ数を指定します。このパラメータは、UFS、NFS、および ZFS が、解決されたパス名の要素をキャッシュするときに使用します。 DNLC は、否定的な検索情報もキャッシュします。これは、キャッシュ内で見つからない名前がキャッシュされることを意味します。
データ型	符号付き整数
デフォルト	$(4 \times (v.v_proc + \text{maxusers}) + 320) + (4 \times (v.v_proc + \text{maxusers}) + 320) / 100$
範囲	0 から MAXINT
単位	DNLC のエントリ
動的か	いいえ
検証	ありません。値を増やすと、ファイルシステムのアンマウントに必要な時間が増えます。これは、アンマウントプロセスでそのファイルシステムのエントリをキャッシュから削除する必要があるためです。
どのような場合に 変更するか	<code>kstat -n dnlcstats</code> コマンドを使用して、DNLC が小さすぎるために DNLC からエントリが削除されたことを知る事ができます。 <code>pick_heuristic</code> パラメータと <code>pick_last</code> パラメータの合計は、キャッシュが小さすぎるために再利用されたエントリ (そうでなければ有効であったはずのエントリ) の数を表します。 <code>ncsize</code> の値が大きすぎると、システムに直接的な影響があります。システムは、 <code>ncsize</code> の値に基づいて DNLC の一連のデータ構造体を割り当てるからです。デフォルトで、システムは、 <code>ncsize</code> に 64 バイトの構造体を割り当てます。 <code>ufs_ninode</code> と <code>nfs:nrnode</code> が明示的に設定されていないと、この値は UFS と NFS にさらに影響を与えます。

コミットレベル 変更の可能性あり

dnlc_dir_enable

説明 大きなディレクトリのキャッシングを有効にします。

注記 - このパラメータは NFS または ZFS ファイルシステムでは無効です。

データ型	符号なし整数
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい。しかし動的には変更しないでください。このパラメータは、元々無効だった場合に有効にできます。または、元々有効だった場合に、無効にできます。しかし、有効にし、無効にし、再び有効にすると、ディレクトリキャッシュが最新の状態を表さないことがあります。
検証	いいえ
どのような場合に 変更するか	ディレクトリキャッシングに既知の問題はありません。しかし、問題が生じた場合は、dnlc_dir_enable を 0 に設定してキャッシングを無効にしてください。
コミットレベル	変更の可能性あり

dnlc_dir_min_size

説明 1 つのディレクトリでキャッシュする最小エントリ数を指定します。

注記 - このパラメータは NFS または ZFS ファイルシステムでは無効です。

データ型	符号なし整数
デフォルト	40
範囲	0 から MAXUINT (無制限)
単位	エントリ
動的か	はい、このパラメータはいつでも変更できます。

検証	なし
どのような場合に 変更するか	小さいディレクトリのキャッシュにおいてパフォーマンスに問題がある場合は、 <code>dnlc_dir_min_size</code> を増やします。個々のファイルシステムに、キャッシングディレクトリの独自の範囲限度があることもある点に留意してください。
コミットレベル	変更の可能性あり

`dnlc_dir_max_size`

説明	1つのディレクトリでキャッシュできるエントリの最大数を指定します。
----	-----------------------------------

注記 - このパラメータは NFS または ZFS ファイルシステムでは無効です。

データ型	符号なし整数
デフォルト	MAXUINT (無制限)
範囲	0 から MAXUINT
動的か	はい、このパラメータはいつでも変更できます。
検証	なし
どのような場合に 変更するか	大きなディレクトリでパフォーマンスに問題がある場合は、 <code>dnlc_dir_max_size</code> を減らします。
コミットレベル	変更の可能性あり

`dnlc_dircache_percent`

説明	DNLC ディレクトリキャッシュが消費できる物理メモリーの最大の割合を計算します。
データ型	Integer
デフォルト	100
範囲	0 から 100
単位	%

動的か	いいえ
検証	ブート時に値の範囲がチェックされ、デフォルト値が適用されます。
どのような場合に 変更するか	システムにメモリー不足が発生し、カーネルメモリー使用量が高くなる場合、この値を下げることを検討してください。デフォルト値でパフォーマンス上の問題が見られる場合、値を増やすことを検討してください。

注記 - DNLC は UFS および ZFS ファイルシステムおよび NFS クライアントによって使用されます。メモリー不足およびカーネルメモリー使用量の増加が発生するか、ARC またはその他のカーネルキャッシュによってメモリーが必要となる場合、高いパフォーマンスを得るために、このチューニング可能パラメータの設定を検討する場合があります。

コミットレベル	変更の可能性あり
---------	----------

TMPFS パラメータ

tmpfs:tmpfs_maxmem

説明	TMPFS がデータ構造体 (tmp ノードとディレクトリエントリ) に使用できるカーネルメモリーの最大量を指定します。
データ型	符号なし long
デフォルト	1 ページまたは物理メモリーの 4% (どちらか大きい方)。
範囲	1 ページのバイト数 (sun4u か sun4v システムの場合は 8192、その他のシステムの場合は 4096) から、TMPFS が最初に使用されたときに存在していたカーネルメモリーの 25%。
単位	バイト
動的か	はい
検証	なし
どのような場合に 変更するか	次のメッセージがコンソールやメッセージファイルに出力される場合には、値を増やします。 tmp_memalloc: tmpfs over memory limit TMPFS がデータ構造体に現在使用しているメモリー量は、tmp_kmemspace フィールドにあります。カーネルデバッグを使用すると、このフィールドを検証できます。

コミットレベル 変更の可能性あり

tmpfs:tmpfs_minfree

説明	TMPFS がシステムの他の部分のために残しておくスワップ空間の最小量を指定します。
データ型	符号付き long
デフォルト	256
範囲	0 からスワップ空間サイズの最大値
単位	ページ
動的か	はい
検証	なし
どのような場合に 変更するか	TMPFS が大量に使用されるシステムで適度なスワップ空間を維持するために、この値を増やすことができます。次のメッセージがコンソールやメッセージファイルに出力された場合は、使用量がこの限度に達したことを示しています。 <i>fs-name: File system full, swap space limit exceeded</i>
コミットレベル	変更の可能性あり

仮想端末

Oracle Solaris ソフトウェアでは、仮想端末 (pty) は次の 2 つの目的で使用されます。

- telnet、rlogin、または rsh コマンドを使用したりリモートログインをサポートする。
- X ウィンドウシステムがコマンドインタプリタウィンドウを作成するときに使用するインタフェースを提供する。

デスクトップワークステーションの場合、仮想端末のデフォルト値で十分です。したがって、チューニングはリモートログオンに使用できる pty の数に焦点を当てます。

pty のデフォルト値は、現在、システムのメモリー容量に基づいて決まります。このデフォルト値を変更しなければならないのは、システムにログインできるユーザー数を制限したり増やしたりする場合だけです。

構成処理では、次の 3 つの関連する変数が使用されます。

- `pt_cnt` – `pty` 数のデフォルトの最大値。
- `pt_pctofmem` – `pty` サポート構造体専用に行うことができるカーネルメモリの割合 (%)。ゼロを指定すると、リモートユーザーがシステムにログインすることはできません。
- `pt_max_pty` – `pty` 数の強い制限の最大値

`pt_cnt` のデフォルト値はゼロで、`pt_max_pty` が設定されていないかぎり、システムは `pt_pctofmem` に指定されたメモリ量に基づいてログインを制限します。`pt_cnt` がゼロでない場合は、この制限に達するまで `pty` が割り当てられます。この制限に達すると、システムは `pt_max_pty` を参照します。`pt_max_pty` の値がゼロ以外の場合、`pt_cnt` と比較されます。`pt_cnt` が `pt_max_pty` より小さい場合は、`pty` を割り当てることができます。`pt_max_pty` がゼロの場合は、`pt_cnt` が、`pt_pctofmem` に基づいてサポートされる `pty` の数と比較されます。`pt_cnt` がこの数より小さければ、`pty` 割り当てが認められます。`pt_pctofmem` に基づいた制限値が有効となるのは、`pt_cnt` と `ptms_ptymax` のデフォルト値が両方ともゼロの場合だけであることを留意してください。

`pty` の強い制限値を、`pt_pctofmem` から計算される最大値と異なるものにするには、`/etc/system` の `pt_cnt` と `ptms_ptymax` に望ましい `pty` 数を設定します。この場合、`ptms_pctofmem` の設定は関連しません。

システムメモリの特定の割合を `pty` サポートのためだけに割り当て、明示的な限度の管理をオペレーティングシステムに任せる場合は、次のようにします。

- `/etc/system` の `pt_cnt` と `ptms_ptymax` を設定しない。
- `/etc/system` の `pt_pctofmem` に望ましい割合 (%) を設定する。たとえば、10% を割り当てる場合、`pt_pctofmem=10`。

このメモリは、`pty` のサポートに使用されるまで実際に割り当てられません。しかし、メモリがいったん割り当てられると、解放されません。

pt_cnt

説明

使用できる `/dev/pts` エントリの数は、システム上で使用できる物理メモリ容量によって決まる限度の範囲内で動的です。`pt_cnt` は、システムがサポートできるログイン数の最小値を決める 3 つの変数のうちの 1 つです。システムがサポートできる `/dev/pts` デバイスのデフォルトの最大数は、ブート時に、指定されたシステムメモリの割合 (`pt_pctofmem`

を参照) に適合する `pty` 構造体の数を計算することによって決められます。`pt_cnt` がゼロの場合、システムはこの最大数まで割り当てます。`pt_cnt` がゼロでない場合は、システムは `pt_cnt` かデフォルトの最大数のうち大きい方まで割り当てます。

データ型	符号なし整数
デフォルト	0
範囲	0 から <code>maxpid</code>
単位	ログイン / ウィンドウ
動的か	いいえ
検証	なし
どのような場合に 変更するか	システムにリモートからログインできるユーザーの数を明示的にコントロールしたい場合
コミットレベル	変更の可能性あり

pt_pctofmem

説明	データ構造体が <code>/dev/pts</code> エントリをサポートするために消費できる物理メモリーの最大の割合を指定します。システムは、 <code>/dev/pts</code> エントリあたり 176 バイトを使用します。
データ型	符号なし整数
デフォルト	5
範囲	0 から 100
単位	%
動的か	いいえ
検証	なし
どのような場合に 変更するか	システムにログインできるユーザーの数を制限するか増やしたい場合。ゼロを指定すると、リモートユーザーがシステムにログインすることはできません。

コミットレベル 変更の可能性あり

pt_max_pty

説明	システムが提供する ptys の最大数を指定します。
データ型	符号なし整数
デフォルト	0 (システムが定義した最大数を使用する)
範囲	0 から MAXUINT
単位	ログイン / ウィンドウ
動的か	はい
検証	なし
暗黙的制約	pt_cnt 以上にすべきです。値が検査されるのは、割り当てられた ptys 数が pt_cnt の値を超過してからです。
どのような場合に 変更するか	システムが、構成値に基づいてより多くのログインをサポートできる場合であっても、サポートするログイン数の絶対的な上限を設定したい場合。
コミットレベル	変更の可能性あり

STREAMS パラメータ

nstrpush

説明	STREAM に追加 (格納) できるモジュールの数を指定します。
データ型	符号付き整数
デフォルト	9
範囲	9 から 16
単位	モジュール
動的か	はい

検証	なし
どのような場合に 変更するか	ソフトウェアベンダーの指定がある場合。STREAM が許可されている プッシュカウントを超えても、メッセージは出されません。プッシュを試み たプログラムに EINVAL という値が返されます。
コミットレベル	変更の可能性あり

strmsgsz

説明	1 つのシステム呼び出しで STREAM に渡し、メッセージのデータ部分に 格納できる最大バイト数を指定します。このサイズを超える write は、複 数のメッセージに分割されます。詳細は、 write(2) を参照してください。
データ型	符号付き整数
デフォルト	65,536
範囲	0 から 262,144
単位	バイト
動的か	はい
検証	なし
どのような場合に 変更するか	putmsg 呼び出しから ERANGE が返された場合。詳細は、 putmsg(2) を 参照してください。
コミットレベル	変更の可能性あり

strctlsz

説明	1 つのシステム呼び出しで STREAM に渡し、メッセージの制御部分に 格納できる最大バイト数を指定します。
データ型	符号付き整数
デフォルト	1024
範囲	0 から MAXINT

単位	バイト
動的か	はい
検証	なし
どのような場合に 変更するか	ソフトウェアベンダーの指定がある場合。この限度を超える と、 <code>putmsg(2)</code> 呼び出しから <code>ERANGE</code> が返されます。
コミットレベル	変更の可能性あり

System V メッセージキュー

System V メッセージキューは、カーネルが作成したキューを使用してメッセージを交換する、メッセージ転送インタフェースを提供します。Oracle Solaris 環境には、メッセージをキューに入れたりキューから取り出したりするためのインタフェースが用意されています。メッセージは、自身の型を持つことができます。キューに入れる場合、メッセージはキューの終わりに置かれます。キューを解除する場合は、指定された型の最初のメッセージがキューから削除されます。型が指定されていない場合は、最初のメッセージが削除されます。

これらのシステムリソースのチューニングに関する詳細は、『[Oracle Solaris 11.2 でのリソースの管理](#)』の第 6 章「リソース制御について」を参照してください。

System V セマフォ

System V セマフォは Oracle Solaris OS で計数型セマフォを提供します。「セマフォ」は、複数のプロセスが共有データオブジェクトにアクセスできるようにする場合に使用するカウンタです。System V セマフォでは、セマフォの標準的な設定/解放操作の他に、必要に応じて増分や減分を行う値を持つことができます (たとえば、使用可能なりソースの数を表すなど)。System V セマフォによって、1 組のセマフォに同時に操作を実行したり、プロセスが停止した場合にそのプロセスによる最後の操作を取り消したりすることもできます。

System V 共有メモリー

System V 共有メモリーでは、プロセスによるセグメントの作成が可能です。連携するプロセスがそのメモリーセグメントに接続し (セグメントに対するアクセス権が必要)、セグメントに含

まれるデータにアクセスできます。この機能はロード可能モジュールとして実装されます。`/etc/system` ファイルのエントリには、`shmsys:` 接頭辞が含まれている必要があります。

DBMS ベンダーは、パフォーマンスを高めるために、*intimate shared memory* (ISM) と呼ばれる特殊な共有メモリーを使用しています。共有メモリーセグメントを ISM セグメントにすると、そのセグメントのメモリーがロックされます。この機能によってより高速な入出力経路をたどることができ、メモリーの使用効率が向上します。セグメントを記述する一連のカーネルリソースは、ISM モードでセグメントに接続するすべてのプロセスによって共有されます。

segspt_minfree

説明	ISM 共有メモリーに割り当てることのできないシステムメモリーのページ数を指定します。
データ型	符号なし long
デフォルト	最初の ISM セグメントが作成される時に使用可能なシステムメモリーの 5%
範囲	物理メモリーの 0 から 50 %
単位	ページ
動的か	はい
検証	ありません。値が小さすぎると、メモリーが ISM セグメントに消費される時に、システムがハングしたりパフォーマンスが大幅に低下することがあります。
どのような場合に 変更するか	大量のメモリーがあるデータベースシステムで ISM を使用する場合、このパラメータの値を引き下げることができます。ISM セグメントが使用されない場合には、このパラメータの効果はありません。大量のメモリーを備えたマシンでは、ほぼ間違いなく、最大値 128M バイト (0x4000) で十分です。
コミットレベル	変更の可能性あり

pr_segdisable

説明	ISM に属する可能性のあるページを回収しようとするときに、ページロックキャッシュのフラッシュを無効にします。
----	---

ロック済みまたはビジー状態の (高負荷 I/O) ページが保留中のページ回収キューに格納されると、ISM で所有されている可能性のある保留中のページを回収させるために、ページ回収スレッドによって `segp_cache` がフラッシュされます。`segp_cache` を定期的または繰り返しフラッシュすると、メモリー負荷の高いマシンのボトルネックとなる可能性があります。

デフォルト動作では、30 秒ごとにページキャッシュがフラッシュされ、ロック済みページがキューで見つかると、タイムアウトは 2 の倍数単位で 1 時間まで指数関数的にバックオフします。

`pr_segp_disable` を有効にしても、システム診断対策の結果として障害が発生しているメモリーページなどを回収するシステムの機能は無効になりません。

データ型	ブール型
デフォルト	1 (無効)
範囲	0 (有効) および 1 (無効)
動的か	いいえ
検証	いいえ

どのような場合に
変更するか

ロック済みまたはビジー状態の (高負荷 I/O) ページが保留中のページ回収キューに格納されると、ISM で所有されている可能性のある保留中ページを回収させるために、ページ回収スレッドによって `segp_cache` がフラッシュされます。`segp_cache` を定期的または繰り返しフラッシュすると、メモリー負荷の高いマシンのボトルネックとなる可能性があります。遅延の影響を受けるデータベースまたは共有メモリーが大きいアプリケーションを使用している場合は、このパラメータを無効にして、`segp` キャッシュのフラッシュを完全にスキップすることを検討してください。

回収できないロック済みのカーネルページの現象は次のとおりです。

- ページ回収が正常に実行されると、短時間の定期的な高い SYS CPU イベントとともに、短時間のデータベース遅延や一時的にデータベースが応答しないイベントが発生しますが、回収が繰り返し失敗するロック済みまたはビジー状態のページにより、ページ回収スレッドのトリガがより低速で続行される可能性があります。

たとえば、回収できないロック済みのメモリーページは、短い間隔で再試行され、1 時間間隔で永続的に繰り返される可能性があります。システムのリブート後に、スケジュール済みのページが回収されるか、または 30 秒 (デフォルトの速度) で再試行が開始される可能性があります。

- `segspt_shmfault`、`segspt_softunlock`、`segspt_shmpagelock`、`segspt_shmfree`、`segspt_shmunmap`、`segspt_shmattach`、および `segspt_dismfault` 構造体をモニタリングすると、短時間の予期しない、または高い `smtx` ロック競合が確認される可能性があります。

コミットレベル 変更の可能性あり

スケジューリング

`disp_rechoose_interval`

説明

以前の `rechoose_interval` パラメータと同様、このパラメータは、プロセスが最後に実行された CPU のすべてのアフィニティーが失われたとみなされるまでの時間を指定します。ただし、このパラメータはより細かい時間の増分で設定します。このパラメータは、非推奨の `rechoose_interval` パラメータの代わりに使用する必要がありますが、`rechoose_interval` パラメータも `/etc/system` ファイルに設定されている場合に引き続き受け付けられます。

この期間が過ぎると、すべての CPU はスレッドスケジューリングの候補と見なされます。このパラメータは、リアルタイムクラスのスレッドには適用されませんが、ほかのすべてのスケジューリングクラスのスレッドに適用されます。

このパラメータの値を変更する場合に、次の手順で、`mdb` を使用します。

1. ナノ秒を、拡大縮小しない時間に変換します。たとえば、5,000,000 ナノ秒ベースの値を、拡大縮小しない時間に変換するには、次の構文を使用します。

```
# mdb -kw
.
.
.
> 0t5000000::time -u
0xb6a444
```

2. `disp_rechoose_interval` を拡大縮小しない時間の値に設定します。たとえば、前の手順で返された値を指定します。

```
> disp_rechoose_interval /Z 0xb6a444
```

```
disp_rechoose_interval: 0x447d998 = 0xb6a444
```

3. `disp_rechoose_interval` が正しい値に設定されていることを確認します。例:

```
> disp_rechoose_interval::print
0xb6a444
```

データ型	符号付き整数
デフォルト	3
範囲	0 から MAXINT
動的か	はい
検証	なし
どのような場合に 変更するか	キャッシュが大きい場合、極めて重要なプロセスがシステムで動作している場合や、データアクセスパターン以外の原因により、一連のプロセスで過度のキャッシュミスが発生していると思われる場合。 このパラメータを変更する前に、プロセッサセットの機能またはプロセッサバインディングの使用を検討してください。詳細は、 psrset(1M) または pbind(1M) を参照してください。
コミットレベル	変更の可能性あり

タイマー

hires_tick

説明	このパラメータを設定すると、Oracle Solaris OS はシステムクロックレートとして、デフォルト値の 100 ではなく 1000 を使用します。
データ型	符号付き整数
デフォルト	0
範囲	0 (無効)、1 (有効)
動的か	いいえ。新しいシステムタイミング変数はブート時に設定されます。ブート後は、このパラメータは参照されません。

検証	なし
どのような場合に 変更するか	10 ミリ秒未満、1 ミリ秒以上の分解能を持つタイムアウトが必要な場合
コミットレベル	変更の可能性あり

timer_max

説明	使用できるPOSIX™ タイマーの数を指定します。
データ型	符号付き整数
デフォルト	1000
範囲	0 から MAXINT
動的か	いいえ。値を増やすと、システムクラッシュを起こす可能性があります。
検証	なし
どのような場合に 変更するか	システムのデフォルトのタイマー数では不十分な場合。アプリケーションは timer_create システムコールの実行時に、EAGAIN エラーを受け取ります。
コミットレベル	変更の可能性あり

SPARC: プラットフォーム固有のパラメータ

次のパラメータは、sun4v および SPARC M-Series sun4u プラットフォームに適用されます。

tsb_alloc_hiwater_factor

説明	<p>tsb_alloc_hiwater を初期化して、変換ストレージバッファ (TSB) に割り当てることのできる物理メモリー量に、次のように上限を設けます。</p> $\text{tsb_alloc_hiwater} = \text{物理メモリー (バイト数)} / \text{tsb_alloc_hiwater_factor}$ <p>TSB に割り当てられたメモリーが tsb_alloc_hiwater の値と等しい場合、TSB メモリー割り当てアルゴリズムはページがマップ解除されると TSB メモリーを再利用しようとします。</p>
----	---

この係数を使用して `tsb_alloc_hiwater` の値を増やす場合は、注意が必要です。システムのハングを防止するには、高水位値が `swapfs_minfree` と `segspt_minfree` の値よりかなり小さくなるようにする必要があります。

データ型	Integer
デフォルト	32
範囲	1 から MAXINIT 係数 1 の場合、すべての物理メモリーを TSB に割り当てることができるようになるので、システムが停止する可能性があります。また、係数が大きすぎると、TSB に割り当てることができるメモリーが残らないので、システムパフォーマンスが低下します。
動的か	はい
検証	なし
どのような場合に 変更するか	非常に大型の共有メモリーセグメントに接続するプロセスがシステムに多数ある場合、このパラメータ値を変更します。ほとんどの場合、この変数のチューニングは不要です。
コミットレベル	変更の可能性あり

default_tsb_size

説明	すべてのプロセスに割り当てる初期変換ストレージバッファ (TSB) のサイズを選択します。
データ型	Integer
デフォルト	デフォルト値は 0 (8K バイト) で、これは 512 エントリに対応します。
範囲	取り得る値は次のとおりです。

値	説明
0	8K バイト
1	16K バイト
3	32K バイト
4	128K バイト
5	256K バイト

値	説明
6	512K バイト
7	1M バイト

動的か	はい
検証	なし
どのような場合に 変更するか	通常、この値を変更する必要はありません。しかし、システム上のプロセスの大半が平均より大きい作業用セットを使用する場合、または常駐セットサイズ (RSS) のサイズ調整が無効な場合は、この値を変更することによって利益が得られることもあります。
コミットレベル	変更の可能性あり

enable_tsb_rss_sizing

説明	TSB 発見的容量調整に基づく常駐セットサイズ (RSS) を有効にします。
データ型	ブール型
デフォルト	1 (TSB のサイズ変更が可能)
範囲	0 (TSB は <code>tsb_default_size</code> のまま) または 1 (TSB のサイズ変更が可能) 0 に設定した場合、 <code>tsb_rss_factor</code> は無視されます。
動的か	はい
検証	はい
どのような場合に 変更するか	0 に設定すると、TSB の増加を防ぐことができます。ほとんどの場合、このパラメータはデフォルト設定のままにしておくべきです。
コミットレベル	変更の可能性あり

tsb_rss_factor

説明	RSS 発見的容量調整の RSS 対 TSB 範囲比を制御します。この係数を 512 で割ると、TSB がサイズ変更候補とみなされるまでに、メモリーに常駐していなければならない TSB 範囲の割合が出ます。
----	---

データ型	Integer
デフォルト	384。これは 75% の値になります。このため、TSB が 3/4 に達するとサイズが増やされます。いくつかの仮想アドレスは通常、TSB の同じスロットにマップされます。したがって、TSB が 100% に達する前に衝突が起こることがあります。
範囲	0 から 512
動的か	はい
検証	なし
どのような場合に 変更するか	TSB での仮想アドレスの衝突による場合など、システムが TSB ミスに起因する過度の数のトラップに直面している場合は、この値を 0 に減らしてもよいかもしれません。 たとえば、 <code>tsb_rss_factor</code> を 384 (事実上は 75%) ではなく 256 (事実上は 50%) に変更すると、状況によっては、TSB における仮想アドレスの衝突を排除できることがあります。特に負荷の大きいシステムでは、カーネルメモリの使用量が増えます。 TSB の動きは、 <code>trapstat -T</code> コマンドでモニターできます。
コミットレベル	変更の可能性あり

近傍性グループのパラメータ

このセクションでは、NUMA (Non-Uniform Memory Architecture) を使用するどの SPARC または x86 システムにも適用できる、汎用的なメモリーチューニング可能パラメータについて説明します。

`lpg_alloc_prefer`

説明	大規模なメモリーページの割り当てを行う際に、要求されたページサイズがローカルのメモリーグループ内ではすぐに利用できないがリモートメモリーグループからであれば要求を満たせるときにヒューリスティックを制御します。 デフォルトでは、ローカルの空きメモリーは断片化されているが、リモートの空きメモリーは断片化されていない場合に、Oracle Solaris OS はリモート大規模ページを割り当てます。このパラメータを 1 に設定した場合、大規模なメモリーページをローカルで割り当てるため、たとえばローカルの
----	---

メモリーグループ内で小さなページを集めて大きなページに合体させるといった追加動作が行われます。

データ型	ブール型
デフォルト	0 (ローカルの空きメモリーが断片化されていて、リモートの空きメモリーが断片化されていない場合は、リモート割り当てを優先する)
範囲	0 (ローカルの空きメモリーが断片化されていて、リモートの空きメモリーが断片化されていない場合は、リモート割り当てを優先する) 1 (ローカルの空きメモリーが断片化されていて、リモートの空きメモリーが断片化されていない場合でも、可能な場合は常にローカル割り当てを優先する)
動的か	いいえ
検証	なし
どのような場合に 変更するか	このパラメータを 1 に設定することが考えられるのは、システム上で長時間動作する複数のプログラムが割り当てる傾向にあるメモリーが単一のプログラムによってアクセスされている場合、または複数プログラムのグループによってアクセスされるメモリーが同じ近傍性グループ (lgroup) 内で使用されていることがわかっている場合です。これらの状況では、ページ合体操作の余分なコストをプログラムの長い実行時間にわたって償却することができます。 このパラメータをデフォルト値 (0) のままにすることが考えられるのは、複数のプログラムが異なる近傍性グループにわたってメモリーを共有する傾向にある場合や、ページが短期間だけ使用される傾向にある場合です。このような状況では、特定の場所における割り当てよりも、要求されたサイズをすばやく割り当てることの方がより重要となります。 TLB ミスは、 <code>trapstat -T</code> コマンドを使用して監視できます。
コミットレベル	不確実

lgrp_mem_pset_aware

説明	プロセスがユーザープロセッサセット内で実行されている場合は、この変数によって、このプロセスのためにランダムに配置されたメモリーがシステム内のすべての lgroup から選択されるのか、またはプロセッサセット内のプロセッサで構成されている lgroup のみから選択されるのかが決定されます。 プロセッサセットの作成についての詳細は、 psrset(1M) を参照してください。
----	---

データ型	ブール型
デフォルト	0。Oracle Solaris OS はシステム内のすべての lgroup からメモリーを選択します
範囲	<ul style="list-style-type: none">■ 0。Oracle Solaris OS はシステム内のすべての lgroup からメモリーを選択します (デフォルト)。■ 1。プロセッサセット内のプロセッサで構成されている lgroup のみからメモリーを選択しようと試みます。最初の試みが失敗した場合は、任意の lgroup 内のメモリーを割り当てることができます。
動的か	いいえ
検証	なし
どのような場合に 変更するか	この値を 1 に設定すると、プロセッサセットがほかのアプリケーションからアプリケーションを切り離すために使用されている場合に、再現性のあるパフォーマンスが得られる可能性があります。
コミットレベル	不確実

◆◆◆ 第 3 章

Oracle Solaris ZFS チューニング可能パラメータ

この章では、システムおよびアプリケーションの要件に応じて考慮する必要がある ZFS チューニング可能パラメータについて説明します。さらに、データベース製品と一緒に ZFS を使用する際のチューニング可能パラメータの推奨事項も提供します。

- 85 ページの「ZFS のチューニングの考慮事項」
- 86 ページの「ZFS ARC パラメータ」
- 87 ページの「ZFS ファイルレベルプリフェッチ」
- 88 ページの「ZFS デバイスの入出力キューの深さ」
- 90 ページの「フラッシュストレージ使用時の ZFS のチューニング」
- 95 ページの「データベース製品に対する ZFS のチューニング」

ほかのタイプのチューニング可能パラメータについては、次を参照してください。

- Oracle Solaris カーネルチューニング可能パラメータ – 第2章「Oracle Solaris カーネルチューニング可能パラメータ」
- NFS チューニング可能パラメータ – 第4章「NFS チューニング可能パラメータ」
- インターネットプロトコルスイートのチューニング可能パラメータ – 第5章「インターネットプロトコル群のチューニング可能パラメータ」
- システム機能のチューニング可能パラメータ – 第6章「システム機能のパラメータ」

ZFS のチューニングの考慮事項

ZFS をチューニングする前に、次の考慮事項を検討してください。

- 一般的に、デフォルト値が最適値です。より適切な値が存在する場合、それがデフォルトになっているはずですが、代替の値は特定のワークロードに役立つ場合もありますが、何らかの別の側面でパフォーマンスを低下させる可能性がきわめて高くなります。場合によっては、破局的な結果となります。

- ZFS のチューニングを適用する前に、ZFS のベストプラクティスに従うようにします。これらの方法は、さまざまな環境で動作することが証明された一連の推奨で、予見できる将来にわたって動作し続けることが見込まれています。したがって、チューニングを行う前に、ベストプラクティスを読んで理解するようにしてください。詳細は、『Oracle Solaris 11.2 での ZFS ファイルシステムの管理』の第 11 章「推奨の Oracle Solaris ZFS プラクティス」を参照してください。
- 別途記載のないかぎり、チューニング可能パラメータはグローバルであり、システム全体にわたって ZFS の動作に影響を及ぼします。

注記 - このリリースの ZFS ARC パラメータをチューニングする前に、Oracle Solaris 11.2 での ZFS とアプリケーションのメモリー管理に関する MOS ドキュメント 166382.1 を参照してください。

ZFS ARC パラメータ

このセクションでは ZFS ARC の動作に関するパラメータについて説明します。

zfs_arc_min

説明	ZFS 適応型置換キャッシュ (ARC) の最小サイズを決定します。 87 ページの「zfs_arc_max」 も参照してください。
データ型	符号なし整数 (64 ビット)
デフォルト	64M バイト
範囲	64M バイトから <code>zfs_arc_max</code>
単位	バイト
動的か	いいえ
検証	はい、範囲が検証されます。
どのような場合に 変更するか	システムのメモリー負荷要求が変動するとき、ZFS ARC は要求の少ない時期はデータをキャッシュし、要求の多い時期は縮小します。ただし、ZFS は <code>zfs_arc_min</code> の値を下回って縮小することはありません。一般には、デフォルト値を変更する必要はありません。

コミットレベル 変更の可能性あり

zfs_arc_max

説明	ZFS 適応型置換キャッシュ (ARC) の最大サイズを決定します。 86 ページの「zfs_arc_min」 も参照してください。
データ型	符号なし整数 (64 ビット)
デフォルト	搭載メモリーが 4G バイト未満のシステムでは、メモリーの 75% 搭載メモリーが 4G バイトを超えるシステムでは、physmem から 1G バイトを引いた値
範囲	64M バイトから physmem
単位	バイト
動的か	いいえ
検証	はい、範囲が検証されます。
どのような場合に 変更するか	将来のメモリー要求が非常に大きく、十分に定義されている場合、ARC がメモリー要求と競合しないように、このパラメータの値を減らして ARC を制限することを検討できます。たとえば、将来の負荷がメモリーの 20% を要求することがわかっている場合、残りの 80% より多くのメモリーを消費しないように ARC を制限することは意味があります。
コミットレベル	変更の可能性あり

ZFS ファイルレベルプリフェッチ

zfs_prefetch_disable

説明	このパラメータは、zfetch と呼ばれるファイルレベルプリフェッチメカニズムを決定します。このメカニズムはファイルの読み取りパターンを参照し、一部の読み取りを予測することによって、アプリケーションの待ち時間を削減します。現在の動作には 2 つの欠点があります。 <ul style="list-style-type: none"> ■ 少量の読み取りからなる順次読み取りパターンでは、キャッシュ内でヒットする頻度が非常に高くなります。この場合、現在の動作は次に
----	--

実行する入出力を検出しようとするために大量の CPU 時間を消費し、一方で、パフォーマンスは CPU の可用性によって大きく支配されます。

- `zfetch` コードが一部の負荷のスケラビリティを制限するということが報告されています。CPU プロファイリングは、ここに記載するように、`lockstat -I` コマンドまたは `er_kernel` を使用して実行できます。

<http://www.oracle.com/technetwork/java/index.html>

プリフェッチは `/etc/system` ファイルの `zfs_prefetch_disable` を設定することによって無効にできます。

デバイスレベルプリフェッチは `zfs_vdev_cache_size` を無効にすると無効化されます。つまり、`zfs_vdev_cache_size` が無効化されると、`vdev cache shift` のチューニングは不要になります。

データ型	ブール型
デフォルト	0 (有効)
範囲	0 (有効) または 1 (無効)
動的か	はい
検証	いいえ
どのような場合に 変更するか	<code>er_kernel</code> の結果、 <code>zfetch_*</code> 機能にかなりの時間がかかることが示されたり、または <code>lockstat</code> によるロックのプロファイリングで <code>zfetch</code> ロックに関する競合が示される場合は、ファイルレベルプリフェッチを無効化することを検討するようにしてください。
コミットレベル	変更の可能性あり

ZFS デバイスの入出力キューの深さ

`zfs_vdev_max_pending`

説明	このパラメータは、各デバイスに対する保留中の同時入出力の最大数を制御します。
----	--

データ型	Integer
デフォルト	10
範囲	0 から MAXINT
動的か	はい
検証	いいえ
どのような場合に 変更するか	<p>LUN が多数のディスクドライブで構成されるストレージレイでは、ZFS キューが読み取り IOPS の制限要因となる可能性があります。この動作は、補助スピンドルが存在するかぎりできるだけ多くの LUN を ZFS ストレージプールに提供するというベストプラクティスの根拠の 1 つになります。つまり、10 個のディスクによるアレイレベルの RAID グループで LUN を作成した場合、5 から 10 個の LUN を使用してストレージプールを構築すると、ZFS は十分な入出力キューを管理できるようになり、この特定のチューニング可能パラメータを設定する必要はありません。</p> <p>ただし、別個のインテントログが使用されておらず、プールが JBOD ディスクで構成されている場合は、ディスクリソースへの競合が発生しているため、小さい <code>zfs_vdev_max_pending</code> 値 (10 など) を使用すると、同期書き込みの待ち時間が改善される可能性があります。別々のインテントログデバイスを使用すると、これらの同期書き込みは、非同期書き込みの深いキューと競合しないため、同期書き込みを多用する負荷に対してこのパラメータをチューニングする必要性が軽減されます。</p> <p>ボリュームが少数のスピンドルで構成されている場合、NVRAM ベースのストレージレイでは、このパラメータをチューニングしても効果は期待されません。ただし、ZFS に提供されたボリュームが多数のスピンドル (10 を超える) で構成されている場合、このパラメータはボリュームで得られる読み取りスループットを制限する可能性があります。この理由は、LUN あたり最大 10 または 35 の入出力がキューに入るため、ストレージスピンドルあたりの入出力は 1 を下回ることになり、個々のディスクがそれらの IOPS を発揮するのに十分ではないためです。この問題は、<code>iostat actv</code> キュー出力が <code>zfs_vdev_max_pending</code> の値に近づいていることとして示されます。</p> <p>デバイスドライバも、LUN あたりの未処理入出力数を制限することがあります。大量の並行 IOPS を処理できるストレージレイ上で LUN を使用している場合、デバイスドライバの制約も並行性を制限することがあります。システムで使用しているドライバの構成を調べてください。たとえば、QLogic ISP2200、ISP2300、および SP212 ファミリー FCI HBA (qlc) ドライバの制限は、<code>/kernel/drv/qlc.conf</code> の <code>execution-throttle</code> パラメータとして説明されています。</p>
コミットレベル	変更の可能性あり

フラッシュストレージ使用時の ZFS のチューニング

次の情報は、Flash SSD、F20 PCIe Accelerator カード、F40 PCIe Accelerator カード、F5100 フラッシュストレージアレイ、および F80 PCIe Accelerator カードに適用されます。

フラッシュストレージで ZFS を使用する場合は、次の一般的なコメントを確認してください。

- ZIL (ZFS インテントログ) には、永続的メモリーを使用するコントローラによって管理された LUN または低遅延ディスク (使用可能な場合) の使用を検討してください。このオプションは、コミットの遅延を短縮するためのフラッシュの使用に比べて、費用対効果が大幅に向上する場合があります。ログデバイスのサイズは、最大書き込みスループットを 10 秒間保持するのに十分な大きさのみにとどめる必要があります。例として、ストレージアレイベースの LUN や、バッテリーで保護された書き込みキャッシュを持つ HBA に接続されたディスクがあります。

このようなデバイスが使用できない場合は、ZFS ストレージプール内のログデバイスとして使用するために、フラッシュデバイスの個別のプールをセグメント分割します。

- F40、F20、および F80 Flash Accelerator カードは 4 つの独立したフラッシュモジュールを内蔵し、それを OS にエクスポートします。F5100 は、最大 80 の独立したフラッシュモジュールを内蔵します。各フラッシュモジュールは、オペレーティングシステムには 1 つのデバイスとして表示されます。SSD は、OS によって 1 つのデバイスとして表示されます。フラッシュデバイスは、特に NFS サーバーで使用された場合、コミットの遅延を削減するために ZFS ログデバイスとして使用される場合があります。たとえば、フラッシュデバイスの 1 つのフラッシュモジュールを ZFS ログデバイスとして使用することによって、1 つの軽量スレッド操作の遅延を 10 分の 1 に短縮できます。大量の同期操作のスループットを向上させるには、さらに多くのフラッシュデバイスをまとめてストライプ化できます。
- ログデバイスは、信頼性のためにミラー化してください。最大の保護を実現するために、ミラーは個別のフラッシュデバイス上に作成してください。F20、F40、および F80 PCIe アクセラレータカードの場合は、ミラーを異なる物理 PCIe カード上に配置することによって最大の保護が実現されます。F5100 ストレージアレイでの最大の保護は、ミラーを個別の F5100 デバイス上に配置することによって実現されます。
- ログデバイスとして使用されていないフラッシュデバイスは、2 番目のレベルのキャッシュデバイスとして使用できます。これは、プライマリディスクストレージから IOPS の負荷を軽減するだけでなく、一般的に使用されるデータの読み取りの遅延を改善させるためにも役立ちます。

ZFS ログデバイスまたはキャッシュデバイスとしてのフラッシュデバイスの追加

ZFS ログデバイスまたはキャッシュデバイスとしてフラッシュデバイスを追加する場合は、次の推奨事項を確認してください。

- `zpool add` コマンドを使用すると、ZFS ログデバイスまたはキャッシュデバイスを既存の ZFS ストレージプールに追加できます。`zpool add` コマンドは、十分に注意して使用してください。ログデバイスを通常のプールデバイスとして誤って追加すると、そのプールを破棄し、最初から復元する作業が必要になります。個々のログデバイス自体をプールから削除できます。
- アクティブなストレージに対してこの操作を実行しようとする前に、`zpool add` コマンドについて十分に理解してください。`zpool add -n` オプションを使用すると、構成を作成することなく構成をプレビューできます。たとえば、次の正しくない `zpool add` プレビュー構文は、デバイスをログデバイスとして追加しようとしています。

```
# zpool add -n tank c4t1d0
vdev verification failed: use -f to override the following errors:
mismatched replication level: pool uses mirror and new vdev is disk
Unable to build pool from specified devices: invalid vdev configuration
```

ログデバイスを既存のプールに追加するための正しい `zpool add` プレビュー構文を次に示します。

```
# zpool add -n tank log c4t1d0
would update 'tank' to the following configuration:
tank
mirror
c4t0d0
c5t0d0
logs
c4t1d0
```

複数のデバイスが指定された場合は、それらがまとめてストライプ化されます。詳細は、次の例または [zpool\(1M\)](#) を参照してください。

フラッシュデバイス `c4t1d0` を ZFS ログデバイスとして追加できます。

```
# zpool add pool log c4t1d0
```

2 つのフラッシュデバイスを使用できる場合は、ミラー化されたログデバイスを追加できます。

```
# zpool add pool log mirror c4t1d0 c4t2d0
```

使用可能なフラッシュデバイスを読み取り用のキャッシュデバイスとして追加できます。

```
# zpool add pool cache c4t3d0
```

キャッシュデバイスはミラー化できず、まとめてストライプ化されます。

```
# zpool add pool cache c4t3d0 c4t4d0
```

フラッシュおよび NVRAM ストレージデバイスの適切な キャッシュフラッシュ動作の保証

ZFS は、ディスクレベルキャッシュを管理するストレージデバイスと一緒に動作するように設計されています。ZFS は一般的にストレージデバイスに対し、キャッシュフラッシュをリクエストすることによって、データを安定したストレージ上に安全に配置するよう要求します。JBOD ストレージの場合、これは設計どおりに動作するため問題はありません。NVRAM ベースの多くのストレージアレイでは、アレイがキャッシュフラッシュリクエストを取得し、それを無視せずに何らかの処理を実際に行う場合、パフォーマンスの問題が発生することがあります。一部のストレージアレイでは、NVRAM 保護によってこれらのキャッシュが安定したストレージと同様に維持されているにもかかわらず、大量のキャッシュをフラッシュします。

ZFS は uberblock 更新のあと、頻繁でないフラッシュを (5 秒程度の間隔で) 実行します。頻繁でないフラッシュはあまり重要でないため、このチューニングの根拠とはなりません。ZFS はまた、アプリケーションが同期書き込みをリクエストするたびフラッシュを実行します (0_DSYNC、fsync、NFS コミットなど)。このタイプのフラッシュが終了するまでアプリケーションが待機するため、パフォーマンスに影響します。実際には、大きく影響します。パフォーマンスの観点からは、これにより NVRAM ベースのストレージを使用するメリットが相殺されます。

キャッシュフラッシュのチューニングは、ログデバイスとして使用された場合のフラッシュデバイスのパフォーマンス向上に役立つことが最近示されました。ZFS に公開されているすべての LUN が NVRAM で保護されたストレージアレイのものであり、かつ保護されていない LUN が将来追加されないことが手順によって保証される場合は、`zfs_nocacheflush` を設定することによって、フラッシュリクエストを発行しないように ZFS をチューニングできます。ZFS に公開された一部の LUN が NVRAM によって保護されていない場合、このチューニングによって、データ損失、アプリケーションレベルの破壊、またはプールの破壊が生じる可能性があります。NVRAM 保護されている一部のストレージアレイでは、キャッシュフラッシュコマンドは無操作であるため、このような状況でチューニングしてもパフォーマンスに差はありません。

最新の OS 変更ではフラッシュリクエストの意味が限定され、ストレージデバイスが適切に保護されている場合は、それらのデバイスにこのリクエストを無視するよう指示されます。この変更には、ディスクドライバに対する修正と、NVRAM デバイスでこの更新内容をサポートするための修正が必要です。NVRAM デバイスがこの機能強化を認識しない場合は、次の手順を使用して、アレイにキャッシュ同期化コマンドを送信しないよう Solaris OS に指示します。この手順を使用する場合は、すべてのターゲット LUN が実際に NVRAM で保護されていることを確認してください。

フラッシュおよび NVRAM デバイスは、自分が不揮発性のデバイスであることや、キャッシュをフラッシュする必要がないことを OS に正しく通知しない場合があります。キャッシュのフラッシュは負荷の大きい操作です。場合によっては、不必要なフラッシュによってパフォーマンスが大幅に低下する場合があります。

次のチューニングエントリを適用する前に、次の `zfs_nocacheflush` 構文の制限を確認してください。

- 次のチューニング構文を `sd.conf` に含めることができますが、ベンダー/製品につき 1 つの `sd-config-list` エントリのみにする必要があります。
- 複数のデバイスエントリが必要な場合は、次の構文を使用すると、ベンダー ID と `sd` チューニング文字列の複数のペアを同じ行に指定できます。

```
#           "012345670123456789012345", "tuning    ",
sd-config-list="|-VID1-||-----PID1-----|", "param1:val1, param2:val2",
               "|-VIDN-||-----PIDN-----|", "param1:val1, param3:val3";
```

前の例で説明されているように、ベンダー ID (VID) 文字列が 8 文字にパディングされ、製品 ID (PID) 文字列が 16 文字にパディングされていることを確認してください。



注意 - デバイスによって、すべてのキャッシュ同期化コマンドが無視されます。自己責任において使用してください。

1. `format` ユーティリティを使用して、ストレージアレイの LUN で `inquiry` サブコマンドを実行します。例:

```
# format
.
.
.
Specify disk (enter its number): x
```

```
format> inquiry
Vendor:   ATA
Product:  Marvell
Revision: XXXX
format>
```

2. アーキテクチャーに基づいて、次のいずれかを選択します。

- すべてのデバイスについて、/kernel/drv/sd.conf ファイルを /etc/driver/drv/sd.conf ファイルにコピーします。
- F40 フラッシュデバイスの場合は、/kernel/drv/sd.conf に次のエントリを追加します。次のエントリで、「ATA」が 8 文字にパディングされ、「3E128-TS2-550B01」に 16 文字が含まれていることを確認してください。文字列の全体の長さは 24 です。

```
sd-config-list="ATA 3E128-TS2-550B01", "disksort:false, cache-nonvolatile:true,
physical-block-size:4096";
```

- F80 フラッシュデバイスの場合は、/kernel/drv/sd.conf に次のエントリを追加します。ATA が 8 文字にパディングされ、3E128-TS2-550B01 に 16 文字が含まれていることを確認してください。文字列の全体の長さは 24 です。

```
sd-config-list="ATA 2E256-TU2-510B00", "disksort:false, cache-nonvolatile:true,
physical-block-size:4096";
```

- F20 および F5100 フラッシュデバイスの場合は、アーキテクチャーに基づいて、次のいずれかを選択します。次のエントリでは、ATA が 8 文字にパディングされ、MARVELL SD88SA02 に 16 文字が含まれています。文字列の全体の長さは 24 です。
- 次のエントリを /etc/driver/drv/sd.conf に追加します。

```
sd-config-list="ATA MARVELL SD88SA02", "throttle-max:32, disksort:false, cache-
nonvolatile:true";
```

3. 示されているように、sd-config-list エントリでベンダー ID (VID) (ここでは ATA) の長さが 8 文字になり、製品 ID (PID) (ここでは MARVELL) の長さが 16 文字になるように、注意深く空白を追加します。

4. システムをリブートします。

パフォーマンスに悪影響を与えることなく、zfs_nocacheflush を元のデフォルト値 (0) にチューニングできます。

5. フラッシュの動作が正しいことを確認します。

確認には、[付録A システムチェックスクリプト](#)に記述されているスクリプトを使用します。

データベース製品に対する ZFS のチューニング

ZFS をデータベース製品と一緒に使用するとき、次の考慮事項を検討してください:

- データベースで入出力に固定ディスクブロックまたは固定レコードサイズを使用している場合、ZFS `recordsize` プロパティをこれに一致するように設定してください。複数のファイルシステムが単一プールを共有している場合であっても、ファイルシステム単位で実行できません。
- ZFS の `copy-on-write` 設計により、`recordsize` を低くチューニングすることは、バッチレポートクエリを犠牲にして OLTP パフォーマンスを改善する方法です。
- ZFS は、ディスク上に格納されているすべてのブロックにチェックサムを実行します。これにより、データベース層でデータのチェックサムを追加して行う必要性が軽減されます。データベース層の代わりに ZFS によってチェックサムを計算する場合は、データがアプリケーションに戻される前に矛盾を捕捉して修正することができます。
- UFS の設計の一部の不備を克服し、データのダブルバッファリングを除去するために、UFS 直接入出力が使用されます。ZFS では、UFS の設計の不備は存在せず、ZFS は `primarycache` および `secondarycache` プロパティを使用して ARC 内のバッファリングデータを管理します。`secondarycache` (L2ARC) プロパティを使用してランダム読み取りを改善するためには、`primarycache` プロパティも有効にする必要があることに注意してください。
- プールのパフォーマンスを維持するには、プール領域の使用率を 90% 以下に維持してください。

Oracle データベース用の ZFS のチューニング

ZFS は単一インスタンスモードのすべての Oracle データベースバージョンについて推奨されます。ZFS は、Oracle RAC データベースが NFS 共有ファイルシステムとして使用可能な場合、これと一緒に使用できます。

ZFS を Oracle データベース用にチューニングすることについての次の推奨事項を検討してください:

- **最新の Oracle Solaris リリースを実行中であることを確認します**
Oracle Solaris 10 9/10 リリースを最低限の開始点として、最新の Oracle Solaris 10 または Oracle Solaris 11 リリースを起動します。
- **必要な場合、ZFS ストレージプール用の LUN を作成します**

使用しているストレージレイ用ツールを使用して、ZFS ストレージプールに提供される LUN を作成します。または、ミラー化された ZFS ストレージプール用にディスク全体を使用することを検討します。詳細は、『Oracle Solaris 11.2 での ZFS ファイルシステムの管理』の第 3 章「Oracle Solaris ZFS ストレージプールの管理」を参照してください。

- 表、索引、取り消し、および一時データ用のデータファイルのストレージプールを作成します
より高いレベルのデータ冗長性を提供するために、ミラー化されたストレージプールの作成を検討してください。例:

```
# zpool status dbpool
```

```
pool: dbpool
```

```
state: ONLINE
```

```
scan: none requested
```

```
config:
```

NAME	STATE	READ	WRITE	CKSUM
dbpool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c0t5000C500335F95E3d0	ONLINE	0	0	0
c0t5000C500335F907Fd0	ONLINE	0	0	0
mirror-1	ONLINE	0	0	0
c0t5000C500335BD117d0	ONLINE	0	0	0
c0t5000C500335DC60Fd0	ONLINE	0	0	0

```
errors: No known data errors
```

多くのコミットを伴う一般的な OLTP データベースなど、再実行ログアクティビティが多いデータベースの場合、別個のログデバイス用に別個の LUN を使用します。

- **archivelog** 用のストレージプールを作成します

使用可能な場合は、システムの内部ディスクがこのタイプの負荷を処理できません。archivelog ファイルシステムは dbpool 内のファイルシステムとすることもできます。

```
# zpool create archivepool c0t5000C500335E106Bd0
```

- **ZFS** ファイルシステムを作成し、次のガイドラインを使用して特定のファイルシステムプロパティを設定します

次のレコードサイズを使用して、データベースコンポーネントを redo、archive、undo、および temp するための別のファイルシステムを作成します。

- Oracle Solaris 11 およびそれ以前のリリース - 128K
- Oracle Solaris 11.1 およびそれ以降のリリース - 1M

一般的な規則は、Oracle データファイルを含むファイルシステムについて、ファイルシステムの `recordsize = db_block_size` を設定します。表データおよび索引コンポーネントについては、8K バイトのレコードサイズを持つファイルシステムを作成します。また、`primarycache` プロパティを使用して、データベースファイルシステムに対するメタデータキャッシュのヒントを提供することも検討してください。ZFS ファイルシステムプロパティの詳細については、『Oracle Solaris 11.2 での ZFS ファイルシステムの管理』の「ZFS のプロパティの概要」を参照してください。

- 8K バイトの `recordsize` を使用して、表データファイルおよび索引データファイル用のファイルシステムを作成します。`primarycache` のデフォルト値を使用します。

```
# zfs create -o recordsize=8k -o mountpoint=/my_db_path/index dbpool/index
# zfs set logbias=throughput dbpool/index
# zfs get primarycache,recordsize,logbias dbpool/index
```

NAME	PROPERTY	VALUE	SOURCE
dbpool/index	primarycache	all	default
dbpool/index	recordsize	8K	local
dbpool/index	logbias	throughput	local

- 一時表スペースおよび undo 表スペース用のファイルシステムを作成します。

Oracle Solaris 11 以前のリリースでは、`recordsize` および `primarycache` のデフォルト値を使用します。

```
# zfs create -o mountpoint=/my_db_path/temp dbpool/temp
# zfs set logbias=throughput dbpool/temp
# zfs create -o mountpoint=/my_db_path/undo dbpool/undo
# zfs set logbias=throughput dbpool/undo
```

Oracle Solaris 11.1 以降のリリースでは、次の `recordsize` およびデフォルトの `primarycache` 値を使用します。

```
# zfs create -o recordsize=1m -o mountpoint=/my_db_path/temp dbpool/temp
# zfs set logbias=throughput dbpool/temp
# zfs create -o recordsize=1m -o mountpoint=/my_db_path/undo dbpool/undo
# zfs set logbias=throughput dbpool/undo
```

- 別個のログデバイスを使用して、再実行ログ用のストレージプールを作成します。多くのコミットを伴う一般的な OLTP データベースなど、再実行ログアクティビティが多いデータベースの場合、別個のログデバイス LUN を使用します。

ディスクを 2 つのスライスにパーティション分割し、64M バイト から 150M バイトの範囲の小さいスライス `s0` を、別個のログデバイス用にします。`s1` スライスには、再実行ログ用の残りのディスク容量が入ります。

```
# zpool create redopool c0t50015179594B6F11d0s1 log c0t50015179594B6F11d0s0
```

```
# zpool status redopool
```

```
pool: redopool
state: ONLINE
scan: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
redopool	ONLINE	0	0	0
c0t50015179594B6F11d0s1	ONLINE	0	0	0
logs				
c0t50015179594B6F11d0s0	ONLINE	0	0	0

```
errors: No known data errors
```

- 再実行プール内に再実行ログ用のファイルシステムを作成します。

Oracle Solaris 11 以前のリリースでは、`recordsize` および `primarycache` のデフォルトのファイルシステム値を使用します。

```
# zfs create -o mountpoint=/my_db_path/redo redopool/redo
```

```
# zfs set logbias=latency redopool/redo
```

Solaris 11.1 以降のリリースでは、`recordsize` およびデフォルトの `primarycache` 値を使用します。

```
# zfs create -o recordsize=1m -o mountpoint=/my_db_path/redo redopool/redo
```

```
# zfs set logbias=latency redopool/redo
```

- アーカイブプール内にアーカイブログファイル用のファイルシステムを作成します。

Oracle Solaris 11 以前のリリースでは、`recordsize` のデフォルト値を使用して圧縮を有効化し、`primarycache` を `metadata` に設定します。

```
# zfs create -o compression=on -o primarycache=metadata -o mountpoint=
/my_db_admin_path/archive archivepool/archive
# zfs get primarycache,recordsize,compressratio,compression,available,
used,quota archivepool/archive
```

NAME	PROPERTY	VALUE	SOURCE
archivepool/archive	primarycache	metadata	local
archivepool/archive	recordsize	128K	default
archivepool/archive	compressratio	1.32x	-
archivepool/archive	compression	on	local
archivepool/archive	available	40.0G	-
archivepool/archive	used	10.0G	-
archivepool/archive	quota	50G	local

Solaris 11.1 以降のリリースでは、圧縮を有効化し、primarycache を metadata に設定して、次の recordsize 値を使用します。

```
# zfs create -o compression=on -o recordsize=1M \
-o mountpoint=/my_db_admin_path/archive archivepool/archive
# zfs get primarycache,recordsize,compressratio,compression,\
available,used,quota archivepool/archive
```

NAME	PROPERTY	VALUE	SOURCE
archivepool/archive	primarycache	all	local
archivepool/archive	recordsize	1M	local
archivepool/archive	compressratio	1.32x	-
archivepool/archive	compression	on	local
archivepool/archive	available	40.0G	-
archivepool/archive	used	10.0G	-
archivepool/archive	quota	50G	local

- データベースファイルシステムが稼働するのに十分なディスク領域が確保されるように割り当て量を設定し、データベースファイルシステムのスナップショットをとることを検討します。また、ダミーファイルシステム上に予約領域を設定して、プールのパフォーマンスを維持するため 10 - 20% のプールスペースを予約します。

```
# zfs set reservation=20gb dbpool/freespace
```

- ストレージアレイとメモリーリソースのチューニングに関する追加の情報については、<http://www.oracle.com/technetwork/server-storage/solaris/config-solaris-zfs-wp-167894.pdf> にあるホワイトペーパーを参照してください。

- 追加の Oracle データベース構成の推奨事項
 - 次のホワイトペーパーの『*Configuring Your Oracle Database on ZFS File Systems*』:
<http://www.oracle.com/technetwork/server-storage/solaris/config-solaris-zfs-wp-167894.pdf>
 - 『*Dynamic SGA Tuning of Oracle Database on Oracle Solaris with DISM*』ホワイトペーパー:
<http://www.oracle.com/technetwork/articles/systems-hardware-architecture/using-dynamic-intimate-memory-sparc-168402.pdf>
- Oracle 11g インストールガイド
 - Oracle Database クイックインストールガイド 11g リリース 2 (11.2) for Oracle Solaris on SPARC (64-Bit)
http://docs.oracle.com/cd/E11882_01/install.112/e24349/toc.htm
 - Oracle Database クイックインストールガイド 11g リリース 2 (11.2) for Oracle Solaris on x86-64 (64-Bit)
http://docs.oracle.com/cd/E11882_01/install.112/e24351/toc.htm

ZFS を MySQL と一緒に使用するときの考慮事項

ZFS を MySQL と一緒に使用するときには、次の考慮事項を確認してください:

- **ZFS recordsize**

OLTP パフォーマンスを高めるには、ZFS recordsize プロパティをストレージエンジンのブロックサイズに一致させます。
- **InnoDB**

データベースアプリケーションなどの既知のアプリケーションのメモリーフットプリントでは、アプリケーションが ZFS キャッシュから必要なメモリーを繰り返し要求する必要がないように、ARC サイズに上限を設定してもかまいません。
- ログ用の別個のプールを作成します。
- my.cnf ファイル内にデータおよびログ用の別のパスを設定します。
- データファイルを作成する前に、InnoDB データファイルに対して ZFS recordsize プロパティを 16K に設定し、InnoDB ログについてはデフォルトの recordsize 値を使用します。

◆◆◆ 第 4 章

NFS チューニング可能パラメータ

このセクションでは、NFS のチューニング可能パラメータについて説明します。

- [101 ページの「NFS 環境のチューニング」](#)
- [102 ページの「NFS モジュールのパラメータ」](#)
- [129 ページの「NFS 関連の SMF 構成パラメータ」](#)
- [129 ページの「rpcmod モジュールパラメータ」](#)

ほかのタイプのチューニング可能パラメータについては、次を参照してください。

- Oracle Solaris カーネルチューニング可能パラメータ – [第2章「Oracle Solaris カーネルチューニング可能パラメータ」](#)
- Oracle Solaris ZFS チューニング可能パラメータ – [第3章「Oracle Solaris ZFS チューニング可能パラメータ」](#)
- インターネットプロトコルスイートのチューニング可能パラメータ – [第5章「インターネットプロトコル群のチューニング可能パラメータ」](#)
- システム機能のチューニング可能パラメータ – [第6章「システム機能のパラメータ」](#)

NFS 環境のチューニング

NFS パラメータは、ブートプロセス中に読み込まれる `/etc/system` ファイルに設定できます。各パラメータには、対応するカーネルモジュールの名前を含めます。詳細は、[13 ページの「Oracle Solaris システムのチューニング」](#)を参照してください。



注意 - パラメータ名や、それが存在するモジュール、デフォルト値は、リリースによって変わることがあります。変更を行なったり、前のリリースの値を適用したりする前に、使用する SunOS リリースのバージョンのドキュメントをチェックしてください。

NFS モジュールのパラメータ

このセクションでは、NFS カーネルモジュールに関連するパラメータについて説明します。

nfs:nfs3_pathconf_disable_cache

説明	NFS バージョン 3 でマウントされたファイルシステムの pathconf 情報をキャッシングするかどうかを制御します。
データ型	整数 (32 ビット)
デフォルト	0 (キャッシングを有効にする)
範囲	0 (キャッシングを有効にする) または 1 (キャッシングを無効にする)
単位	ブール値
動的か	はい
検証	なし
どのような場合に 変更するか	pathconf 情報はファイルごとにキャッシュされます。しかし、サーバーが特定のファイルの情報を動的に変更することがある場合は、このパラメータを使用してキャッシングを無効にします。クライアントがキャッシュエントリを検証する方法はありません。
コミットレベル	変更の可能性あり

nfs:nfs_allow_preepoch_time

説明	<p>タイムスタンプが正しくなかったり「負」であるファイルをクライアントから表示できるようにするかどうかを制御します。</p> <p>従来、NFS クライアントも NFS サーバーも、返されるファイルの時間範囲を確認していませんでした。伝送されるタイムスタンプ値は符号なしの 32 ビット long です。したがって、あらゆる値が有効でした。</p> <p>64 ビットの Solaris カーネルでは、タイムスタンプ値は符号付きの 64 ビット long です。時間フィールドがフルの 32 ビットの時間を表しているのか、時間フィールドが負の時間、つまり、1970 年 1 月 1 日より前を表しているのかを判別できません。</p> <p>32 ビットから 64 ビットに変換するときに、時間値に符号を付けるかどうかを決定することはできません。時間値が本当に負数の場合は、値に</p>
----	---

符号を付けるべきです。しかし、時間値がフルの 32 ビット時間値を本当に表している場合は、時間値に符号を付けるべきではありません。この問題は、フルの 32 ビット時間値を無効にすることによって解決できます。

データ型	整数 (32 ビット)
デフォルト	0 (32 ビットのタイムスタンプを無効にする)
範囲	0 (32 ビットのタイムスタンプを無効にする) または 1 (32 ビットのタイムスタンプを有効にする)
単位	ブール値
動的か	はい
検証	なし
どのような場合に 変更するか	正常な操作が行われていても、ファイルによっては、タイムスタンプ値がはるかに離れた将来や過去の日付に設定されることがあります。NFS でマウントされたファイルシステムを使用してこれらのファイルにアクセスすることが望ましい場合は、このパラメータを 1 にすれば、タイムスタンプ値をチェックなしで受け取ることができます。
コミットレベル	変更の可能性あり

nfs:nfs_cots_timeo

説明	トランスポートプロトコルとして TCP などの接続型トランスポートを使用している、NFS バージョン 2 でマウントされたファイルシステムのデフォルトの RPC タイムアウトを制御します。
データ型	符号付き整数 (32 ビット)
デフォルト	600 (60 秒)
範囲	0 から $2^{31} - 1$
単位	1/10 秒
動的か	はい。ただし、ファイルシステムの RPC タイムアウトは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	なし

どのような場合に 変更するか	TCP では、要求と応答が適切に転送されるよう機能します。しかし、特に遅いネットワークにおいて往復時間が非常に長くなると、NFS バージョン 2 のクライアントがタイムアウトになってしまう可能性があります。 クライアントが無用にタイムアウトしてしまうことを防ぐには、このパラメータを増やします。ただし、値の範囲が非常に広いので、この値を大きくしすぎると、再転送が長い間検出されないおそれがあります。
コミットレベル	変更の可能性あり

nfs:nfs3_cots_timeo

説明	トランスポートプロトコルとして TCP などの接続型トランスポートを使用している、NFS バージョン 3 でマウントされたファイルシステムのデフォルトの RPC タイムアウトを制御します。
データ型	符号付き整数 (32 ビット)
デフォルト	600 (60 秒)
範囲	0 から $2^{31} - 1$
単位	1/10 秒
動的か	はい。ただし、ファイルシステムの RPC タイムアウトは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	なし
どのような場合に 変更するか	TCP では、要求と応答が適切に転送されるよう機能します。しかし、特に遅いネットワークにおいて往復時間が非常に長くなると、NFS バージョン 3 のクライアントがタイムアウトになってしまう可能性があります。 クライアントが無用にタイムアウトしてしまうことを防ぐには、このパラメータを増やします。ただし、値の範囲が非常に広いので、この値を大きくしすぎると、再転送が長い間検出されないおそれがあります。
コミットレベル	変更の可能性あり

nfs:nfs4_cots_timeo

説明	トランスポートプロトコルとして TCP などの接続型トランスポートを使用している、NFS バージョン 4 でマウントされたファイルシステムのデフォルトの RPC タイムアウトを制御します。
----	--

NFS バージョン 4 のプロトコル仕様では、同一 TCP 接続での再転送は認められません。したがって、このパラメータでは主に強制的アンマウント操作の検出、サーバーがどの程度迅速に新しいサーバーにフェイルオーバーしたかという検出など、クライアントが特定のイベントにどの程度迅速に応答するかを制御します。

データ型	符号付き整数 (32 ビット)
デフォルト	600 (60 秒)
範囲	0 から $2^{31} - 1$
単位	1/10 秒
動的か	はい。ただし、このパラメータは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	なし
どのような場合に 変更するか	TCP では、要求と応答が適切に転送されるよう機能します。しかし、特に遅いネットワークにおいて往復時間が非常に長くなると、NFS バージョン 4 のクライアントがタイムアウトになってしまう可能性があります。 クライアントが無用にタイムアウトしてしまうことを防ぐには、このパラメータを増やします。ただし、値の範囲が非常に広いので、この値を大きくしすぎると、再転送が長い間検出されないおそれがあります。
コミットレベル	変更の可能性あり

nfs:nfs_do_symlink_cache

説明	NFS バージョン 2 でマウントされたファイルシステムに対して、シンボリックリンクファイルの内容をキャッシュするかどうかを制御します。
データ型	整数 (32 ビット)
デフォルト	1 (キャッシングを有効にする)
範囲	0 (キャッシングを無効にする) または 1 (キャッシングを有効にする)
単位	ブール値
動的か	はい
検証	なし

どのような場合に 変更するか	サーバーが、シンボリックリンクファイルの内容を変更してもそのファイルの更新タイムスタンプを更新しない場合や、タイムスタンプの精度が粗すぎる場合には、シンボリックリンクファイルの内容が変更されても、クライアントにはその変更が長い間見られないことがあります。その場合、このパラメータを使用して、シンボリックリンク内容のキャッシングを無効にします。そうすることによって、クライアント上で動作しているアプリケーションに変更がただちに認識されるようにします。
コミットレベル	変更の可能性あり

nfs:nfs3_do_symlink_cache

説明	NFS バージョン 3 でマウントされたファイルシステムに対して、シンボリックリンクファイルの内容をキャッシュするかどうかを制御します。
データ型	整数 (32 ビット)
デフォルト	1 (キャッシングを有効にする)
範囲	0 (キャッシングを無効にする) または 1 (キャッシングを有効にする)
単位	ブール値
動的か	はい
検証	なし
どのような場合に 変更するか	サーバーが、シンボリックリンクファイルの内容を変更してもそのファイルの更新タイムスタンプを更新しない場合や、タイムスタンプの精度が粗すぎる場合には、シンボリックリンクファイルの内容が変更されても、クライアントにはその変更が長い間見られないことがあります。その場合、このパラメータを使用して、シンボリックリンク内容のキャッシングを無効にします。そうすることによって、クライアント上で動作しているアプリケーションに変更がただちに認識されるようにします。
コミットレベル	変更の可能性あり

nfs:nfs_dynamic

説明	UDP などのコネクションレストランスポートを使用する、NFS バージョン 2 でマウントされたファイルシステムに対して、「動的再転送」と呼ばれる機能を有効にするかどうかを制御します。この機能は、サーバーの応答時
----	--

	間をモニタリングし、RPC タイムアウトと読み取り / 書き込みの転送サイズを調整することによって、再転送の回数を減らそうとするものです。
データ型	整数 (32 ビット)
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	なし
どのような場合に 変更するか	このパラメータは変更しないでください。
コミットレベル	変更の可能性あり

nfs:nfs3_dynamic

説明	UDP などのコネクションレストランスポートを使用する、NFS バージョン 3 でマウントされたファイルシステムに対して、「動的再転送」と呼ばれる機能を有効にするかどうかを制御します。この機能は、サーバーの応答時間をモニタリングし、RPC タイムアウトと読み取り / 書き込みの転送サイズを調整することによって、再転送の回数を減らそうとするものです。
データ型	整数 (32 ビット)
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
単位	ブール値
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	なし
どのような場合に 変更するか	このパラメータは変更しないでください。
コミットレベル	変更の可能性あり

nfs:nfs_lookup_neg_cache

説明	NFS バージョン 2 でマウントされたファイルシステムに対して、ネガティブ名前キャッシュを使用するかどうかを制御します。このネガティブ名前キャッシュを使用すると、ルックアップされたファイル名が存在しない場合には、そのファイル名が記録されます。このキャッシュは、存在しないことがすでにわかっているファイル名のルックアップ要求が、ネットワークを介して行われないようにする目的で使用されます。
データ型	整数 (32 ビット)
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
単位	ブール値
動的か	はい
検証	なし
どのような場合に 変更するか	<p>このキャッシュが正しく機能するためには、ネガティブエントリが使用前に厳密に検証されなければなりません。整合性メカニズムは、読み取り専用でマウントされたファイルシステムに関しては多少緩和されています。これは、サーバー上のファイルシステムが変更されないか、変更されるにしてもそのペースが緩やかであり、クライアントへの伝達がゆっくりであつても支障はないと想定されているからです。この場合は、この整合性メカニズムが通常の属性キャッシュメカニズムになります。</p> <p>読み取り専用でクライアントにマウントされているファイルシステムがサーバーで変更されたら、その変更をクライアントでただちに必要める場合は、このパラメータを使用してネガティブキャッシュを無効にします。</p> <p><code>nfs:nfs_disable_rmdir_cache</code> パラメータを無効にする場合は、通常、このパラメータも無効にします。詳細は、118 ページの「<code>nfs:nfs_disable_rmdir_cache</code>」を参照してください。</p>
コミットレベル	変更の可能性あり

nfs:nfs3_lookup_neg_cache

説明	NFS バージョン 3 の読み取り専用でマウントされたファイルシステムに対して、ネガティブ名前キャッシュを使用するかどうかを制御します。この
----	--

	ネガティブ名前キャッシュを使用すると、ルックアップされたファイル名が存在しない場合には、そのファイル名が記録されます。このキャッシュは、存在しないことがすでにわかっているファイル名のルックアップ要求が、ネットワークを介して行われないようにする目的で使用されます。
データ型	整数 (32 ビット)
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
単位	ブール値
動的か	はい
検証	なし
どのような場合に 変更するか	<p>このキャッシュが正しく機能するためには、ネガティブエントリが使用前に厳密に検証されなければなりません。整合性メカニズムは、読み取り専用でマウントされたファイルシステムに関しては多少緩和されています。これは、サーバー上のファイルシステムが変更されないか、変更されるにしてもそのペースが緩やかであり、クライアントへの伝達がゆっくりであっても支障はないと想定されているからです。この場合は、この整合性メカニズムが通常の属性キャッシュメカニズムになります。</p> <p>ネガティブキャッシュエントリは、読み取り専用でマウントされたファイルシステムに対してのみ使用されます。サーバー上のファイルシステムが変更されないか、変更が非常に遅いと想定されているため、変更がクライアントに伝達されるのが遅くても問題ありません。この場合は、この整合性メカニズムが通常の属性キャッシュメカニズムになります。</p> <p>読み取り専用でクライアントにマウントされているファイルシステムがサーバーで変更されたら、その変更をクライアントでただちに見る必要がある場合は、このパラメータを使用してネガティブキャッシュを無効にします。</p> <p><code>nfs:nfs_disable_rddir_cache</code> パラメータを無効にする場合は、通常、このパラメータも無効にします。詳細は、118 ページの「<code>nfs:nfs_disable_rddir_cache</code>」を参照してください。</p>
コミットレベル	変更の可能性あり

nfs:nfs4_lookup_neg_cache

説明	NFS バージョン 4 でマウントされたファイルシステムに対して、ネガティブ名前キャッシュを使用するかどうかを制御します。このネガティブ名前
----	--

キャッシュを使用すると、ルックアップされたファイル名が存在しない場合には、そのファイル名が記録されます。このキャッシュは、存在しないことがすでにわかっているファイル名のルックアップ要求が、ネットワークを介して行われないようにする目的で使用されます。

データ型	整数 (32 ビット)
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
単位	ブール値
動的か	はい
検証	なし

どのような場合に
変更するか

このキャッシュが正しく機能するためには、ネガティブエントリが使用前に厳密に検証されなければなりません。整合性メカニズムは、読み取り専用でマウントされたファイルシステムに関しては多少緩和されています。これは、サーバー上のファイルシステムが変更されないか、変更されるにしてもそのペースが緩やかであり、クライアントへの伝達がゆっくりであつても支障はないと想定されているからです。この場合は、この整合性メカニズムが通常の属性キャッシュメカニズムになります。

読み取り専用でクライアントにマウントされているファイルシステムがサーバーで変更されたら、その変更をクライアントでただちに見る必要がある場合は、このパラメータを使用してネガティブキャッシュを無効にします。

`nfs:nfs_disable_rddir_cache` パラメータを無効にする場合は、通常、このパラメータも無効にします。詳細は、[118 ページ](#)の「`nfs:nfs_disable_rddir_cache`」を参照してください。

コミットレベル

変更の可能性あり

`nfs:nfs_max_threads`

説明

NFS バージョン 2 クライアントの非同期入出力を行うカーネルスレッドの数を制御します。NFS は RPC に基づくものであり、RPC はもともと同期して動作する機能であるため、呼び出し側のスレッドと非同期に NFS 操作を行うには、個別の実行コンテキストが必要です。

非同期に実行できる操作には、先読みのための読み取り、`readdir` 先読みのための `readdir`、`putpage` および `pageio` 操作のための書き込み、

	コミット、およびクライアントがファイルの使用を停止したときに実行するクリーンアップ操作のための非アクティブ化があります。
データ型	符号なし short
デフォルト	8
範囲	0 から $2^{15} - 1$
単位	スレッド
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	なし
どのような場合に 変更するか	ある時点で存在する同時入出力操作の数を増やしたり、減らしたりする場合。たとえば、帯域幅が非常に狭いネットワークでは、NFS クライアントによるネットワークの過負荷を防止するためにこの値を減らすことがあるかもしれません。また、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント / サーバリソースをいっそう有効に活用できるようになります。
コミットレベル	変更の可能性あり

nfs:nfs3_max_threads

説明	<p>NFS バージョン 3 クライアントの非同期入出力を行うカーネルスレッドの数を制御します。NFS は RPC に基づくものであり、RPC はもともと同期して動作する機能であるため、呼び出し側のスレッドと非同期に NFS 操作を行うには、個別の実行コンテキストが必要です。</p> <p>非同期に実行できる操作には、先読みのための読み取り、readdir 先読みのための readdir、putpage および pageio 要求のための書き込み、およびコミットがあります。</p>
データ型	符号なし short
デフォルト	8
範囲	0 から $2^{15} - 1$
単位	スレッド

動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	なし
どのような場合に 変更するか	ある時点で存在する同時入出力操作の数を増やしたり、減らしたりする場合。たとえば、帯域幅が非常に狭いネットワークでは、NFS クライアントによるネットワークの過負荷を防止するためにこの値を減らすことがあるかもしれません。また、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント / サーバーリソースをいっそう有効に活用できるようになります。
コミットレベル	変更の可能性あり

nfs:nfs4_max_threads

説明	<p>NFS バージョン 4 クライアントの非同期入出力を行うカーネルスレッドの数を制御します。NFS は RPC に基づくものであり、RPC はもともと同期して動作する機能であるため、呼び出し側のスレッドと非同期に NFS 操作を行うには、個別の実行コンテキストが必要です。</p> <p>非同期に実行できる操作には、先読みのための読み取り、後書き、ディレクトリの先読み、およびクライアントがファイルの使用を停止したときに実行するクリーンアップ操作があります。</p>
データ型	符号なし short
デフォルト	8
範囲	0 から $2^{15} - 1$
単位	スレッド
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	なし
どのような場合に 変更するか	ある時点で存在する同時入出力操作の数を増やしたり、減らしたりする場合。たとえば、帯域幅が非常に狭いネットワークでは、NFS クライアントによるネットワークの過負荷を防止するためにこの値を減らすことがあるかもしれません。また、広帯域幅のネットワークにおいて、クライアント

とサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント / サーバリソースをいっそう有効に活用できるようになります。

コミットレベル 変更の可能性あり

nfs:nfs_nra

説明	ファイルへの順次アクセスが検出されたときに NFS バージョン 2 クライアントによってキューに入れられる先読み操作の数を制御します。これらの先読み操作では、並行性が高まり、読み取りのスループットが向上します。個々の先読み要求は、通常、ファイルデータの 1 論理ブロックに対するものです。
データ型	整数 (32 ビット)
デフォルト	4
範囲	0 から $2^{31} - 1$
単位	論理ブロック
動的か	はい
検証	なし
どのような場合に 変更するか	特定のファイルに対してある時点で存在する先読み要求の数を増やしたり、減らしたりする場合。たとえば、ネットワークの帯域幅が非常に狭い場合やクライアントのメモリーが少ない場合は、NFS クライアントによるネットワークの過負荷やシステムメモリーの使いすぎを防止するために、この値を減らすことができるかもしれません。また、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント / サーバリソースをいっそう有効に活用できるようになります。
コミットレベル	変更の可能性あり

nfs:nfs3_nra

説明	ファイルへの順次アクセスが検出されたときに NFS バージョン 3 クライアントによってキューに入れられる先読み操作の数を制御します。これら
----	--

の先読み操作では、並行性が高まり、読み取りのスループットが向上します。個々の先読み要求は、通常、ファイルデータの 1 論理ブロックに対するものです。

データ型	整数 (32 ビット)
デフォルト	4
範囲	0 から $2^{31} - 1$
単位	論理ブロック(119 ページの「nfs:nfs3_bsize」を参照)
動的か	はい
検証	なし

どのような場合に
変更するか

特定のファイルに対してある時点で存在する先読み要求の数を増やしたり、減らしたりする場合。たとえば、ネットワークの帯域幅が非常に狭い場合やクライアントのメモリーが少ない場合は、NFS クライアントによるネットワークの過負荷やシステムメモリーの使いすぎを防止するために、この値を減らすことができるかもしれません。または、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント / サーバーリソースをいっそう有効に活用できるようになります。

コミットレベル 変更の可能性あり

nfs:nrnode

説明

NFS クライアントの `rnode` キャッシュのサイズを制御します。

NFS バージョン 2、3、および 4 のクライアントのいずれでも使用される `rnode` は、NFS クライアント上のファイルを記述する中心的なデータ構造体です。`rnode` には、サーバー上のファイルを識別するファイルハンドルが含まれています。`rnode` にはさらに、ネットワークからサーバーへの呼び出しを回避するために NFS クライアントが使用する、各種キャッシュへのポインタも含まれています。個々の `rnode` は `vnode` と 1 対 1 で対応しています。`vnode` には、ファイルデータがキャッシュされます。

NFS クライアントは、キャッシュされたデータやメタデータが破棄されないように、最少数の `rnode` を維持しようとします。`rnode` の再利用や解放が行われると、キャッシュされたデータやメタデータは破棄されなければなりません。

データ型 整数 (32 ビット)

デフォルト	このパラメータのデフォルト値は 0 で、 <code>nrnode</code> の値が <code>ncsize</code> パラメータの値に設定されるべきであることを示しています。実際、 <code>nrnode</code> の値が正でないと、 <code>nrnode</code> には <code>ncsize</code> が設定されます。
範囲	1 から $2^{31} - 1$
単位	<code>rnode</code>
動的か	いいえ。この値は、 <code>/etc/system</code> ファイルにパラメータを追加するか、パラメータを変更し、その後システムをリブートすることによってのみ変更できます。
検証	<code>rnode</code> キャッシュが使用可能なメモリーの 25% を超えないような最大値をシステムは強制します。
どのような場合に 変更するか	<code>rnode</code> の作成や破棄は動的に行われるため、システムは、システムのメモリーの要求や同時にアクセスされるファイルの数が増えるに従って、キャッシュのサイズを自動的に調整して、 <code>nrnode</code> サイズキャッシュを決定する傾向があります。しかし、アクセスするファイルの組み合わせが前もって予測できる場合など、状況によっては、 <code>nrnode</code> の値を設定することもあります。たとえば、NFS クライアントが少数の非常に大きいファイルにアクセスする場合、 <code>nrnode</code> を小さい値に設定すると、システムメモリーでは <code>rnode</code> の代わりにファイルデータをキャッシュできます。または、クライアントが多数の小さいファイルにアクセスする場合は、 <code>nrnode</code> の値を増やして、ファイルメタデータを格納できるように最適化すると、メタデータを要求するネットワーク呼び出しの数を減らすことができます。推奨はできませんが、 <code>nrnode</code> の値を 1 に設定すると、 <code>rnode</code> キャッシュを事実上無効にできます。この値は 1 <code>rnode</code> だけのキャッシュをクライアントに指示するので、結果的に頻繁に再利用されることになります。
コミットレベル	変更の可能性あり

`nfs:nfs_shrinkreaddir`

説明	<p>以前の一部の NFS サーバーでは、NFS バージョン 2 の <code>READDIR</code> 要求で 1024 バイトより大きいディレクトリ情報を求めるものが正しく処理されなかったことがありました。これは、サーバーの実装にバグがあったためです。このパラメータは、NFS バージョン 2 クライアントでの対処方法を含んでいます。</p> <p>このパラメータが有効化されると、クライアントは、1024 バイトよりも大きいディレクトリ情報を求める <code>READDIR</code> 要求を生成しなくなります。このパラメータを無効にすると、送信されるサイズは、<code>getdents</code> システム呼び出しを使用するか、または <code>NFS_MAXDATA</code> (8192 バイト) を使用して渡される</p>
----	--

	サイズのどちらか小さい方に設定されます。詳細は、 getdents(2) を参照してください。
データ型	整数 (32 ビット)
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
単位	ブール値
動的か	はい
検証	なし
どのような場合に 変更するか	NFS バージョン 2 専用のサーバーが使用され、ディレクトリの読み取りで相互運用性に問題がある場合は、このパラメータの値を調べてください。このパラメータを有効にすると、ディレクトリを読み取るアプリケーションのパフォーマンスが多少低下することがあります。
コミットレベル	変更の可能性あり

nfs:nfs3_shrinkreaddir

説明	<p>以前の一部の NFS サーバーでは、NFS バージョン 3 の READDIR 要求で 1024 バイトより大きいディレクトリ情報を求めるものが正しく処理されなかったことがありました。これは、サーバーの実装にバグがあったためです。このパラメータは、NFS バージョン 3 クライアントでの対処方法を含んでいます。</p> <p>このパラメータが有効化されると、クライアントは、1024 バイトよりも大きいディレクトリ情報を求める READDIR 要求を生成しなくなります。このパラメータを無効にすると、送信されるサイズは、getdents システム呼び出しを使用するか、または <code>MAXBSIZE</code> (8192 バイト) を使用して渡されるサイズのどちらか小さい方に設定されます。詳細は、getdents(2) を参照してください。</p>
データ型	整数 (32 ビット)
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
単位	ブール値
動的か	はい

検証	なし
どのような場合に 変更するか	NFS バージョン 3 専用のサーバーが使用され、ディレクトリの読み取りで相互運用性に問題がある場合は、このパラメータの値を調べてください。このパラメータを有効にすると、ディレクトリを読み取るアプリケーションのパフォーマンスが多少低下することがあります。
コミットレベル	変更の可能性あり

nfs:nfs_write_error_interval

説明	NFS クライアントが受け取った ENOSPC および EDQUOT 書き込みエラーのログ間隔を制御します。このパラメータは、NFS バージョン 2、3、および 4 のクライアントに影響を与えます。
データ型	long 整数 (64 ビット)
デフォルト	5 秒
範囲	0 から $2^{63} - 1$
単位	秒
動的か	はい
検証	なし
どのような場合に 変更するか	クライアントによってログされるメッセージ量に応じてこのパラメータの値を増減します。たとえば、サーバーのファイルシステムが満杯で頻繁に使用されているときに出力される「out of space」メッセージを減らす場合は、このパラメータの値を増やせるかもしれません。
コミットレベル	変更の可能性あり

nfs:nfs_write_error_to_cons_only

説明	NFS 書き込みエラーをシステムコンソールと syslog に記録するか、それともシステムコンソールだけに記録するかを制御します。このパラメータは、NFS バージョン 2、3、および 4 のクライアントのメッセージに影響を与えます。
データ型	整数 (32 ビット)

デフォルト	0 (システムコンソールと syslog)
範囲	0 (システムコンソールと syslog) または 1 (システムコンソール)
単位	ブール値
動的か	はい
検証	なし
どのような場合に 変更するか	syslogd デーモンによってロギングされるメッセージを含むファイルシステムがいっぱいになるのを防ぐには、このパラメータの値を調べます。このパラメータを有効にすると、メッセージはシステムコンソールに出力されるだけで、syslog メッセージファイルにはコピーされません。
コミットレベル	変更の可能性あり

nfs:nfs_disable_rddir_cache

説明	READDIR 要求と READDIRPLUS 要求に対する応答を格納するために、キャッシュを使用するかどうかを制御します。このキャッシュを使用すると、ディレクトリ情報を取得するためにサーバーを繰り返し呼び出すことがなくなります。
データ型	整数 (32 ビット)
デフォルト	0 (キャッシングを有効にする)
範囲	0 (キャッシングを有効にする) または 1 (キャッシングを無効にする)
単位	ブール値
動的か	はい
検証	なし
どのような場合に 変更するか	ファイルやディレクトリがサーバーに作成されたりサーバーから削除されてもサーバーがディレクトリの変更時間を更新しないために、相互運用性に問題がある場合は、このパラメータの値を調べます。ディレクトリにファイルを追加しても新しい名前が表示されなかったり、ディレクトリからファイルを削除しても古い名前が削除されない場合は、この問題があります。 このパラメータは、NFS バージョン 2、3、および 4 でマウントされたファイルシステムのキャッシングに適用されます。このパラメータは NFS でマウントされたすべてのファイルシステムに適用されるため、キャッシング

をファイルシステムごとに有効にしたり、無効にしたりすることはできません。

このパラメータを無効にする場合は、DNLC ネガティブキャッシュに不良エントリが発生しないように、次のパラメータも無効にするようにしてください。

- 108 ページの「`nfs:nfs_lookup_neg_cache`」
- 108 ページの「`nfs:nfs3_lookup_neg_cache`」
- 109 ページの「`nfs:nfs4_lookup_neg_cache`」

コミットレベル 変更の可能性あり

`nfs:nfs3_bsize`

説明	NFS バージョン 3 のクライアントが使用する論理ブロックサイズを制御します。このブロックサイズは、クライアントが入出力を行うときにサーバーに対して読み取りや書き込みを行うデータ量を表します。
データ型	符号なし整数 (32 ビット)
デフォルト	32,768 (32K バイト)
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、ファイルシステムのブロックサイズは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	ありません。このパラメータの設定が小さすぎたり、大きすぎたりすると、システムの動作が異常になることがあります。このパラメータには、各プラットフォームの <code>PAGESIZE</code> より小さい値を設定しないでください。さらに、このパラメータの値が大きすぎると、メモリー割り当てが許可されるまで待つ間に、システムがハングすることがあります。
どのような場合に 変更するか	データ転送サイズの最大値を変更したい場合は、このパラメータの値を調べてください。このパラメータは、 <code>nfs:nfs3_max_transfer_size</code> パラメータと連携して変更してください。転送サイズを増やしたい場合は両方のパラメータを増やします。転送サイズを減らしたい場合は、通常、このパラメータを減らすだけで十分です。
コミットレベル	変更の可能性あり

nfs:nfs4_bsize

説明	NFS バージョン 4 のクライアントが使用する論理ブロックサイズを制御します。このブロックサイズは、クライアントが入出力を行うときにサーバーに対して読み取りや書き込みを行うデータ量を表します。
データ型	符号なし整数 (32 ビット)
デフォルト	32,768 (32K バイト)
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、ファイルシステムのブロックサイズは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	ありません。このパラメータの設定が小さすぎたり、大きすぎたりすると、システムの動作が異常になることがあります。このパラメータには、各プラットフォームの PAGESIZE より小さい値を設定しないでください。さらに、このパラメータの値が大きすぎると、メモリー割り当てが許可されるまで待つ間に、システムがハングすることがあります。
どのような場合に 変更するか	データ転送サイズの最大値を変更したい場合は、このパラメータの値を調べてください。このパラメータは、 <code>nfs:nfs4_max_transfer_size</code> パラメータとセットで変更してください。転送サイズを増やしたい場合は両方のパラメータを増やします。転送サイズを減らしたい場合は、通常、このパラメータを減らすだけで十分です。
コミットレベル	変更の可能性あり

nfs:nfs_async_clusters

説明	<p>NFS バージョン 2 クライアントによって生成される非同期要求の組み合わせを制御します。非同期要求には、<code>read-ahead</code>、<code>putpage</code>、<code>pageio</code>、<code>readdir-ahead</code> という 4 つのタイプがあります。クライアントは、これらのタイプをラウンドロビンに処理し、特定の要求タイプだけを優先することがないようにします。</p> <p>しかし、書き込みの一括化 (書き込みをまとめる) などの NFS バージョン 2 サーバーの機能の中には、既存の NFS バージョン 2 クライアントの特定の動作に依存するものがあります。特に、この機能では、クライアントが複数の WRITE 要求をほぼ同時に送信することに依存します。</p>
----	--

キューから要求を 1 度に 1 つずつ取り出したのでは、クライアントのパフォーマンスを高めるために設けられたこのサーバー機能が生かされません。

そこで、このパラメータを使用して、タイプが変更されるまでに送信される、各要求タイプの要求数を制御します。

データ型	符号なし整数 (32 ビット)
デフォルト	1
範囲	0 から $2^{31} - 1$
単位	非同期要求
動的か	はい。ただし、ファイルシステムに対するクラスタ設定は、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	ありません。しかし、このパラメータに 0 を設定すると、キューに入れられている特定の要求タイプの要求がすべて処理されてから、次のタイプが処理されます。これによって、アルゴリズムの公平性の部分が実質的に無効にされます。
どのような場合に 変更するか	あるタイプについて次のタイプへ移行する前に生成される非同期要求の数を増やす場合。これによって、クライアントからの要求のクラスタに依存するサーバーの機能が生かされる場合があります。
コミットレベル	変更の可能性あり

nfs:nfs3_async_clusters

説明	<p>NFS バージョン 3 クライアントによって生成される非同期要求の組み合わせを制御します。非同期要求には、read-ahead、putpage、pageio、readdir-ahead、commit という 5 つのタイプがあります。クライアントは、これらのタイプをラウンドロビンに処理し、特定の要求タイプだけを優先することがないようにします。</p> <p>しかし、書き込みの一括化 (書き込みをまとめる) などの NFS バージョン 3 サーバーの機能の中には、既存の NFS バージョン 3 クライアントの一定の動作に依存するものがあります。特に、この機能では、クライアントが複数の WRITE 要求をほぼ同時に送信することに依存します。キューから要求を 1 度に 1 つずつ取り出したのでは、クライアントのパフォーマンスを高めるために設けられたこのサーバー機能が生かされません。</p>
----	--

そこで、このパラメータを使用して、タイプが変更されるまでに送信される、各要求タイプの要求数を制御します。

データ型	符号なし整数 (32 ビット)
デフォルト	1
範囲	0 から $2^{31} - 1$
単位	非同期要求
動的か	はい。ただし、ファイルシステムに対するクラスタ設定は、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	ありません。しかし、このパラメータに 0 を設定すると、キューに入れられている特定の要求タイプの要求がすべて処理されてから、次のタイプが処理されます。この値によって、アルゴリズムの公平性の部分が実質的に無効にされます。
どのような場合に 変更するか	あるタイプについて次のタイプへ移行する前に生成される非同期操作の数を増やす場合。これによって、クライアントからの操作のクラスタに依存するサーバーの機能が生かされる場合があります。
コミットレベル	変更の可能性あり

nfs:nfs4_async_clusters

説明	<p>NFS バージョン 4 クライアントによって生成される非同期要求の組み合わせを制御します。非同期要求には、read-ahead、putpage、pageio、readdir-ahead、commit、および inactive という 6 つのタイプがあります。クライアントは、これらのタイプをラウンドロビンに処理し、特定の要求タイプだけを優遇することがないようにします。</p> <p>しかし、書き込みの一括化 (書き込みをまとめる) などの NFS バージョン 4 サーバーの機能の中には、既存の NFS バージョン 4 クライアントの一定の動作に依存するものがあります。特に、この機能では、クライアントが複数の WRITE 要求をほぼ同時に送信することに依存します。キューから要求を 1 度に 1 つずつ取り出したのでは、クライアントのパフォーマンスを高めるために設けられたこのサーバー機能が生かされません。</p> <p>そこで、このパラメータを使用して、タイプが変更されるまでに送信される、各要求タイプの要求数を制御します。</p>
----	---

データ型	符号なし整数 (32 ビット)
デフォルト	1
範囲	0 から $2^{31} - 1$
単位	非同期要求
動的か	はい。ただし、ファイルシステムに対するクラスタ設定は、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	ありません。しかし、このパラメータに 0 を設定すると、キューに入れられている特定の要求タイプの要求がすべて処理されてから、次のタイプが処理されます。これによって、アルゴリズムの公平性の部分が実質的に無効にされます。
どのような場合に 変更するか	あるタイプについて次のタイプへ移行する前に生成される非同期要求の数を増やす場合。これによって、クライアントからの要求のクラスタに依存するサーバーの機能が生かされる場合があります。
コミットレベル	変更の可能性あり

nfs:nfs_async_timeout

説明	非同期入出力要求を実行するスレッドが終了するまで、なにもしない休眠状態を続けることのできる時間の長さを制御します。実行する要求がないと各スレッドは休眠状態に入ります。このタイマーが切れる前に新しい要求が到着しないと、スレッドは休眠から起きて終了します。要求が届くと、スレッドは起き上がって再び要求がなくなるまで要求を実行します。その後、スレッドは休眠状態に戻り、次の要求が届くか、またはタイマーが満了するまで待ちます。
データ型	整数 (32 ビット)
デフォルト	6000 (1 分を 60 秒 * 100Hz として表す)
範囲	0 から $2^{31} - 1$
単位	Hz(一般にクロックは 100Hz で動作する)
動的か	はい

検証	ありません。しかし、このパラメータに正以外の値を設定すると、スレッドが、自身が処理する要求がキューになくなるとすぐに終了します。
どのような場合に 変更するか	システムでのアプリケーションの動作を正確に把握し、非同期入出力要求の割合を予測できる場合は、次のどちらかの方法によってこのパラメータをチューニングすることで、パフォーマンスをある程度最適化することができます。 <ul style="list-style-type: none"> ■ スレッドの終了までの時間を短くして、カーネルリソースの解放を早くする。 ■ スレッドの終了までの時間を長くして、スレッドの作成や破棄にかかるオーバーヘッドを減らす。
コミットレベル	変更の可能性あり

nfs:nacache

説明	NFS クライアント上のファイルアクセスキャッシュにアクセスするハッシュキューの数を調整します。ファイルアクセスキャッシュは、ユーザーがアクセスしようとするファイルに関する、ユーザーの持つファイルアクセス権を格納します。キャッシュそのものは動的に割り当てられます。しかし、キャッシュに対するインデックスを作成するためのハッシュキューは、静的に割り当てられます。このアルゴリズムでは、アクティブファイルごとに 1 つのアクセスキャッシュエントリが、ハッシュバケットごとにこれらの 4 つのアクセスキャッシュエントリがあるものとみなします。したがって、このパラメータの値には、デフォルトで <code>nrnode</code> パラメータの値が設定されます。
データ型	整数 (32 ビット)
デフォルト	このパラメータのデフォルト値は 0 です。この値は <code>nacache</code> の値に <code>nrnode</code> パラメータの値が設定されるべきであることを示しています。
範囲	1 から $2^{31} - 1$
単位	アクセスキャッシュエントリ
動的か	いいえ。この値は、 <code>/etc/system</code> ファイルにパラメータを追加するか、パラメータを変更し、その後システムをリブートすることによってのみ変更できます。
検証	ありません。しかし、このパラメータに負の値を設定すると、システムは、おそらく、非常に多くのハッシュキューをシステムに割り当てようとし、その間におそらくハングします。

どのような場合に 変更するか	1つのファイルごとに1つのアクセスキャッシュエントリがあるという基本的な前提が損われるおそれがある場合は、このパラメータの値を検討します。複数のユーザーが同じファイルにほぼ同時にアクセスするタイムシェアリングモードのシステムでは、この前提が損なわれる可能性があります。このような場合には、予想されるアクセスキャッシュのサイズを増やすことが、キャッシュへのハッシュアクセスの効率性を保つ上で役立つことがあります。
コミットレベル	変更の可能性あり

nfs:nfs3_jukebox_delay

説明	NFS バージョン 3 クライアントが前回の要求で NFS3ERR_JUKEBOX エラーを受け取ってから、新しい要求を送信するまでに待機する時間の長さを制御します。NFS3ERR_JUKEBOX エラーは、通常、何らかの理由でファイルが一時的に使用できないときにサーバーから返されます。このエラーは、通常、階層型ストレージ、CD やテープといったジュークボックスに関連しています。
データ型	long 整数 (64 ビット)
デフォルト	1000 (10 秒を 10 秒 * 100Hz で表す)
範囲	64 ビットプラットフォームでは 0 から $2^{63} - 1$
単位	Hz(一般にクロックは 100Hz で動作する)
動的か	はい
検証	なし
どのような場合に 変更するか	このパラメータの値を調べ、必要ならサーバーが示す動作に合わせて値を調整します。再送信を繰り返すことによるネットワークオーバーヘッドを減らすためにファイルを使用できる遅延を長くする場合は、この値を増やします。ファイルが使用可能になったことを検出する場合の遅延を短くするには、この値を減らします。
コミットレベル	変更の可能性あり

nfs:nfs3_max_transfer_size

説明	NFS バージョン 3 の READ、WRITE、REaddir、または REaddirPLUS 要求のデータ部分の最大サイズを制御します。このパラメータは、サーバー
----	---

	が返す要求の最大サイズとクライアントが生成する要求の最大サイズの両方を制御します。
データ型	整数 (32 ビット)
デフォルト	1,048,576 (1M バイト)
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	<p>ありません。しかし、サーバー側の最大転送サイズに 0 を設定すると、クライアントはおそらく、誤作動するか、単にサーバーに要求を送信しないかのどちらかになる可能性があります。</p> <p>また、UDP 転送ポート経由の NFS を使用する場合にも、転送サイズの最大値に制限があります。UDP には、1 データグラム当たり 64K バイトという厳しい制限値があります。この 64K バイトには、要求のデータ部分のほかに、RPC ヘッダーやその他の NFS 情報も含まれている必要があります。この制限値が大きすぎると、UDP エラーのためにクライアントとサーバーの通信に問題が発生することがあります。</p>
どのような場合に 変更するか	<p>ネットワークを介して送信するデータのサイズをチューニングする場合。通常、<code>nfs:nfs3_bsize</code> パラメータもこのパラメータの変更が反映されるように更新すべきです。</p> <p>たとえば、転送サイズを 32K バイトより大きい値に増やす場合は、その増加した値が反映されるように <code>nfs:nfs3_bsize</code> を更新します。そうしないと、ネットワーク上で送信される要求のサイズは変わりません。詳細は、119 ページの「nfs:nfs3_bsize」を参照してください。</p> <p>転送サイズをデフォルト値より小さくする場合は、<code>mount</code> コマンドの <code>-wsize</code> または <code>-rsize</code> オプションをファイルシステム単位で使用します。</p>
コミットレベル	変更の可能性あり

nfs:nfs4_max_transfer_size

説明	NFS バージョン 4 の READ、WRITE、REaddir、または REaddirPLUS 要求のデータ部分の最大サイズを制御します。このパラメータは、サーバーが返す要求の最大サイズとクライアントが生成する要求の最大サイズの両方を制御します。
----	--

データ型	整数 (32 ビット)
デフォルト	32,768 (32 K バイト)
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	<p>ありません。しかし、サーバー側の最大転送サイズに 0 を設定すると、クライアントはおそらく、誤作動するか、単にサーバーに要求を送信しないかのどちらかになる可能性があります。</p> <p>また、UDP トランスポート経由の NFS を使用する場合にも、転送サイズの最大値に制限があります。UDP の最大値については、125 ページの「<code>nfs:nfs3_max_transfer_size</code>」を参照してください。</p>
どのような場合に 変更するか	<p>ネットワークを介して送信するデータのサイズをチューニングする場合。通常、<code>nfs:nfs4_bsize</code> パラメータもこのパラメータの変更が反映されるように更新すべきです。</p> <p>たとえば、転送サイズを 32K バイトより大きい値に増やす場合は、その増加した値が反映されるように <code>nfs:nfs4_bsize</code> を更新します。そうしないと、ネットワーク上で送信される要求のサイズは変わりません。詳細は、120 ページの「<code>nfs:nfs4_bsize</code>」を参照してください。</p> <p>転送サイズをデフォルト値より小さくする場合は、<code>mount</code> コマンドの <code>-wsize</code> または <code>-rsize</code> オプションをファイルシステム単位で使用します。</p>
コミットレベル	変更の可能性あり

`nfs:nfs3_max_transfer_size_clts`

説明	NFS バージョン 3 の UDP を介した <code>READ</code> 、 <code>WRITE</code> 、 <code>REaddir</code> 、または <code>REaddirPlus</code> 要求のデータ部分の最大サイズを制御します。このパラメータは、サーバーが返す要求の最大サイズとクライアントが生成する要求の最大サイズの両方を制御します。
データ型	整数 (32 ビット)
デフォルト	32,768 (32 K バイト)

範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	ありません。しかし、サーバー側の最大転送サイズに 0 を設定すると、クライアントはおそらく、誤作動するか、単にサーバーに要求を送信しないかのどちらかになる可能性があります。
どのような場合に 変更するか	このパラメータは変更しないでください。
コミットレベル	変更の可能性あり

nfs:nfs3_max_transfer_size_cots

説明	NFS バージョン 3 の TCP を介した READ、WRITE、REaddir、または REaddirPLUS 要求のデータ部分の最大サイズを制御します。このパラメータは、サーバーが返す要求の最大サイズとクライアントが生成する要求の最大サイズの両方を制御します。
データ型	整数 (32 ビット)
デフォルト	1,048,576 バイト
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	ありません。しかし、サーバー側の最大転送サイズに 0 を設定すると、クライアントはおそらく、誤作動するか、単にサーバーに要求を送信しないかのどちらかになる可能性があります。
どのような場合に 変更するか	1M バイトを超える転送サイズが必要な場合以外、このパラメータを変更しないでください。
コミットレベル	変更の可能性あり

NFS 関連の SMF 構成パラメータ

Oracle Solaris 11.2 では、`network/nfs/server` サービスに `nfs-props` プロパティグループが含まれており、これにより、NFS 認証キャッシュのリフレッシュの制御および `mountd` `netgroup` キャッシュの制御を行う構成可能なパラメータを提供しています。

- [129 ページの「server_authz_cache_refresh」](#)
- [129 ページの「netgroup_refresh」](#)

これらのプロパティの取得と設定には、`sharectl` コマンドを使用します。

```
# sharectl get -p server_authz_cache_refresh nfs
server_authz_cache_refresh=600
$ sharectl set -p server_authz_cache_refresh=1 nfs
```

SMF コマンドを使用してこれらのプロパティを取得および設定することもできますが、`network/nfs/server` サービスをリフレッシュする必要があります。

```
# svccfg -s nfs/server:default setprop nfs-props/server_authz_cache_refresh=1
# svcprop -p nfs-props/server_authz_cache_refresh svc:/network/nfs/server:default
1
# svcadm restart nfs/server:default
```

server_authz_cache_refresh

このパラメータは、NFS 認証キャッシュのリフレッシュを制御します。この整数プロパティのデフォルト値は 600、最小値は 0、最大値は `INT32_MAX` です。ゼロ (0) は、期限がないことを意味します。

netgroup_refresh

このパラメータは `mountd` ネットグループキャッシュを制御します。この整数プロパティのデフォルト値は 600、最小値は 0、最大値は `INT32_MAX` です。ゼロ (0) は、期限がないことを意味します。

rpcmod モジュールパラメータ

このセクションでは、`rpcmod` モジュールの NFS パラメータについて説明します。

rpcmod:clnt_max_conns

説明	個々の NFS サーバーと通信するときに、NFS クライアントが使用する TCP 接続の数を制御します。1 つの接続で RPC を多重化できるように、カーネル RPC が構築されます。しかし、必要な場合には複数の接続を使用できます。
データ型	整数 (32 ビット)
デフォルト	1
範囲	1 から $2^{31} - 1$
単位	接続
動的か	はい
検証	なし
どのような場合に 変更するか	一般には、1 つの接続だけでネットワーク帯域幅全体を使いこなすことができます。しかし、ネットワークが提供する帯域幅を TCP が 1 つのストリームだけで利用できない場合は、複数の接続を使えば、クライアントとサーバー間のスループットが向上することがあります。 接続数の増加にはそれなりの影響があります。接続数が増えると、各接続を維持するために必要なカーネルリソースの使用量も増えます。
コミットレベル	変更の可能性あり

rpcmod:clnt_idle_timeout

説明	クライアントとサーバー間の接続が終了するまでにアイドル状態を維持できる、クライアント側の時間の長さを制御します。
データ型	long 整数 (64 ビット)
デフォルト	300,000 ミリ秒 (5 分)
範囲	0 から $2^{63} - 1$
単位	ミリ秒
動的か	はい

検証	なし
どのような場合に 変更するか	クライアント側でどのくらいの間アイドル状態であれば接続を閉じるかを 変更する場合は、このパラメータを使用します。システムリソースが浪費され るのを防ぐために、接続を閉じるまでの時間を短縮する場合などです。
コミットレベル	変更の可能性あり

rpcmod:svc_idle_timeout

説明	クライアントとサーバー間の接続が終了するまでにアイドル状態を維持で きる、サーバー側の時間の長さを制御します。
データ型	long 整数 (64 ビット)
デフォルト	360,000 ミリ秒 (6 分)
範囲	0 から $2^{63} - 1$
単位	ミリ秒
動的か	はい
検証	なし
どのような場合に 変更するか	サーバー側でどのくらいの間アイドル状態であれば接続を閉じるかを 変更する場合は、このパラメータを使用します。システムリソースが浪費され るのを防ぐために、接続を閉じるまでの時間を短縮する場合などです。
コミットレベル	変更の可能性あり

rpcmod:svc_default_stksize

説明	カーネル RPC サービス スレッドに対するカーネルスタックのサイズを設 定します。
データ型	整数 (32 ビット)
デフォルト	デフォルト値は 0 です。この場合、スタックサイズはシステムデフォルト に設定されます。
範囲	0 から $2^{31} - 1$

単位	バイト
動的か	はい。新しく割り当てられるすべてのスレッドに適用されます。スタックサイズはスレッドの作成時に設定されます。したがって、このパラメータの変更は、既存のスレッドには適用されず、新しく割り当てられるすべてのスレッドに適用されます。
検証	なし
どのような場合に 変更するか	呼び出し深度が非常に深いため、スタックがオーバーフローし、レッドゾーンの障害が発生するおそれがある場合。トランスポートに対する呼び出し深度が比較的深く、ローカルファイルシステムに対する呼び出しの深さが深いという組み合わせは、NFS サービススレッドのスタックがオーバーフローを起こすことがあります。 このパラメータには、プラットフォームのハードウェア <code>pagesize</code> の倍数を設定する必要があります。
コミットレベル	変更の可能性あり

rpcmod:maxdupreqs

説明	コネクションレストランスポートにおける RPC レベルの再転送を検出する、重複要求キャッシュのサイズを制御します。このキャッシュは、クライアントネットワークアドレス、RPC の手順番号、プログラム番号、バージョン番号、およびトランザクション ID でインデックス化されています。このキャッシュにより、非べき等であるかもしれない再転送要求の処理が防止されます。
データ型	整数 (32 ビット)
デフォルト	8192
範囲	1 から $2^{31} - 1$
単位	要求
動的か	キャッシュのサイズは動的に決められますが、キャッシュへの高速アクセスを可能にするハッシュキューのサイズは静的に決められます。キャッシュのサイズを著しく大きくすると、キャッシュ内のエントリの検索に長い時間がかかることがあります。 このパラメータの値を 0 に設定しないでください。0 に設定すると、NFS サーバーが非べき等の要求を処理できなくなります。
検証	なし

どのような場合に 変更するか	NFS クライアントで不正な障害エラーが検出された場合は、このパラメータの値を調べます。たとえば、ディレクトリの作成が失敗したのに、実際にはディレクトリが作成されている場合は、再転送された MKDIR 要求をサーバーが検出しなかった可能性があります。 キャッシュのサイズは、サーバーの負荷に見合ったものでなければなりません。キャッシュには非べき等の要求が格納されるため、キャッシュでは、要求全体の一部だけしか管理する必要がありません。キャッシュは、クライアントによる再転送を検出できるだけの間、情報を保持していません。一般に、コネクションレスタランスポートのクライアントのタイムアウトは比較的短く、1 秒から 20 秒くらいです。
コミットレベル	変更の可能性あり

rpcmod:cotsmaxdupreqs

説明	接続型トランスポートにおける RPC レベルの再転送を検出する、重複要求キャッシュのサイズを制御します。このキャッシュは、クライアントネットワークアドレス、RPC の手順番号、プログラム番号、バージョン番号、および、トランザクション ID でインデックス化されています。このキャッシュにより、非べき等であるかもしれない再転送要求の処理が防止されます。
データ型	整数 (32 ビット)
デフォルト	8192
範囲	1 から $2^{31} - 1$
単位	要求
動的か	はい
検証	キャッシュのサイズは動的に決められますが、キャッシュへの高速アクセスを可能にするハッシュキューのサイズは静的に決められます。キャッシュのサイズを著しく大きくすると、キャッシュ内のエントリの検索に長い時間がかかることがあります。 このパラメータの値を 0 に設定しないでください。0 に設定すると、NFS サーバーが非べき等の要求を処理できなくなります。
どのような場合に 変更するか	NFS クライアントで不正な障害エラーが検出された場合は、このパラメータの値を調べます。たとえば、ディレクトリの作成に失敗したにもかかわらず、実際にはディレクトリが作成されている場合は、再転送された MKDIR 要求をサーバーが検出しなかった可能性があります。 キャッシュのサイズは、サーバーの負荷に見合ったものでなければなりません。キャッシュには非べき等の要求が格納されるため、キャッシュで

は、要求全体の一部だけしか管理する必要がありません。キャッシュは、クライアント側の再転送を検出できるだけの間、情報を保持していなければなりません。一般に、コネクション型のトランスポートのクライアントのタイムアウトは非常に長く、1 分くらいです。したがって、エントリは、キャッシュに比較的長く留まる必要があります。

コミットレベル

変更の可能性あり

◆◆◆ 第 5 章

インターネットプロトコル群のチューニング可能パラメータ

この章では、さまざまなインターネットプロトコルスイートのプロパティを説明します。

- [136 ページの「IP チューニング可能パラメータ」](#)
- [147 ページの「TCP チューニング可能パラメータ」](#)
- [163 ページの「UDP チューニング可能パラメータ」](#)
- [165 ページの「IPQoS チューニング可能パラメータ」](#)
- [166 ページの「SCTP チューニング可能パラメータ」](#)
- [177 ページの「ルート別のメトリック」](#)

ほかのタイプのチューニング可能パラメータについては、次を参照してください。

- [Oracle Solaris カーネルチューニング可能パラメータ - 第2章「Oracle Solaris カーネルチューニング可能パラメータ」](#)
- [Oracle Solaris ZFS チューニング可能パラメータ - 第3章「Oracle Solaris ZFS チューニング可能パラメータ」](#)
- [NFS チューニング可能パラメータ - 第4章「NFS チューニング可能パラメータ」](#)
- [システム機能のチューニング可能パラメータ - 第6章「システム機能のパラメータ」](#)

IP パラメータのチューニングの概要

この章で説明するチューニングパラメータはすべて、`ipadm` コマンド構文を使って設定できます。

```
# ipadm set-prop -p parameter ip|ipv4|ipv6|tcp|udp|sctp
```

例:

```
# ipadm set-prop -p extra_priv_ports=1047 tcp
```

```
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY          PERM CURRENT    PERSISTENT  DEFAULT    POSSIBLE
tcp  extra_priv_ports    rw  1047          1047        2049,4045  1-65535
```

詳細は、[ipadm\(1M\)](#) を参照してください。

IP パラメータの妥当性検証

このセクションで紹介するすべてのパラメータを対象に、パラメータ範囲内であるかどうかのチェックが行われます。パラメータ範囲は、各パラメータの説明に記載されています。

RFC (Internet Request for Comments)

インターネットのプロトコルと標準の仕様は、RFC ドキュメントに記述されています。RFC は次のサイトから確認できます。

<https://www.ietf.org/rfc.html>

このサイトでは、IETF Repository Retrieval 検索フィールドに RFC 番号または Internet-Draft ファイル名を入力することによって、RFC トピックを参照できます。

IP チューニング可能パラメータ

`_icmp_err_interval` と `_icmp_err_burst`

説明 IP で ICMP エラーメッセージを生成する頻度を制御します。IP は、`_icmp_err_interval` の間に最大で `ip_icmp_err_burst` の IP エラーメッセージを生成します。

`_icmp_err_interval` パラメータは、サービス拒否攻撃から IP を保護するためのものです。パラメータの値を 0 に設定すると、レート制限が無効になります。エラーメッセージの生成処理は無効になりません。

デフォルト `_icmp_err_interval` は 100 ミリ秒
`_icmp_err_burst` は 10 エラーメッセージ

範囲 `_icmp_err_interval` は 0 から 99,999 ミリ秒

`_icmp_err_burst` は 1 から 99,999 のエラーメッセージ

動的か	はい
どのような場合に 変更するか	診断の目的でエラーメッセージの生成頻度を増やしたい場合
コミットレベル	変更の可能性あり

`_respond_to_echo_broadcast` と `_respond_to_echo_multicast (ipv4 または ipv6)`

説明	IP がブロードキャスト ICMPv4 エコー要求または IPv6 マルチキャスト ICMPv6 エコー要求に応答するかどうかを制御します。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に 変更するか	セキュリティー上の理由でこの動作を行いたくない場合、無効にします
コミットレベル	変更の可能性あり

`send_redirects (ipv4 or ipv6)`

説明	IPv4 または IPv6 が、ICMPv4 または ICMPv6 リダイレクトメッセージを送信するかどうかを制御します。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に 変更するか	セキュリティー上の理由でこの動作を行いたくない場合、無効にします
コミットレベル	変更の可能性あり

forwarding (ipv4 または ipv6)

説明	IPv4 または IPv6 が、パケットをソース IPv4 ルーティングオプションを指定して転送するか、IPv6 ルーティングヘッダーを指定して転送するかを制御します。
デフォルト	オフ
範囲	オフまたはオン
動的か	はい
どのような場合に 変更するか	サービス妨害攻撃を防ぐためにこのパラメータは無効のままにします。
コミットレベル	変更の可能性あり

ttl

説明	IP 接続上で、アウトバウンド IPv4 パケットの IPv4 ヘッダーの TTL 値を制御します。
デフォルト	255
範囲	1 から 255
動的か	はい
どのような場合に 変更するか	通常、この値を変更する必要はありません。
コミットレベル	変更の可能性あり

hoplimit (ipv6)

説明	IP 接続上で、アウトバウンド IPv6 パケットの IPv6 ヘッダーのホップ制限値を設定します。
デフォルト	255
範囲	1 から 255

動的か	はい
どのような場合に 変更するか	通常、この値を変更する必要はありません。
コミットレベル	変更の可能性あり

_addrs_per_if

説明	実インタフェースに対応する論理 IP インタフェースの最大数を指定します。
デフォルト	256
範囲	1 から 8,192
動的か	はい
どのような場合に 変更するか	この値は変更しないでください。論理インタフェースの数を増やす必要がある場合は、例外的に値を増やすことができるかもしれませんが、この変更が IP のパフォーマンスに悪影響を及ぼす可能性があります。
コミットレベル	変更の可能性あり

hostmodel (ipv4 または ipv6)

説明	マルチホームシステム上の IPv4 または IPv6 パケットの送受信動作を制御します。このプロパティの値には、 <code>weak</code> 、 <code>strong</code> 、および <code>src-priority</code> を指定できます。デフォルト値は <code>weak</code> です。
デフォルト	<code>weak</code>
範囲	<code>weak</code> 、 <code>strong</code> 、または <code>src-priority</code> <ul style="list-style-type: none"> ■ <code>weak</code> <ul style="list-style-type: none"> ■ 送信パケット - 送信パケットの発信元アドレスは、送信インタフェースに構成されているアドレスに一致する必要はありません。 ■ 着信パケット - 着信パケットの宛先アドレスは、着信インタフェースに構成されているアドレスに一致する必要はありません。 ■ <code>strong</code>

- 送信パケット - 送信パケットの発信元アドレスは、送信インタフェースに構成されているアドレスに一致する必要があります。
- 着信パケット - 着信パケットの宛先アドレスは、着信インタフェースに構成されているアドレスに一致する必要があります。
- src-priority
 - 送信パケット - パケットの IP 着信先への経路が複数ある場合は、パケットの IP 発信元アドレスが送信インタフェース上に構成されている経路が優先されます。
そのような経路がない場合、弱い ES の場合と同様に、フォールバックによって最適な経路が選択されます。
 - 着信パケット - 着信パケットの宛先アドレスが、ホストのいずれかのインタフェースに構成されている必要があります。

動的か	はい
どのような場合に 変更するか	厳密なネットワーキングドメイン (たとえばファイアウォールや VPN ノードなど) を通過するインタフェースがマシンにある場合は、このパラメータに strong を設定します。
コミットレベル	変更の可能性あり

重複アドレスの検出に関連した IP チューニング可能パラメータ

ネットワークの重複アドレス検出 (DAD) を実行するには次のパラメータを構成します。

`_arp_defend_interval / _ndp_defend_interval`

説明	システムが、ネットワークの重複アドレスを検出するために IPv4 ARP および IPv6 NDP のアドレス宣言をブロードキャストする間隔。
デフォルト	300,000 ミリ秒
範囲	0-360,000
動的か	はい
どのような場合に 変更するか	なし

コミットレベル 変更の可能性あり

`_arp_defend_period / _ndp_defend_period`

説明	非請求の ARP または NDP アドレス保護メッセージがいずれかの物理ネットワークインタフェースで生成される時間間隔。これらのパラメータは、「 <code>_arp_defend_rate / _ndp_defend_rate</code> 」と連携して機能します。 これらのパラメータは、通常の ARP または NDP 解決、または競合が検出されたときのアドレス保護には適用されません。その代わりに、これらのパラメータは、自発的な競合検出トラフィックに対してのみ実装されます。
デフォルト	3,600 秒
範囲	0-3,600
動的か	はい
どのような場合に 変更するか	なし
コミットレベル	変更の可能性あり

`_arp_defend_rate / _ndp_defend_rate`

説明	いずれかの物理ネットワークインタフェースにおいて 1 時間で生成可能な非請求の ARP または NDP アドレス保護メッセージの数。時間間隔を変更するには、「 <code>_arp_defend_period / _ndp_defend_period</code> 」を構成します。 これらのパラメータは、通常の ARP または NDP 解決、または競合が検出されたときのアドレス保護には適用されません。その代わりに、これらのパラメータは、自発的な競合検出トラフィックに対してのみ実装されます。
デフォルト	100 メッセージ / 時
範囲	0-20,000
動的か	はい
どのような場合に 変更するか	なし
コミットレベル	変更の可能性あり

`_arp_fastprobe_count`

説明	転送 - 停止シーケンスにおいて、重複するアドレスを検出するために、この数のプローブが転送されたあと停止します。時間の長さは、「 <code>_arp_fastprobe_interval</code> 」で定義されます。このパラメータは、重複アドレスのプローブを高速化するために使用されます。
デフォルト	3 パケット
範囲	0-20
動的か	はい
どのような場合に 変更するか	なし
コミットレベル	変更の可能性あり

`_arp_fastprobe_interval`

説明	重複するアドレスを検出するための、ある一定のプローブ数の送出間隔である「 <code>_arp_probe_interval</code> 」と同じ機能。基盤となるドライバがリンクアップまたはリンクダウンのイベントを正しく報告できる場合、システムは、IP インタフェースを起動するプロセスを高速化するために、このパラメータをプローブの送出間隔として使用します。このパラメータは、「 <code>_arp_fastprobe_count</code> 」とともに使用します。
デフォルト	150 ミリ秒
範囲	10-20,000
動的か	はい
どのような場合に 変更するか	なし
コミットレベル	変更の可能性あり

`_arp_probe_count`

説明	転送 - 停止シーケンスにおいて、重複するアドレスを検出するために、この数のプローブが転送されたあと停止します。停止の長さは、「 <code>_arp_probe_interval</code> 」によって決まります。停止時間が経過すると、プローブが再開されます。
----	--

デフォルト	3 パケット
範囲	0-20
動的か	はい
どのような場合に 変更するか	なし
コミットレベル	変更の可能性あり

`_arp_probe_interval`

説明	重複アドレスを検出するための、ある一定のプロブ数の送出時間間隔。各間隔のあとに送信されるプロブの数は、「 <code>_arp_probe_count</code> 」によって定義されます。
デフォルト	1,500 ミリ秒
範囲	10-20,000
動的か	はい
どのような場合に 変更するか	なし
コミットレベル	変更の可能性あり

`arp_publish_count/ndp_unsolicit_count`

説明	ネットワークピアのアドレスキャッシュを更新するための非請求のアドレス宣言ごとに送信される IPv4 ARP および IPv6 NDP の各パケット数。ローカル IP アドレスが正常に稼働すると、この宣言が送信され、「 <code>arp_publish_interval / ndp_unsolicit_interval</code> 」パラメータによって制御される間隔で送信されるようになります。
デフォルト	3 パケット
範囲	1-20
動的か	はい
どのような場合に 変更するか	なし

コミットレベル 変更の可能性あり

arp_publish_interval / ndp_unsolicit_interval

説明 ローカル IP アドレスが正常に稼働したあと、システムが、IPv4 ARP および IPv6 NDP の各非請求アドレス宣言を送出する時間間隔。この宣言は、ネットワークピアのアドレスキャッシュを更新するために送信されます。各宣言におけるパケットの数は、「arp_publish_count/ndp_unsolicit_count」パラメータによって制御されます。

デフォルト 2,000 ミリ秒

範囲 1,000-20,000

動的か はい

どのような場合に
変更するか なし

コミットレベル 変更の可能性あり

_defend_interval

説明 ローカルアドレスが、ほかのシステムの IP アドレスと競合していることが検出されたとき、そのローカルアドレスをシステムが保護する時間長。この時間間隔内におけるアドレス保護の試行回数は、「_max_defend」によって定義されます。

デフォルト 30 秒

範囲 0-999,999

動的か はい

どのような場合に
変更するか なし

コミットレベル 変更の可能性あり

_dup_recovery

説明 非一時アドレスがリモートシステムの同じアドレスと競合しているため、システムがその非一時アドレスをダウン状態としてマーキングしたあとのプ

ローブの送信間隔。ローカルシステムは定期的にプローブを送出して、競合が残っているかどうかをテストします。プローブの応答が返ってこない場合は、競合が解消されたとみなされ、アドレスが再度アップ状態としてマーキングされます。

デフォルト	300,000 ミリ秒
範囲	0-360,000
動的か	はい
どのような場合に 変更するか	なし
コミットレベル	変更の可能性あり

_max_defend

説明 IP アドレスがほかのシステムの IP アドレスと競合している場合に、その IP アドレスが保護される回数。アドレスの保護は、「[_defend_interval](#)」で指定された時間内に実行されます。

デフォルト	3 回
範囲	0-1,000
動的か	はい
どのような場合に 変更するか	なし
コミットレベル	変更の可能性あり

_max_temp_defend

説明 一時ローカルアドレスまたは DHCP によって管理されているアドレスがほかのシステムの IP アドレスと競合しているとき、システムがそのアドレスを保護する回数。[_max_temp_defend](#) の値を超えると、システムはそのアドレスを放棄します。

デフォルト	1 回
範囲	0-1,000
動的か	はい

どのような場合に 変更するか	なし
コミットレベル	変更の可能性あり

特別な注意を要する IP チューニング可能パラメータ

次のパラメータの変更は非推奨です。

`_pathmtu_interval`

説明	IP がパス最大転送単位 (PMTU) 検出情報をフラッシュしてから PMTU をふたたび検出開始するまでの間隔をミリ秒単位で指定します。PMTU の検出については、RFC 1191 を参照してください。
デフォルト	1200 ミリ秒 (20 分)
範囲	2-999,999,999
動的か	はい
どのような場合に 変更するか	この値は変更しないでください。
コミットレベル	変更の可能性あり

`_icmp_return_data_bytes (ipv4 または ipv6)`

説明	IPv4 や IPv6 は、ICMPv4 または ICMPv6 のエラーメッセージを送信するときに、エラーメッセージの原因になったパケットの IP ヘッダーを含めます。このパラメータでは、パケットのうち IPv4 や IPv6 のヘッダーを除いてあと何バイトを ICMPv4 や ICMPv6 のエラーメッセージに含めるかを制御します。
デフォルト	IPv4 の場合 64 IPv6 の場合 1,280
範囲	IPv4 の場合 8 から 65,536 IPv6 の場合 8 から 1,280
動的か	はい

どのような場合に 変更するか	この値は変更しないでください。ただし、ICMP エラーメッセージを含む情報を増やすとネットワークの問題を診断する上で役立つことがあります。この機能が必要な場合は、値を増やします。
コミットレベル	変更の可能性あり

TCP チューニング可能パラメータ

`_deferred_ack_interval`

説明	直接接続していないホストに対する TCP 遅延肯定応答 (ACK) タイマーのタイムアウト値を指定します。 RFC 1122 の 4.2.3.2 を参照してください。
デフォルト	100 ミリ秒
範囲	1 ミリ秒から 60,000 ミリ秒
動的か	はい
どのような場合に 変更するか	このパラメータには、500 ミリ秒を超える値を設定しないでください。次の場合は、値を増やします。 <ul style="list-style-type: none"> ■ ネットワークリンクが遅く (57.6 Kbps 未満)、最大セグメントサイズ (MSS) が 512 バイトを超える ■ この間隔が複数の TCP セグメントを受信するには短すぎる
コミットレベル	変更の可能性あり

`_local_dack_interval`

説明	直接接続しているホストに対する TCP 遅延肯定応答 (ACK) タイマーのタイムアウト値を指定します。 RFC 1122 の 4.2.3.2 を参照してください。
デフォルト	50 ミリ秒
範囲	10 から 500 ミリ秒
動的か	はい

どのような場合に 変更するか	このパラメータには、500 ミリ秒を超える値を設定しないでください。 次の場合は、値を増やします。 <ul style="list-style-type: none">■ ネットワークリンクが遅く (57.6 Kbps 未満)、最大セグメントサイズ (MSS) が 512 バイトを超える■ この間隔が複数の TCP セグメントを受信するには短すぎる
コミットレベル	変更の可能性あり

_deferred_acks_max

説明	肯定応答 (ACK) が生成される前にリモート宛先 (直接接続していない) から受け取られる TCP セグメントの最大数を指定します。TCP セグメントは、個々の接続の最大セグメントサイズ (MMS) 単位で表されます。このパラメータに 0 か 1 を設定すると、すべてのセグメントが 1 MSS の長さであるとみなされ、遅延 ACK は使用されなくなります。実際の数は、接続ごとに動的に計算されます。この値はデフォルトの最大値です。
デフォルト	2
範囲	0 から 16
動的か	はい
どのような場合に 変更するか	この値は変更しないでください。ただし、遅延 ACK の影響でネットワークトラフィックが著しく混雑するような状況では、この値を減らします。ただし、2 より小さくしないでください。
コミットレベル	変更の可能性あり

_local_dacks_max

説明	肯定応答 (ACK) が生成される前に宛先 (直接接続している) から受け取られる TCP セグメントの最大数を指定します。TCP セグメントは、個々の接続の最大セグメントサイズ (MMS) 単位で表されます。このパラメータに 0 か 1 を設定すると、すべてのセグメントが 1 MSS の長さであるとみなされ、遅延 ACK は使用されなくなります。実際の数は、接続ごとに動的に計算されます。この値はデフォルトの最大値です。
デフォルト	8
範囲	0 から 16

動的か	はい
どのような場合に 変更するか	この値は変更しないでください。ただし、遅延 ACK の影響でネットワークトラフィックが著しく混雑するような状況では、この値を減らします。ただし、2 より小さくしないでください。
コミットレベル	変更の可能性あり

`_wscale_always`

説明	このパラメータが有効になっていると (デフォルトの設定)、ウィンドウスケールオプションの値が 0 の場合でも、TCP は常にウィンドウスケールオプションを指定して SYN セグメントを送信します。ウィンドウスケールオプションの指定された SYN セグメントを受信すると、パラメータが無効になっている場合でも、TCP は、ウィンドウスケールオプションを指定して SYN セグメントに応答します。オプションの値は受信ウィンドウサイズに従って設定されます。 ウィンドウスケールオプションについては、RFC 1323 を参照してください。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に 変更するか	ウィンドウスケールオプションをサポートしていない古い TCP スタックとの相互運用性の問題がある場合は、このパラメータを無効にしてください。
コミットレベル	変更の可能性あり

`_tstamp_always`

説明	1 が設定されていると、TCP は常にタイムスタンプオプションを指定して SYN セグメントを送信します。2 に設定すると、TCP 接続がアクティブまたはパッシブのどちらかでオープンされたかとは関係なく、タイムスタンプは完全に無効化されます。TCP は、タイムスタンプオプションの指定された (0 の場合もある) SYN セグメントを受信すると、タイムスタンプオプションを指定して SYN セグメントに応答します。
デフォルト	0 (無効)

範囲	0 (無効)、1 (有効)、または 2 (TCP 接続のオープン方法に関係なく無効)
動的か	はい
どのような場合に 変更するか	往復時間 (RTT) や TCP シーケンス番号ラップアラウンドを正確に測定したい場合、有効にします。 このオプションを有効にする理由については、RFC 1323 を参照してください。
コミットレベル	変更の可能性あり

send_buf

説明	デフォルトの送信ウィンドウサイズをバイト数で指定します。ルートごとに異なる値を設定する方法については、後述の177 ページの「 ルート別のメトリック 」を参照してください。151 ページの「 max_buf 」も参照してください。
デフォルト	49,152
範囲	4,096 から 151 ページの「 max_buf 」の現在の値
動的か	はい
どのような場合に 変更するか	アプリケーションは <code>setsockopt(3XNET)</code> <code>SO_SNDBUF</code> を使って、送信バッファを接続ごとに変更できます。
コミットレベル	変更の可能性あり

recv_buf

説明	デフォルトの受信ウィンドウサイズをバイト数で指定します。ルートごとに異なる値を設定する方法については、後述の177 ページの「 ルート別のメトリック 」を参照してください。151 ページの「 max_buf 」と 162 ページの「 _recv_hiwat_minmss 」も参照してください。
デフォルト	128,000
範囲	2,048 から 151 ページの「 max_buf 」の現在の値
動的か	はい

どのような場合に 変更するか	アプリケーションは <code>setsockopt(3XNET)</code> <code>SO_RCVBUF</code> を使って、受信バッファを接続ごとに変更できます。
コミットレベル	変更の可能性あり

`max_buf`

説明	送信および受信バッファサイズの最大値をバイト数で指定します。このパラメータは、 <code>setsockopt(3XNET)</code> を使用するアプリケーションによって設定される送信バッファサイズと受信バッファサイズを制御します。
デフォルト	1,048,576
範囲	128,000 から 1,073,741,824
動的か	はい
どのような場合に 変更するか	高速ネットワーク環境で TCP 接続を行う場合は、ネットワークリンクの速度に合わせて値を増やします。
コミットレベル	変更の可能性あり

`_cwnd_max`

説明	TCP 輻輳ウィンドウ (cwnd) の最大値をバイト数で指定します。TCP 輻輳ウィンドウについては、RFC 1122 と 2581 を参照してください。
デフォルト	1,048,576
範囲	128 から 1,073,741,824
動的か	はい
どのような場合に 変更するか	アプリケーションが <code>setsockopt(3XNET)</code> を使用してウィンドウサイズに <code>_cwnd_max</code> より大きい値を設定しようとしても、使用される実際のウィンドウが <code>_cwnd_max</code> を超えることはありません。したがって、 <code>_max_buf</code> は <code>_cwnd_max</code> より大きくするべきです。
コミットレベル	変更の可能性あり

_slow_start_initial

説明	輻輳ウィンドウ (cwnd) の初期サイズの最大値を TCP 接続の MSS 単位で指定します。 輻輳ウィンドウの初期サイズがどのように計算されるかについては、RFC 2414 を参照してください。
デフォルト	10
範囲	1 から 10
動的か	はい
どのような場合に 変更するか	この値は変更しないでください。 特殊な状況下で cwnd の初期サイズがネットワークの輻輳を招く場合は、この値を減らします。
コミットレベル	変更の可能性あり

_local_slow_start_initial

説明	輻輳ウィンドウ (cwnd) の初期サイズを、直接接続されたホスト間の TCP 接続の最大セグメントサイズ (MSS) 単位で指定します。
デフォルト	10
範囲	1 から 16,384
動的か	はい
どのような場合に 変更するか	初期ウィンドウが大きい方がアプリケーションにメリットがある場合、このパラメータを増やします。
コミットレベル	変更の可能性あり

_slow_start_after_idle

説明	輻輳ウィンドウが 1 再送タイムアウト (RTO) の間アイドルにされた (セグメントをまったく受信しなかった) あとの輻輳ウィンドウのサイズを TCP 接続の MSS 単位で指定します。 輻輳ウィンドウの初期サイズがどのように計算されるかについては、RFC 2414 を参照してください。
----	--

デフォルト	4
範囲	1 から 16,384
動的か	はい
どのような場合に 変更するか	詳細は、152 ページの「 <code>_slow_start_initial</code> 」を参照してください。
コミットレベル	変更の可能性あり

sack

説明	2 が設定されていると、TCP は常に選択的肯定応答 (SACK) 許可オプションを指定して SYN セグメントを送信します。SACK 許可オプションとして値 1 が指定されている SYN セグメントを受信した場合、TCP は SACK 許可オプションを指定して応答します。値 0 が設定されている場合は、着信セグメントに SACK 許可オプションが指定されているかどうかにかかわらず、TCP は SACK 許可オプションを送信しません。SACK オプションについては、RFC 2018 を参照してください。
デフォルト	アクティブ
範囲	なし、パッシブ、またはアクティブ
動的か	はい
どのような場合に 変更するか	SACK 処理を行うと TCP 再送のパフォーマンスが向上するため、自発的に有効にします。自発的に有効にすると相手方が混乱するおそれがある場合は、1 を設定します。この場合、SACK 処理は、着信接続で SACK 処理が許可されているときにのみ行われます。
コミットレベル	変更の可能性あり

_rev_src_routes

説明	0 が設定されていると、TCP は、セキュリティ上の理由により、着信接続に対して IP ソースルーティングオプションを逆方向に使用しません。1 が設定されている場合は、通常どおりソースルーティングを逆方向に使用します。
デフォルト	0 (無効)

範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に 変更するか	診断のために IP ソースルーティングが必要な場合は、有効にします。
コミットレベル	変更の可能性あり

`_time_wait_interval`

説明	TCP 接続を TIME-WAIT 状態に保つ時間をミリ秒で指定します。 RFC 1122 の 4.2.2.13 を参照してください。
デフォルト	600,000 (60 秒)
範囲	1 秒から 600,000 ミリ秒
動的か	はい
どのような場合に 変更するか	この値は 60 秒より小さくしないでください。 このパラメータの変更方法については、RFC 1122 の 4.2.2.13 を参照 してください。
コミットレベル	変更の可能性あり

`ecn`

説明	ECN (Explicit Congestion Notification、明示的輻輳通知) のサ ポートを制御します。 このパラメータが 0 に設定されていると、TCP は、ECN メカニズムをサ ポートしている接続先とのネゴシエーションを行いません。 接続開始時にこのパラメータが 1 に設定されていると、TCP は、ECN メカニズムをサポートしていることを接続先に通知しません。 ただし、接続先が SYN セグメントで ECN メカニズムをサポートしてい ることを示した場合、TCP は、新しい着信接続要求を受けた際に、ECN メカニズムをサポートしていることを接続先に通知します。 このパラメータを 2 に設定すると、TCP は接続を受け付けた時点で ECN メカニズムに関して接続先とネゴシエーションを行います。さら に、TCP は自発的な送信接続を行う際に、送信する SYN セグメント内 で、ECN メカニズムをサポートしていることを示します。
----	--

	ECN については、RFC 3168 を参照してください。
デフォルト	パッシブ
範囲	なし、パッシブ、アクティブ
動的か	はい
どのような場合に 変更するか	TCP は、ECN を利用して、輻輳制御の処理を効率化できます。ただし、このメカニズムにより、既存の TCP 実装やファイアウォール、NAT などのネットワークデバイスが混乱する場合があります。混乱するデバイスは IETF 非準拠です。 これらのデバイスを考慮し、このパラメータのデフォルト値は 1 に設定されています。ただし、まれに、受動的に有効にした場合でも問題が生じる場合があります。必要がある場合以外は、パラメータを 0 に設定しないでください。
コミットレベル	変更の可能性あり

`_conn_req_max_q`

説明	<code>accept(3SOCKET)</code> によって受け付けられるのを待っている TCP リスナーの、保留状態の TCP 接続のデフォルトの最大数を指定します。 156 ページ の「 <code>_conn_req_max_q0</code> 」も参照してください。
デフォルト	128
範囲	1 から 4,294,967,295
動的か	はい
どのような場合に 変更するか	複数の接続要求を受けることのある Web サーバーのようなアプリケーションでは、着信頻度に応じてこのデフォルト値を増やすことができます。 このパラメータに著しく大きい値を設定しないでください。保留状態の TCP 接続はメモリーを過剰に使用することがあります。さらに、保留状態の TCP 接続の数が多すぎて、アプリケーションが接続要求を適時に処理できない場合は、新しい着信要求が拒否されることがあります。 <code>_conn_req_max_q</code> を増やしても、アプリケーションでそれだけの数の保留状態の TCP 接続を持てるとは限りません。アプリケーションでは、 <code>listen(3SOCKET)</code> を使用して、保留状態の TCP 接続の最大数をソケットごとに変更できます。このパラメータは、アプリケーションが <code>listen()</code> を使用して設定できる最大値を表します。つまり、このパラメータ

々に非常に大きな値を設定しても、あるソケットに対する実際の最大数は、`listen()` に指定された値によっては `_conn_req_max_q` よりもはるかに少ないことがあります。

コミットレベル 変更の可能性あり

`_conn_req_max_q0`

説明	<p>単一の TCP リスナーが持つことができる、不完全な (3 段階ハンドシェイクがまだ終わっていない) 保留状態の TCP 接続のデフォルトの最大数を指定します。</p> <p>TCP の 3 段階ハンドシェイクについては、RFC 793 を参照してください。155 ページの「<code>_conn_req_max_q</code>」も参照してください。</p>
デフォルト	1,024
範囲	0 から 4,294,967,295
動的か	はい
どのような場合に 変更するか	<p>きわめて多くの接続要求を受信することがある Web サーバーのようなアプリケーションでは、着信頻度に応じてこのデフォルト値を増やすことができます。</p> <p><code>_conn_req_max_q0</code> と、各ソケットについて保留状態にある接続の最大数との関係は、次のとおりです。</p> <p>接続要求を受信すると、TCP はまず、受け付けられるのを待っている保留状態の TCP 接続 (3 段階ハンドシェイクが終わっている) の数が、そのリスナーに対する最大数 (N) を超えていないかをチェックします。接続数が超えていると、その要求は拒否されます。超えていなければ、は、不完全な保留状態の TCP 接続の数が、N と <code>tcp_conn_req_max_q0</code> の合計を超えていないかをチェックします。そうでなければ、その要求は受け付けられます。それ以外の場合、もっとも古い不完全な保留状態の TCP 要求がドロップされます。</p>
コミットレベル	変更の可能性あり

`_conn_req_min`

説明	<p>受け付けられるのを待っている、単一のリスナーの保留状態の TCP 接続の最大数のデフォルトの最小値。これは、1 つのアプリケーションが使用できる <code>listen(3SOCKET)</code> のもっとも小さい最大値です。</p>
----	---

デフォルト	1
範囲	1 から 1,024
動的か	はい
どのような場合に 変更するか	このパラメータにより、 listen(3SOCKET) を使用するアプリケーションが保留状態の TCP 接続の最大数を過度に小さく設定するのを防ぐことができます。この値は、着信接続要求の頻度に応じて増やすことができます。
コミットレベル	変更の可能性あり

`_rst_sent_rate_enabled`

説明	このパラメータに 1 が設定されている場合、RST セグメントの最大送信速度は、 <code>ipadm</code> パラメータ <code>_rst_sent_rate</code> によって制御されます。このパラメータに 0 が設定されている場合、RST セグメントの送信時に速度を制御することはできません。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に 変更するか	このチューニング可能パラメータは、RST セグメントの送信速度を制限することで、TCP に対するサービス拒否攻撃を防止します。この速度制御は、RFC 793 に厳密に準拠する必要がある場合にのみ、無効になります。
コミットレベル	変更の可能性あり

`_rst_sent_rate`

説明	TCP が 1 秒間に送信できる最大 RST セグメント数を設定します。
デフォルト	40
範囲	0 から 4,294,967,295
動的か	はい

どのような場合に 変更するか	TCP 環境では、正当な理由により、デフォルト値より多くの RST が生成される場合があります。このような場合は、このパラメータのデフォルト値を引き上げます。
コミットレベル	変更の可能性あり

smallest_anon_port

説明	このパラメータは、TCP が一時的なポートとして選択できる最小ポート番号を制御します。アプリケーションでは、指定されたプロトコルを使って接続を作成する際に、ポート番号を指定しないで一時的なポートを使用することができます。一時的なポートは、特定のアプリケーションに関連付けられていません。接続が閉じられると、ポート番号を別のアプリケーションで再利用できます。
Unit	ポート番号
デフォルト	32,768
範囲	1,024 から 65,535
動的か	はい
どのような場合に 変更するか	より広範囲の一時的なポートが必要な場合。
コミットレベル	変更の可能性あり

largest_anon_port

説明	このパラメータは、TCP が一時的なポートとして選択できる最大ポート番号を制御します。アプリケーションでは、指定されたプロトコルを使って接続を作成する際に、ポート番号を指定しないで一時的なポートを使用することができます。一時的なポートは、特定のアプリケーションに関連付けられていません。接続が閉じられると、ポート番号を別のアプリケーションで再利用できます。
Unit	ポート番号
デフォルト	65,535
範囲	32,768 から 65,535

動的か	はい
どのような場合に 変更するか	より広範囲の一時的なポートが必要な場合。
コミットレベル	変更の可能性あり

特別な注意を要する TCP パラメータ

次のパラメータの変更は非推奨です。

`_keepalive_interval`

説明	<p>この <code>ipadm</code> パラメータは、システム全体で TCP 接続がアイドル状態になってから最初にプローブが送信されるまでの間隔を設定します。</p> <p>Solaris は、RFC 1122 に記述されている TCP キープアライブメカニズムをサポートします。このメカニズムは、TCP ソケットで <code>SO_KEEPALIVE</code> ソケットオプションを設定することで有効になります。</p> <p>ソケットで <code>SO_KEEPALIVE</code> が有効な場合、TCP 接続が 2 時間 (<code>tcp_keepalive_interval</code> パラメータのデフォルト値) アイドル状態になると最初のキープアライブプローブが送信されます。ピアがプローブに 8 分間応答しない場合、TCP 接続が終了します。詳細は、160 ページの「<code>_rexmit_interval_initial</code>」を参照してください。</p> <p>また、個々のアプリケーションに <code>TCP_KEEPALIVE_THRESHOLD</code> ソケットオプションを使用してデフォルトの間隔をオーバーライドすれば、各アプリケーションがソケットごとに独自の間隔を持つようになります。オプションの値は、ミリ秒単位の符号なし整数です。<code>tcp(7P)</code> も参照してください。</p>
デフォルト	2 時間
範囲	10 秒から 10 日
単位	符号なし整数 (ミリ秒)
動的か	はい
どのような場合に 変更するか	この値は変更しないでください。この値を小さくすると、不必要なネットワークトラフィックが生じる可能性があるとともに、ネットワークの一時的な問題のために未完了のまま接続が終了してしまう可能性も高くなります。
コミットレベル	変更の可能性あり

`_ip_abort_interval`

説明	<p>TCP 接続に対するデフォルトの合計再送タイムアウト値をミリ秒で指定します。ある 接続で、TCP が <code>tcp_ip_abort_interval</code> の間再転送を行なっても、この間に相手側のエンドポイントから肯定応答をまったく受け取らないと、この接続は閉じられます。</p> <p>TCP の再送タイムアウト (RTO) の計算については、RFC 1122 の 4.2.3 を参照してください。160 ページの「<code>_rexmit_interval_max</code>」も参照してください。</p>
デフォルト	5 分
範囲	500 ミリ秒から 1193 時間
動的か	はい
どのような場合に 変更するか	この値は変更しないでください。例外については、160 ページの「 <code>_rexmit_interval_max</code> 」を参照してください。
コミットレベル	変更の可能性あり

`_rexmit_interval_initial`

説明	<p>TCP 接続に対するデフォルトの初期再送タイムアウト値をミリ秒で指定します。ルートごとに異なる値を設定する方法については、後述の177 ページの「<code>ルート別のメトリック</code>」を参照してください。</p>
デフォルト	1,000 ミリ秒
範囲	1 ミリ秒から 20,000 ミリ秒
動的か	はい
どのような場合に 変更するか	この値は変更しないでください。この値を小さくすると、不要な再転送が行われるおそれがあります。
コミットレベル	変更の可能性あり

`_rexmit_interval_max`

説明	<p>デフォルトの最大再送タイムアウト値 (RTO) をミリ秒で指定します。すべての TCP 接続に対して、計算された RTO がこの値を超えることはありません。160 ページの「<code>_ip_abort_interval</code>」も参照してください。</p>
----	--

デフォルト	6,000 ミリ秒
範囲	1 ミリ秒から 7,200,000 ミリ秒
動的か	はい
どのような場合に 変更するか	通常のネットワーク環境では、この値を変更しないでください。 単一の接続の往復時間 (RTT) が 10 秒程度になるような特別な状況では、この値を増やすことができます。この値を変更する場合は、合わせて <code>_ip_abort_interval</code> パラメータも変更する必要があります。 <code>_ip_abort_interval</code> には、 <code>_rexmit_interval_max</code> の 4 倍以上の値を指定します。
コミットレベル	変更の可能性あり

`_rexmit_interval_min`

説明	デフォルトの最小再送タイムアウト値 (RTO) をミリ秒で指定します。すべての TCP 接続に対して、計算された RTO がこの値を下回ることはできません。 160 ページ の「 <code>_rexmit_interval_max</code> 」も参照してください。
デフォルト	200 ミリ秒
範囲	1 ミリ秒から 7,200,000 ミリ秒
動的か	はい
どのような場合に 変更するか	通常のネットワーク環境では、この値を変更しないでください。 TCP の RTO 計算は、RTT のもっとも大きい変動に対処できます。単一の接続の往復時間 (RTT) が 10 秒程度になるような特別な状況では、この値を増やすことができます。この値を変更する場合は、合わせて <code>_rexmit_interval_max</code> パラメータも変更する必要があります。 <code>_rexmit_interval_max</code> には、 <code>_rexmit_interval_min</code> の 8 倍以上の値を指定します。
コミットレベル	変更の可能性あり

`_rexmit_interval_extra`

説明	計算された再送タイムアウト値 (RTO) に追加する定数をミリ秒で指定します。
----	---

デフォルト	0 ミリ秒
範囲	0 から 7,200,000 ミリ秒
動的か	はい
どのような場合に 変更するか	この値は変更しないでください。 計算された RTO が接続に対して適切でない場合は、不要な再転送を 避けるためにこの値を変更することができます。
コミットレベル	変更の可能性あり

_tstamp_if_wscale

説明	このパラメータに 1 が設定され、かつ、ある接続に対するウィンドウスケールオプションが有効になっていると、TCP は、その接続の timestamp オプションも有効にします。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に 変更するか	この値は変更しないでください。一般に、TCP を高速ネットワークで使用 する場合は、シーケンス番号のラップアラウンドに対する保護が必要になり ます。この場合、timestamp オプションが必要になります。
コミットレベル	変更の可能性あり

_recv_hiwat_minmss

説明	デフォルトの最小受信ウィンドウサイズを制御します。最小値 は、_recv_hiwat_minmss に、接続の最大セグメントサイズ (MSS) を掛 けた値です。
デフォルト	8
範囲	1 から 65,536
動的か	はい

どのような場合に 変更するか	この値は変更しないでください。この値を変更する必要がある場合は、4 より小さい値にしないでください。
コミットレベル	変更の可能性あり

UDP チューニング可能パラメータ

send_buf

説明	UDP ソケットのデフォルトの送信バッファサイズを指定します。詳細 は、 164 ページの「max_buf」 を参照してください。
デフォルト	57,344 バイト
範囲	1,024 から 164 ページの「max_buf」 の現在の値
動的か	はい
どのような場合に 変更するか	このサイズは、アプリケーションで <code>setsockopt(3XNET)</code> <code>SO_SNDBUF</code> を 使用してソケットごとに変更できます。一般には、デフォルト値を変更す る必要はありません。
コミットレベル	変更の可能性あり

recv_buf

説明	UDP ソケットのデフォルトの受信バッファサイズを指定します。詳細 は、 164 ページの「max_buf」 を参照してください。
デフォルト	57,344 バイト
範囲	128 から 164 ページの「max_buf」 の現在の値
動的か	はい
どのような場合に 変更するか	このサイズは、アプリケーションで <code>setsockopt(3XNET)</code> <code>SO_RCVBUF</code> を 使用してソケットごとに変更できます。一般には、デフォルト値を変更す る必要はありません。
コミットレベル	変更の可能性あり

max_buf

説明	UDP ソケットの送信および受信バッファサイズの最大値を指定します。これは、アプリケーションが <code>getsockopt(3SOCKET)</code> を使用して設定する送信バッファおよび受信バッファの最大値を制御します。
デフォルト	2,097,152
範囲	65,536 から 1,073,741,824
動的か	はい
どのような場合に 変更するか	高速ネットワーク環境で接続を確立する場合は、ネットワークリンク速度に合わせて、このパラメータの値を大きくします。
コミットレベル	変更の可能性あり

smallest_anon_port

説明	このパラメータは、UDP が一時的なポートとして選択できる最小ポート番号を制御します。アプリケーションでは、指定されたプロトコルを使って接続を作成する際に、ポート番号を指定しないで一時的なポートを使用することができます。一時的なポートは、特定のアプリケーションに関連付けられていません。接続が閉じられると、ポート番号を別のアプリケーションで再利用できます。
Unit	ポート番号
デフォルト	32,768
範囲	1,024 から 65,535
動的か	はい
どのような場合に 変更するか	より広範囲の一時的なポートが必要な場合。
コミットレベル	変更の可能性あり

largest_anon_port

説明	このパラメータは、UDP が一時的なポートとして選択できる最大ポート番号を制御します。アプリケーションでは、指定されたプロトコルを使って接
----	---

続を作成する際に、ポート番号を指定しないで一時的なポートを使用することができます。一時的なポートは、特定のアプリケーションに関連付けられていません。接続が閉じられると、ポート番号を別のアプリケーションで再利用できます。

Unit	ポート番号
デフォルト	65,535
範囲	32,768 から 65,535
動的か	はい
どのような場合に 変更するか	より広範囲の一時的なポートが必要な場合。
コミットレベル	変更の可能性あり

IPQoS チューニング可能パラメータ

_policy_mask

説明 IPQoS 処理を有効または無効 にします。有効または無効にするコールアウト位置は、次のとおりです。転送アウトバウンド、転送インバウンド、ローカルアウトバウンド、またはローカルインバウンド。このパラメータは、次のようなビットマスクになっています。

使用し ない	使用し ない	使用し ない	使用し ない	転送アウト バウンド	転送イン バウンド	ローカル アウトバウ ンド	ローカルイ ンバウンド
X	X	X	X	0	0	0	0

どの位置でも 1 であれば、その特定のコールアウト位置で IPQoS 処理をマスク、すなわち無効にします。たとえば、0x01 の値は、すべてのローカルインバウンドパケットの IPQoS 処理を無効にします。

デフォルト	0 (すべてのコールアウト位置で IPQoS 処理が有効)
範囲	0 (0x00) から 15 (0x0F)。15 の場合、すべてのコールアウト位置の IPQoS 処理が無効

動的か	はい
どのような場合に 変更するか	いずれかのコールアウト位置で IPQoS 処理を有効または無効にしたい 場合
コミットレベル	変更の可能性あり

SCTP チューニング可能パラメータ

_max_init_retr

説明	SCTP 終端が INIT チャンクの再送信位置で行う最大試行回数を制御 します。SCTP 終端は、SCTP 設定構造で、この値をオーバーライドでき ます。
デフォルト	8
範囲	0 から 128
動的か	はい
どのような場合に 変更するか	INIT 再送回数は 166 ページの「_pa_max_retr」に依存しま す。_max_init_retr が _pa_max_retr 以下であれば理想的です。
コミットレベル	変更の可能性あり

_pa_max_retr

説明	SCTP 接続のすべてのパスを経由する最大再送回数を制御します。こ の値を超過すると、SCTP 接続は終了します。
デフォルト	10
範囲	1 から 128
動的か	はい
どのような場合に 変更するか	すべてのパスを経由する最大再送回数は、パスの数と各パスの最大再 送回数によって決定されます。_pa_max_retr は、使用可能なすべてのパ スの 167 ページの「_pp_max_retr」の合計に設定するべきです。たと えば、宛先までのパス数が 3 で、これらのパスの最大再送回数がそれ

ぞれ 5 回である場合、`_pa_max_retr` には 15 以下の値を設定する必要があります (RFC 2960, Section 8.2 の Note を参照)。

コミットレベル 変更の可能性あり

`_pp_max_retr`

説明 特定のパスを経由する最大再送回数を制御します。この数値を超過したパスがあると、パス (宛先) に到達できません。

デフォルト 5

範囲 1 から 128

動的か はい

どのような場合に
変更するか この値を 5 より小さい値に変更しないでください。

コミットレベル 変更の可能性あり

`_cwnd_max`

説明 SCTP 接続の輻輳ウィンドウの最大値を制御します。

デフォルト 1,048,576

範囲 128 から 1,073,741,824

動的か はい

どのような場合に
変更するか アプリケーションが `setsockopt(3XNET)` を使用してウィンドウサイズに `_cwnd_max` より大きい値を設定しようとしても、使用される実際のウィンドウが `_cwnd_max` を超えることはありません。したがって、172 ページの「`max_buf`」は `_cwnd_max` より大きくする必要があります。

コミットレベル 変更の可能性あり

`_ipv4_ttl`

説明 SCTP 接続上で、アウトバウンド IPv4 パケットの IP バージョン 4 ヘッダーの TTL 値を制御します。

デフォルト	64
範囲	1 から 255
動的か	はい
どのような場合に 変更するか	通常、この値を変更する必要はありません。
コミットレベル	変更の可能性あり

_ipv6_hoplimit

説明	SCTP 接続上で、アウトバウンド IPv6 パケットの IPv6 ヘッダーのホップ制限値を設定します。
デフォルト	60
範囲	0 から 255
動的か	はい
どのような場合に 変更するか	通常、この値を変更する必要はありません。
コミットレベル	変更の可能性あり

_heartbeat_interval

説明	HEARTBEAT チャンクからハードビートに対応したアイドル状態の宛先までの間隔を計算します。 SCTP 終端は、相手側のアイドル状態の宛先転送アドレスまでの到達性をモニターするため、定期的に HEARTBEAT チャンクを送信します。
デフォルト	30 秒
範囲	0 から 86,400 秒
動的か	はい
どのような場合に 変更するか	RFC 2960 のセクション 8.3 を参照してください。
コミットレベル	変更の可能性あり

`_new_secret_interval`

説明	新しいシークレットを生成するタイミングを判定します。生成されたシークレットから、Cookie の MAC を計算できます。
デフォルト	2 分
範囲	0 から 1,440 分
動的か	はい
どのような場合に 変更するか	RFC 2960 のセクション 5.1.3 を参照してください。
コミットレベル	変更の可能性あり

`_initial_mtu`

説明	IP ヘッダー長を含めた SCTP パケットの初期最大送信サイズを判定します。
デフォルト	1500 バイト
範囲	68 から 65,535
動的か	はい
どのような場合に 変更するか	基底リンクが 1500 バイト以上のフレームサイズをサポートする場合は、このパラメータの値を大きくします。
コミットレベル	変更の可能性あり

`_deferred_ack_interval`

説明	SCTP 遅延肯定応答 (ACK) タイマーのタイムアウト値をミリ秒で設定します。
デフォルト	100 ミリ秒
範囲	1 から 60,000 ミリ秒
動的か	はい

どのような場合に 変更するか	RFC 2960 のセクション 6.2 を参照してください。
コミットレベル	変更の可能性あり

_ignore_path_mtu

説明	パス MTU 検出の有効/無効を切り替えます。
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に 変更するか	パスにおける MTU の変更を無視する場合、このパラメータを有効にします。パス MTU が減った場合、このパラメータを有効にすると、IP 分割が行われます。
コミットレベル	変更の可能性あり

_initial_ssthresh

説明	相手側の宛先アドレスの初期スロースタートしきい値を設定します。
デフォルト	1,048,576
範囲	1,024 から 4,294,967,295
動的か	はい
どのような場合に 変更するか	RFC 2960 のセクション 7.2.1 を参照してください。
コミットレベル	変更の可能性あり

send_buf

説明	デフォルトの送信バッファサイズをバイト数で指定します。 172 ページの「max_buf」 も参照してください。
デフォルト	102,400

範囲	8,192 から 172 ページの「max_buf」の現在の値
動的か	はい
どのような場合に 変更するか	アプリケーションは <code>setsockopt(3XNET)</code> <code>SO_SNDBUF</code> を使って、送信バッファを接続ごとに変更できます。
コミットレベル	変更の可能性あり

_xmit_lowat

説明	送信ウィンドウサイズの下限値を制御します。
デフォルト	8,192
範囲	8,192 から 1,073,741,824
動的か	はい
どのような場合に 変更するか	通常、この値を変更する必要はありません。このパラメータは、ソケットの送信バッファを書き込み可能にするために必要な最小限のサイズを設定します。必要に応じて、170 ページの「send_buf」と一致するようにこのパラメータを変更できます。
コミットレベル	変更の可能性あり

recv_buf

説明	デフォルトの受信バッファサイズをバイト数で指定します。172 ページの「max_buf」も参照してください。
デフォルト	102,400
範囲	8,192 から 172 ページの「max_buf」の現在の値
動的か	はい
どのような場合に 変更するか	アプリケーションは <code>setsockopt(3XNET)</code> <code>SO_RCVBUF</code> を使って、受信バッファを接続ごとに変更できます。
コミットレベル	変更の可能性あり

max_buf

説明	送信および受信バッファサイズの最大値をバイト数で制御します。これは、アプリケーションが <code>getsockopt(3SOCKET)</code> を使用して設定する送信バッファおよび受信バッファの最大値を制御します。
デフォルト	1,048,576
範囲	102,400 から 1,073,741,824
動的か	はい
どのような場合に 変更するか	高速ネットワーク環境で接続を確立する場合は、ネットワークリンク速度に合わせて、このパラメータの値を大きくします。
コミットレベル	変更の可能性あり

_rto_min

説明	相手側のすべての宛先アドレスの再送タイムアウト (RTO) の下限値をミリ秒で設定します。
デフォルト	1,000
範囲	500 から 60,000
動的か	はい
どのような場合に 変更するか	RFC 2960 のセクション 6.3.1 を参照してください。
コミットレベル	変更の可能性あり

_rto_max

説明	相手側のすべての宛先アドレスの再送タイムアウト (RTO) の上限値をミリ秒で設定します。
デフォルト	60,000
範囲	1,000 から 60,000,000

動的か	はい
どのような場合に 変更するか	RFC 2960 のセクション 6.3.1 を参照してください。
コミットレベル	変更の可能性あり

`_rto_initial`

説明	相手側のすべての宛先アドレスの初期再送タイムアウト (RTO) 値をミリ秒で制御します。
デフォルト	3,000
範囲	1,000 から 60,000,000
動的か	はい
どのような場合に 変更するか	RFC 2960 のセクション 6.3.1 を参照してください。
コミットレベル	変更の可能性あり

`_cookie_life`

説明	Cookie の寿命をミリ秒で設定します。
デフォルト	60,000
範囲	10 から 60,000,000
動的か	はい
どのような場合に 変更するか	通常、この値を変更する必要はありません。このパラメータは、 172 ページ の「 <code>_rto_max</code> 」に合わせて変更できます。
コミットレベル	変更の可能性あり

`_max_in_streams`

説明	SCTP 接続 1 個あたりに許可された最大インバウンドストリーム数を制御します。
----	---

デフォルト	32
範囲	1 から 65,535
動的か	はい
どのような場合に 変更するか	RFC 2960 のセクション 5.1.1 を参照してください。
コミットレベル	変更の可能性あり

`_initial_out_streams`

説明	SCTP 接続 1 個あたりに許可された最大アウトバウンドストリーム数を制御します。
デフォルト	32
範囲	1 から 65,535
動的か	はい
どのような場合に 変更するか	RFC 2960 のセクション 5.1.1 を参照してください。
コミットレベル	変更の可能性あり

`_shutack_wait_bound`

説明	SHUTDOWN チャンクの送信後、SHUTDOWN ACK を待機する最大待ち時間をミリ秒で制御します。
デフォルト	60,000
範囲	0 から 300,000
動的か	はい
どのような場合に 変更するか	通常、この値を変更する必要はありません。このパラメータは、 172 ページ の「 <code>_rto_max</code> 」に合わせて変更できます。
コミットレベル	変更の可能性あり

`_maxburst`

説明	1 つのバーストで送信されるセグメント数の制限値を設定します。
デフォルト	4
範囲	2 から 8
動的か	はい
どのような場合に 変更するか	このパラメータを変更する必要はありません。この値はテスト目的で変更 する場合があります。
コミットレベル	変更の可能性あり

`_addip_enabled`

説明	SCTP 動的アドレス構成の有効/無効を切り替えます。
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に 変更するか	動的アドレス構成が必要な場合は有効にします。セキュリティ上、この パラメータはテスト目的以外では有効にしないでください。
コミットレベル	変更の可能性あり

`_prsctp_enabled`

説明	SCTP に対する部分的な信頼の拡張 (RFC 3758) の有効/無効を切り 替えます。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に 変更するか	ご使用の SCTP 環境で部分的な信頼がサポートされていない場合、無 効にします。

コミットレベル 変更の可能性あり

smallest_anon_port

説明	このパラメータは、SCTP が一時的なポートとして選択できる最小ポート番号を制御します。アプリケーションでは、指定されたプロトコルを使って接続を作成する際に、ポート番号を指定しないで一時的なポートを使用することができます。一時的なポートは、特定のアプリケーションに関連付けられていません。接続が閉じられると、ポート番号を別のアプリケーションで再利用できます。
Unit	ポート番号
デフォルト	32,768
範囲	1,024 から 65,535
動的か	はい
どのような場合に 変更するか	より広範囲の一時的なポートが必要な場合。
コミットレベル	変更の可能性あり

largest_anon_port

説明	このパラメータは、SCTP が一時的なポートとして選択できる最大ポート番号を制御します。アプリケーションでは、指定されたプロトコルを使って接続を作成する際に、ポート番号を指定しないで一時的なポートを使用することができます。一時的なポートは、特定のアプリケーションに関連付けられていません。接続が閉じられると、ポート番号を別のアプリケーションで再利用できます。
Unit	ポート番号
デフォルト	65,535
範囲	32,768 から 65,535
動的か	はい
どのような場合に 変更するか	より広範囲の一時的なポートが必要な場合。

コミットレベル 変更の可能性あり

ルート別のメトリック

ルート別のメトリックを使用して、一定のプロパティを IPv4 や IPv6 のルーティングテーブルエントリに関連付けることができます。

たとえば、システムに、Fast Ethernet と Gigabit Ethernet という 2 つの異なるネットワークインタフェースがあるとします。recv_maxbuf のシステムデフォルト値は 128,000 バイトです。このデフォルト値は Fast Ethernet インタフェースには十分ですが、Gigabit Ethernet インタフェースには不十分な可能性があります。

recv_maxbuf のシステムデフォルト値を増やす代わりに、Gigabit Ethernet インタフェースのルーティングエントリに別のデフォルトの TCP 受信ウィンドウサイズを関連付けることができます。これにより、そのルートを通るすべての TCP 接続には、大きくした受信ウィンドウサイズが使用されます。

たとえば、IPv4 を使用する次のようなルーティングテーブルがあるとします (netstat -rn)。

```
Routing Table: IPv4
Destination      Gateway          Flags Ref    Use    Interface
-----
192.123.123.0    192.123.123.4   U      1      4     net0
192.123.124.0    192.123.124.4   U      1      4     net1
default          192.123.123.1   UG     1      8
```

この例では、次の処理が行われます。

```
# route change -net 192.123.124.0 -recvpipe x
```

この結果、net1 リンクの 192.123.124.0 ネットワークへのすべての接続には、デフォルトの受信ウィンドウサイズ 128,000 の代わりに、受信バッファサイズ *x* が使用されます。

宛先が a.b.c.d ネットワークにあり、そのネットワーク固有のルーティングエントリがない場合は、そのネットワークに接頭辞ルートを追加し、メトリックを変更できます。例:

```
# route add -net a.b.c.d 192.123.123.1 -netmask w.x.y.z
# route change -net a.b.c.d -recvpipe y
```

接頭辞ルートのゲートウェイがデフォルトのルーターであることに留意してください。そのネットワークへのすべての接続は受信バッファサイズ *y* を使用します。複数のインタフェースがある場合は、-ifp 引数を使用して、使用するインタフェースを指定します。それによって、特定

の宛先に対してどのインタフェースを使用するかを制御できます。メトリックを確認するには `route(1M) get` コマンドを使用します。

◆◆◆ 第 6 章

システム機能のパラメータ

この章では、各種システム機能のデフォルト値を設定する大部分のパラメータについて説明します。

ほかのタイプのチューニング可能パラメータについては、次を参照してください。

- Oracle Solaris カーネルチューニング可能パラメータ – [第2章「Oracle Solaris カーネルチューニング可能パラメータ」](#)
- Oracle Solaris ZFS チューニング可能パラメータ – [第3章「Oracle Solaris ZFS チューニング可能パラメータ」](#)
- NFS チューニング可能パラメータ – [第4章「NFS チューニング可能パラメータ」](#)
- インターネットプロトコルスイートのチューニング可能パラメータ – [第5章「インターネットプロトコル群のチューニング可能パラメータ」](#)

システムのデフォルトのパラメータ

さまざまなシステム機能の動作は、その機能が起動時に読み込む一連の値によって制御されます。各機能の値は、`/etc/default` ディレクトリにある機能のファイルに保存するか、サービス管理機能 (SMF) 構成リポジトリのサービスインスタンスのプロパティに保存することができます。SMF サービスとプロパティの詳細は、『[Oracle Solaris 11.2 でのシステムサービスの管理](#)』を参照してください。

電源管理プロパティの設定に関する詳細は、『[Oracle Solaris 11.2 でのシステム情報、プロセス、およびパフォーマンスの管理](#)』を参照してください。

autofs

`sharectl` コマンドを使用して、SMF `autofs` プロパティを表示または構成できます。例:

```
# sharectl get autofs
timeout=600
automount_verbose=false
automountd_verbose=false
nobrowse=false
trace=0
environment=
# sharectl set -p timeout=200 autofs
```

詳細は、[sharectl\(1M\)](#) を参照してください。

cron

この機能を利用して、cron ロギングの有効/無効を切り替えることができます。

devfsadm

現在、このファイルは使用されていません。

dhcpageant

DHCP のクライアント使用率は、dhcpageant デーモンによって提供されます。ipadm を使用して、DHCP アドレスオブジェクトを作成する場合や、ipadm で、DHCP からそのネットワーク構成を受信するように構成されたインタフェースが識別される場合、dhcpageant が起動され、そのインタフェースのアドレスが管理されます。

詳細は、[dhcpageant\(1M\)](#) の「FILES」セクションにある `/etc/default/dhcpageant` の情報を参照してください。

fs

ファイルシステム管理コマンドには、汎用的な部分とファイルシステム固有の部分があります。ファイルシステムのタイプが `-F` オプションで明示的に指定されていない場合は、デフォルトが使用されます。その値はこのファイルに指定されています。詳細は、[default_fs\(4\)](#) の「Description」セクションを参照してください。

ftp

この機能を利用して、`ls` コマンドに RFC 959 `MLST` コマンドの動作を設定できます。デフォルトの `ls` の動作は、以前の Solaris リリースから変更されていません。

詳細は、[ftp\(4\)](#) を参照してください。

inetinit

この機能を利用して、TCP シーケンス番号を構成したり、6to4 リレールーターのサポートを有効または無効にすることができます。

init

システム初期化プロパティは次の SMF サービスに含まれるようになりました。

```
svc:/system/environment:init
```

同様の構文を使用して、`TZ` や `LANG` などのシステム初期化プロパティを表示し、構成できます。

```
# svccfg -s svc:/system/environment:init
svc:/system/environment:init> setprop
Usage: setprop pg/name = [type:] value
setprop pg/name = [type:] ([value...])
```

Set the `pg/name` property of the currently selected entity. Values may be enclosed in double-quotes. Value lists may span multiple lines.

```
svc:/system/environment:init> listprop
umask                                application
umask/umask                          astring    022
umask/value_authorization            astring    solaris.smf.value.environment
environment                          application
environment/LANG                    astring
environment/LC_ALL                  astring
.
```

詳細は、[init\(1M\)](#) の「FILES」セクションを参照してください。

ipsec

この機能では、IKE デーモンのデバッグ情報や `ikeadm` の特権レベルなどのパラメータを構成できます。

kbd

キーボード構成プロパティは次の SMF サービスに含まれるようになりました。

```
svc:/system/keymap:default
```

同様の構文を使用して、キーボードプロパティを表示し、構成します。

```
# svccfg -s svc:/system/keymap:default
svc:/system/keymap:default> setprop
Usage: setprop pg/name = [type:] value
setprop pg/name = [type:] ([value...])
```

Set the pg/name property of the currently selected entity. Values may be enclosed in double-quotes. Value lists may span multiple lines.

```
svc:/system/keymap:default> listprop
general                framework
general/complete      astring
general/enabled        boolean    false
keymap                  system
keymap/console_beeper_freq integer    900
keymap/kbd_beeper_freq integer    2000
keymap/keyboard_abort  astring   enable
keymap/keyclick        boolean   false
.
.
.
```

詳細は、[kbd\(1\)](#) を参照してください。

keyserv

詳細は、[keyserv\(1M\)](#) の「FILES」セクションにある `/etc/default/keyserv` の情報を参照してください。

login

詳細は、[login\(1\)](#) の「FILES」セクションにある `/etc/default/login` の情報を参照してください。

mpathd

この機能を利用して、`in.mpathd` 構成パラメータを設定できます。

詳細は、[in.mpathd\(1M\)](#) を参照してください。

nfs

`sharectl` コマンドを使用して、SMF NFS プロパティーを表示または構成できます。例:

```
# sharectl get nfs
servers=1024
lockd_listen_backlog=32
lockd_servers=1024
lockd_retransmit_timeout=5
grace_period=90
server_versmin=2
server_versmax=4
client_versmin=2
client_versmax=4
server_delegation=on
nfsmapid_domain=
# sharectl set -p grace_period=60 nfs
```

詳細は、[nfs\(4\)](#) を参照してください。

nfslogd

詳細は、[nfslogd\(1M\)](#) の「Description」セクションを参照してください。

nss

この機能を利用して、`initgroups(3C)` 参照パラメータを構成できます。

詳細は、[nss\(4\)](#) を参照してください。

passwd

詳細は、[passwd\(1\)](#) の「FILES」セクションにある `/etc/default/passwd` の情報を参照してください。

su

詳細は、[su\(1M\)](#) の「FILES」セクションにある `/etc/default/su` の情報を参照してください。

syslog

詳細は、[syslogd\(1M\)](#) の「FILES」セクションにある `/etc/default/syslogd` の情報を参照してください。

tar

`-f` 関数修飾子の詳細は、[tar\(1\)](#) を参照してください。

TAPE 環境変数がなく、いずれかの引数が数字で、かつ `-f` が指定されていない場合、`archiveN` 文字列と一致する数値が `/etc/default/tar` ファイルで検索されます。このファイルの `archiveN` 文字列の値は、出力デバイス、ブロック化因数、サイズとして使用されます。

例:

```
% tar -c 2 /tmp/*
```

このコマンドは、`/etc/default/tar` ファイルで `archive2` と指定されているデバイスに出力を書き込みます。

telnetd

このファイルは、Telnet 接続の際に表示されるデフォルトのバナー (BANNER) を識別します。

utmpd

utmpd デーモンは /var/adm/utmpx (Solaris の以前のバージョンでは /var/adm/utmp) をモニターし、pututxline(3C) で root 以外のプロセスによって挿入された utmp エントリがプロセスの終了前に確実にクリーンアップされるようにします。

/etc/default/utmpd の次の 2 つのエントリがサポートされています。

- SCAN_PERIOD - utmpd が /proc をチェックしてモニター対象のプロセスが生きているかどうかを確認する間隔 (スリープ秒数)。デフォルト値は 300 です。
- MAX_FDS - utmpd がモニターしようとするプロセスの最大数。デフォルト値は 4096 です。通常、この値を変更する必要はありません。

◆◆◆ 付録 A

システムチェックスクリプト

システムのフラッシュ動作の確認

このスクリプトを使用すると、ZFS とフラッシュストレージをチューニングしたあと、システムでフラッシュが正しく動作していることを容易に確認できます。詳細は、[92 ページの「フラッシュおよび NVRAM ストレージデバイスの適切なキャッシュフラッシュ動作の保証」](#)を参照してください。指示された手順が完了したら、次のスクリプトを実行します。

```
#!/bin/ksh
#
#cd /dev/rdisk
#for d in *d0; do
# /export/home/admin1/bin/sdflush.sh $d
#done
#
#
if [[ $# -ne 1 ]]; then
    echo "Usage: $0 cctx..."
    exit 1;
fi

sd=`iostat -x $1 2>&1 | grep sd | nawk '{print $1}' | sed s/sd/^`
printf "Value for %s : " $1
echo '*sd_state::softstate 0t'$sd' | ::print struct sd_lun un_phy_blocksize' \
    | mdb -k

#echo '*sd_state::softstate 0t'$sd' | ::print struct sd_lun un_f_suppress_cache_flush' \
#echo '*sd_state::softstate 0t'$sd' | ::print struct sd_lun un_phy_blocksize' \
```


索引

数字・記号

`_addip_enabled`, 175
`_addrs_per_if`, 139
`_arp_defend_interval`, 140
`_arp_defend_period`, 141
`_arp_defend_rate`, 141
`_arp_fastprobe_count`, 142
`_arp_fastprobe_interval`, 142
`_arp_probe_count`, 142
`_arp_probe_interval`, 143
`_arp_publish_count`, 143
`_arp_publish_interval`, 144
`_conn_req_max_q`, 155
`_conn_req_max_q0`, 156
`_conn_req_min`, 156
`_cookie_life`, 173
`_cwnd_max`, 151, 167
`_defend_interval`, 144
`_deferred_ack_interval`, 147, 169
`_deferred_acks_max`, 148
`_dup_recovery`, 145
`_heartbeat_interval`, 168
`_icmp_err_burst`, 136
`_icmp_err_interval`, 136
`_icmp_return_data_bytes`, 146
`_ignore_path_mtu`, 170
`_initial_mtu`, 169
`_initial_out_streams`, 174
`_initial_ssthresh`, 170
`_ip_abort_interval`, 160
`_ipv4_ttl`, 167
`_ipv6_hoplimit`, 168
`_keepalive_interval`, 159
`_local_dack_interval`, 147
`_local_dacks_max`, 148
`_local_slow_start_initial`, 152
`_max_defend`, 145
`_max_in_streams`, 173
`_max_init_retr`, 166
`_max_temp_defend`, 145
`_ndp_defend_interval`, 140
`_ndp_defend_period`, 141
`_ndp_defend_rate`, 141
`_ndp_unsolicit_count`, 143
`_ndp_unsolicit_interval`, 144
`_new_secret_interval`, 169
`_pathmtu_interval`, 146
`_policy_mask`, 165
`_pp_max_retr`, 167
`_prsrctp_enabled`, 175
`_recv_hiwat_minmss`, 162
`_respond_to_echo_broadcast`, 137
`_respond_to_echo_multicast`, 137
`_rev_src_routes`, 153
`_rexmit_interval_extra`, 161
`_rexmit_interval_initial`, 160
`_rexmit_interval_max`, 160
`_rexmit_interval_min`, 161
`_rst_sent_rate`, 157
`_rst_sent_rate_enabled`, 157
`_rto_max`, 172, 173
`_rto_min`, 172
`_shutack_wait_bound`, 174
`_slow_start_after_idle`, 152
`_slow_start_initial`, 152
`_time_wait_interval`, 154
`_tstamp_always`, 149
`_tstamp_if_wscale`, 162

_wscale_always, 149
_xmit_lowat, 171

A

autofs, 179
autoup, 29

C

cron, 180

D

ddi_msix_alloc_limit パラメータ, 54
default_stksize, 23
default_tsb_size, 79
desfree, 39
dhcpageant, 180
disp_rechoose_interval, 76
dnlc_dir_enable, 65
dnlc_dir_max_size, 66
dnlc_dir_min_size, 65
dnlc_dircache_percent, 66
doiflush, 30
dopageflush, 30

E

ecn, 155
enable_tsb_rss_sizing, 80

F

fastscan, 44
forwarding, 138
fs, 180
fsflush, 27
ftp, 181

H

handspreadpages, 46

hires_tick, 77
hoplimit (ipv6), 138
hostmodel, 139

I

inetinit, 181
init, 181
intr_force, 57
intr_throttling, 58
ip_queue_fanout, 56
ip_queue_worker_wait, 55
ipcl_conn_hash_size, 55
ipsec, 182

K

kbd, 182
keyserv, 182
kmem_flags, 50
kmem_stackinfo, 51

L

largest_anon_port, 158, 165, 176
lgrp_mem_pset_aware, 82
logevent_max_q_sz, 25
login, 183
lotsfree, 38
lpg_alloc_prefer, 81
lwp_default_stksize, 24

M

max_buf (SCTP), 172
max_buf (TCP), 151
max_buf (UDP), 164
max_nprocs, 34
maxpgio, 47
maxphys, 61
maxpid, 33
maxuprc, 35
maxusers, 31

min_percent_cpu, 45
 minfree, 40
 moddebug, 52
 mpathd, 183
 mr_enable, 56

N

ncsize, 64
 nfs_max_threads, 110
 nfs:nacache, 124
 nfs:nfs3_async_clusters, 122
 nfs:nfs3_bsize, 119
 nfs:nfs3_cots_timeo, 104
 nfs:nfs3_do_symlink_cache, 106
 nfs:nfs3_dynamic, 107
 nfs:nfs3_jukebox_delay, 125
 nfs:nfs3_lookup_neg_cache, 108
 nfs:nfs3_max_threads, 111
 nfs:nfs3_max_transfer_size, 125
 nfs:nfs3_max_transfer_size_clts, 127
 nfs:nfs3_max_transfer_size_cots, 128
 nfs:nfs3_nra, 113
 nfs:nfs3_pathconf_disable_cache, 102
 nfs:nfs3_shrinkreaddir, 116
 nfs:nfs4_async_clusters, 122
 nfs:nfs4_bsize, 120
 nfs:nfs4_cots_timeo, 104
 nfs:nfs4_lookup_neg_cache, 109
 nfs:nfs4_max_threads, 112
 nfs:nfs4_max_transfer_size, 126
 nfs:nfs_allow_preepoch_time, 102
 nfs:nfs_async_clusters, 121
 nfs:nfs_async_timeout, 123
 nfs:nfs_cots_timeo, 103
 nfs:nfs_disable_rmdir_cache, 118
 nfs:nfs_do_symlink_cache, 105
 nfs:nfs_dynamic, 106
 nfs:nfs_lookup_neg_cache, 108
 nfs:nfs_nra, 113
 nfs:nfs_shrinkreaddir, 115
 nfs:nfs_write_error_interval, 117
 nfs:nfs_write_error_to_cons_only, 117

nfs:nrnode, 114
 nfslogd, 183
 ngroups_max, 35
 noexec_user_stack, 26
 nss, 184
 nstrpush, 71

O

Oracle データベースチューニング
 ZFS ファイルシステム, 95

P

pageout_reserve, 41
 pages_before_pager, 46
 pages_pp_maximum, 42
 passwd, 184
 physmem, 22
 pidmax, 33
 pr_seg disable, 74
 primarycache
 ZFS ファイルシステムプロパティ, 95
 pt_cnt, 69
 pt_max_pty, 71
 pt_pctofmem, 70

R

recordsize
 ZFS ファイルシステムプロパティ, 95
 recv_buf (SCTP), 171
 recv_buf (TCP), 150
 recv_buf (UDP), 163
 reserved_procs, 33
 rlim_fd_cur, 63
 rlim_fd_max, 62
 rpcmod:clnt_idle_timeout, 130
 rpcmod:clnt_max_conns, 130
 rpcmod:cotsmaxdupreqs, 133
 rpcmod:maxdupreqs, 132
 rpcmod:svc_default_stksize, 131
 rpcmod:svc_idle_timeout, 131

rx_copy_threshold, 61
rx_limit_per_intr, 59
rx_queue_number, 58
rx_ring_size, 60

S

sack, 153
sctp_maxburst, 175
secondarycache
 ZFS ファイルシステムプロパティ, 95
segspt_minfree, 74
send_buf (SCTP), 170
send_buf (TCP), 150
send_buf (UDP), 163
send_redirects, 137
slowscan, 45
smallest_anon_port, 158, 164, 176
strmsgsz, 72, 72
su, 184
swapfs_minfree, 49
swapfs_reserve, 48
syslog, 184

T

tar, 184
throttlefree, 41
timer_max, 78
tmpfs_maxkmem, 67
tmpfs_minfree, 68
tsb_alloc_hiwater, 78
tsb_rss_factor, 80
ttl (ipv4), 138
tune_t_fsflushr, 28
tune_t_minarmem, 43
tx_copy_threshold, 60
tx_queue_number, 57
tx_ring_size, 59

U

utmpd, 185

Z

zfs_arc_max, 87
zfs_arc_min, 86
zfs_prefetch_disable, 88
ZFS ファイルシステム
 Oracle データベース用のチューニング, 95
ZFS ファイルシステムプロパティ
 primarycache, 95
 recordsize, 95
 secondarycache, 95