

Oracle® Solaris 11.1 カーネルのチューン アップ・リファレンスマニュアル

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

はじめに	13
1 Oracle Solaris システムのチューニングの概要	17
Oracle Solaris システムチューニングの新機能	17
Oracle Solaris システムのチューニング	19
チューニング可能パラメータの説明形式	20
Oracle Solaris カーネルのチューニング	22
/etc/system ファイル	22
kldb コマンド	23
mdb コマンド	23
Oracle Solaris の特殊な tune および var 構造体	24
Oracle Solaris システム構成情報の表示	25
sysdef コマンド	25
kstat ユーティリティ	25
2 Oracle Solaris カーネルチューニング可能パラメータ	27
チューニング可能パラメータ情報の記載箇所	28
カーネルとメモリーの一般的なパラメータ	28
physmem	28
default_stksize	29
lwp_default_stksize	30
logevent_max_q_sz	31
segkpsize	32
noexec_user_stack	33
fsflush とそれに関連するパラメータ	33
fsflush	34
tune_t_fsflushr	34

autoup	35
dopageflush	36
doiflush	37
プロセス規模調整パラメータ	38
maxusers	38
reserved_procs	39
pidmax	40
max_nprocs	41
maxuprc	42
ngroups_max	42
ページング関連パラメータ	43
lotsfree	45
desfree	46
minfree	47
throttlefree	49
pageout_reserve	50
pages_pp_maximum	51
tune_t_minarmem	52
fastscan	52
slowscan	53
min_percent_cpu	54
handspreadpages	54
pages_before_pager	55
maxpgio	56
スワッピング関連パラメータ	57
swapfs_reserve	57
swapfs_minfree	58
カーネルメモリアロケータ	59
kmem_flags	59
kmem_stackinfo	61
一般的なドライバパラメータ	62
moddebug	62
ddi_msix_alloc_limit	64
ネットワークドライバパラメータ	65
igb パラメータ	65
ixgbe パラメータ	66

一般的な入出力パラメータ	70
maxphys	70
rlim_fd_max	71
rlim_fd_cur	72
一般的なファイルシステムパラメータ	73
ncsize	73
dnlc_dir_enable	74
dnlc_dir_min_size	74
dnlc_dir_max_size	75
dnlc_dircache_percent	75
TMPFS パラメータ	76
tmpfs:tmpfs_maxkmem	76
tmpfs:tmpfs_minfree	77
仮想端末	78
pt_cnt	79
pt_pctofmem	79
pt_max_pty	80
STREAMS パラメータ	81
nstrpush	81
strmsgsz	81
strctlsz	82
System V メッセージキュー	82
System V セマフォ	83
System V 共有メモリー	83
segspt_minfree	83
スケジューリング	84
disp_rechoose_interval	84
タイマー	86
hires_tick	86
timer_max	86
SPARC システム固有のパラメータ	87
tsb_alloc_hiwater_factor	87
default_tsb_size	88
enable_tsb_rss_sizing	88
tsb_rss_factor	89
近傍性グループのパラメータ	90

lpg_alloc_prefer	90
lgrp_mem_pset_aware	91
3 Oracle Solaris ZFS チューニング可能パラメータ	93
チューニング可能パラメータ情報の記載箇所	93
ZFS のチューニングの考慮事項	94
ZFS ARC パラメータ	94
zfs_arc_min	94
zfs_arc_max	95
ZFS ファイルレベルプリフェッチ	96
zfs_prefetch_disable	96
ZFS デバイスの入出力キューの深さ	97
zfs_vdev_max_pending	97
フラッシュストレージ使用時の ZFS のチューニング	98
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	99
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	101
データベース製品に対する ZFS のチューニング	103
Oracle データベース用の ZFS のチューニング	104
ZFS を MySQL と一緒に使用するときの考慮事項	108
4 NFS チューニング可能パラメータ	109
チューニング可能パラメータ情報の記載箇所	109
NFS 環境のチューニング	109
NFS モジュールのパラメータ	110
nfs:nfs3_pathconf_disable_cache	110
nfs:nfs4_pathconf_disable_cache	110
nfs:nfs_allow_preepoch_time	111
nfs:nfs_cots_timeo	112
nfs:nfs3_cots_timeo	113
nfs:nfs4_cots_timeo	114
nfs:nfs_do_symlink_cache	115
nfs:nfs3_do_symlink_cache	115
nfs:nfs4_do_symlink_cache	116

nfs:nfs_dynamic	117
nfs:nfs3_dynamic	117
nfs:nfs_lookup_neg_cache	118
nfs:nfs3_lookup_neg_cache	119
nfs:nfs4_lookup_neg_cache	120
nfs:nfs_max_threads	121
nfs:nfs3_max_threads	122
nfs:nfs4_max_threads	123
nfs:nfs_nra	124
nfs:nfs3_nra	125
nfs:nfs4_nra	126
nfs:nrnode	127
nfs:nfs_shrinkreaddir	128
nfs:nfs3_shrinkreaddir	129
nfs:nfs_write_error_interval	130
nfs:nfs_write_error_to_cons_only	131
nfs:nfs_disable_rmdir_cache	131
nfs:nfs3_bsize	132
nfs:nfs4_bsize	133
nfs:nfs_async_clusters	134
nfs:nfs3_async_clusters	135
nfs:nfs4_async_clusters	136
nfs:nfs_async_timeout	137
nfs:nacache	138
nfs:nfs3_jukebox_delay	139
nfs:nfs3_max_transfer_size	140
nfs:nfs4_max_transfer_size	141
nfs:nfs3_max_transfer_size_clts	142
nfs:nfs3_max_transfer_size_cots	143
rpcmod モジュールのパラメータ	144
rpcmod:clnt_max_conns	144
rpcmod:clnt_idle_timeout	144
rpcmod:svc_idle_timeout	145
rpcmod:svc_default_stksize	145
rpcmod:maxdupreqs	146
rpcmod:cotsmaxdupreqs	147

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	149
	チューニング可能パラメータ情報の記載箇所	149
	IPパラメータのチューニングの概要	149
	IPパラメータの妥当性検証	150
	RFC (Internet Request for Comments)	150
	IPチューニング可能パラメータ	151
	_icmp_err_intervalと_icmp_err_burst	151
	_respond_to_echo_broadcastと_respond_to_echo_multicast (ipv4 または ipv6)	151
	_send_redirects (ipv4 または ipv6)	152
	forwarding (ipv4 または ipv6)	152
	ttl	153
	hoplimit (ipv6)	153
	_addr_per_if	153
	hostmodel (ipv4 または ipv6)	154
	ip_queue_fanout	155
	特別な注意を要するIPチューニング可能パラメータ	156
	TCPチューニング可能パラメータ	157
	_deferred_ack_interval	157
	_local_dack_interval	158
	_deferred_acks_max	158
	_local_dacks_max	159
	_wscale_always	159
	_tstamp_always	160
	send_buf	161
	recv_buf	161
	max_buf	162
	_cwnd_max	162
	_slow_start_initial	163
	_local_slow_start_initial	163
	_slow_start_after_idle	163
	sack	164
	_rev_src_routes	165
	_time_wait_interval	165
	ecn	166
	_conn_req_max_q	167

_conn_req_max_q0	168
_conn_req_min	169
_rst_sent_rate_enabled	169
_rst_sent_rate	170
/etc/system ファイルに設定する TCP/IP パラメータ	171
特別な注意を要する TCP パラメータ	172
UDP チューニング可能パラメータ	177
send_buf	177
recv_buf	178
max_buf	178
smallest_anon_port	179
largest_anon_port	179
IPQoS チューニング可能パラメータ	180
_policy_mask	180
SCTP チューニング可能パラメータ	181
_max_init_retr	181
_pa_max_retr	181
_pp_max_retr	182
_cwnd_max	182
_ipv4_ttl	183
_heartbeat_interval	183
_new_secret_interval	184
_initial_mtu	184
_deferred_ack_interval	184
_ignore_path_mtu	185
_initial_ssthresh	185
send_buf	186
_xmit_lowat	186
recv_buf	186
max_buf	187
_rto_min	187
_rto_max	188
_rto_initial	188
_cookie_life	188
_max_in_streams	189
_initial_out_streams	189

_shutack_wait_bound	190
_maxburst	190
_addip_enabled	190
_prsctp_enabled	191
smallest_anon_port	191
largest_anon_port	192
ルート別のメトリック	192
6 システム機能のパラメータ	195
システムのデフォルトのパラメータ	196
autofs	196
cron	196
devfsadm	196
dhcpgent	196
fs	197
ftp	197
inetinit	197
init	197
ipsec	198
kbd	198
keyserv	198
login	199
mpathd	199
nfs	199
nfslogd	199
nss	199
passwd	199
su	200
syslog	200
tar	200
telnetd	200
utmpd	200
A チューニング可能パラメータの変更履歴	201
カーネルパラメータ	201

カーネルとメモリーの一般的なパラメータ (Oracle Solaris 11 および 11.1)	201
ページング関連パラメータ (Oracle Solaris 11)	202
プロセスサイジングのチューニング可能パラメータ (Oracle Solaris 11.1)	202
スワッピング関連パラメータ (Oracle Solaris 11.1)	203
一般的なファイルシステムパラメータ (Oracle Solaris 11.1)	203
一般的なドライバパラメータ (Oracle Solaris 11)	203
ネットワークドライバパラメータ (Oracle Solaris 11)	203
TCP/IP チューニング可能パラメータ (Oracle Solaris 11 および 11.1)	204
[tcp,sctp,udp]_smallest_anon_port および	
[tcp,sctp,udp]_largest_anon_port	204
_local_slow_start_initial	204
IP パラメータの名前の変更 (Oracle Solaris 11)	204
TCP パラメータの名前の変更 (Oracle Solaris 11)	205
UDP パラメータの名前の変更 (Oracle Solaris 11)	207
SCTP パラメータの名前の変更 (Oracle Solaris 11)	207
廃止または削除されたパラメータ (Oracle Solaris 11 および 11.1)	208
NCA パラメータ	209
consistent_coloring	209
rstchown	209
廃止された TCP/IP モジュールのパラメータ (Oracle Solaris 11)	210
B このマニュアルの改訂履歴	211
最新バージョン: Oracle Solaris 11.1 リリース	211
Oracle Solaris リリースで追加または変更されたパラメータ	211
索引	213

はじめに

『Oracle Solaris 11.1 カーネルのチューンアップ・リファレンスマニュアル』では、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。

このマニュアルには、SPARC システムを対象とする情報と x86 システムを対象とする情報が含まれています。

注 - この Oracle Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャをサポートしています。サポートされるシステムは、<http://www.oracle.com/webfolder/technetwork/hcl/index.html> の『Oracle Solaris Hardware Compatibility List』に記載されています。このドキュメントでは、プラットフォームにより実装が異なる場合は、それを特記します。

対象読者

このドキュメントは、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、熟練した Oracle Solaris システム管理者を対象としています。Oracle Solaris チューニング可能パラメータの変更に関するガイドラインについては、19 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。

内容の紹介

このドキュメントは、次の各章と付録から構成されています。

章	説明
第1章「Oracle Solaris システムのチューニングの概要」	Oracle Solaris システムのチューニングの概要。このドキュメントで使用するカーネルのチューニング可能パラメータの書式の説明も含む
第2章「Oracle Solaris カーネルチューニング可能パラメータ」	Oracle Solaris カーネルのチューニング可能パラメータの説明 (カーネルメモリー、ファイルシステム、プロセスサイズ、ページングのパラメータなど)

章	説明
第3章「Oracle Solaris ZFS チューニング可能パラメータ」	Oracle Solaris ZFS のチューニング可能パラメータについての説明。この章には、データベース製品のための ZFS チューニング情報も含まれています。
第4章「NFS 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の説明 (シンボリックリンクのキャッシュや、動的再転送、RPC セキュリティーのパラメータなど)
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」	TCP/IP のチューニング可能パラメータの説明 (IP 転送やソースルーティング、バッファサイジングのパラメータなど)
第6章「システム機能のパラメータ」	特定のシステム機能のデフォルト値の設定パラメータの説明。変更するには、 <code>/etc/default</code> ディレクトリ内のファイルを編集する
付録 A 「チューニング可能パラメータの変更履歴」	変更または廃止されたパラメータの履歴
付録 B 「このマニュアルの改訂履歴」	このマニュアルの改訂履歴 (Oracle Solaris 最新リリースを含む)

Oracle Solaris のチューニングに関するほかの情報源

この表に、Oracle Solaris のチューニングに関するほかの情報源を示します。

チューニングに関する情報源	参照先
詳細な技術ホワイトペーパー	http://www.oracle.com/technetwork/server-storage/solaris/overview/index.html

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> を参照してください。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。

表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表 P-1 表記上の規則

字体	説明	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上的コンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上的コンピュータ出力と区別して示します。	machine_name% su Password:
<i>aabbcc123</i>	プレースホルダ: 実際に使用する特定の名称または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
AaBbCc123	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第6章を参照してください。 キャッシュは、ローカルに格納されるコピーです。 ファイルを保存しないでください。 注: いくつかの強調された項目は、オンラインでは太字で表示されます。

コマンド例のシェルプロンプト

次の表に、Oracle Solaris OS に含まれるシェルの UNIX システムプロンプトおよびスーパーユーザーのプロンプトを示します。コマンド例のシェルプロンプトは、そのコマンドを標準ユーザーで実行すべきか特権ユーザーで実行すべきかを示します。

表 P-2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	#

表 P-2 シェルプロンプト (続き)

シェル	プロンプト
C シェル	machine_name%
C シェルのスーパーユーザー	machine_name#

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。

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

Oracle Solaris 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 11.1 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。

- Oracle Solaris 11.1: Oracle Solaris ZFS チューニング可能パラメータ情報は、第 3 章「Oracle Solaris ZFS チューニング可能パラメータ」にあります。
- Oracle Solaris 11.1: `maxusers`、`max_nprocs`、`ngroups_max`、`pidmax`、および `segkpsize` パラメータが改訂されました。詳細は、第 2 章「Oracle Solaris カーネルチューニング可能パラメータ」を参照してください。
- Oracle Solaris 11.1: `dnlc_dircache_percent` パラメータは新規パラメータです。詳細は、75 ページの「`dnlc_dircache_percent`」を参照してください。
- Oracle Solaris 11: 以前 `/etc/system` ファイルで設定していた `rstchown` パラメータは廃止されています。このパラメータを `/etc/system` ファイルで設定すると、次のエラーメッセージが表示されます。

```
sorry, variable 'rstchown' is not defined in the 'kernel'
```

このパラメータは ZFS `rstchown` ファイルシステムプロパティおよび一般ファイルシステムのマウントオプションに置き換えられました。詳細は、『[Oracle Solaris 11.1 の管理: ZFS ファイルシステム](#)』および `mount(1M)` を参照してください。

- Oracle Solaris 11: 以前に `/etc/default` ディレクトリのファイルを編集することによって構成していた次のシステム構成パラメータが、SMF サービスに変更されました。

- `autofs`
- `init`
- `kbd`
- `nfs`

SMF サービスプロパティの変更については、[第 6 章「システム機能のパラメータ」](#) を参照してください。

- Oracle Solaris 11: ネットワークプロパティの設定用に、`ipadm` コマンドが、`ndd` コマンドに置き換まりました。TCP、IP、UDP および SCTP プロパティは次のように設定します。

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

さらに、ほとんどのネットワークのチューニング可能パラメータの名前が、`ipadm` 形式に密接に関連するように、若干変更されました。

詳細は、[149 ページの「IP パラメータのチューニングの概要」](#) を参照してください。

- Oracle Solaris 11: このリリースには `disp_rechoose_interval` パラメータが含まれません。詳細は、[84 ページの「`disp_rechoose_interval`」](#) を参照してください。
- Oracle Solaris 11: このリリースには `ngroups_max` パラメータに関する記述が含まれません。詳細は、[42 ページの「`ngroups_max`」](#) を参照してください。
- Oracle Solaris 11: このリリースには、`zfs_arc_min` および `zfs_arc_max` パラメータに関する記述が含まれます。詳細は、[94 ページの「`zfs_arc_min`」](#) および [95 ページの「`zfs_arc_max`」](#) を参照してください。
- Oracle Solaris 11: このリリースには、いくつかの `igb` および `ixgbe` ネットワークドライバパラメータが含まれます。詳細は、[65 ページの「`igb` パラメータ」](#) および [66 ページの「`ixgbe` パラメータ」](#) を参照してください。
- Oracle Solaris 11: このリリースには `ddi_msix_alloc_limit` パラメータが含まれません。このパラメータは、デバイスインスタンスが割り当てることができる MSI-X 割り込みの数を増やすために使用できます。詳細は、[64 ページの「`ddi_msix_alloc_limit`」](#) を参照してください。
- Oracle Solaris 11: このリリースには `kmem_stackinfo` パラメータが含まれます。このパラメータを有効にして、カーネルスレッドスタックの使用状況を監視できます。詳細は、[61 ページの「`kmem_stackinfo`」](#) を参照してください。

- Oracle Solaris 11: このリリースでは、メモリー近傍性グループのパラメータが提供されます。これらのパラメータについての詳細は、90 ページの「[近傍性グループのパラメータ](#)」を参照してください。

Oracle Solaris システムのチューニング

Oracle Solaris OS は、SPARC および x86 プロセッサで動作する、マルチスレッドでスケラブルな UNIX オペレーティングシステムです。Solaris は、システムの負荷に自動的に対応するため、最小限のチューニングしか必要ありません。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カーネルのチューニングオプションについて詳しく説明します。

Solaris カーネルは、常にロードされているコア部分と、参照が発生するとロードされるロード可能モジュールから構成されています。このガイドのカーネルに関する部分で参照されている変数の多くは、コア部分にあります。しかし、ロード可能モジュールの変数もいくつかあります。

システムのチューニングをする際に考慮しなければならないのは、さまざまなシステムパラメータ(またはシステム変数)を設定する行為は、処理効率を高めるという目的にとって、多くの場合、一番効率の良くない行為だということです。もっとも効果的なチューニング方法は、通常、アプリケーションの動作を変更することです。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も効果があります。このガイドに記載された変数の値を1つ変更しただけで、システムパフォーマンスに意味のある影響が現れることは、ごく限られた場合にしか起きません。

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。このドキュメントで述べ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 の新しいリリースに移行する場合は、空の `/etc/system` ファイルで開始することをお勧めします。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

各チューニング可能パラメータの説明形式は、次のとおりです。

- パラメータ名
- 説明
- データ型
- デフォルト
- 範囲
- 単位
- 動的か
- 検証
- 暗黙的制約
- どのような場合に変更するか
- ゾーン構成
- コミットレベル
- 変更履歴

パラメータ名

/etc/system ファイルに入力するか、または /etc/default/facility ファイルに指定されているとりの名前。

ほとんどのパラメータ名は、コロン (:) を伴わない `parameter` の形式をとります。このような名前は、カーネルのコア部分内の変数を表しています。名前にコロンが含まれている場合、コロンの左側の文字列はロード可能モジュールの名前を表し、コロンの右側の文字列はモジュール内の変数名を表します。コロンの右側の文字列はモジュール内のパラメータ名を示します。例:

`module_name:variable`

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値として使用する値です。
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	動作中のシステムで mdb、または kmdb デバッガを使用してパラメータを変更できる場合は「はい」です。パラメータがブート時の初期化だけの場合は「いいえ」です。
検証	システムが、/etc/system 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せん。
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) のマニュアルページを参照してください。
変更履歴	(オプション) 存在する場合は、付録 A または付録 B の「変更履歴」への参照が示されます。

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 ファイルの変更	22 ページの「/etc/system ファイル」
カーネルデバッガ (kldb) の使用	23 ページの「kldb コマンド」
モジュールデバッガ (mdb) の使用	23 ページの「mdb コマンド」
ipadm コマンドによる TCP/IP パラメータの設定	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 下のファイルの変更	第 6 章「システム機能のパラメータ」

/etc/system ファイル

/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。

例 - /etc/system でのパラメータの設定

次の /etc/system 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 = 0x780000000
```

適正でない値からの復元

値を変更する前に /etc/system ファイルのコピーを作成しておけば、不正な値を簡単に元の値に戻せます。例:

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

/etc/system 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い状態になった場合は、次のコマンドでブートします。

```
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後、`/etc/system` ファイルを修正できません。

システムの回復の詳細は、『[Oracle Solaris の管理: 一般的なタスク](#)』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\(1\)](#) のマニュアルページ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\(1\)](#) のマニュアルページを参照してください。

例 – `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
```

モジュラーデバッガの使用についての詳細は、『[Oracle Solaris Modular Debugger Guide](#)』を参照してください。

kldb デバッガまたは mdb デバッガを使用する場合、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後は、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。動作中のシステム上で `mdb` を使うか、あるいは `kadb` でブートし、カーネルデバッガですべての構造体やデータアイテムを調べることができます。

詳細は、`mdb(1)` または `kadb(1M)` のマニュアルページを参照してください。

sysdef コマンド

`sysdef` コマンドは、メモリー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 バイトのメモリーを備えた 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)` のマニュアルページを参照してください。

kstat ユーティリティ

`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ーを読んだり、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)` のマニュアルページ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

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。

- 28 ページの「カーネルとメモリーの一般的なパラメータ」
- 33 ページの「`fsflush` とそれに関連するパラメータ」
- 38 ページの「プロセス規模調整パラメータ」
- 43 ページの「ページング関連パラメータ」
- 57 ページの「スワッピング関連パラメータ」
- 59 ページの「カーネルメモリアロケータ」
- 62 ページの「一般的なドライバパラメータ」
- 65 ページの「ネットワークドライバパラメータ」
- 70 ページの「一般的な入出力パラメータ」
- 73 ページの「一般的なファイルシステムパラメータ」
- 76 ページの「TMPFS パラメータ」
- 78 ページの「仮想端末」
- 81 ページの「STREAMS パラメータ」
- 82 ページの「System V メッセージキュー」
- 83 ページの「System V セマフォ」
- 83 ページの「System V 共有メモリー」
- 84 ページの「スケジューリング」
- 86 ページの「タイマー」
- 87 ページの「SPARC システム固有のパラメータ」
- 90 ページの「近傍性グループのパラメータ」

チューニング可能パラメータ情報の記載箇所

チューニング可能パラメータ	参照先
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群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」

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は第3章「Oracle Solaris ZFS チューニング可能パラメータ」に移動しました。

physmem

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)などのメモリーは考慮されません。したがって、physmem の値は、より小さなメモリー量を表わすよう、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。30ページの「[lwp_default_stksize](#)」も参照してください。

データ型

整数

デフォルト

- SPARCシステムでは PAGESIZE の 3 倍。
- x64 システムでは PAGESIZE の 5 倍。

範囲

最小値はデフォルト値です。

- SPARCシステムでは PAGESIZE の 3 倍。
- x64 システムでは PAGESIZE の 5 倍。

最大値はデフォルト値の 32 倍です。

単位

getpagesize パラメータから戻った値の倍数になるバイト数です。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

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。

```
Illegal stack size, Using N
```

N の値は、default_stksize のデフォルト値です。

どのような場合に変更するか

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。

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。

データ型

整数

デフォルト

- 32,768 (SPARC プラットフォーム)
- 20,480 (x64 プラットフォーム)

範囲

最小値はデフォルト値です。

- SPARCシステムでは PAGESIZE の 3 倍。
- x64 システムでは PAGESIZE の 5 倍。

最大値はデフォルト値の 32 倍です。

単位

getpagesize パラメータから戻った値の倍数になるバイト数です。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

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。

```
Illegal stack size, Using N
```

どのような場合に変更するか	N の値は、 <code>lwp_default_stksize</code> のデフォルト値です。 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。
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> デー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、他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	整数
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、 <code>ddi_log_sysevent(9F)</code> および <code>sysevent_post_event(3SYSEVENT)</code> のマニュアルページ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。
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 (x86)
単位	ページ
動的か	いいえ
検証	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。 SPARC システムでは、segkpsize 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。x86 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。
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。2G バイトのデフォルトサイズでは、65,536 を超えるカーネルスレッド用に 24K バイトのスタックを作成できます。これより大きな数が必要な場合や、スタックサイズを増やすことが必要な場合は、物理メモリーが十分にあれば segkpsize を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

変更履歴 詳細は、202 ページの「[segkpsize](#)」を参照してください。

noexec_user_stack

説明	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。
	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。
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。値が設定された後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) のマニュアルページ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、34 ページの「`tune_t_fsflushr`」と 35 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	<code>autoup</code> 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	<code>autoup</code> は <code>tune_t_fsflushr</code> の整数倍でなければなりません。最小でも <code>autoup</code> は <code>tune_t_fsflushr</code> 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はいくつかあります。

- 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

説明	fsflush 呼び出しで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、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ので、同期化は <i>fsflush</i> が呼び出されるたびに行われます。 N は <i>fsflush</i> 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。その後で tune_t_fsflushr や autoup 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	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。 システムがリポートされる際に消えたり状態の一貫性がどうなっても構わないファイルは、TMPFS ファイルシステム (/tmp など) に置いた方がいいでしょう。システム上の i 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iノードの更新が行われません。

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を行い、一貫性を保つことを望むこともあるでしょう。

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nCPU x 8のどちらか大きい方

範囲

/etc/system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nCPU x 8のどちらか大きい方まで

単位	/etc/system ファイル内に設定されている場合、1 から 4096 と nCPU x 8 のどちらか大きい方まで ユーザー
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後、maxusers が再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。 out of processes 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、maxusers 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セクション約できます。 ■ 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。73 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
変更履歴	詳細は、202 ページの「maxusers」を参照してください。

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には使用されません。
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	<p>使用可能な最大プロセスIDの値を指定します。</p> <p>pidmaxでは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。</p> <p>/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。</p>
データ型	符号付き整数
デフォルト	3,000
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。 41 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
変更履歴	詳細は、202 ページの「pidmax」を参照してください。

max_nprocs

説明	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。 この値は、他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。 <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	maxusers が /etc/system ファイル内で設定されている場合は、 $10 + (16 \times \text{maxusers})$ maxusers が /etc/system ファイル内で設定されていない場合は、30,000 と $10 + (128 \times \text{CPU の数})$ のどちらか大きい方

範囲	26 から maxpid の値
動的か	いいえ
検証	はい。値が maxpid を超える場合、maxpid 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
変更履歴	詳細は、 202 ページ の「max_nprocs」を参照してください。

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	max_nprocs - reserved_procs
範囲	1 から max_nprocs - reserved_procs
単位	プロセス数
動的か	いいえ
検証	はい。この値は max_nprocs - reserved_procs 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が多くて)。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。 out of per-user processes for uid N
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
変更履歴	詳細は、202 ページの「ngroups_max」を参照してください。

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。システムの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、52 ページの「fastscan」および 53 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、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` の間のメモリー量の一部を調べ、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い場合は、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	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、 <code>getpagesize(3C)</code> を参照してください。 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。
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	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	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になりません。 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値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、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性があるかもしれません。 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、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

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	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。 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。
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	<code>desfree</code> が <code>lotsfree</code> より大きい場合、 <code>desfree</code> は <code>lotsfree/2</code> 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	<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。
副次的な影響	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。 <ul style="list-style-type: none"> ■ 使用可能なメモリーが <code>desfree</code> を超えない限り、非同期の入出力要求が処理されません。したがって、<code>desfree</code> の値を増やすと、増やす前か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が増えることがあります。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、 <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。
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	<code>desfree/2</code>
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、 <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。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、 <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	<code>minfree</code> が <code>desfree</code> より大きい場合、 <code>minfree</code> は <code>desfree/2</code> 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	<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	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、 <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。
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、 <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。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、 <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	<code>throttlefree</code> が <code>desfree</code> よりも大きい場合、 <code>throttlefree</code> は <code>minfree</code> に設定されません。メッセージは表示されません。
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	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、 <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、 <code>getpagesize(3C)</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

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	throttlefree / 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 throttlefree / 2 より大きい場合、pageout_reserve は throttlefree / 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 tune_t_minarmem+100 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	<p>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。</p> <p>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。詳細は、mlock(3C)、mlockall(3C)、および memcntl(2) のマニュアルページ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

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	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。 <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。

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。

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、 <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	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ことがある場合や、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>handsreadpages</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。このパラメータを変更する場合、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>handsreadpages</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>handsreadpages 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に指定します。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。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	40
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せん。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ー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ー不足の解消が早くなる場合があります。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せん。 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	システムの他の部分のために、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、 87 ページ の「 <code>tsb_alloc_hiwater_factor</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

変更履歴

詳細は、203 ページの「`swapfs_minfree`」を参照してください。

カーネルメモリーアロケータ

Oracle Solaris カーネルメモリーアロケータは、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。カーネルメモリーアロケータは、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。カーネルメモリーアロケータは、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 カーネルメモリーアロケータには、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、CONTENTS 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、そのバッファー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。バッファーの一部が変更されている場合は、そのバッファーを前に割り当て、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ー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、バッファーが解放されたら、パターンをチェックして、データがバッファー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、バッファーが解放されると、バッファーの内容を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。

フラグ	設定	説明
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 <code>kmem_stackinfo</code> 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。
----	---

	<ul style="list-style-type: none"> ■ 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監視する場合。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 Modular Debugger Guide 』の第5章「 Built-In Commands 」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。

- 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	2
範囲	1 から 16
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

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 パラメータ

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 <code>/etc/driver/drv/ixgbe.conf</code> 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 <code>/etc/driver/drv/ixgbe.conf</code> 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/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	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す。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の向上は望めません。
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 <code>libc(3LIB)</code> の <code>stdio(3C)</code> 関数を指します。 <code>select</code> はデフォルトで、<code>fd_set</code> につき 1024 の記述子に制限されます。詳細は、<code>select(3C)</code> 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 <code>fd_set</code> 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 <code>fd_set</code> サイズは 65,536 で、変更することはできません。

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。

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、自身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。それには、 `setrlimit()` 呼び出しを使用するか、プロセスを実行するシェルで `limit` コマンドを実行します。「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 `setrlimit` 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せん。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ことができます。 <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 - 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せん。または、元々有効だった場合に、無効にできません。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 - 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 - 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	整数
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。
<hr/>	
	注-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。
<hr/>	
コミットレベル	変更の可能性あり
変更履歴	詳細は、 203 ページ の「 <code>dnlc_dircache_percent</code> 」を参照してください。

TMPFS パラメータ

tmpfs:tmpfs_maxkmem

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	512
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。 fs-name: File system full, swap space limit exceeded
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で使用できるカーネルメモリーの割合 (%)。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限り、システムは pt_pctofmem に指定されたメモリー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に留意してください。

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。

システムメモリーの特定の割合を 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% を割り当てると、`pt_pctofmem=10`。

このメモリーは、`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、メモリー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ー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ーの割合 (`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 `maxpid`

単位

ログイン/ウィンドウ

動的吗

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 `/dev/pts` 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、`/dev/pts` 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 <code>EINVAL</code> 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 <code>write</code> は、複数のメッセージに分割されます。詳細は、 write(2) のマニュアルページ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) のマニュアルページ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、 putmsg(2) 呼び出しから ERANGE 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ついては、『[Oracle Solaris のシステム管理 \(Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理\)](#)』の第 6 章「リソース制御 (概要)」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、使用可能なリ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とき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

スケジューリング

disp_rechoose_interval

説明	以前の <code>rechoose_interval</code> 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 <code>rechoose_interval</code> パラメータの代わりに使用する必要がありますが、 <code>rechoose_interval</code> パラメータも <code>/etc/system</code> 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) のマニュアルページを参照してください。
コミットレベル	変更の可能性あり
変更履歴	詳細は、 202 ページ の「 <code>disp_rechoose_interval</code> 」を参照してください。

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せん。
コミットレベル	変更の可能性あり

SPARC システ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

tsb_alloc_hiwater を初期化して、変換ストレージバッファ (TSB) に割り当てることができる物理メモリー量に、次のように上限を設けます。

$$\text{tsb_alloc_hiwater} = \text{物理メモリー (バイト数)} / \text{tsb_alloc_hiwater_factor}$$

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マッピングされていないページとして TSB メモリーを再利用しようとします。

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。システム停止を防止するには、高位境界値が 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

整数

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 整数

デフォルト デフォルト値は0(8Kバイト)で、これは512エンタリに対応します。

範囲 指定可能な値は、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
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	整数
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、カーネルメモリーの使用量が増えます。

TSBの動きは、`trapstat -T` コマンドで監視できます。

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の誤動作は、 <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、 <code>psrset(1M)</code> 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せん。
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

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。

- 94 ページの「ZFS のチューニングの考慮事項」
- 94 ページの「ZFS ARC パラメータ」
- 96 ページの「ZFS ファイルレベルプリフェッチ」
- 97 ページの「ZFS デバイスの入出力キューの深さ」
- 98 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」
- 103 ページの「データベース製品に対する ZFS のチューニング」

チューニング可能パラメータ情報の記載箇所

チューニング可能パラメータ	参照先
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群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」

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。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 11.1の管理: ZFSファイルシステム』の第12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。

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。95 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 zfs_arc_max
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 zfs_arc_min 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
変更履歴	詳細は、201 ページの「zfs_arc_min」を参照してください。

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。94 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
変更履歴	詳細は、201 ページの「zfs_arc_max」を参照してください。

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。

- 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://developers.sun.com/prodtech/cc/articles/perftools.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	er_kernel の結果、zfetch_* 機能にかなりの時間がかかることが示されたり、または lockstat によるロックのプロファイリングで zfetch 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	整数
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 zfs_vdev_max_pending 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</p>

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。

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アレイでは、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、`iostat actv` キュー出力が `zfs_vdev_max_pending` の値に近づいていることとして示されます。

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アレイ上で 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 FCHBA (qlc) ドライバの制限は、`/kernel/drv/qlc.conf` の `execution-throttle` パラメータとして説明されています。

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、および F5100 フラッシュストレージアレイに適用されます。

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 および F20 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 および F40 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 (`O_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 デバイスが未対応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。

- F40 フラッシュデバイスの場合、`/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-non:true";
```

- F20 および F5100 フラッシュデバイスの場合、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。

- 多くの SPARC アーキテクチャー - /kernel/drv/ssd.conf に次のエントリを追加します。

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

- x64 およびいくつかの SPARC ドライバー - /kernel/drv/sd.conf に次のエントリを追加します。

```
ssd-config-list="ATA    MARVELL SD88SA02","throttle-max:32, disksort:false, cache-non: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。

フラッシュデバイスの SCSI マッピング解除に関する考慮事項

Solaris 11.1 OS には、SCSI マッピング解除ルーチンの過剰な呼び出しを発生させていた問題があります。この特定の問題は、フラッシュのパフォーマンスに悪影響を与えます。回避方法として、次のようにマッピング解除機能を無効にします。

- /etc/system ファイル内に次のエントリを含めます。
set zfs:zfs_unmap_ignore_size=0
- システムをリブートします。

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す。
- 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ーを犠牲にして 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 **Solaris** リリースを実行中であることを確認します
Solaris 10 9/10 リリースを最低限の開始点として、最新の Solaris 10 または 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。詳細については、『[Oracle Solaris 11.1 の管理: ZFS ファイルシステム](#)』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。

128K バイトのデフォルトレコードサイズを使用して、再実行、アーカイブ、取り消し、および一時データベースコンポーネント用のファイルシステムを別々に作成します。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 ファイルシステムのプロパティの詳細については、『[Oracle Solaris 11.1 の管理: ZFS ファイルシステム](#)』の「[ZFS のプロパティの紹介](#)」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
```

- デフォルトの `recordsize` および `primarycache` の値を使用して、一時表スペースおよび取り消し表スペース用のファイルシステムを作成します。

```
# 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
```

- 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。recordsize および primarycache には、デフォルトのファイルシステムの値を使用しません。

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

- アrchiveプール内に archivelog ファイル用のファイルシステムを作成し、圧縮を有効にして、recordsize にはデフォルト値を使用し、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

- ストレージレイ入出力キューのチューニングを検討します (HDS または EMC ストレージレイを持つシステムの場合)

ZFS は読み取りおよび書き込み入出力を集約し、デバイスを処理するドライバレベルにそれを送信する前に入出力の優先度を管理します。zfs_vdev_max_pending パラメータは、ZFS がすべてのストレージプールデバイスに送信する最大の入出力数を定義します。

レガシーのストレージ環境では、`ssd_max_throttle` および `sd_max_throttle` パラメータは、ドライバがストレージに送信できる最大の同時入出力数を定義します。`zfs_vdev_max_pending` のデフォルト値を `[s]sd_max_throttle` パラメータの値と同じに設定することで、ZFS が入出力を別の不要な SD 層のキューに入れることを回避します。

既存の環境の `/etc/system` ファイル内に `ssd:ssd_max_throttle` または `sd:ssd_max_throttle` がある場合、`zfs:zfs_vdev_max_pending` を同じ値に設定します。たとえば、ストレージアレイ管理者が、次の設定を要求したとします。

```
set ssd:ssd_max_throttle=20
```

この場合、このパラメータを次のように設定します。

```
set ssd:ssd_max_throttle=20
set zfs:zfs_vdev_max_pending=20
```

このパラメータを設定すると、ZFS が各 LUN のキューを制御できます。つまり、ストレージ内の保留中の入出力の合計数は、次のように拡張できます。

```
number of LUNs * ZFS_VDEV_MAX_PENDING
```

- 十分なメモリーリソースおよびスワップリソースを割り当てます

`zfs_arc_max` パラメータを低い値にチューニングすることによって ZFS のメモリー消費を削減できますが、データベースで活発に使用される部分についてはメタデータをキャッシュするために十分なメモリーをプロビジョニングすることを推奨します。これは 8K バイトの ZFS レコードでは 1.5% と推定され、大きさに比例して大きいレコードでは増加し、小さいレコードでは減少します。インデックスファイルを保持するファイルシステムは、メモリー不足の場合に最後に無効化されるため、ファイルシステムキャッシングによって最大のメリットを受けません。`zfs_arc_max` パラメータはバイト単位で、10 進数または 16 進数の値を受け入れます。次の例では、このパラメータを 2G バイトに設定します。

```
set zfs:zfs_arc_max=2147483648
or
set zfs:zfs_arc_max=0x80000000
```

メモリー不足が原因でアプリケーションが失敗することを防ぐために、ある程度のスワップ空間を構成する必要があります。システムメモリー全体と同等のスワップの量があれば、この目的のためには常に十分です。このスワップ空間は使用される予定はありませんが、予約領域として必要になります。スワップ空間を増やす方法については、『[Oracle Solaris 11.1 の管理: ZFS ファイルシステム](#)』の「ZFS スワップデバイスおよびダンプデバイスを管理する」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あるように、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。

- 109 ページの「NFS 環境のチューニング」
- 110 ページの「NFS モジュールのパラメータ」
- 144 ページの「rpcmod モジュールのパラメータ」

チューニング可能パラメータ情報の記載箇所

チューニング可能パラメータ	参照先
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群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」

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、19 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:nfs4_pathconf_disable_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</p>
----	---

	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せん。
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ため、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ため、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:nfs4_do_symlink_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、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監視し、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監視し、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、ロ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ロックアップ要求が、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。

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、[131 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: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ず。

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するようにします。詳細は、131 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、ロ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>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、131 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: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:nfs3_max_threads

説明	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。
	非同期に実行できる操作には、先読みのための読み取り、 <code>readdir</code> 先読みのための <code>readdir</code> 、 <code>putpage</code> および <code>pageio</code> 要求のための書き込み、およびコミットがあります。
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	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。
----	---

	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。
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: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が。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が。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: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 (132 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が。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:nfs4_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 4 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 (133 ページの「 <code>nfs:nfs4_bsize</code> 」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が。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が。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、最小限の `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 です。これは `nrnode` の値が `ncsize` 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、`nrnode` の値が正でないと、`nrnode` には `ncsize` が設定されません。

範囲

1 から $2^{31} - 1$

単位

`rnode`

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> の値を設定できることもあります。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となります。
コミットレベル	変更の可能性あり

nfs:nfs_shrinkreaddir

説明	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いことがありました。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。
----	---

	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 NFS_MAXDATA (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) のマニュアルページ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 バイトよりも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いことがありました。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 MAXBSIZE (8192 バイト) を使用して渡されるサ</p>
----	---

	イズのどちらか小さい方に設定されます。詳細は、 getdents(2) のマニュアルページ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 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ギン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	長整数 (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ギン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 <code>syslog</code> 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 <code>syslog</code>)
範囲	0 (システムコンソールと <code>syslog</code>) 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	<code>syslogd</code> 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、 <code>syslog</code> 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	<code>REaddir</code> 要求と <code>REaddirplus</code> 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	<p>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。</p> <p>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。</p> <p>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。</p> <ul style="list-style-type: none"> ■ 118 ページの「<code>nfs:nfs_lookup_neg_cache</code>」 ■ 119 ページの「<code>nfs:nfs3_lookup_neg_cache</code>」 ■ 120 ページの「<code>nfs:nfs4_lookup_neg_cache</code>」
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、その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: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、その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 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 <code>WRITE</code> 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: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 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</p>
----	--

	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: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 <code>NFS3ERR_JUKEBOX</code> 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。 <code>NFS3ERR_JUKEBOX</code> 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	長整数 (64 ビット)
デフォルト	1000 (10 秒を $10 \text{ 秒} * 100\text{Hz}$ 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 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 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</p>

どのような場合に変更するか	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。
	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。
	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、 132 ページ の「 <code>nfs:nfs3_bsize</code> 」を参照してください。
コミットレベル	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。 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	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。

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、[140 ページ](#)の「`nfs:nfs3_max_transfer_size`」を参照してください。

どのような場合に変更するか

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、`nfs:nfs4_bsize` 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。

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 `nfs:nfs4_bsize` 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、[133 ページ](#)の「`nfs:nfs4_bsize`」を参照してください。

転送サイズをデフォルト値より小さくする場合は、`mount` コマンドの `-wsize` または `-rsize` オプションをファイルシステム単位で使用します。

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 `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

検証	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。
どのような場合に変更するか	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。
コミットレベル	このパラメータは変更しないでください。 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

rpcmod モジュールのパラメータ

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	長整数 (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	長整数 (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に、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	<p>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。</p> <p>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。</p>
検証	なし
どのような場合に変更するか	<p>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。</p> <p>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す。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、情報を保持していなければなりません。一般に、コネクションレスト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。</p>
コミットレベル	変更の可能性あり

rpcmod:cotsmaxdupreqs

説明	<p>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。</p>
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	<p>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。</p> <p>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。</p>
どのような場合に変更するか	<p>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。</p> <p>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す。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。</p>
コミットレベル	変更の可能性あり

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。

- 151 ページの「IP チューニング可能パラメータ」
- 157 ページの「TCP チューニング可能パラメータ」
- 177 ページの「UDP チューニング可能パラメータ」
- 180 ページの「IPQoS チューニング可能パラメータ」
- 181 ページの「SCTP チューニング可能パラメータ」
- 192 ページの「ルート別のメトリック」

チューニング可能パラメータ情報の記載箇所

チューニング可能パラメータ	参照先
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」

IP パラメータのチューニングの概要

ipadm コマンドを使って、この章で説明するすべてのチューニングパラメータを設定できます。ただし、次に示すパラメータは例外です。

- 171 ページの「ipcl_conn_hash_size」
- 172 ページの「ip_queue_worker_wait」
- 155 ページの「ip_queue_fanout」

これらのパラメータは、`/etc/system` ファイル内にもみ設定できます。

`ipadm` コマンドを使って TCP/IP パラメータを設定するには、次の構文を使用します。

```
# 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。

`ndd` コマンドを使って TCP/IP パラメータを設定するには、次の構文を使用します。

```
# ndd -set driver parameter value
```

詳細は、[ndd\(1M\)](#) のマニュアルページ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、 <code>_icmp_err_interval</code> の間に最大で <code>ip_icmp_err_burst</code> の IP エラーメッセージを生成します。
	<code>_icmp_err_interval</code> 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	<code>_icmp_err_interval</code> は 100 ミリ秒 <code>_icmp_err_burst</code> は 10 エラーメッセージ
範囲	<code>_icmp_err_interval</code> は 0 から 99,999 ミリ秒 <code>_icmp_err_burst</code> 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
変更履歴	詳細は、 204 ページの「IP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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
変更履歴	詳細は、 204 ページの「IP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

_send_redirects (ipv4 または 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
変更履歴	詳細は、 204 ページの「IP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

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	0 (無効またはオフ)
範囲	0 (無効またはオフ) または 1 (有効またはオン)
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
変更履歴	詳細は、 204 ページの「IP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

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
変更履歴	詳細は、 204 ページの「IP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

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	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
変更履歴	詳細は、 204 ページの「IP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

_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。ただし、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

変更履歴

詳細は、[204 ページの「IP パラメータの名前の変更 \(Oracle Solaris 11\)」](#)を参照してください。

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、`weak`、`strong`、および `src-priority` を指定できます。デフォルト値は `weak` です。

デフォルト

`weak`

範囲

`weak`、`strong`、または `src-priority`

■ `weak`

- 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。

■ `strong`

- 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。

	<p>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。</p> <ul style="list-style-type: none"> ■ 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
変更履歴	詳細は、204 ページの「IP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

ip_queue_fanout

説明	<p>queue と TCP/IP 接続 を関連付けるモードを判定します。</p> <p>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 queue との接続が関連付けられます。</p>
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	<p>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。</p> <p>このプロパティは、<code>/etc/system</code> ファイル内でのみ設定できます。</p>
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

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 を再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

2-999999999

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

変更履歴

詳細は、[204 ページの「IP パラメータの名前の変更 \(Oracle Solaris 11\)」](#) を参照してください。

`_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 1280

範囲

IPv4 の場合 8-6,636

IPv6 の場合 8-1,280

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 エラーメッセージに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
変更履歴	詳細は、 204 ページの「IP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

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 1 分
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
変更履歴	詳細は、 205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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
変更履歴	詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_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。実際の数値は、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
変更履歴	詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_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。実際の数値は、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
変更履歴	詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_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、ウィンドスケールオプションを指定して SYN セグメントに回答します。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
変更履歴	詳細は、 205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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。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 セグメントに回答します。
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	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
変更履歴	詳細は、 205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

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192ページの「ルート別のメトリック」を参照してください。162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 162 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) 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
変更履歴	詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

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192ページの「ルート別のメトリック」を参照してください。162ページの「max_buf」と177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 162 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) 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
変更履歴	詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

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、 setsockopt(3XNET) 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
変更履歴	詳細は、 205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

_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 setsockopt(3XNET) 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
変更履歴	詳細は、 205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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
変更履歴	詳細は、 205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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
変更履歴	詳細は、 204 ページの「_local_slow_start_initial」 を参照してください。

`_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、163 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
変更履歴	詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

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	2 (自発的に有効にする)
範囲	0 (無効)、1 (受動的に有効にする)、2 (自発的に有効にする) のいずれか
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せん。

コミットレベル	変更の可能性あり
変更履歴	詳細は、 205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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
変更履歴	詳細は、 205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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 10 分
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

変更履歴

詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

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

1 (受動的に有効にする)

範囲

0 (無効)、1 (受動的に有効にする)、2 (自発的に有効にする) のいずれか

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

コミットレベル	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。
変更履歴	変更の可能性あり 詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_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。168 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。

`_conn_req_max_q` 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、`listen(3SOCKET)` 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 `listen()` 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
変更履歴	詳細は、 205 ページ の「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_conn_req_max_q0`

説明	単一の TCP リスナーが持つことができる、不完全な (3 段階ハンドシェー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。
	TCP の 3 段階ハンドシェークについては、 RFC 793 を参照してください。 167 ページ の「 <code>_conn_req_max_q</code> 」も参照してください。
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 段階ハンドシェークが終わっている) 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
変更履歴	詳細は、 205 ページ の「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_conn_req_min`

説明	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。
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、 <code>listen(3SOCKET)</code> 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
変更履歴	詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_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
変更履歴	詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

_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
変更履歴	詳細は、 205 ページ の「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

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。
単位	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
変更履歴	詳細は、 204 ページ の「 [tcp,sctp,udp]_smallest_anon_port および [tcp,sctp,udp]_largest_anon_port 」を参照してください。

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。
単位	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
変更履歴	詳細は、 204 ページ の「 <code>[tcp,sctp,udp]_smallest_anon_port</code> および <code>[tcp,sctp,udp]_largest_anon_port</code> 」を参照してください。

/etc/system ファイルに設定する TCP/IP パラメータ

次のパラメータは、`/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

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

この `ipadm` 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。

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 `SO_KEEPALIVE` ソケットオプションを設定することで有効になります。

ソケットで `SO_KEEPALIVE` が有効な場合、TCP 接続が 2 時間 (`tcp_keepalive_interval` 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、174 ページの「`_rexmit_interval_initial`」を参照してください。

また、個々のアプリケーションに `TCP_KEEPALIVE_THRESHOLD` 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できます。オプションの値は、ミリ秒単位の符号なし整数です。`tcp(7P)` のマニュアルページも参照してください。

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

変更履歴

詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

_ip_abort_interval

説明

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 `tcp_ip_abort_interval` 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。

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。175 ページの「`_rexmit_interval_max`」も参照してください。

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、175 ページの「`_rexmit_interval_max`」を参照してください。

コミットレベル

変更の可能性あり

変更履歴

詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

_rexmit_interval_initial

説明

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192 ページの「ルート別のメトリック」を参照してください。

デフォルト

1000 ミリ秒

範囲

1 ミリ秒から 20000 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

変更履歴

詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_rexmit_interval_max`

説明	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できません。174 ページの「 <code>_ip_abort_interval</code> 」も参照してください。
デフォルト	6000 ミリ秒
範囲	1 ミリ秒から 7200000 ミリ秒
動的か	はい
どのような場合に変更するか	通常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べきです。 <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
変更履歴	詳細は、205 ページの「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_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。175 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 7200000 ミリ秒
動的か	はい
どのような場合に変更するか	通常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

することができます。この値を変更する場合は、合わせて `_rexmit_interval_max` パラメータも変更する必要があります。 `_rexmit_interval_max` には、 `_rexmit_interval_min` 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

変更履歴

詳細は、[205 ページの「TCP パラメータの名前の変更 \(Oracle Solaris 11\)」](#)を参照してください。

`_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 7200000 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

変更履歴

詳細は、[205 ページの「TCP パラメータの名前の変更 \(Oracle Solaris 11\)」](#)を参照してください。

`_tstamp_if_wsacle`

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

コミットレベル	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、 <code>timestamp</code> オプションが必要になります。
変更履歴	変更の可能性あり
	詳細は、 205 ページ の「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。
<code>_recv_hiwat_minmss</code>	
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、 <code>_recv_hiwat_minmss</code> 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
変更履歴	詳細は、 205 ページ の「TCP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

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、 178 ページ の「 <code>max_buf</code> 」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 178 ページ の「 <code>max_buf</code> 」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) SO_SNDBUF</code> を使用してソケット

コミットレベル	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。
変更履歴	変更の可能性あり 詳細は、 207 ページの「UDP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

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、 178 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 178 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) 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
変更履歴	詳細は、 207 ページの「UDP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

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
変更履歴	詳細は、 207 ページ の「UDP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

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。
単位	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
変更履歴	詳細は、 204 ページ の「 <code>[tcp,sctp,udp]_smallest_anon_port</code> および <code>[tcp,sctp,udp]_largest_anon_port</code> 」を参照してください。

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。
単位	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
変更履歴	詳細は、 204 ページ の「 <code>[tcp,sctp,udp]_smallest_anon_port</code> および <code>[tcp,sctp,udp]_largest_anon_port</code> 」を参照してください。

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 181 ページの「 <code>_pa_max_retr</code> 」に依存します。 <code>_max_init_retr</code> が <code>_pa_max_retr</code> 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
変更履歴	詳細は、207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

`_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せん。 <code>_pa_max_retr</code> は、使用可能なすべてのパスの 182 ページの「 <code>_pp_max_retr</code> 」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

コミットレベル	合、 <code>_pa_max_retr</code> 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)。
変更履歴	変更の可能性あり 詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

`_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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

`_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 <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、187 ページの「 max_buf 」は <code>_cwnd_max</code> 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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

_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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

_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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

_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監視するため、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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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 範囲	2 分 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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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 範囲	1500 バイト 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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

_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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

_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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

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。187 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 187 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) 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
変更履歴	詳細は、207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

_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、186 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
変更履歴	詳細は、207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

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。187 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 187 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) 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
変更履歴	詳細は、207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

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
変更履歴	詳細は、207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」を参照してください。

_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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

_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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

_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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

_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、188 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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

`_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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

`_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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

`_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 範囲	60,000 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、 188 ページの「_rto_max」 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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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 範囲	4 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
変更履歴	詳細は、 207 ページの「SCTP パラメータの名前の変更 (Oracle Solaris 11)」 を参照してください。

`_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 範囲	0 (無効) 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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

_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
変更履歴	詳細は、207 ページの「 SCTP パラメータの名前の変更 (Oracle Solaris 11) 」を参照してください。

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。
単位	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
変更履歴	詳細は、 204 ページ の「 <code>[tcp,sctp,udp]_smallest_anon_port</code> および <code>[tcp,sctp,udp]_largest_anon_port</code> 」を参照してください。

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。
単位	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
変更履歴	詳細は、 204 ページ の「 <code>[tcp,sctp,udp]_smallest_anon_port</code> および <code>[tcp,sctp,udp]_largest_anon_port</code> 」を参照してください。

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)。

```
192.123.123.0      192.123.123.4      U        1        4 hme0
192.123.124.0      192.123.124.4      U        1        4 ge0
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、ge0 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。

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。

- 196 ページの「autofs」
- 196 ページの「cron」
- 196 ページの「devfsadm」
- 196 ページの「dhcpagent」
- 197 ページの「fs」
- 197 ページの「ftp」
- 197 ページの「inetinit」
- 197 ページの「init」
- 198 ページの「ipsec」
- 198 ページの「kbd」
- 198 ページの「keyserv」
- 199 ページの「login」
- 199 ページの「mpathd」
- 199 ページの「nfs」
- 199 ページの「nfslogd」
- 199 ページの「nss」
- 199 ページの「passwd」
- 200 ページの「su」
- 200 ページの「syslog」
- 200 ページの「tar」
- 200 ページの「telnetd」
- 200 ページの「utmpd」

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 11.1 でのサービスと障害の管理](#)』の「[SMF 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 11.1 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](#)』の第5章「[システムコンソール、端末デバイス、および電源サービスの管理 \(タスク\)](#)」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\(1M\)](#) のマニュアルページ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。

dhcpgent

DHCP のクライアント使用率は、`dhcpgent` 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、`dhcpgent` 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、[dhcagent\(1M\)](#) の「FILES」セクションにある `/etc/default/dhcagent` 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 NLST 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/value_authorization  astring                solaris.smf.value.environment
umask/umask          astring                022
upgrade             application
```

upgrade/skip_init_upgrade	boolean	false
upgrade/value_authorization	astring	solaris.smf.value.environment
environment	application	
environment/LANG	astring	C
.		
.		

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\(1\)](#)のマニュアルページ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\(1\)](#)のマニュアルページの「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\(1M\)](#)のマニュアルページ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\(4\)](#)のマニュアルページ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\(4\)](#)のマニュアルページ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\(1\)](#) のマニュアルページ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` (および Solaris の以前のバージョンでは `/var/adm/utmp` も) を監視し、`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消去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、監視するプロセスがまだ活着しているかどうかの `/proc` のチェックとチェックの合間に休眠する秒数。デフォルト値は 300 です。
- `MAX_FDS` - utmpd が監視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。

チューニング可能パラメータの変更履歴

この章には、特定のチューニング可能パラメータの変更履歴を記載します。このセクションにパラメータが記載されていれば、旧リリースから変更されていることを意味します。また、元の機能がすでに削除されているパラメータもリストされていません。

- 201 ページの「カーネルパラメータ」
- 204 ページの「TCP/IP チューニング可能パラメータ (Oracle Solaris 11 および 11.1)」
- 208 ページの「廃止または削除されたパラメータ (Oracle Solaris 11 および 11.1)」

カーネルパラメータ

カーネルとメモリの一般的なパラメータ (Oracle Solaris 11 および 11.1)

zfs_arc_min

このパラメータの説明は、Oracle Solaris 11 リリースで新しく記載されました。詳細は、94 ページの「zfs_arc_min」を参照してください。

Oracle Solaris 11.1 では、zfs_arc_min パラメータ情報は第 3 章「Oracle Solaris ZFS チューニング可能パラメータ」に移動しました。

zfs_arc_max

このパラメータの説明は、Oracle Solaris 11 リリースで新しく記載されました。詳細は、95 ページの「zfs_arc_max」を参照してください。

Oracle Solaris 11.1 では、zfs_arc_max パラメータ情報は第 3 章「Oracle Solaris ZFS チューニング可能パラメータ」に移動しました。

disp_rechoose_interval

このパラメータは、Oracle Solaris 11 リリースで新しく追加されました。詳細は [84 ページ](#) の「`disp_rechoose_interval`」を参照してください。

ページング関連パラメータ (Oracle Solaris 11)

fastscan

`fastscan` のデフォルト値が明確にされました。詳細は、[52 ページ](#) の「`fastscan`」を参照してください。

プロセスサイジングのチューニング可能パラメータ (Oracle Solaris 11.1)

ngroups_max

このパラメータは、Oracle Solaris 11 リリースで新しく記載されました。

Oracle Solaris 11.1 リリースでは、「どのような場合に変更するか」の説明に、このパラメータが 16 より大きく設定された場合の `AUTH_SYS` 認証に関する相互作用情報が含まれるように改訂されました。詳細は、[42 ページ](#) の「`ngroups_max`」を参照してください。

max_nprocs

このパラメータのデフォルト値が、Oracle Solaris 11.1 リリースで更新されました。詳細は、[41 ページ](#) の「`max_nprocs`」を参照してください。

maxusers

このパラメータのデフォルト値が、Oracle Solaris 11.1 リリースで更新されました。詳細は、[38 ページ](#) の「`maxusers`」を参照してください。

pidmax

このパラメータのデフォルト値が、Oracle Solaris 11.1 リリースで更新されました。詳細は、[40 ページ](#) の「`pidmax`」を参照してください。

segkpsize

このパラメータのデフォルト値が、Oracle Solaris 11.1 リリースで更新されました。詳細は、[32 ページ](#) の「`segkpsize`」を参照してください。

スワッピング関連パラメータ (Oracle Solaris 11.1)

swaps_minfree

このパラメータ情報の「どのような場合に変更するか」セクションが、Oracle Solaris 11.1 リリースで更新されました。詳細は、[58 ページの「swaps_minfree」](#)を参照してください。

一般的なファイルシステムパラメータ (Oracle Solaris 11.1)

dnlc_dircache_percent

このパラメータは、Oracle Solaris 11.1 リリースで新しく追加されました。詳細は、[75 ページの「dnlc_dircache_percent」](#)を参照してください。

一般的なドライバパラメータ (Oracle Solaris 11)

ddi_msix_alloc_limit

このパラメータは、Oracle Solaris 11 リリースで新しく記載されました。詳細は、[64 ページの「ddi_msix_alloc_limit」](#)を参照してください。

ネットワークドライバパラメータ (Oracle Solaris 11)

igb パラメータ

Oracle Solaris 11 リリースでは **igb** ネットワークドライバパラメータが提供されています。詳細は、[65 ページの「igb パラメータ」](#)を参照してください。

ixgbe パラメータ

Oracle Solaris 11 リリースでは **ixgbe** ネットワークドライバパラメータが提供されています。詳細は、[66 ページの「ixgbe パラメータ」](#)を参照してください。

TCP/IP チューニング可能パラメータ (Oracle Solaris 11 および 11.1)

[tcp,sctp,udp]_smallest_anon_port および [tcp,sctp,udp]_largest_anon_port

これらのパラメータの説明は、Oracle Solaris 11 リリースで新たに追加されました。

- 191 ページの「smallest_anon_port」
- 192 ページの「largest_anon_port」
- 170 ページの「smallest_anon_port」
- 171 ページの「largest_anon_port」
- 179 ページの「smallest_anon_port」
- 179 ページの「largest_anon_port」

_local_slow_start_initial

このパラメータは、Oracle Solaris 11.1 リリースで新しく記載されました。詳細は、163 ページの「_local_slow_start_initial」を参照してください。

IP パラメータの名前の変更 (Oracle Solaris 11)

次の IP パラメータの名前が IP プロパティに変更されました。

IP プロパティを設定するには、次のような構文を使用します。

```
# ipadm set-prop -p _icmp_err_interval=100 ip
```

IP プロパティ情報を表示するには、次のような構文を使用します。

```
# ipadm show-prop -p _icmp_err_interval ip
PROTO PROPERTY          PERM CURRENT    PERSISTENT  DEFAULT  POSSIBLE
ip   _icmp_err_interval  rw   100           100         100      0-99999
```

表 A-1 IP パラメータの名前の変更

以前の IP パラメータ名	IP プロパティ名
ip_addrs_per_if	_addrs_per_if
ip_forwarding	forwarding (IPv4)
ip6_forwarding	forwarding (IPv6)

表 A-1 IP パラメータの名前の変更 (続き)

以前の IP パラメータ名	IP プロパティ名
ip_forward_src_routed	_forward_src_routed (IPv4)
ip6_forward_src_routed	_forward_src_routed (IPv6)
ip_icmp_err_interval	_icmp_err_interval
ip_icmp_err_burst	_icmp_err_burst
ip_icmp_return_data_bytes	_icmp_return_data_bytes (IPv4)
ip6_icmp_return_data_bytes	_icmp_return_data_bytes (IPv6)
ip_ire_pathmtu_interval	_pathmtu_interval
ip_respond_to_echo_broadcast	_respond_to_echo_broadcast (IPv4)
ip6_respond_to_echo_broadcast	_respond_to_echo_broadcast (IPv6)
ip_respond_to_echo_multicast	_respond_to_echo_multicast (IPv4)
ip6_respond_to_echo_multicast	_respond_to_echo_multicast (IPv6)
ip_send_redirects	_send_redirects (IPv4)
ip6_send_redirects	_send_redirects (IPv6)
ip_strict_dst_multihoming	hostmodel

TCP パラメータの名前の変更 (Oracle Solaris 11)

次の TCP パラメータの名前が TCP プロパティに変更されました。

TCP プロパティを設定するには、次のような構文を使用します。

```
# ipadm set-prop -p _deferred_ack_interval=100 tcp
```

TCP プロパティ情報を表示するには、次のような構文を使用します。

```
# ipadm show-prop -p _deferred_ack_interval tcp
PROTO PROPERTY          PERM CURRENT    PERSISTENT  DEFAULT  POSSIBLE
tcp    _deferred_ack_interval rw    100          --          100      1-60000
```

表 A-2 TCP パラメータの名前の変更

以前の TCP パラメータ名	TCP プロパティ名
tcp_deferred_ack_interval	_deferred_ack_interval
tcp_local_dack_interval	_local_dack_interval
tcp_deferred_acks_max	_deferred_acks_max

表 A-2 TCP パラメータの名前の変更 (続き)

以前の TCP パラメータ名	TCP プロパティ名
tcp_local_dacks_max	_local_dacks_max
tcp_wscale_always	_wscale_always
tcp_tstamp_always	_tstamp_always
tcp_xmit_hiwat	send_buf
tcp_recv_hiwat	recv_buf
tcp_max_buf	max_buf
tcp_cwnd_max	_cwnd_max
tcp_slow_start_initial	_slow_start_initial
tcp_slow_start_after_idle	_slow_start_after_idle
tcp_sack_permitted	sack
tcp_rev_src_routes	_rev_src_routes
tcp_time_wait_interval	_time_wait_interval
tcp_ecn_permitted	ecn
tcp_conn_req_max_q	_conn_req_max_q
tcp_conn_req_max_q0	_conn_req_max_q0
tcp_conn_req_min	_conn_req_min
tcp_rst_sent_rate_enabled	_rst_sent_rate_enabled
tcp_rst_sent_rate	_rst_sent_rate
tcp_keepalive_interval	_keepalive_interval
tcp_ip_abort_interval	_ip_abort_interval
tcp_rexmit_interval_initial	_rexmit_interval_initial
tcp_rexmit_interval_max	_rexmit_interval_max
tcp_rexmit_interval_min	_rexmit_interval_min
tcp_rexmit_interval_extra	_rexmit_interval_extra
tcp_tstamp_if_wscale	_tstamp_if_wscale
tcp_recv_hiwat_minmss	_recv_hiwat_minmss

UDP パラメータの名前の変更 (Oracle Solaris 11)

次の UDP パラメータの名前が UDP プロパティに変更されました。

UDP プロパティを設定するには、次のような構文を使用します。

```
# ipadm set-prop -p send_buf=57344 udp
```

UDP プロパティ情報を表示するには、次のような構文を使用します。

```
# ipadm show-prop -p send_buf udp
PROTO PROPERTY          PERM CURRENT    PERSISTENT  DEFAULT  POSSIBLE
udp   send_buf            rw   57344         57344       57344     1024-2097152
```

表 A-3 UDP パラメータの名前の変更

以前の UDP パラメータ名	UDP プロパティ名
udp_max_buf	max_buf
udp_xmit_hiwat	send_buf
udp_recv_hiwat	recv_buf

SCTP パラメータの名前の変更 (Oracle Solaris 11)

次の SCTP パラメータの名前が SCTP プロパティに変更されました。

SCTP プロパティを設定するには、次のような構文を使用します。

```
# ipadm set-prop -p _max_init_retr=8 sctp
```

SCTP プロパティ情報を表示するには、次のような構文を使用します。

```
# ipadm show-prop -p _max_init_retr sctp
PROTO PROPERTY          PERM CURRENT    PERSISTENT  DEFAULT  POSSIBLE
sctp  _max_init_retr      rw   8             8           8       0-128
```

表 A-4 SCTP パラメータの名前の変更

以前の SCTP パラメータ名	SCTP プロパティ名
sctp_max_init_retr	_max_init_retr
sctp_pa_max_retr	_pa_max_retr
sctp_pp_max_retr	_pp_max_retr
sctp_cwnd_max	_cwnd_max
sctp_ipv4_ttl	_ipv4_ttl

表 A-4 SCTP パラメータの名前の変更 (続き)

以前の SCTP パラメータ名	SCTP プロパティ名
sctp_heartbeat_interval	_heartbeat_interval
sctp_new_secret_interval	_new_secret_interval
sctp_initial_mtu	_initial_mtu
sctp_deferred_ack_interval	_deferred_ack_interval
sctp_ignore_path_mtu	_ignore_path_mtu
sctp_initial_ssthresh	_initial_ssthresh
sctp_ipv6_hoplimit	_ipv6_hoplimit
sctp_xmit_lowat	_xmit_lowat
sctp_xmit_hiwat	send_buf
sctp_recv_hiwat	recv_buf
sctp_max_buf	max_buf
sctp_rto_min	_rto_min
sctp_rto_max	_rto_max
sctp_rto_initial	_rto_initial
sctp_cookie_life	_cookie_life
sctp_max_in_streams	_max_in_streams
sctp_initial_out_streams	_initial_out_streams
sctp_shutack_wait_bound	_shutack_wait_bound
sctp_maxburst	_maxburst
sctp_addip_enabled	_addip_enabled
sctp_prsctp_enabled	_prsctp_enabled

廃止または削除されたパラメータ (Oracle Solaris 11 および 11.1)

次のセクションでは、最新の Oracle Solaris リリースで廃止または削除されたパラメータについて説明します。

NCA パラメータ

ネットワークキャッシュアクセラレータ (NCA) パラメータ情報が廃止され、削除されました。

consistent_coloring

このパラメータは Oracle Solaris 11.1 リリースから廃止されています。

rstchown

このパラメータは Oracle Solaris 11 リリースから廃止されています。

説明	<p>chown システム呼び出しの POSIX 挙動が有効かどうかを示します。POSIX 挙動は次のとおりです。</p> <ul style="list-style-type: none"> ■ プロセスは、UID 0 で動作していない限りファイルの所有者を変更できない。 ■ プロセスは、UID 0 で動作していない限り、ファイルのグループ所有権を、現在メンバーになっていないグループに変更することはできない。 <p>詳細は、chown(2) のマニュアルページを参照してください。</p>
データ型	符号付き整数
デフォルト	1 (POSIX 挙動が使用されている)
範囲	0 (POSIX 挙動が強制されない) または 1 (POSIX 挙動が使用される)
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	POSIX 挙動が適切でない場合。POSIX 挙動をオフにすると、さまざまなセキュリティーホールの可能性が出てくる点に留意してください。オフにすると、ユーザーがファイルの所有権を別のユーザーに変更する可能性も生じます。その場合は、新たに所有者になったユーザーかシステム管理者の介入なしにはそのファイルの所有権を元に戻せません。

コミットレベル

廃止または互換性がなくなる可能性あり

廃止された TCP/IP モジュールのパラメータ (Oracle Solaris 11)

ip_multidata_outbound

このパラメータは Oracle Solaris 11 リリースで廃止されました。

tcp_mdt_max_pbufs

このパラメータは Oracle Solaris 11 リリースで廃止されました。

このマニュアルの改訂履歴

このセクションでは、このマニュアルの改訂履歴について説明しています。

- 211 ページの「最新バージョン: Oracle Solaris 11.1 リリース」
- 211 ページの「Oracle Solaris リリースで追加または変更されたパラメータ」

最新バージョン: Oracle Solaris 11.1 リリース

このマニュアルの最新バージョンは、Oracle Solaris 11.1 リリースに該当します。

Oracle Solaris リリースで追加または変更されたパラメータ

次のセクションでは、追加、変更、または廃止された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。

- Oracle Solaris 11.1: Oracle Solaris ZFS チューニング可能パラメータ情報は、第3章「Oracle Solaris ZFS チューニング可能パラメータ」にあります。
- Oracle Solaris 11.1: `maxusers`、`max_nprocs`、`ngroups_max`、`pidmax`、および `segkpsize` パラメータが Oracle Solaris 11.1 リリースで改訂されました。詳細は、第2章「Oracle Solaris カーネルチューニング可能パラメータ」を参照してください。
- Oracle Solaris 11.1: `dnlc_dircache_percent` パラメータは Oracle Solaris 11.1 リリースの新規パラメータです。詳細は、75 ページの「`dnlc_dircache_percent`」を参照してください。
- Oracle Solaris 11: `rstchown` パラメータは廃止されています。詳細は、17 ページの「Oracle Solaris システムチューニングの新機能」を参照してください。

- Oracle Solaris 11: TCP、IP、UDP、および SCTP プロパティの設定用に `ipadm` コマンドが、`ndd` コマンドに置き換われました。さらに、ネットワークパラメータの名前が、`ipadm` 形式と密接に関連するように変更されました。詳細は、[149 ページの「IP パラメータのチューニングの概要」](#)を参照してください。
- Oracle Solaris 11: このリリースには `disp_rechoose_interval` パラメータが含まれます。詳細は、[84 ページの「disp_rechoose_interval」](#)を参照してください。
- Oracle Solaris 11: このリリースには `ngroups_max` パラメータに関する記述が含まれます。詳細は、[42 ページの「ngroups_max」](#)を参照してください。
- Oracle Solaris 11: このリリースには、`zfs_arc_min` および `zfs_arc_max` パラメータに関する記述が含まれます。詳細は、[94 ページの「zfs_arc_min」](#) および [95 ページの「zfs_arc_max」](#)を参照してください。
- Oracle Solaris 11: このリリースには、いくつかの `igb` および `ixgbe` ネットワークドライバパラメータが含まれています。詳細は、[65 ページの「igb パラメータ」](#) および [66 ページの「ixgbe パラメータ」](#)を参照してください。
- Oracle Solaris 11: このリリースには `ddi_msix_alloc_limit` パラメータが含まれます。このパラメータは、デバイスインスタンスが割り当てることができる MSI-X 割り込みの数を増やすために使用できます。詳細は、[64 ページの「ddi_msix_alloc_limit」](#)を参照してください。
- Oracle Solaris 11: このリリースには `kmem_stackinfo` パラメータが含まれます。このパラメータを有効にして、カーネルスレッドスタックの使用状況を監視できます。詳細は、[61 ページの「kmem_stackinfo」](#)を参照してください。
- Oracle Solaris 11: このリリースでは、メモリー近傍性グループのパラメータが提供されます。これらのパラメータについての詳細は、[90 ページの「近傍性グループのパラメータ」](#)を参照してください。
- Oracle Solaris 11: パラメータの情報が `sun4v` システムを含むように更新されています。詳細は、次のマニュアルページを参照してください。
 - [70 ページの「maxphys」](#)
 - [76 ページの「tmpfs:tmpfs_maxkmem」](#)
 - [87 ページの「SPARC システム固有のパラメータ」](#)

索引

A

_addip_enabled, 190
_addrs_per_if, 153
autofs, 196
autoup, 35

C

_conn_req_max_q, 167
_conn_req_max_q0, 168
_conn_req_min, 169
_cookie_life, 188
cron, 196
_cwnd_max, 162,182

D

ddi_msix_alloc_limit パラメータ, 64
default_stksize, 29
default_tsb_size, 88
_deferred_ack_interval, 157,184
_deferred_acks_max, 158
desfree, 46
dhcpageant, 196
disp_rechoose_interval, 84,202
dnlc_dir_enable, 74
dnlc_dir_max_size, 75
dnlc_dir_min_size, 74
dnlc_dircache_percent, 75
doiflush, 37

dopageflush, 36

E

ecn, 166
enable_tsb_rss_sizing, 88

F

fastscan, 52
forwarding, 152
fs, 197
fsflush, 34
ftp, 197

H

handspreadpages, 54
_heartbeat_interval, 183
hires_tick, 86
hoplimit (ipv6), 153
hostmodel, 154

I

_icmp_err_burst, 151
_icmp_err_interval, 151
_icmp_return_data_bytes, 156
_ignore_path_mtu, 185

inetinit, 197
init, 198
_initial_mtu, 184
_initial_out_streams, 189
_initial_ssthresh, 185
intr_force, 65
intr_throttling, 67
_ip_abort_interval, 174
ip_squeue_fanout, 155
ip_squeue_worker_wait, 172
ipcl_conn_hash_size, 171
ipsec, 198
_ipv4_ttl, 183
_ipv6_hoplimit, 183

K

kbd, 198
_keepalive_interval, 173
keyserv, 198
kmem_flags, 59
kmem_stackinfo, 61

L

largest_anon_port, 171, 179, 192
lgrp_mem_pset_aware, 91
_local_dack_interval, 158
_local_dacks_max, 159
_local_slow_start_initial, 163
logevent_max_q_sz, 31
login, 199
lotsfree, 45
lpg_alloc_prefer, 90
lwp_default_stksize, 30

M

max_buf (SCTP), 187
max_buf (TCP), 162
max_buf (UDP), 178
_max_in_streams, 189

_max_init_retr, 181
max_nprocs, 41, 202
maxpgio, 56
maxphys, 70
maxpid, 40
maxuprc, 42
maxusers, 38, 202
min_percent_cpu, 54
minfree, 47
moddebug, 62
mpathd, 199
mr_enable, 65

N

ncsize, 73
nnd, 150
_new_secret_interval, 184
nfs:nacache, 138
nfs:nfs3_async_clusters, 136
nfs:nfs3_bsize, 132
nfs:nfs3_cots_timeo, 113
nfs:nfs3_do_symlink_cache, 115
nfs:nfs3_dynamic, 117
nfs:nfs3_jukebox_delay, 139
nfs:nfs3_lookup_neg_cache, 119
nfs:nfs3_max_threads, 123
nfs:nfs3_max_transfer_size, 140
nfs:nfs3_max_transfer_size_clts, 142
nfs:nfs3_max_transfer_size_cots, 143
nfs:nfs3_nra, 125
nfs:nfs3_pathconf_disable_cache, 110
nfs:nfs3_shrinkreaddir, 129
nfs:nfs4_async_clusters, 137
nfs:nfs4_bsize, 133
nfs:nfs4_cots_timeo, 114
nfs:nfs4_do_symlink_cache, 116
nfs:nfs4_lookup_neg_cache, 120
nfs:nfs4_max_threads, 124
nfs:nfs4_max_transfer_size, 141
nfs:nfs4_nra, 126
nfs:nfs4_pathconf_disable_cache, 110
nfs:nfs_allow_preepoch_time, 111
nfs:nfs_async_clusters, 134

nfs:nfs_async_timeout, 137
 nfs:nfs_cots_timeo, 112
 nfs:nfs_disable_rddir_cache, 131
 nfs:nfs_do_symlink_cache, 115
 nfs:nfs_dynamic, 117
 nfs:nfs_lookup_neg_cache, 118
 nfs:nfs_nra, 124
 nfs:nfs_shrinkreaddir, 129
 nfs:nfs_write_error_interval, 130
 nfs:nfs_write_error_to_cons_only, 131
 nfs:nrnode, 127
 nfs_max_threads, 122
 nfslogd, 199
 ngroups_max, 42
 noexec_user_stack, 33
 nss, 199
 nstrpush, 81

O

Oracle データベースチューニング, ZFS ファイルシステム, 104

P

pageout_reserve, 50
 pages_before_pager, 55
 pages_pp_maximum, 51
 passwd, 199
 _pathmtu_interval, 156
 physmem, 28
 pidmax, 40, 202
 _policy_mask, 180
 _pp_max_retr, 182
 primarycache, ZFS ファイルシステムプロパティ, 104
 _prscpt_enabled, 191
 pt_cnt, 79
 pt_max_pty, 80
 pt_pctofmem, 79

R

recordsize, ZFS ファイルシステムプロパティ, 103
 recv_buf (SCTP), 186
 recv_buf (TCP), 161
 recv_buf (UDP), 178
 _recv_hiwat_minmss, 177
 reserved_procs, 39
 _respond_to_echo_broadcast, 151
 _respond_to_echo_multicast, 151
 _rev_src_routes, 165
 _rexmit_interval_extra, 176
 _rexmit_interval_initial, 174
 _rexmit_interval_max, 175
 _rexmit_interval_min, 175
 rlim_fd_cur, 72
 rlim_fd_max, 71
 rpcmod:clnt_idle_timeout, 144
 rpcmod:clnt_max_conns, 144
 rpcmod:cotsmaxdupreqs, 147
 rpcmod:maxdupreqs, 146
 rpcmod:svc_default_stksize, 145
 rpcmod:svc_idle_timeout, 145
 _rst_sent_rate, 170
 _rst_sent_rate_enabled, 169
 rstchown, 209
 _rto_max, 188
 _rto_min, 187
 rx_copy_threshold, 70
 rx_limit_per_intr, 68
 rx_queue_number, 67
 rx_ring_size, 69

S

sack, 164
 sctp_maxburst, 190
 secondarycache, ZFS ファイルシステムプロパティ, 104
 segkpsize, 202
 segspt_minfree, 83
 send_buf (SCTP), 186
 send_buf (TCP), 161
 send_buf (UDP), 177

_send_redirects, 152
_shutack_wait_bound, 190
_slow_start_after_idle, 163
_slow_start_initial, 163
slowsan, 53
smallest_anon_port, 170, 179, 191
strmsgsz, 81, 82
su, 200
swapfs_minfree, 58, 203
swapfs_reserve, 57
syslog, 200

T

tar, 200
throttlefree, 49
_time_wait_interval, 165
timer_max, 86
tmpfs_maxkmem, 76
tmpfs_minfree, 77
tsb_alloc_hiwater, 87
tsb_rss_factor, 89
_tstamp_always, 160
_tstamp_if_wscale, 176
ttl (ipv4), 153
tune_t_fsflushr, 34
tune_t_minarmem, 52
tx_copy_threshold, 69
tx_queue_number, 66
tx_ring_size, 68

U

utmpd, 200

W

_wscale_always, 159

X

_xmit_lowat, 186

Z

zfs_arc_max, 95, 201
zfs_arc_min, 94, 201
zfs_prefetch_disable, 96
ZFS ファイルシステム, Oracle データベース用の
チューニング, 104
ZFS ファイルシステムプロパティ
primarycache, 104
recordsize, 103
secondarycache, 104