

Oracle® Solaris チューニング可能パラ メータ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
Solaris 10 リリースでの Oracle Solaris システムのチューニング	19
デフォルトのスタックサイズ	19
System V IPC 構成	19
NFSv4 パラメータ	21
追加または変更された TCP/IP パラメータ	21
SPARC: 変換記憶バッファ (TSB) パラメータ	23
SCTP チューニング可能パラメータ	24
Oracle Solaris システムのチューニング	24
チューニング可能パラメータの説明形式	25
Oracle Solaris カーネルのチューニング	27
/etc/system ファイル	27
kndb コマンド	28
mdb コマンド	28
Oracle Solaris の特殊な tune および var 構造体	29
Oracle Solaris システム構成情報の表示	30
sysdef コマンド	30
kstat ユーティリティ	30
2 Oracle Solaris カーネルチューニング可能パラメータ	31
チューニング可能パラメータ情報の記載箇所	32
カーネルとメモリーの一般的なパラメータ	32
physmem	32
default_stksize	33

lwp_default_stksize	34
logevent_max_q_sz	35
segkpsize	36
noexec_user_stack	37
fsflush とそれに関連するパラメータ	38
fsflush	38
tune_t_fsflushr	39
autoup	39
dopageflush	41
doiflush	41
プロセス規模調整パラメータ	42
maxusers	43
reserved_procs	44
pidmax	45
max_nprocs	45
maxuprc	46
ngroups_max	47
ページング関連パラメータ	48
lotsfree	49
desfree	50
minfree	52
throttlefree	53
pageout_reserve	54
pages_pp_maximum	55
tune_t_minarmem	56
fastscan	56
slowscan	57
min_percent_cpu	58
handspreadpages	58
pages_before_pager	59
maxpgio	60
スワッピング関連パラメータ	61
swapfs_reserve	61
swapfs_minfree	62
カーネルメモリアロケータ	63
kmem_flags	63

一般的なドライバパラメータ	65
moddebug	65
ddi_msix_alloc_limit	67
一般的な入出力パラメータ	68
maxphys	68
rlim_fd_max	69
rlim_fd_cur	70
一般的なファイルシステムパラメータ	71
ncsize	71
rstchown	72
dnlc_dir_enable	73
dnlc_dir_min_size	73
dnlc_dir_max_size	74
UFS パラメータ	75
bufhwm および bufhwm_pct	75
ndquot	77
ufs_ninode	77
ufs_WRITES	80
ufs_LW および ufs_HW	80
freebehind	81
smallfile	82
ufs_delete_hiwat	83
TMPFS パラメータ	84
tmpfs:tmpfs_maxkmem	84
tmpfs:tmpfs_minfree	84
仮想端末	85
pt_cnt	86
pt_pctofmem	87
pt_max_pty	87
STREAMS パラメータ	88
nstrpush	88
strmsgsz	88
strctlsz	89
System V メッセージキュー	90
System V セマフォ	90
System V 共有メモリー	91

segspt_minfree	91
pr_segp_disable	92
スケジューリング	94
rechoose_interval	94
タイマー	94
hires_tick	94
timer_max	95
SPARC システム固有のパラメータ	95
consistent_coloring	95
tsb_alloc_hiwater_factor	97
default_tsb_size	98
enable_tsb_rss_sizing	99
tsb_rss_factor	99
近傍性グループのパラメータ	100
lpg_alloc_prefer	100
lgrp_mem_pset_aware	102
Solaris ボリュームマネージャーのパラメータ	103
md_mirror:md_resync_bufsz	103
md:mirrored_root_flag	103
3 Oracle Solaris ZFS チューニング可能パラメータ	107
チューニング可能パラメータ情報の記載箇所	107
ZFS のチューニングの考慮事項	108
ZFS ARC パラメータ	108
zfs_arc_min	108
zfs_arc_max	109
ZFS ファイルレベルプリフェッチ	110
zfs_prefetch_disable	110
ZFS デバイスの入出力キューの深さ	111
zfs_vdev_max_pending	111
ZFS およびキャッシュフラッシュ	112
zfs_nocacheflush	113
ZFS メタデータの圧縮	114
zfs_mdcomp_disable	114
データベース製品に対する ZFS のチューニング	115

Oracle データベース用の ZFS のチューニング	116
ZFS を MySQL と一緒に使用するときの考慮事項	120
4 NFS チューニング可能パラメータ	121
チューニング可能パラメータ情報の記載箇所	121
NFS 環境のチューニング	121
NFS モジュールのパラメータ	122
nfs:nfs3_pathconf_disable_cache	122
nfs:nfs4_pathconf_disable_cache	122
nfs:nfs_allow_preepoch_time	123
nfs:nfs_cots_timeo	124
nfs:nfs3_cots_timeo	125
nfs:nfs4_cots_timeo	126
nfs:nfs_do_symlink_cache	127
nfs:nfs3_do_symlink_cache	127
nfs:nfs4_do_symlink_cache	128
nfs:nfs_dynamic	129
nfs:nfs3_dynamic	129
nfs:nfs_lookup_neg_cache	130
nfs:nfs3_lookup_neg_cache	131
nfs:nfs4_lookup_neg_cache	132
nfs:nfs_max_threads	133
nfs:nfs3_max_threads	134
nfs:nfs4_max_threads	135
nfs:nfs_nra	136
nfs:nfs3_nra	137
nfs:nfs4_nra	137
nfs:nrnode	138
nfs:nfs_shrinkreaddir	140
nfs:nfs3_shrinkreaddir	141
nfs:nfs_write_error_interval	142
nfs:nfs_write_error_to_cons_only	142
nfs:nfs_disable_rmdir_cache	143
nfs:nfs3_bsize	144
nfs:nfs4_bsize	145

nfs:nfs_async_clusters	146
nfs:nfs3_async_clusters	147
nfs:nfs4_async_clusters	148
nfs:nfs_async_timeout	149
nfs:nacache	150
nfs:nfs3_jukebox_delay	151
nfs:nfs3_max_transfer_size	152
nfs:nfs4_max_transfer_size	153
nfs:nfs3_max_transfer_size_clts	154
nfs:nfs3_max_transfer_size_cots	155
rpcmod モジュールのパラメータ	155
rpcmod:clnt_max_conns	155
rpcmod:clnt_idle_timeout	156
rpcmod:svc_idle_timeout	157
rpcmod:svc_default_stksize	157
rpcmod:maxdupreqs	158
rpcmod:cotsmaxdupreqs	159
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	161
チューニング可能パラメータ情報の記載箇所	161
IPパラメータのチューニングの概要	161
IPパラメータの妥当性検証	162
RFC (Internet Request for Comments)	162
IPチューニング可能パラメータ	162
ip_icmp_err_interval と ip_icmp_err_burst	162
ip_respond_to_echo_broadcast と ip6_respond_to_echo_multicast	163
ip_send_redirects と ip6_send_redirects	163
ip_forward_src_routed と ip6_forward_src_routed	164
ip_addrs_per_if	164
ip_strict_dst_multihoming と ip6_strict_dst_multihoming	165
ip_multidata_outbound	165
ip_queue_fanout	166
ip_soft_rings_cnt	167
ip_pmtu_min	168
特別な注意を要する IP チューニング可能パラメータ	168

TCP チューニング可能パラメータ	170
tcp_deferred_ack_interval	170
tcp_local_dack_interval	170
tcp_deferred_acks_max	171
tcp_local_dacks_max	171
tcp_wscale_always	172
tcp_tstamp_always	173
tcp_xmit_hiwat	173
tcp_rcv_hiwat	174
tcp_max_buf	174
tcp_cwnd_max	174
tcp_slow_start_initial	175
tcp_slow_start_after_idle	175
tcp_sack_permitted	176
tcp_rev_src_routes	177
tcp_time_wait_interval	177
tcp_ecn_permitted	177
tcp_conn_req_max_q	178
tcp_conn_req_max_q0	179
tcp_conn_req_min	180
tcp_rst_sent_rate_enabled	181
tcp_rst_sent_rate	181
tcp_mdt_max_pbufs	181
tcp_naglim_def	182
tcp_smallest_anon_port	183
tcp_largest_anon_port	183
/etc/system ファイルに設定する TCP/IP パラメータ	184
特別な注意を要する TCP パラメータ	185
UDP チューニング可能パラメータ	189
udp_xmit_hiwat	189
udp_rcv_hiwat	190
udp_smallest_anon_port	190
udp_largest_anon_port	191
udp_do_checksum	191
特別な注意を要する UDP パラメータ	192
IPQoS チューニング可能パラメータ	192

ip_policy_mask	192
SCTP チューニング可能パラメータ	193
sctp_max_init_retr	193
sctp_pa_max_retr	193
sctp_pp_max_retr	194
sctp_cwnd_max	194
sctp_ipv4_ttl	195
sctp_heartbeat_interval	195
sctp_new_secret_interval	195
sctp_initial_mtu	196
sctp_deferred_ack_interval	196
sctp_ignore_path_mtu	196
sctp_initial_ssthresh	197
sctp_xmit_hiwat	197
sctp_xmit_lowat	197
sctp_recv_hiwat	198
sctp_max_buf	198
sctp_ipv6_hoplimit	199
sctp_rto_min	199
sctp_rto_max	199
sctp_rto_initial	200
sctp_cookie_life	200
sctp_max_in_streams	200
sctp_initial_out_streams	201
sctp_shutack_wait_bound	201
sctp_maxburst	201
sctp_addip_enabled	202
sctp_prsctp_enabled	202
sctp_smallest_anon_port	202
sctp_largest_anon_port	203
ルート別のメトリック	203
6 システム機能のパラメータ	205
システムのデフォルトのパラメータ	206
autofs	206

cron	206
devfsadm	206
dhcpcagent	206
fs	207
ftp	207
inetinit	207
init	207
ipsec	207
kbd	207
keyserv	208
login	208
lu	208
mpathd	208
nfs	209
nfslogd	209
nss	209
passwd	209
power	209
rpc.nisd	209
su	209
syslog	209
sys-suspend	210
tar	210
telnetd	210
utmpd	210
yppasswdd	211
A チューニング可能パラメータの変更履歴	213
カーネルパラメータ	213
カーネルとメモリの一般的なパラメータ (Oracle Solaris 10)	213
ページング関連パラメータ	214
プロセスのサイズのチューニング可能パラメータ	214
UFS パラメータ	214
一般的なドライバパラメータ	214
一般的な入出力のチューニング可能パラメータ	215

fsflush とそれに関連するパラメータ	215
ページング関連のチューニング可能パラメータ	215
一般的なファイルシステムパラメータ	215
TMPFS パラメータ	215
SPARC システム固有のパラメータ (Solaris 10 リリース)	216
NFS チューニング可能パラメータ	216
nfs:nfs3_nra (Solaris 10 リリース)	216
TCP/IP チューニング可能パラメータ	216
ip_forward_src_routed と ip6_forward_src_routed (Solaris 10 リリース)	216
ip_multidata_outbound (Solaris 10 リリース)	217
ip_queue_fanout (Solaris 10 11/06 リリース)	217
ip_queue_worker_wait (Solaris 10 11/06 リリース)	217
ip_soft_rings_cnt (Solaris 10 11/06 リリース)	217
ip_queue_write (Solaris 10 リリース)	217
tcp_local_dack_interval (Solaris 10 リリース)	217
[tcp,sctp,udp]_smallest_anon_port および [tcp,sctp,udp]_largest_anon_port	217
tcp_naglim_def (Solaris 10 リリース)	218
udp_do_checksum (Solaris 10 リリース)	218
廃止または削除されたパラメータ (Oracle Solaris 10)	218
rstchown	218
System V メッセージキューパラメータ	219
System V セマフォのパラメータ	223
System V 共有メモリーパラメータ	228
B このマニュアルの改訂履歴	231
最新バージョン: <i>Oracle Solaris 10 1/13</i> リリース	231
Oracle Solaris リリースで追加または変更されたパラメータ	231
索引	233

はじめに

『Oracle Solaris 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 のカーネルと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ー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。本書では、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。

本書の x86 に関する用語については、以下を参照してください。

- 「x86」は、64 ビットおよび 32 ビットの x86 互換製品系列を指します。
 - 「x64」は、具体的には 64 ビット x86 互換 CPU を指します。
 - 「32 ビット x86」は、x86 をベースとするシステムに関する 32 ビット特有の情報を指します。
-

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、24 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)
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、バッファサイズのパラメータなど)
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、画面上のコンピュータ出力、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、画面上の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前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。
<i>AaBbCc123</i>	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

Oracle Solaris OS に含まれるシェルで使用する、UNIX のシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例のシェルプロンプトから、通常ユーザーと特権ユーザーのどちらがコマンドを実行すべきかがわかります。

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	#
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「Solaris 10 リリースでの Oracle Solaris システムのチューニング」
- 24 ページの「Oracle Solaris システムのチューニング」
- 25 ページの「チューニング可能パラメータの説明形式」
- 27 ページの「Oracle Solaris カーネルのチューニング」
- 29 ページの「Oracle Solaris の特殊な tune および var 構造体」
- 30 ページの「Oracle Solaris システム構成情報の表示」
- 30 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 10 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 10 1/13:** 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 10 8/11:** 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 ZFS 管理ガイド』および `mount(1M)` を参照してください。

- **Oracle Solaris 10 8/11:** このリリースには `ngroups_max` パラメータの説明が含まれています。詳細は、47 ページの「`ngroups_max`」を参照してください。

- **Solaris 10 10/09:** 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、108 ページの「`zfs_arc_min`」および 109 ページの「`zfs_arc_max`」を参照してください。
- **Solaris 10 10/09:** 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、100 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。
- **Solaris 10 5/08:** 95 ページの「SPARC システム固有のパラメータ」セクションの変換格納バッファのパラメータがより適切な情報を提供するように改訂されました。このリリースでは、次のパラメータが変更されています。
 - 98 ページの「`default_tsb_size`」
 - 99 ページの「`enable_tsb_rss_sizing`」
 - 99 ページの「`tsb_rss_factor`」
- **Solaris 10 8/07:** 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。
 - 68 ページの「`maxphys`」
 - 84 ページの「`tmpfs:tmpfs_maxkmem`」
 - 95 ページの「SPARC システム固有のパラメータ」
- **Solaris 10 8/07:** IP インスタンスプロジェクトによって、ゾーンを排他的 IP ゾーンとして構成し、一部の LAN または VLAN からそのゾーンへの排他的アクセスを割り当てることができます。

以前の共有 IP ゾーンの動作が、引き続きデフォルトの動作になっています。排他的 IP ゾーンとは、TCP/IP チューニング可能パラメータを含む TCP/IP の状態およびポリシーが、すべての面において排他的に IP ゾーン単位になっていることを意味します。

IP インスタンス機能の導入によって、次に挙げる TCP パラメータは `PRIV_SYS_NET_CONFIG` 特権を必要とするため、大域ゾーンでのみ設定可能になります。

- 166 ページの「`ip_queue_fanout`」
- 185 ページの「`ip_queue_worker_wait`」

その他の TCP、IP、SCTP パラメータおよびルートメトリックは、`PRIV_SYS_IP_CONFIG` 特権のみを必要とします。各排他的 IP ゾーンは、これらのパラメータを独自のセットで制御します。共有 IP ゾーンでは、TCP、IP、SCTP、およびルートパラメータは大域ゾーンによって制御されます。これは、これらのパラメータの設定が大域ゾーンとすべての共有 IP ゾーン間で共有されるためです。

Solaris ゾーンでの IP インスタンスの使用の詳細は、『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』を参照してください。

Solaris 10 リリースでの Oracle Solaris システムのチューニング

このセクションでは、Oracle Solaris 10 リリースの重要なチューニング拡張機能について説明します。

- 19 ページの「デフォルトのスタックサイズ」
- 19 ページの「System V IPC 構成」
- 21 ページの「NFSv4 パラメータ」
- 21 ページの「追加または変更された TCP/IP パラメータ」
- 23 ページの「SPARC: 変換記憶バッファ (TSB) パラメータ」
- 24 ページの「SCTP チューニング可能パラメータ」

デフォルトのスタックサイズ

新しいパラメータ `default_stksize` では、あらゆるスレッド、カーネル、またはユーザーのデフォルトスタックサイズを指定します。 `lwp_default_stksize` パラメータも引き続き利用できますが、すべてのカーネルスタックに作用するわけではありません。 `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。詳細は、33 ページの「`default_stksize`」を参照してください。

System V IPC 構成

Oracle Solaris 10 リリースでは、すべての System V IPC 機能を自動構成することも、またはリソース制御によって制御することもできます。共有できる機能はメモリー、メッセージキュー、およびセマフォアです。

リソース制御によって、ローカルシステム上で、またはネームサービス環境において、プロジェクト単位またはプロセス単位で IPC を設定できます。

旧リリースの Solaris では、カーネルのチューニング可能パラメータで IPC 機能を制御していました。これらの機能のデフォルト値を変更するには、`/etc/system` ファイルを変更してシステムをリブートしなければなりません。

しかし、リソース制御で IPC 機能を制御するようになったので、システムの稼働中に IPC 機能の構成を変更できます。

これまで動作させるためにシステムチューニングが必要だったアプリケーションの多くは、デフォルト値の増大とリソースの自動割り当てにより、チューニングしなくても動作する可能性があります。

次の表では、現在は廃止された IPC チューニング可能パラメータと、代わりに使用できる可能性のあるリソース制御を識別しています。廃止された IPC チューニング可能パラメータとリソース制御との重要な違いは、IPC チューニング可能パラメータがシステム単位で設定されていたのに対し、リソース制御は各プロジェクトまたは各プロセス単位で設定されることです。

リソース制御	廃止されたチューニング可能パラメータ	旧デフォルト値	最大値	新デフォルト値
<code>process.max-msg-qbytes</code>	<code>msgsys:msginfo_msgmnb</code>	4096	ULONG_MAX	65536
<code>process.max-msg-messages</code>	<code>msgsys:msginfo_msgtql</code>	40	UINT_MAX	8192
<code>process.max-sem-ops</code>	<code>semsys:seminfo_semopm</code>	10	INT_MAX	512
<code>process.max-sem-nsems</code>	<code>semsys:seminfo_semmsl</code>	25	SHRT_MAX	512
<code>project.max-shm-memory</code>	<code>shmsys:shminfo_shmmax*</code>	0x800000	UINT64_MAX	物理メモリーの 1/4
<code>project.max-shm-ids</code>	<code>shmsys:shminfo_shmmni</code>	100	2 ²⁴	128
<code>project.max-msg-ids</code>	<code>msgsys:msginfo_msgmni</code>	50	2 ²⁴	128
<code>project.max-sem-ids</code>	<code>semsys:seminfo_semmni</code>	10	2 ²⁴	128

* `project.max-shm-memory` リソース制御は 1 つのプロジェクトにおける共有メモリー量の合計を制限していますが、以前は `shmsys:shminfo_shmmax` パラメータが 1 つの共有メモリーセグメントのサイズを制限していました。

リソース制御の詳細は、『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』の「[使用可能なリソース制御](#)」を参照してください。

廃止されたパラメータは、Oracle Solaris システムの `/etc/system` ファイルに引き続き含めることができます。その場合、これらのパラメータは、Oracle Solaris の以前のリリースの場合と同様に、デフォルトのリソース制御値の初期化に使用されます。詳細は、[218 ページ](#)の「[廃止または削除されたパラメータ \(Oracle Solaris 10\)](#)」を参照してください。ただし、古いパラメータはできるだけ使用しないでください。

関連のある次のパラメータは削除されました。Oracle Solaris システムの `/etc/system` ファイルにこれらのパラメータが含まれている場合は、コメントになります。

<code>semsys:seminfo_semmns</code>	<code>semsys:seminfo_semvmx</code>
<code>semsys:seminfo_semmnu</code>	<code>semsys:seminfo_semaem</code>
<code>semsys:seminfo_semume</code>	<code>semsys:seminfo_semusz</code>
<code>semsys:seminfo_semmap</code>	<code>shmsys:shminfo_shmseg</code>

```
shmsys:shminfo_shmmin                msgsys:msginfo_msgmap
msgsys:msginfo_msgseg                msgsys:msginfo_msgssz
msgsys:msginfo_msgmax
```

現在使用できるリソース制御の一覧については、[rctldm\(1M\)](#) のマニュアルページを参照してください。リソース制御の構成については、[project\(4\)](#) および『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』の第 6 章「リソース制御 (概要)」を参照してください。

NFSv4 パラメータ

Oracle Solaris 10 リリースでは、NFSv4 プロトコル用の次のパラメータが新しく追加されました。

- 122 ページの「[nfs:nfs4_pathconf_disable_cache](#)」
- 126 ページの「[nfs:nfs4_cots_timeo](#)」
- 128 ページの「[nfs:nfs4_do_symlink_cache](#)」
- 132 ページの「[nfs:nfs4_lookup_neg_cache](#)」
- 135 ページの「[nfs:nfs4_max_threads](#)」
- 137 ページの「[nfs:nfs4_nra](#)」
- 145 ページの「[nfs:nfs4_bsize](#)」
- 148 ページの「[nfs:nfs4_async_clusters](#)」
- 153 ページの「[nfs:nfs4_max_transfer_size](#)」

NFSv4 パラメータについては、122 ページの「[NFS モジュールのパラメータ](#)」を参照してください。

追加または変更された TCP/IP パラメータ

Oracle Solaris 10 リリースでは、次の IP パラメータを利用できます。

- 185 ページの「[ip_queue_worker_wait](#)」
- 166 ページの「[ip_queue_fanout](#)」
- 184 ページの「[ipcl_conn_hash_size](#)」

Oracle Solaris 10 リリースでは、次の TCP パラメータを利用できます。

- 181 ページの「[tcp_rst_sent_rate_enabled](#)」
- 181 ページの「[tcp_rst_sent_rate](#)」
- 181 ページの「[tcp_mdt_max_pbufs](#)」

この Oracle Solaris リリースでは、次の TCP/IP パラメータは廃止されています。

- [ipc_tcp_conn_hash_size](#)

- `tcp_compression_enabled`
- `tcp_conn_hash_size`
- `ip_forwarding`
- `ip6_forwarding`
- `xxx_forwarding`

IP 転送に関する変更

この Oracle Solaris リリースでは、`ndd` コマンドで次のチューニング可能パラメータを設定する代わりに、`routedm` コマンドまたは `ifconfig` コマンドを使用して、IP 転送を有効にしたり無効にしたりするようになりました。

- `ip_forwarding`
- `ip6_forwarding`
- `xxx_forwarding`

`ndd` コマンドの代わりに `routedm` コマンドおよび `ifconfig` コマンドを使用して IP 転送を設定することによって、次の利点が得られます。

- すべての設定値がリブート後も維持されます。
- 新しい `ifconfig router` および `-router` コマンドを `/etc/hostname.interface` ファイルに、インタフェースの初期構成時に実行される他の `ifconfig` コマンドとともに組み込むことができます。

システムのすべてのインタフェース上で IPv4 または IPv6 パケットを転送できるようにするには、次のコマンドを使用します。

```
# routedm -e ipv4-forwarding
```

```
# routedm -e ipv6-forwarding
```

システムのすべてのインタフェース上で IPv4 または IPv6 パケット転送を無効にするには、次のコマンドを使用します。

```
# routedm -d ipv4-forwarding
```

```
# routedm -d ipv6-forwarding
```

旧リリースの Solaris では、システムのすべてのインタフェース上で IPv4 または IPv6 パケットを転送できるようにする場合、次のように入力します。

```
# ndd -set /dev/ip ip_forwarding 1
```

```
# ndd -set /dev/ip ip6_forwarding 1
```

旧リリースの Solaris では、システムのすべてのインタフェース上で IPv4 または IPv6 パケット転送を無効にする場合、次のように入力します。

```
# ndd -set /dev/ip ip_forwarding 0
```

```
# ndd -set /dev/ip ip6_forwarding 0
```

特定の IPv4 インタフェースまたは IPv6 インタフェースで IP 転送を有効にするには、インタフェースに合わせて次のような構文を使用します。例として、bge0 インタフェースを使用します。

```
# ifconfig bge0 router
```

```
# ifconfig bge0 inet6 router
```

特定の IPv4 インタフェースまたは IPv6 インタフェースで IP 転送を無効にするには、インタフェースに合わせて次のような構文を使用します。例として、bge0 インタフェースを使用します。

```
# ifconfig bge0 -router
```

```
# ifconfig bge0 inet6 -router
```

これまで、特定のインタフェースで IP 転送を有効にするには、次のように入力していました。

```
# ndd -set /dev/ip bge0:ip_forwarding 1
```

```
# ndd -set /dev/ip bge0:ip_forwarding 1
```

これまで、特定のインタフェースで IP 転送を無効にするには、次のように入力していました。

```
# ndd -set /dev/ip ip_forwarding 0
```

```
# ndd -set /dev/ip ip6_forwarding 0
```

実行システム上で前のいずれかの routeadm 設定値を有効にする場合は、次のコマンドを使用します。

```
# routeadm -u
```

詳細は、[routeadm\(1M\)](#) および [ifconfig\(1m\)](#) のマニュアルページを参照してください。

SPARC: 変換記憶バッファ (TSB) パラメータ

Oracle Solaris 10 リリースでは、変換格納バッファ (TSB) をチューニングするための新しいパラメータが追加されました。TSB パラメータについては、[95 ページ](#) の「[SPARC システム固有のパラメータ](#)」を参照してください。

SCTP チューニング可能パラメータ

この Oracle Solaris リリースには、TCP と同様のサービスを提供する、信頼性の高いトランスポートプロトコルであるストリーム制御伝送プロトコル (SCTP) が用意されています。SCTP チューニング可能パラメータについては、[193 ページの「SCTP チューニング可能パラメータ」](#)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。Oracle Solaris 10 リリースでは、System V IPC (セマフォ、共有メモリー、およびメッセージキュー) に関係するすべてのチューニング可能パラメータが変更されているため、使用環境で変更するようにしてください。詳細は、[19 ページの「System V IPC 構成」](#)を参照してください。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おりの名前。

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せん。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

次の区別で、符号付きまたは符号なし short 整数または long 整数を指定。

- 32ビットのカーネルが動作しているシステムでは、long 整数のサイズは整数と同じです。

	<ul style="list-style-type: none"> ▪ 64 ビットのカーネルが動作しているシステムでは、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
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	27 ページの「/etc/system ファイル」
カーネルデバッグ (kldb) の使用	28 ページの「kldb コマンド」
モジュールデバッグ (mdb) の使用	28 ページの「mdb コマンド」
ndd 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 = 0x78000000
```

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す。

システムの回復の詳細は、『[Oracle Solaris の管理: 基本管理](#)』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 zfs sd pcisch sockfs ip hook neti sctp arp
usbaf fcp fctl md lofs cpc random crypto fcip nca logindmux ptm ufs sPPP nfs ]
> ::memstat
Page Summary          Pages          MB  %Tot
-----
Kernel                95193          743   37%
ZFS File Data         96308          752   38%
Anon                   28132          219   11%
Exec and libs          1870           14    1%
Page cache             1465            11    1%
Free (cachelist)       4242            33    2%
Free (freelist)        28719          224   11%

Total                  255929          1999
Physical               254495          1988
> $q
```

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

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

たとえば、UFS モジュールがロードされている場合、各デバッグは `ufs:ufs_WRITES` を `ufs_WRITES` としてアクセスします。 `ufs:` 接頭辞が必要なのは、`/etc/system` ファイルに設定する場合です。

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 kmdb 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、16G バイトのメモリーを備えた SPARC 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。

```
20840448      maximum memory allowed in buffer cache (bufhwm)
15898        maximum number of processes (v.v_proc)
99           maximum global priority in sys class (MAXCLSPRI)
15893        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 ユーティリティー

kstat データ構造体群は、さまざまなカーネルのサブシステムやドライバによって維持されています。この構造体群は、カーネル内のデータをユーザープログラムに提供するメカニズムを提供します。このメカニズムを利用する場合、プログラムはカーネルのメモリーを読んだり、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。

- 32 ページの「カーネルとメモリーの一般的なパラメータ」
- 38 ページの「`fsflush` とそれに関連するパラメータ」
- 42 ページの「プロセス規模調整パラメータ」
- 48 ページの「ページング関連パラメータ」
- 61 ページの「スワッピング関連パラメータ」
- 63 ページの「カーネルメモリアロケータ」
- 65 ページの「一般的なドライバパラメータ」
- 68 ページの「一般的な入出力パラメータ」
- 71 ページの「一般的なファイルシステムパラメータ」
- 75 ページの「UFS パラメータ」
- 84 ページの「TMPFS パラメータ」
- 85 ページの「仮想端末」
- 88 ページの「STREAMS パラメータ」
- 90 ページの「System V メッセージキュー」
- 90 ページの「System V セマフォ」
- 91 ページの「System V 共有メモリー」
- 94 ページの「スケジューリング」
- 94 ページの「タイマー」
- 95 ページの「SPARC システム固有のパラメータ」
- 100 ページの「近傍性グループのパラメータ」
- 103 ページの「Solaris ボリュームマネージャーのパラメータ」

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。34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。
- x86 システムでは PAGESIZE の 2 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。
- x86 システムでは PAGESIZE の 2 倍。
- x86 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

- 8192 (x86 プラットフォーム)
- 24,576 (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。
- x86 システムでは PAGESIZE の 2 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、`lwp_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

変更履歴

詳細は、[214 ページの「`lwp_default_stksize` \(Solaris 10 リリース\)」](#)を参照してください。

logevent_max_q_sz

説明

キューに格納して `syseventd` デモンへの配信を待機させることのできる、システムイベントの最大数です。システムイベントキューのサイズがこの制限に達すると、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	ddi_log_sysevent と sysevent_post_event によってシステムイベントが生成されるたびに、システムイベントフレームワークはこの値をチェックします。 詳細は、 ddi_log_sysevent(9F) および sysevent_post_event(3SYSEVENT) 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。このパラメータを設定できるのは、64ビットカーネルが動作しているシステムに限られます。64ビットカーネルが動作しているシステムは、デフォルトで、24Kバイトのスタックサイズを使用します。
データ型	符号なし long
デフォルト	64 ビットカーネル、2G バイト 32 ビットカーネル、512M バイト
範囲	64 ビットカーネル、512M バイトから 24G バイト
単位	ページ
動的か	いいえ
検証	値が、最小および最大サイズ (64 ビットシステムでは 512M バイトおよび 24G バイト) と比較されません。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。

どのような場合に変更するか	キャッシュの作成で実際に使用されるサイズは、検査後、 <code>segkpsize</code> に指定されている値か、物理メモリーの50%のうち、小さい方です。
コミットレベル	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。このデフォルトサイズによって、65,536以上のカーネルスレッドに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は、物理メモリーが十分にあれば <code>segkpsize</code> を増やすことができます。
	変更の可能性あり

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。64ビットカーネルおよび32ビットカーネルが動作するシステムで32ビットアプリケーションのスタックを実行不能にするには、このパラメータの設定が必要です。

注-このパラメータは、64ビットのSPARCおよびAMD64アーキテクチャーでのみ効果的です。

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、 <code>mprotect(2)</code> 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、213 ページの「 <code>noexec_user_stack (Solaris 10 リリース)</code> 」を参照してください。

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いならば、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、39 ページの「`tune_t_fsflushr`」と 39 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、 <code>tune_t_flushr</code> とともに制御します。
	さらに、 <code>autoup</code> 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

	<p>ます。B_DELWRI フラグが付いているバッファ (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 <i>autoup</i> 秒を超えると書き出されます。autoup の値を増やすと、バッファがメモリーに置かれている時間が長くなります。</p>
データ型	符号付き整数
デフォルト	30
範囲	1 から MAXINT
単位	秒
動的か	いいえ
検証	autoup がゼロ以下の場合、30 に再設定され、警告メッセージが出力されます。この確認が行われるのはブート時だけです。
暗黙的制約	<p>autoup は tune_t_fsflushr の整数倍でなければなりません。最小でも autoup は tune_t_fsflushr 値の 6 倍以上でなければなりません。そうでないと、fsflush が呼び出されるたびに余計なメモリーが走査されます。</p> <p>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。</p>
どのような場合に変更するか	<p>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。</p> <ul style="list-style-type: none"> ■ 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 $\text{tune_t_fsflushr} / \text{autoup}$ 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
変更履歴	詳細は、215 ページの「 dopageflush (Solaris 10 リリース) 」を参照してください。

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
----	---

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。 システムがリブ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	<p>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。</p> <ul style="list-style-type: none"> ■ 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容量または 2048 のどちらか小さい方
範囲	<p>/etc/system ファイルに設定されていない場合は、物理メモリーに基づいて、1 から 2048。</p> <p>/etc/system ファイルに設定されている場合は、1 から 4096。</p>
単位	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に 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	<p>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。</p> <p>out of processes</p> <p>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。</p>

- 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。71 ページの「`ncsize`」を参照してください。

コミットレベル

変更の可能性あり

reserved_procs

説明

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

データ型

符号付き整数

デフォルト

5

範囲

5 から MAXINT

単位

プロセス数

動的か

いいえ。最初のパラメータ計算の後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	30,000
範囲	5 から 999,999
単位	プロセス数
動的か	いいえ。pidmax の値を設定するためにブート時だけ使用されます。
検証	はい。reserved_procs の値と 999,999 に対して値を比較します。reserved_procs より小さい場合、または 999,999 より大きい場合、値は 999,999 に設定されます。
暗黙的制約	max_nprocs に対して範囲の検査が行われ、max_nprocs は常にこの値以下に保たれます。
どのような場合に変更するか	システム上で 30,000 を超える数のプロセスをサポートできるようにするために必要です。 45 ページの「max_nprocs」 も参照してください。
コミットレベル	変更の可能性あり

max_nprocs

説明	<p>システム上に作成できるプロセスの最大数を指定します。システムプロセスとユーザープロセスを含みます。/etc/system に指定した任意の値が maxuprc の計算に使用されます。</p>
----	--

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。

- 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)
- UFS のディスク割り当て構造体を割り当てるとき (ndquot が指定されていない場合)
- 構成されたシステム V セマフォールによって使用されるメモリーの総量がシステム限度を超えていないか確認するとき
- x86 プラットフォーム向けのハードウェアアドレス変換のリソースを構成するとき

データ型	符号付き整数
デフォルト	maxusers が /etc/system ファイル内で設定されている場合は、10 + (16 x maxusers)
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、 214 ページの「max_nprocs (Solaris 10 リリース)」 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 <code>max_nprocs - reserved_procs</code> と比較され、2つの値のうちの小さい方に設定されます。
どのような場合に変更するか	1 ユーザーが作成できるプロセスの数を強く制限するために、デフォルト値より小さい値を指定したい場合 (システムが作成できるプロセスの数が多くて)。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。 <code>ngroups_max</code> 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せん。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	グループの最大数を増やす場合。 ある特定のユーザーに 16 より多いグループが割り当てられている場合は、NFS 環境で <code>AUTH_SYS</code> 資格に関する問題が発生する可能性があ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
変更履歴	詳細は、214 ページの「 <code>ngroups_max (Solaris 10 リリース)</code> 」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、[56 ページの「fastscan」](#) および [57 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ません。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秒間スケジューリングされます。

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`、`slowsan`、および `handsreadpages` 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せん。詳細は、 getpagesize(3C) 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	lotsfree 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	<p>ページ要求が急激に増えるような場合には、メモリアルゴリズムが要求に対応できないことがあります。これを回避するには、早期にメモリーの回収を開始するのも1つの方法です。これは、ページングシステムにいくらか余裕を与えることとなります。</p> <p>経験則によると、このパラメータは、システムが2-3秒で割り当てる必要がある量の2倍にします。このパラメータの適正值は負荷によって異なります。DBMS サーバーはデフォルトの設定で支障がないはずですが、ファイルシステムの入出力負荷が非常に大きい場合は、このパラメータを調整する必要があるかもしれません。</p> <p>負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。許容される最小値は512K バイトであり、getpagesize によって返されるページサイズに基づくページ数で表されます。</p>
コミットレベル	変更の可能性あり

desfree

説明	システム上で常時解放しておくべきメモリー容量を指定します。
データ型	符号なし整数
デフォルト	lotsfree / 2
範囲	<p>最小値は、256K バイトまたは物理メモリーの 1/128 のどちらか大きい方であり、getpagesize によって返されるページサイズに基づくページ数で表されます。</p> <p>最大値は物理メモリーのページ数です。この最大値は物理メモリーの 15% 以内であるべきです。シス</p>

	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、増やす前なら処理されたであろう要求が拒否されることがあります。 ■ NFS の非同期書き込みが、同期書き込みとして実行されます。 ■ スワッパーが本来より早く立ち上がり、そのスワッパーの動作が、積極的な動作をする方向に傾きます。 ■ システムに前もって読み込む実行可能ページの数が本来よりも少なくなる可能性があります。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	desfree / 2
範囲	最小値は、128K バイトまたは物理メモリーの 1/256 のどちらか大きい方であり、getpagesize によって返されるページサイズに基づくページ数で表されます。 最大値は物理メモリーのページ数です。この最大値は物理メモリーの 7.5% 以内であるべきです。システムは、「検証」のセクションで記述している場合以外は、この範囲を強制しません。
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、/etc/system ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検証	minfree が desfree より大きい場合、minfree は desfree / 2 に設定されます。メッセージは表示されません。
暗黙的制約	lotsfree が desfree よりも大きく、desfree が minfree よりも大きいという関係が常に維持されるようにする必要があります。
どのような場合に変更するか	一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。許容される最小値は 128K バイトであり、getpagesize によって返されるページサイズに基づくページ数で表されます。
コミットレベル	変更の可能性あり

throttlefree

説明	要求を満たせるだけのメモリーがある場合でも、メモリー割り当て要求ブロッキングをスリープ状態にするメモリーレベルを指定します。
データ型	符号なし整数
デフォルト	minfree
範囲	<p>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。</p> <p>最大値は物理メモリーのページ数です。この最大値は物理メモリーの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。</p>
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、 <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> に設定されず。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、/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	<p>/etc/system 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 tune_t_minarmem+100 よりも小さい場合、この値は tune_t_minarmem+100 へリセットされます。</p> <p>/etc/system 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、SHARE_MMU 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 (mlock、mlockall、および memcntl) 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>handspreadpages</code> です。
データ型	符号なし long
デフォルト	<code>fastscan</code>
範囲	1 からシステムの物理メモリーの最大ページ数
単位	ページ
動的か	はい。このパラメータを変更する場合、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>handspreadpages</code> の新しい値がいったん認識されると、 <code>reset_hands</code> はゼロに設定されます。
検証	値は物理メモリー容量と <code>handspreadpages value</code> のどちらか小さい方に設定されます。
どのような場合に変更するか	ページが再利用されるまで置いておく時間を長くする場合。この値を増やすと2つの段階の間の時間が長くなるため、ページが再利用されるまでの時間が増えます。
コミットレベル	変更の可能性あり

pages_before_pager

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

コミットレベル	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。
変更履歴	変更の可能性あり 詳細は、 215 ページの「maxpgio (Solaris 10 リリース)」 を参照してください。

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

swaps_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、97 ページの「`tsb_alloc_hiwater_factor`」を参照してください。

コミットレベル

変更の可能性あり

カーネルメモリーアロケータ

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

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
変更履歴	詳細は、 214 ページ の「 <code>ddi_msix_alloc_limit</code> 」を参照してください。

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 <code>maxphys</code> 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 (<code>sun4u</code> または <code>sun4v</code>) または 57,344 (<code>x86</code>)。ワイド転送をサポートする <code>sd</code> ドライバは 1,048,576 を使用します。 <code>ssd</code> 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 <code>MAXINT</code>
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	<code>raw</code> 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、 <code>maxphys</code> を変更してもパフォーマンスの向上は望めません。 UFS ファイルシステムとの間で入出力を行う際、常に大量 (64K バイト超) のデータの読み取り/書き込みが行われる場合も、このパラメータの変更を検討することをお勧めします。ファイルシステムは、連続性が向上するように最適化する必要があります。たとえば、シリンダグループのサイズを増やし、シリンダグループあたりの <code>i</code> ノード数を減らします。UFS では、転送する最大の入出力サイズに対して 1M バイトの内部制限が課せられています。

コミットレベル	変更の可能性あり
変更履歴	詳細は、215 ページの「 maxphys (Solaris 10 リリース) 」を参照してください。

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、select(3C) を参照してください。32 ビットアプリケーションコードは、より大きな <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

コミットレベル	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。 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、 <code>rlim_fd_max</code> 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、 <code>setrlimit()</code> 呼び出しを使用するか、プロセスを実行するシェルで <code>limit</code> 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	<code>rlim_fd_max</code> と比較します。 <code>rlim_fd_cur</code> が <code>rlim_fd_max</code> より大きい場合、 <code>rlim_fd_cur</code> は <code>rlim_fd_max</code> にリセットされます。
どのような場合に変更するか	1プロセス当たりのデフォルトのオープンファイル数が十分でない場合。この値を増やす意味は、プログラムで <code>setrlimit</code> を使用して自身で使用できる最大ファイル記述子数を増やす必要がなくなる点にあります。
コミットレベル	変更の可能性あり

一般的なファイルシステムパラメータ

ncsize

説明	ディレクトリ名検索キャッシュ (DNLC) のエントリ数を指定します。このパラメータは、UFS、NFS、および ZFS が、解決されたパス名の要素をキャッシュするときに使用します。 DNLC は、否定的な検索情報もキャッシュしません。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。32 ビットカーネルが動作しているシステムは <code>ncsize</code> に 36 バイトの構造体を、64 ビットカーネルが動作しているシステムは <code>ncsize</code> に 64 バイトの構造体をそれぞれ割り当てます。 <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
変更履歴	詳細は、 215 ページの「ncsize (Solaris 10 リリース)」 を参照してください。

rstchown

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

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す。または、元々有効だった場合に、無効にできます。しかし、有効にし、無効にし、再び有効にすると、ディレクトリキャッシュが最新の状態を表さないことがあります。
検証	いいえ
どのような場合に変更するか	ディレクトリキャッシングに既知の問題はありません。しかし、問題が生じた場合は、dnlc_dir_enable を 0 に設定してキャッシングを無効にしてください。
コミットレベル	変更の可能性あり

dnlc_dir_min_size

説明	1つのディレクトリでキャッシュする最小エントリ数を指定します。
	注 - 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。たとえば、UFSではディレクトリの最小は <code>ufs_min_dir_cache</code> バイトです(1エントリ当たり16バイトとして、およそ1024エントリ)。
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

UFSパラメータ

bufhwm および bufhwm_pct

説明	<p>入出力バッファのキャッシングに使用するメモリの最大量を指定します。バッファは、ファイルシステムのメタデータ(スーパーブロック、iノード、間接ブロック、ディレクトリ)の書き込みに使用されます。割り当てられるメモリー量(Kバイト単位)がbufhwmを超えるまで、必要に応じてバッファが割り当てられます。超過した時点で、要求を満たせるだけのバッファが回復されるまで、バッファキャッシュからメタデータが破棄されます。</p> <p>歴史的経緯により、bufhwmにはufs: 接頭辞は不要です。</p>
データ型	符号付き整数
デフォルト	物理メモリーの2%。
範囲	80Kバイトから物理メモリーの20%または2Tバイトのどちらか小さい方。つまり、bufhwm_pctは1から20にできます。
単位	bufhwm: Kバイト bufhwm_pct: 物理メモリーの%。
動的か	<p>いいえ。bufhwm および bufhwm_pct はシステムの初期設定時に限って評価され、ハッシュ容量が算出されます。これらのパラメータから算出された限度(バイト数)は、データ構造体に格納され、バッファの割り当てや解放に応じて、この値が調整されず。</p> <p>動作しているシステムのロック手順に従わずにこの値を調整すると、正しくない動作を招くおそれがあります。</p> <p>稼働中に bufhwm または bufhwm_pct を変更しても無効です。</p>
検証	<p>bufhwm が下限の 80 Kバイトに満たない場合、または上限(物理メモリーの20%、2Tバイト、またはカーネルヒープの最大量の1/4のいずれか小さい方)を超える場合は、上限にリセットされません。無効な値を試みると、システムコンソールと /var/adm/messages ファイルに次のメッセージが出力されます。</p> <pre>"binit: bufhwm (value attempted) out of range (range start..range end). Using N as default."</pre>

「value attempted」は、/etc/system ファイルに指定された値、またはカーネルデバッグを使用することによって指定された値です。Nは使用可能なシステムメモリーに基づいてシステムが計算した値です。

同様に、bufhwm_pct が1から20%という許容範囲外の値に設定された場合は、デフォルトの2%にリセットされます。その場合は、次のメッセージがシステムコンソールと /var/adm/messages ファイルに出力されます。

```
"binit: bufhwm_pct(value attempted) out of range(0..20).
      Using 2 as default."
```

bufhwm と bufhwm_pct がどちらもゼロ以外の値に設定されている場合は、bufhwm が優先されます。

どのような場合に
変更するか

バッファは必要になったときにのみ割り当てられるので、デフォルト値に対する負担増は、バッファハッシュヘッダー用に必要となる制御構造体の割り当てです。これらの構造体は、32ビットカーネルでは想定されるバッファ当たり52バイト、64ビットカーネルでは想定されるバッファ当たり96バイトを消費します。

512Mバイトの64ビットカーネルでは、ハッシュチェーン数は $10316 / 32 == 322$ であり、次の2の累乗の512まで拡大されます。したがって、ハッシュヘッダーは 512×96 バイト、すなわち48Kバイトを消費します。ハッシュヘッダー割り当てでは、バッファが32Kバイトであることが前提です。

バッファプール内でまだ割り当てられていないメモリー量を知るには、カーネルデバッグを使用して、カーネルの bfreelist 構造体を調べます。この構造体で調べるフィールドは b_bufsize であり、これが残っているはずのメモリー(バイト数)です。mdb コマンドを使用し、buf マクロで調べる例を示します。

```
# mdb -k
Loading modules: [ unix krtld genunix ip nfs ipc ]
> bfreelist::print "struct buf" b_bufsize
b_bufsize = 0x225800
```

6Gバイトのメモリーを装備したこのシステムでは、bufhwm のデフォルト値は122277です。実際に要求されるバッファサイズは通常、1Kバイトより大きいので、ヘッダー構造体の数を判断することはできません。しかし、一部の領域は、このシステムに割り当てられた制御構造体からうまく回収されることがあります。

512Mバイトのシステムでは、同じ構造体は、10144Kバイトのうち4Kバイトだけが割り当てられていないことを示します。ま

た、`kstat -n biostats` で `biostats` の `kstat` を調べると、このシステムでは、`buffer_cache_hits` と `buffer_cache_lookups` の割合も適切であることが分かったとします。これらの情報は、このシステムのデフォルト設定であることを示します。

コミットレベル 変更の可能性あり

ndquot

説明	割り当て対象となる UFS ファイルシステム用の割り当て構造体の数を指定します。このパラメータは、1 つまたは複数の UFS ファイルで割り当てが有効になっているときだけ適用されます。歴史的経緯により、 <code>ufs:</code> 接頭辞は不要です。
データ型	符号付き整数
デフォルト	$((\text{maxusers} \times 40) / 4) + \text{max_nprocs}$
範囲	0 から MAXINT
単位	割り当て構造体
動的か	いいえ
検証	ありません。値が大きすぎると、システムがハングします。
どのような場合に変更するか	デフォルトの割り当て構造体数では十分でない場合。このような状況は、コンソールやメッセージログに出力される次のメッセージから判別できます。
コミットレベル	<code>dquot table full</code> 変更の可能性あり

ufs_ninode

説明	メモリー内で維持すべき <code>i</code> ノードの数を指定します。 <code>i</code> ノードはファイルシステム単位ではなく、UFS 全体でキャッシュされます。
	この場合の鍵となるパラメータは <code>ufs_ninode</code> です。このパラメータを使用して、 <code>i</code> ノードキャッシュの処理に関係する 2 つの鍵となる境界値が計算されます。高位境界値は $\text{ufs_ninode} / 2$ 、下位境界値は $\text{ufs_ninode} / 4$ で計算されます。

システムがiノードの処理が終わると、次のどちらかが起こる可能性があります。

- iノードによって参照されるファイルがもはやシステムにないため、そのiノードが削除される。iノードが削除されると、その空間はiノードキャッシュに戻され、別のiノード(ディスクから読み込まれるか、新規ファイル用に作成されるもの)用に使用されます。
- ファイルは存在するが、実行プロセスに参照されていない。iノードはアイドルキューに入れます。参照されていたページはメモリーに残ります。

iノードをアイドルリングする場合、カーネルはこのアイドルリング処理を一定の時期まで先送りします。ファイルシステムがロギングファイルシステムの場合も、カーネルはiノードの削除を先送りします。2つのカーネルスレッドがこの先送り処理を引き受けます。それぞれのスレッドが一方のキューを処理します。

先送りされていた処理が終わると、システムはそのiノードを削除キューかアイドルキューに入れます。それぞれのキューには、そのキューを処理できるスレッドがあります。iノードがキューに入られると、キューの占有率が下位境界値と比較され、占有率が下位境界値を超えていると、そのキューに関連するスレッドが起こされます。キューが起こされると、スレッドがキューを調べ、iノードに結びつけられたページがあればディスクに書き出し、iノードを解放します。スレッドは、起こされた時にキューにあったiノードの50%を削除すると停止します。

アイドルスレッドの処理が負荷に追いつかない場合は、2つめのメカニズムが使用されます。システムは、vnodeを見つける必要があると、ufs_vgetルーチンを実行します。vgetは「最初に」アイドルキューの長さを調べます。長さが高位境界値を超えていると、アイドルキューから2つのiノード取り出し、アイドルリングします(ページをフラッシュし、iノードを解放する)。vgetは、自身が使用するiノードを取得する「前に」これを行います。

	システムは、コア内にページがないiノードをアイドルリストの先頭に置き、ページがあるiノードをアイドルリストの終わりに置くことによって最適化を図ります。しかし、リストの順序に関し、それ以外の処理は行いません。iノードは常にアイドルキューの先頭から削除されます。
	iノード全体がキューから削除されるのは、同期、アンマウント、または再マウントが行われるときだけです。
	歴史的経緯により、このパラメータにはufs: 接頭辞は必要ありません。
データ型	符号付き整数
デフォルト	ncsize
範囲	0からMAXINT
単位	iノード
動的か	はい
検証	ufs_ninodeが0以下の場合、この値はncsizeに設定されます。
どのような場合に変更するか	デフォルトのiノード数では足りない場合。kstat -n inode_cacheで報告されるmaxsize reachedフィールドがkstatのmaxsizeフィールドより大きい場合は、ufs_ninodeの値が小さすぎる可能性があります。iノードのアイドルリングが多すぎる場合も、問題になることがあります。 iノードのアイドルリングが多すぎるかどうかは、kstat -n inode_cacheを使用し、inode_cache kstatを調べることで判断できません。thread idlesはバックグラウンドスレッドがアイドルリングしたiノード数を、vget idlesはiノードを使用する前の要求プロセスによるアイドル数をそれぞれ表しています。
コミットレベル	変更の可能性あり

ufs_WRITES

説明	ufs_WRITES がゼロ以外の場合、1つのファイルに対する書き込み未処理のバイト数が調べられます。ufs_HWを参照し、書き込みを行うべきか、未処理のバイト数がufs_LWになるまで書き込みを延期すべきかが判定されます。未処理のバイト数のトータルはファイルごとに管理されるため、あるファイルの未処理のバイト数が限度を超えても、それが他のファイルに影響を与えることはありません。
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	UFSの書き込みスロットル(抑制)を全体的にオフにしたい場合。十分な入出力能力がない場合は、このパラメータを無効にすると、ディスクに対するサービスキューが長くなるおそれがあります。
コミットレベル	変更の可能性あり

ufs_LW および ufs_HW

説明	<p>ufs_HWでは、単一ファイル境界値の未処理バイト数を指定します。未処理のバイト数がこの値を上回り、ufs_WRITESが設定されていると、書き込みは延期されます。書き込みの延期は、書き込みを行うスレッドを、条件変数で眠らせることで行われます。</p> <p>ufs_LWは1つのファイルに対する未処理のバイト数の境界値です。これを下回ると、他の処理が休眠状態となっている原因の条件変数が切り替えられます。書き込みが終了し、バイト数がufs_LWを下回ると、条件変数が切り替わり、それによってその変数で待機しているすべてのスレッドが立ち上がり、それぞれの書き込みを行おうとします。</p>
----	---

データ型	符号付き整数
デフォルト	ufs_LWの場合は8 x 1024 x 1024、ufs_HWの場合は16 x 1024 x 1024
範囲	0からMAXINT
単位	バイト
動的か	はい
検証	なし
暗黙的制約	ufs_LWとufs_HWは、ufs_WRITESがゼロでないときだけ意味があります。たとえばufs_LWとufs_HWが近すぎると複数のスレッドが立ち上がってもいずれも書き込みを実行できないことがあったり、あるいはufs_LWとufs_HWが離れすぎていると複数のスレッドが必要以上に待たされることがあるなどの不要な問題を避けるために、ufs_HWとufs_LWはともに変更するようにしてください。
どのような場合に変更するか	ファイルシステムがストライプ化ボリュームから構成されている場合は、これらの値の変更を検討します。使用可能な合計帯域幅がufs_HWの現在の値を簡単に超える可能性があります。残念ながら、このパラメータはファイルシステムごとに設定されるものではありません。
	ufs_throttlesが通常ではない値のときにも、このパラメータの変更を検討するかもしれません。現在、ufs_throttlesにアクセスできるのは、カーネルデバッグを使用した場合だけです。
コミットレベル	変更の可能性あり

freebehind

説明	freebehind アルゴリズムを有効にします。このアルゴリズムが有効な場合、システムはメモリー使用率が高いときに順次入出力を検出すると、新しく読み取ったブロックに関してファイルシステム キャッシュを迂回します。
データ型	ブール型
デフォルト	1(有効)

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	freebehind アルゴリズムが頻繁に発生する場合。重要な順次システムファイル動作が予想されない場合、freebehind を無効にすると、大きさに関係なくすべてのファイルがファイルシステムのページキャッシュで維持される候補になります。さらに細かいチューニングについては、smallfile を参照してください。
コミットレベル	変更の可能性あり

smallfile

説明	<p>ファイルの大きさが超えると freebehind アルゴリズムでキャッシュに保持しない候補になる、しきい値を決定します。</p> <p>大容量メモリーシステムには、深刻なメモリー要求を行わずに、10M バイトのファイルを何千もキャッシュするのに十分なメモリーがあります。しかし、この状況はあくまでもアプリケーションに強く依存します。</p> <p>smallfile パラメータと freebehind パラメータの目的は、キャッシュによるメモリー不足を頻繁に引き起こすことなく、キャッシュ情報を再利用することです。</p>
データ型	符号付き整数
デフォルト	0
範囲	0 から 2,147,483,647
動的か	はい
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中程度の大きさのファイルを順次読み取り、バッファリングによる大きな利益が予想され、なおかつシステムが特にメモリー不足に

コミットレベル	陥っていない場合、 <code>smallfile</code> を増やします。中程度の大きさのファイルとは、32Kバイトから2Gバイトのファイルです。 変更の可能性あり
---------	--

ufs_delete_hiwat

説明	マウント済みの UFS ファイルシステムごとに、UFS ファイルシステム削除キューのサイズを指定します。
データ型	符号なし整数
デフォルト	0 (無制限)
範囲	0 から 2^{64}
動的か	いいえ
検証	なし
どのような場合に変更するか	このパラメータを変更すると、UFS 削除キューのパフォーマンスが影響を受ける可能性があるため、一般に変更することはお勧めしません。ただし、キューに格納されたすべての操作を実行する必要がある場合に、削除キューのサイズによって同期操作のパフォーマンスが影響を受けます。 構成によっては、UFS ファイルシステムがより高速に同期操作に応答するように、削除キューのサイズを制限することを検討してください。この場合、値を数十万に設定するようにしてください。たとえば、262,144 と入力します。
コミットレベル	変更の可能性あり
変更履歴	詳細は、 214 ページ の「 <code>ufs_delete_hiwat</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。 <code>tmp_memalloc: tmpfs over memory limit</code> TMPFS がデータ構造体に現在使用しているメモリー量は、 <code>tmp_kmemspace</code> 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
変更履歴	詳細は、 215 ページの「tmpfs:tmpfs_maxkmem (Solaris 10 リリース)」 を参照してください。

tmpfs:tmpfs_minfree

説明	TMPFS がシステムの他の部分のために残しておくスワップ空間の最小量を指定します。
データ型	符号付き long
デフォルト	256

範囲	0からスワップ空間サイズの最大値
単位	ページ
動的か	はい
検証	なし
どのような場合に 変更するか	TMPFSが大量に使用されるシステムで適度なスワップ空間を維持するために、この値を増やすことができます。次のメッセージがコンソールやメッセージファイルに出力された場合は、使用量がこの限度に達したことを示しています。 <code>fs-name: File system full, swap space limit exceeded</code>
コミットレベル	変更の可能性あり

仮想端末

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

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

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

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

構成処理では、次の3つの関連する変数が使用されます。

- `pt_cnt` – pty 数のデフォルトの最大値
- `pt_pctofmem` – pty サポート構造体専用に見え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 `pct_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を留意してください。

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

システムメモリーの特定の割合を`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%を割り当てる場合、`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 <code>maxpid</code>
単位	ログイン/ウィンドウ
動的か	いいえ

検証	なし
どのような場合に変更するか	システムにリモートからログインできるユーザーの数を明示的にコントロールしたい場合
コミットレベル	変更の可能性あり

pt_pctofmem

説明	データ構造体が /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。64 ビットカーネルのシステムでは /dev/pts エントリ当たり 176 バイトを消費します。32 ビットカーネルのシステムでは /dev/pts エントリ当たり 112 バイトを消費します。
データ型	符号なし整数
デフォルト	5
範囲	0 から 100
単位	%
動的か	いいえ
検証	なし
どのような場合に変更するか	システムにログインできるユーザーの数を制限するか増やしたい場合。ゼロを指定すると、リモートユーザーがシステムにログインすることはできません。
コミットレベル	変更の可能性あり

pt_max_pty

説明	システムが提供する ptys の最大数を指定します。
データ型	符号なし整数
デフォルト	0 (システムが定義した最大数を使用する)
範囲	0 から MAXUINT
単位	ログイン/ウィンドウ
動的か	はい

検証	なし
暗黙的制約	pt_cnt 以上にすべきです。値が検査されるのは、割り当てられた ptys 数が pt_cnt の値を超過してからです。
どのような場合に変更するか	システムが、構成値に基づいてより多くのログインをサポートできる場合であっても、サポートするログイン数の絶対的な上限を設定したい場合。
コミットレベル	変更の可能性あり

STREAMS パラメータ

nstrpush

説明	STREAM に追加 (格納) できるモジュールの数を指定します。
データ型	符号付き整数
デフォルト	9
範囲	9 から 16
単位	モジュール
動的か	はい
検証	なし
どのような場合に変更するか	ソフトウェアベンダーの指定がある場合、STREAM が許可されているプッシュカウントを超えても、メッセージは出されません。プッシュを試みたプログラムに EINVAL という値が返されます。
コミットレベル	変更の可能性あり

strmsgsz

説明	1 つのシステム呼び出しで STREAM に渡し、メッセージのデータ部分に格納できる最大バイト数を指定します。このサイズを超える write
----	--

	は、複数のメッセージに分割されます。詳細は、 write(2) 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	<code>putmsg</code> 呼び出しから <code>ERANGE</code> 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 <code>STREAM</code> 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 <code>MAXINT</code>
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 <code>ERANGE</code> が返されます。
コミットレベル	変更の可能性あり

System V メッセージキュー

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

Oracle Solaris 10 リリースの System V メッセージキューについては、[19 ページ](#)の「[System V IPC 構成](#)」を参照してください。

これらのシステムリソースをチューニングする方法については、『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』の第 6 章「[リソース制御 \(概要\)](#)」を参照してください。

廃止された System V メッセージキューの古い情報は、[218 ページ](#)の「[廃止または削除されたパラメータ \(Oracle Solaris 10\)](#)」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。

Oracle Solaris 10 リリースでのセマフォリソースの変更については、[19 ページ](#)の「[System V IPC 構成](#)」を参照してください。

Oracle Solaris 10 リリースでの新しいリソース制御の使用の詳細は、『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』の第 6 章「[リソース制御 \(概要\)](#)」を参照してください。

廃止された System V セマフォパラメータの古い情報は、[218 ページ](#)の「[廃止または削除されたパラメータ \(Oracle Solaris 10\)](#)」を参照してください。

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。

Oracle Solaris 10 リリースの共有メモリーリソースの変更については、[19 ページ](#)の「[System V IPC 構成](#)」を参照してください。

Oracle Solaris 10 リリースでの新しいリソース制御の使用方法の詳細は、『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』の第 6 章「[リソース制御 \(概要\)](#)」を参照してください。

廃止された System V 共有メモリーパラメータの古い情報は、[218 ページ](#)の「[廃止または削除されたパラメータ \(Oracle Solaris 10\)](#)」を参照してください。

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

pr_segpg_disable

説明

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

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

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

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

データ型	ブール型
デフォルト	1 (無効)
範囲	0 (有効) および 1 (無効)
動的か	いいえ
検証	いいえ
どのような場合に変更するか	ロック済みまたはビジー状態の (高負荷 I/O) ページが保留中のページ回収キューに格納されると、ISM で所有されている可能性のある保留中ページを回収

させるために、ページ回収スレッドによって `segp_cache` がフラッシュされます。`segp_cache` を定期的または繰り返しフラッシュすると、メモリー負荷の高いマシンのボトルネックとなる可能性があります。

遅延の影響を受けるデータベースまたは共有メモリーが大きいアプリケーションを使用している場合は、このパラメータを無効にして、`segp` キャッシュのフラッシュを完全にスキップすることを検討してください。

回収できないロック済みのカーネルページの現象は次のとおりです。

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

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

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

コミットレベル

変更の可能性あり

スケジューリング

rechoose_interval

説明

プロセスが最後に実行していた CPU に対するすべての親和性を失ったとみなされるまでの、クロック刻みの数。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ず。このパラメータは、タイムシェアリングクラスのスレッドに対してのみ意味を持ちます。リアルタイムスレッドは、最初の使用可能な CPU に対してスケジュールされます。

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

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	32
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

consistent_coloring

説明	UltraSPARC プラットフォームでさまざまなページ配置ポリシーを使用する機能が利用できません。ページ配置ポリシーは、L2 キャッシュの使用
----	--

が最適化されるように物理ページアドレスを割り当てようとするものです。デフォルトアルゴリズムとしてどのアルゴリズムが選択されたとしても、特定のアプリケーション群にとって、そのアルゴリズムが別のアルゴリズムよりも適していない可能性があります。このパラメータは、システムのすべてのプロセスに適用される配置アルゴリズムを変更します。

メモリーは、L2 キャッシュのサイズに基づいて区画に分割されます。マップされていないページでページフォルトが最初起こると、ページ配置コードは1つの区画から1つのページを割り当てます。選択されるページは、次の3つのアルゴリズムのどれが使用されているかによって異なります。

- ページ彩色 - ページが選択される区画は、仮想アドレスのさまざまなビットに基づいて決められます。このアルゴリズムを使用するには、`consistent_coloring` をゼロに設定します。このアルゴリズムでは、プロセス別の履歴はありません。
- 仮想アドレス=物理アドレス - プログラム内の連続するページに、連続する区画からページを選択します。このアルゴリズムを使用するには、`consistent_coloring` に1を設定します。このアルゴリズムでは、プロセス別の履歴はありません。
- 区画飛び越し - プログラム内の連続するページに、通常、1つおきの区画からページを割り当てます。ただし、このアルゴリズムは、ときには2つ以上の区画を飛び越すこともあります。このアルゴリズムを使用するには、`consistent_coloring` に2を設定します。各プロセスは、無作為に選択された区画から開始し、割り当てられた最後の区画のプロセスごとの記録が保管されます。

動的か
検査

はい

ありません。値が2より大きいと、コンソールに一連の「WARNING: AS_2_BIN: bad consistent coloring value」メッセージが表示されます。その後、ただちにシステムが停止します。復旧には、電源を再投入する必要があります。

どのような場合に変更するか	システムの主な作業負荷が、長い時間動作するハイパフォーマンスコンピューティング (HPC) アプリケーションである場合。この値を変更すると、パフォーマンスが向上することがあります。ファイルサーバーやデータベースサーバー、それに多数のアクティブプロセスが動作するシステム (たとえばコンパイルやタイムシェアリングサーバーなど) では、この値を変更しても効果はありません。
コミットレベル	変更の可能性あり

tsb_alloc_hiwater_factor

説明	<p>tsb_alloc_hiwater を初期化して、変換ストレージバッファ (TSB) に割り当てることができる物理メモリー量に、次のように上限を設けます。</p> $\text{tsb_alloc_hiwater} = \text{物理メモリー (バイト数)} / \text{tsb_alloc_hiwater_factor}$ <p>TSB に割り当てられたメモリーが tsb_alloc_hiwater の値と等しい場合、TSB メモリー割り当てアルゴリズムは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。</p> <p>この係数を使用して tsb_alloc_hiwater の値を増やす場合は、注意が必要です。システム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。</p>
データ型	整数
デフォルト	32
範囲	1 から MAXINIT
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

変更履歴 詳細は、216 ページの「[default_tsb_size \(Solaris 10 リリース\)](#)」を参照してください。

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> は無視されません。
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せん。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
変更履歴	詳細は、216 ページの「 enable_tsb_rss_sizing (Solaris 10 リリース) 」を参照してください。

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	<p>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。</p> <p>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。</p> <p>TSB の動きは、<code>trapstat -T</code> コマンドでモニターできます。</p>
コミットレベル	変更の可能性あり
変更履歴	詳細は、 216 ページの「tsb_rss_factor (Solaris 10 リリース)」 を参照してください。

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	<p>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。</p> <p>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</p>
----	---

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

Solaris ボリュームマネージャーのパラメータ

md_mirror:md_resync_bufsz

説明	RAID 1 ボリューム (ミラー) の再同期に使用するバッファのサイズを 512 バイト単位のブロック数で設定します。設定する値を大きくすると、再同期の速度が速くなります。
データ型	整数
デフォルト	デフォルト値は 128 です。大規模なシステムでは、ミラー再同期速度を上げるためにより高い値を使用することがあります。
範囲	128 から 2048
単位	ブロック (512 バイト)
動的か	いいえ
検証	なし
どのような場合に変更するか	Solaris ボリュームマネージャーの RAID 1 ボリューム (ミラー) を使用していて、ミラー再同期の速度を上げたい場合。全体的なシステムパフォーマンスに対し十分なメモリーがある場合は、この値を増加させても他のパフォーマンス上の問題は発生しません。 ミラー再同期の速度を上げる場合は、十分なパフォーマンスとなるまで、このパラメータの値を (128 ブロックずつ) 増加させてください。かなり大きなシステムや新しいシステムでは、この値を 2048 とするのが適当と考えられます。旧式のシステムでこの値を高く設定するとシステムがハングアップする場合があります。
コミットレベル	変更の可能性あり

md:mirrored_root_flag

説明	複製定数に関する Solaris ボリュームマネージャー要件をオーバーライドし、状態データ
----	---

	ベースの有効な複製が利用できる場合は、Solaris ボリュームマネージャーを強制的に起動させます。
	デフォルト値は「無効」で、すべての複製の過半数を使用可能にして同期させてからでなければ、Solaris ボリュームマネージャーは起動しません。
データ型	ブール値
デフォルト	0(無効)
範囲	0(無効)、1(有効)
動的か	いいえ
検証	なし
どのような場合に変更するか	このパラメータの使用はサポートされません。 次の3つの条件がすべて当てはまる場合、危険を承知のうえでこのパラメータを有効にする Solaris ボリュームマネージャーのユーザーもいます。 <ul style="list-style-type: none"> ■ root (/) またはその他のシステムに不可欠なファイルシステムがミラー化されている場合 ■ 使用できるディスクまたはコントローラが2つだけである ■ システムの無人リブートが必要である このパラメータを有効にすると、システムの状態(どちら側のミラーが良好または「保守(Maintenance)」状態なのかなどを含む)が正確に反映されていない古い複製を使用して、システムがブートする可能性があります。その結果、データが破損したり、システムが破損したりすることもあります。 このパラメータを変更するのは、データの一貫性や整合性よりシステムの可用性の方が重要な場合に限定してください。障害が発生していないかどうか、注意深くシステムをモニターしてください。障害、保守(Maintenance)、またはホットスワップボリュームの数をできるだけ低く抑えることによって、危険性を軽減できます。

コミットレベル

状態データベースの複製については、『Solaris Volume Manager 管理ガイド』の第6章「状態データベース(概要)」を参照してください。

変更の可能性あり

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。

- 108 ページの「ZFS のチューニングの考慮事項」
- 108 ページの「ZFS ARC パラメータ」
- 110 ページの「ZFS ファイルレベルプリフェッチ」
- 111 ページの「ZFS デバイスの入出力キューの深さ」
- 112 ページの「ZFS およびキャッシュフラッシュ」
- 114 ページの「ZFS メタデータの圧縮」
- 115 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 ZFS 管理ガイド](#)』の第11章「[推奨のOracle Solaris ZFS プラクティス](#)」を参照してください。
- 別途記載のないかぎり、チューニング可能パラメータはグローバルであり、システム全体にわたってZFSの動作に影響を及ぼします。

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。109 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	物理メモリーの 1/32 または 64M バイトのどちらか大きい方の値 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

コミットレベル	し、ZFS は <code>zfs_arc_min</code> の値を下回って縮小することはありません。一般には、デフォルト値を変更する必要はありません。
変更履歴	変更の可能性あり 詳細は、213 ページの「 <code>zfs_arc_min</code> 」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。108 ページの「 <code>zfs_arc_min</code> 」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、 <code>physmem</code> から 1G バイトを引いた値
範囲	64M バイトから <code>physmem</code>
単位	バイト
動的か	いいえ
検証	はい、範囲が検証されます。
どのような場合に変更するか	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、213 ページの「 <code>zfs_arc_max</code> 」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://www.oracle.com/technetwork/java/index.html>

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

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

データ型

ブール型

デフォルト

0 (有効)

範囲

0 (有効) または 1 (無効)

動的か

はい

検証

いいえ

どのような場合に変更するか	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

ZFS およびキャッシュフラッシュ

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) 実行しません。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。

`zfs_nocacheflush`

説明	このパラメータは、システム全体についての ZFS 書き込みキャッシュフラッシュを制御します。
	Oracle の Sun ハードウェアでは、このパラメータをチューニングしないでください。キャッシュフラッシュをチューニングする必要がある場合は、ハードウェアデバイスごとにチューニングすることを検討してください。下の一般的な指示を参照してください。ZFS によって送信されたキャッシュフラッシュを無視するようストレージデバイスに指示するための操作方法については、ストレージベンダーにお問い合わせください。
データ型	ブール型
デフォルト	0
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	キャッシュフラッシュチューニングは、ログデバイスとして使用した場合の一部の SSD のパフォーマンスを高めることが最近わかりました。このチューニング構文を <code>sd.conf</code> に含めることができますが、ベンダー/製品につき 1 つの <code>sd-config-list</code> エントリのみにする必要があります。例:

```
sd-config-list = "ATA TX43E10100GB0LSI", "throttle-max:32, disksort:false, cache-nonvolatile:true";
```

最終手段として、ZFS に公開されているすべての LUN が、NVRAM 保護されたストレージアレイから

のものであって、保護されない 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、`zfs_nocacheflush` を設定することによって、フラッシュリクエストを実行しないように ZFS をチューニングすることができます。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。

コミットレベル

変更の可能性あり

ZFS メタデータの圧縮

`zfs_mdcomp_disable`

説明	このパラメータは ZFS メタデータの圧縮を制御します (間接ブロックのみ)。ZFS データブロック圧縮は、ファイルシステムごとに設定できる ZFS <code>compression</code> プロパティによって制御されます。
データ型	ブール型
デフォルト	0
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	一般的に、メタデータ圧縮は有効にするようにしてください。メタデータを扱うことの多い負荷の場合、このデフォルトによって、CPU によるわずかな追加計算と引き換えに、いくらかの容量 (数パーセント) が得られることが期待されます。しかし、メタデータの圧縮をオンにするための、より大きな動機づけが存在します。数百万個ものオブジェクトに拡大したあと、わずか数個に縮小する

ディレクトリの場合、メタデータ圧縮によって大量の容量(10倍を超える)が節約されます。

フラッシュストレージデバイスに対する入出力は 4K バイトの境界に合わせられています。メタデータ圧縮が有効になると、フラッシュストレージデバイスに対する入出力の位置が合わなくなることがあります。フラッシュデバイスをプライマリストレージ用に使っている場合、入出力の位置合わせの問題を解決するためにメタデータ圧縮を無効にすることを検討することができます。フラッシュデバイスに別個のログデバイスを使用することは、位置合わせの問題による影響を受けません。

コミットレベル

変更の可能性あり

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 80% 以下に維持してください。

Oracle データベース用の ZFS のチューニング

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

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

- 最新の **Oracle Solaris** リリースを実行中であることを確認します

Oracle Solaris 10 9/10 リリースを最低限の開始点として、最新の Oracle Solaris 10 または Oracle Solaris 11 リリースを起動します。

- 必要な場合、**ZFS ストレージプール用の LUN** を作成します

使用しているストレージレイ用ツールを使用して、ZFS ストレージプールに提供される LUN を作成します。または、ミラー化された ZFS ストレージプール用にディスク全体を使用することを検討します。詳細については、『[Oracle Solaris ZFS 管理ガイド](#)』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は、『Oracle Solaris ZFS 管理ガイド』の「ZFS のプロパティの概要」を参照してください。

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

```
# zfs create -o recordsize=8k -o mountpoint=/my_db_path/index dbpool/index
# zfs set logbias=throughput dbpool/index
# zfs get primarycache,recordsize,logbias dbpool/index
NAME                PROPERTY          VALUE             SOURCE
dbpool/index        primarycache      all               default
dbpool/index        recordsize        8K               local
dbpool/index        logbias           throughput        local
```

- 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す。

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

- アーカイブプール内に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]ssd_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:sd_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す。`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 ZFS 管理ガイド』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ないように、ARC サイズに上限を設定してもかまいません。
- ログ用の別個のプールを作成します。
- my.cnf ファイル内にデータおよびログ用の別のパスを設定します。
- データファイルを作成する前に、InnoDB データファイルに対して ZFS recordsize プロパティを 16K に設定し、InnoDB ログについてはデフォルトの recordsize 値を使用します。

◆◆◆ 第 4 章

NFS チューニング可能パラメータ

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

- 121 ページの「NFS 環境のチューニング」
- 122 ページの「NFS モジュールのパラメータ」
- 155 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、24 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>しかし、32 ビットの Solaris カーネルが動作しているシステムでは、タイムスタンプの値は符号付きの 32 ビット long です。このため、タイムスタンプが 1970 年 1 月 1 日より前の表示 (つまり「昔」の) になっていることがあります。</p> <p>64 ビットの Solaris カーネルが動作しているシステムでの問題は、これとは多少異なります。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>
----	--

32 ビットから 64 ビットに変換するとき、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、時間値に符号を付けるべきではありません。この問題は、フルの 32 ビット時間値を無効にすることによって解決できます。

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

nfs:nfs_cots_timeo

説明	トランスポートプロトコルとして TCP などの接続型トランスポートを使用している、NFS バージョン 2 でマウントされたファイルシステムのデフォルトの RPC タイムアウトを制御します。
データ型	符号付き整数 (32 ビット)
デフォルト	600 (60 秒)
範囲	0 から $2^{31} - 1$
単位	1/10 秒
動的か	はい。ただし、ファイルシステムの RPC タイムアウトは、ファイルシステムのマウント時に設定され

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

nfs:nfs3_cots_timeo

説明	<p>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。</p>
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	<p>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。</p>
検証	なし
どのような場合に変更するか	<p>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。</p>

クライアントが無用にタイムアウトしてしまうことを防ぐには、このパラメータを増やします。ただし、値の範囲が非常に広いので、この値を大きくしすぎると、再転送が長い間検出されないおそれがあります。

コミットレベル 変更の可能性あり

nfs:nfs4_cots_timeo

説明

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

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

データ型 符号付き整数 (32 ビット)

デフォルト 600 (60 秒)

範囲 0 から $2^{31} - 1$

単位 1/10 秒

動的か はい。ただし、このパラメータは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。

検証 なし

どのような場合に変更するか

TCP では、要求と応答が適切に転送されるよう機能します。しかし、特に遅いネットワークにおいて往復時間が非常に長くなると、NFS バージョン 4 のクライアントがタイムアウトになってしまう可能性があります。

クライアントが無用にタイムアウトしてしまうことを防ぐには、このパラメータを増やします。ただ

し、値の範囲が非常に広いため、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モニタリングし、RPCタイムアウトと読み取り/書き込みの転送サイズを調整することによって、再転送の回数を減らそうとするものです。

データ型

整数 (32 ビット)

デフォルト

1 (有効)

範囲

0 (無効)、1 (有効)

動的か

はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。

検証

なし

どのような場合に変更するか

このパラメータは変更しないでください。

コミットレベル

変更の可能性あり

nfs:nfs3_dynamic

説明

UDPなどのコネクションレストランスポートを使用する、NFSバージョン3でマウントされたファイルシステムに対して、「動的再転送」と呼ばれる機能を有効にするかどうかを制御します。この機能は、サーバーの応答時間をモニタリングし、RPC

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

nfs:nfs_lookup_neg_cache

説明	NFS バージョン 2 でマウントされたファイルシステムに対して、ネガティブ名前キャッシュを使用するかどうかを制御します。このネガティブ名前キャッシュを使用すると、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、[143 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 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

プール値

動的か

はい

検証

なし

どのような場合に変更するか	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。
コミットレベル	変更の可能性あり

nfs:nfs4_lookup_neg_cache

説明	NFS バージョン 4 でマウントされたファイルシステムに対して、ネガティブ名前キャッシュを使用するかどうかを制御します。このネガティブ名前キャッシュを使用すると、ルックアップされたファイル名が存在しない場合には、そのファイル名が記録されます。このキャッシュは、存在しないことがすでにわかっているファイル名のルックアップ要求が、ネットワークを介して行われないようにする目的で使用されます。
データ型	整数 (32 ビット)
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
単位	ブール値
動的か	はい
検証	なし
どのような場合に変更するか	このキャッシュが正しく機能するためには、ネガティブエントリが使用前に厳密に検証されなければなりません。整合性メカニズムは、読み取り専用でマウントされたファイルシステムに関しては多少緩和されています。これは、サーバー上のファイルシステムが変更されないか、変更されるにしてもそのペースが緩やかであり、クライアントへの伝達がゆっくりであ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、[143 ページ](#)の「`nfs:nfs_disable_rddir_cache`」を参照してください。

コミットレベル

変更の可能性あり

`nfs:nfs_max_threads`

説明

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

非同期に実行できる操作には、先読みのための読み取り、`readdir` 先読みのための `readdir`、`putpage` および `pageio` 操作のための書き込み、コミット、およびクライアントがファイルの使用を停止したときに実行するクリーンアップ操作のための非アクティブ化があります。

データ型

符号なし short

デフォルト

8

範囲

0 から $2^{15} - 1$

単位

スレッド

動的か

はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。

検証

なし

どのような場合に変更するか

ある時点で存在する同時入出力操作の数を増やしたり、減らしたりする場合。たとえば、帯域幅が非常に狭いネットワークでは、NFS クライアントによるネットワークの過負荷を防止するためにこの値を減

らすことがあるかもしれません。また、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント/サーバ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、`readdir` 先読みのための `readdir`、`putpage` および `pageio` 要求のための書き込み、およびコミットがあります。

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、

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、読み取りのストループットが向上します。個々の先読み要求は、通常、ファイルデータの 1 論理ブロックに対するものです。

データ型

整数 (32 ビット)

デフォルト

4

範囲

0 から $2^{31} - 1$

単位

論理ブロック

動的か

はい

検証

なし

どのような場合に変更するか

特定のファイルに対してある時点で存在する先読み要求の数を増やしたり、減らしたりする場合。たとえば、ネットワークの帯域幅が非常に狭い場合やクライアントのメモリーが少ない場合は、NFS クライアントによるネットワークの過負荷やシステムメモリーの使いすぎを防止するために、この値を減らすことができるかもしれません。また、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント/サーバ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 (144 ページの「 nfs:nfs3_bsize 」を参照)
動的か	はい
検証	なし
どのような場合に変更するか	特定のファイルに対してある時点で存在する先読み要求の数を増やしたり、減らしたりする場合。たとえば、ネットワークの帯域幅が非常に狭い場合やクライアントのメモリーが少ない場合は、NFS クライアントによるネットワークの過負荷やシステムメモリーの使いすぎを防止するために、この値を減らすことができるかもしれません。または、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント/サーバーリソースをいっそう有効に活用できるようになります。
コミットレベル	変更の可能性あり
変更履歴	詳細は、216 ページの「 nfs:nfs3_nra (Solaris 10 リリース) 」を参照してください。

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 (145 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。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:nrnode

説明	NFS クライアントの <code>rnode</code> 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 <code>rnode</code> 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。 <code>rnode</code> 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。 <code>rnode</code> 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
----	--

	<p><code>rnode</code> は <code>vnode</code> と 1 対 1 で対応しています。 <code>vnode</code> には、ファイルデータがキャッシュされます。</p> <p>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 <code>rnode</code> を維持しようとしています。 <code>rnode</code> 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。</p>
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 <code>nrnode</code> の値が <code>ncsize</code> パラメータの値に設定されるべきであることを示しています。実際、 <code>nrnode</code> の値が正でないと、 <code>nrnode</code> には <code>ncsize</code> が設定されません。
範囲	1 から $2^{31} - 1$
単位	<code>rnode</code>
動的か	いいえ。この値は、 <code>/etc/system</code> ファイルにパラメータを追加するか、パラメータを変更し、その後システムをリブートすることによってのみ変更できます。
検証	<code>rnode</code> キャッシュが使用可能なメモリーの 25% を超えないような最大値をシステムは強制します。
どのような場合に変更するか	<p><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。</p>

推奨はできませんが、`nrnode` の値を 1 に設定すると、`rnode` キャッシュを事実上無効にできます。この値は 1 `rnode` 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 `READDIR` 要求で 1024 バイトより大きいディレクトリ情報を求めるものが正しく処理されな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

以前の一部の NFS サーバーでは、NFS バージョン 3 の READDIR 要求で 1024 バイトより大きいディレクトリ情報を求めるものが正しく処理されな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。

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	long 整数 (32 ビットプラットフォームでは32 ビット、64 ビットプラットフォームでは64 ビット)
デフォルト	5 秒
範囲	32 ビットプラットフォームでは0から $2^{31} - 1$ 64 ビットプラットフォームでは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 syslog に記録するか、それともシステムコンソールだけに記録するかを制御します。このパラメータは、NFS バージョン2、3、および4のクライアントのメッセージに影響を与えます。
データ型	整数 (32 ビット)
デフォルト	0 (システムコンソールと syslog)
範囲	0 (システムコンソールと syslog) または 1 (システムコンソール)
単位	ブール値
動的か	はい

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

nfs:nfs_disable_rddir_cache

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

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

- 130 ページの「nfs:nfs_lookup_neg_cache」
- 131 ページの「nfs:nfs3_lookup_neg_cache」
- 132 ページの「nfs:nfs4_lookup_neg_cache」

コミットレベル

変更の可能性あり

nfs:nfs3_bsize

説明

NFS バージョン 3 のクライアントが使用する論理ブロックサイズを制御します。このブロックサイズは、クライアントが入出力を行うときにサーバーに対して読み取りや書き込みを行うデータ量を表します。

データ型

符号なし整数 (32 ビット)

デフォルト

32,768 (32K バイト)

範囲

0 から $2^{31} - 1$

単位

バイト

動的か

はい。ただし、ファイルシステムのブロックサイズは、ファイルシステムのマウント時に設定されません。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、nfs:nfs3_max_transfer_size 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、read-ahead、putpage、pageio、readdir-ahead 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 WRITE 要求をほぼ同時に送信することに依存します。キューから要求を1度に1つずつ取り出したのでは、クライアントのパフォーマンスを高めるために設けられたこのサーバー機能が生かされません。</p> <p>そこで、このパラメータを使用して、タイプが変更されるまでに送信される、各要求タイプの要求数を制御します。</p>
データ型	符号なし整数 (32 ビット)
デフォルト	1
範囲	0 から $2^{31} - 1$
単位	非同期要求
動的か	はい。ただし、ファイルシステムに対するクラスタ設定は、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	ありません。しかし、このパラメータに0を設定すると、キューに入れられている特定の要求タイプの要求がすべて処理されてから、次のタイプが処理されます。これによって、アルゴリズムの公平性の部分が実質的に無効にされます。
どのような場合に変更するか	あるタイプについて次のタイプへ移行する前に生成される非同期要求の数を増やす場合。これに

	よって、クライアントからの要求のクラスタに依存するサーバーの機能が生かされる場合があります。
コミットレベル	変更の可能性あり

nfs:nfs3_async_clusters

説明	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。
	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。
	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	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。 しかし、書き込みの一括化(書き込みをまとめる)などの NFS バージョン4サーバーの機能の中には、既存の NFS バージョン4クライアントの一定の動作に依存するものがあります。特に、この機能では、クライアントが複数の WRITE 要求をほぼ同時に送信することに依存します。キューから要求を1度に1つずつ取り出したのでは、クライアントのパフォーマンスを高めるために設けられたこのサーバー機能が生かされません。 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

検証	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。 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。

- 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 `nrnode` 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 `nacache` の値に `nrnode` 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、`/etc/system` 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

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

nfs:nfs3_jukebox_delay

説明	NFS バージョン3クライアントが前回の要求で NFS3ERR_JUKEBOX エラーを受け取ってから、新しい要求を送信するまでに待機する時間の長さを制御します。NFS3ERR_JUKEBOX エラーは、通常、何らかの理由でファイルが一時的に使用できないときにサーバーから返されます。このエラーは、通常、階層型ストレージ、CD やテープといったジュークボックスに関連しています。
データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	1000 (10 秒を 10 秒 * 100Hz で表す)
範囲	32 ビットプラットフォームでは 0 から $2^{31} - 1$ 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

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 には、1 データグラム当たり 64K バイトという厳しい制限値があります。この 64K バイトには、要求のデータ部分のほかに、RPC ヘッダーやその他の NFS 情報も含まれている必要があります。この制限値が大きすぎると、UDP エラーのためにクライアントとサーバーの通信に問題が発生することがあります。

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:nfs3_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:nfs3_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、[144 ページ](#) の「`nfs:nfs3_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: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、[152 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	<p>ネットワークを介して送信するデータのサイズをチューニングする場合。通常、<code>nfs:nfs4_bsize</code> パラメータもこのパラメータの変更が反映されるように更新すべきです。</p> <p>たとえば、転送サイズを 32K バイトより大きい値に増やす場合は、その増加した値が反映されるように <code>nfs:nfs4_bsize</code> を更新します。そうしないと、ネットワーク上で送信される要求のサイズは変わりません。詳細は、145 ページ の「<code>nfs:nfs4_bsize</code>」を参照してください。</p> <p>転送サイズをデフォルト値より小さくする場合は、<code>mount</code> コマンドの <code>-wsize</code> または <code>-rsize</code> オプションをファイルシステム単位で使用します。</p>
コミットレベル	変更の可能性あり

`nfs:nfs3_max_transfer_size_clts`

説明	<p>NFS バージョン 3 の UDP を介した <code>READ</code>、<code>WRITE</code>、<code>REaddir</code>、または <code>REaddirplus</code> 要求のデータ部分の最大サイズを制御します。このパラメータは、サーバーが返す要求の最大サイズとクライアントが生成する要求の最大サイズの両方を制御します。</p>
データ型	整数 (32 ビット)
デフォルト	32,768 (32 K バイト)
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検証	ありません。しかし、サーバー側の最大転送サイズに 0 を設定すると、クライアントはおそらく、誤作動するか、単にサーバーに要求を送信しないかのどちらかになる可能性があります。
どのような場合に変更するか	このパラメータは変更しないでください。

コミットレベル 変更の可能性あり

nfs:nfs3_max_transfer_size_cots

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

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	<p>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。</p> <p>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。</p>
コミットレベル	変更の可能性あり

rpcmod:clnt_idle_timeout

説明	クライアントとサーバー間の接続が終了するまでにアイドル状態を維持できる、クライアント側の時間の長さを制御します。
データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	300,000 ミリ秒 (5分)
範囲	<p>32 ビットプラットフォームでは 0 から $2^{31} - 1$</p> <p>64 ビットプラットフォームでは 0 から $2^{63} - 1$</p>
単位	ミリ秒
動的か	はい
検証	なし

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

rpcmod:svc_idle_timeout

説明	クライアントとサーバー間の接続が終了するまでにアイドル状態を維持できる、サーバー側の時間の長さを制御します。
データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	360,000 ミリ秒 (6分)
範囲	32 ビットプラットフォームでは 0 から $2^{31} - 1$ 64 ビットプラットフォームでは 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	1024
範囲	1 から $2^{31} - 1$
単位	要求
動的か	キャッシュのサイズは動的に決められますが、キャッシュへの高速アクセスを可能にするハッシュキューのサイズは静的に決められま

	<p>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	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>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、コネクションレスタランスポートのクライアントのタイムアウトは比較的短く、1秒から20秒くらいです。</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、TCP、IP、UDP、SCTP などのインターネットプロトコル群の各種パラメータについて説明します。

- 162 ページの「IP チューニング可能パラメータ」
- 170 ページの「TCP チューニング可能パラメータ」
- 189 ページの「UDP チューニング可能パラメータ」
- 192 ページの「IPQoS チューニング可能パラメータ」
- 193 ページの「SCTP チューニング可能パラメータ」
- 203 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 カーネル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

IP 転送関連の最新情報については、21 ページの「追加または変更された TCP/IP パラメータ」を参照してください。

ndd 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。

- 184 ページの「`ipcl_conn_hash_size`」

- 185 ページの「`ip_squeue_worker_wait`」

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。

`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
```

詳細は、`ndd(1M)` のマニュアルページを参照してください。

SMF フレームワークは、システムサービスを管理するための方法を提供しますが、`ndd` コマンドは引き続きシステム起動スクリプトに含まれています。起動スクリプトの作成については、『Oracle Solaris の管理: 基本管理』の「実行制御スクリプトの使用」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 のコピーは、<ftp://ftp.rfc-editor.org/in-notes> から取得可能です。このサイトの `rfc-index.txt` ファイル中で RFC のトピックを探してください。

IP チューニング可能パラメータ

`ip_icmp_err_interval` と `ip_icmp_err_burst`

説明

IP で IPv4 または IPv6 ICMP エラーメッセージを生成する頻度を制御します。IP

は、`ip_icmp_err_interval` の間に最大で

`ip_icmp_err_burst` の IPv4 または IPv6 ICMP エラーメッセージを生成します。

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

デフォルト	ip_icmp_err_interval は100 ミリ秒
	ip_icmp_err_burst は10 エラーメッセージ
範囲	ip_icmp_err_interval は0 から 99,999 ミリ秒
	ip_icmp_err_burst は1 から 99,999 のエラーメッセージ
動的か	はい
どのような場合に変更するか	診断の目的でエラーメッセージの生成頻度を増やしたい場合
コミットレベル	変更の可能性あり

ip_respond_to_echo_broadcast と ip6_respond_to_echo_multicast

説明	IPv4 や IPv6 が、ブロードキャスト ICMPv4 エコー要求またはマルチキャスト 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

ip_send_redirects と ip6_send_redirects

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

ip_forward_src_routed と ip6_forward_src_routed

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)
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	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、216 ページの「 ip_forward_src_routed と ip6_forward_src_routed (Solaris 10 リリース) 」を参照してください。

ip_addr_per_if

説明	実インタフェースに対応する論理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 8192
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

ip_strict_dst_multihoming と ip6_strict_dst_multihoming

説明	非転送インタフェースに到着したパケットを、そのインタフェース上に明示的に構成されていない IP アドレス向けとして受け入れるかどうかを制御します。ip_forwarding が有効になっているか、該当するインタフェースに対し xxx:ip_forwarding が有効になっていると、このパラメータは無視されます (そのパケットが実際に転送されるため)。
	RFC 1122 の 3.3.4.2 を参照してください。
デフォルト	0 (緩やかなマルチホーミング)
範囲	0 = オフ (緩やかなマルチホーミング) 1 = オン (厳密なマルチホーミング)
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 1 を設定します。
コミットレベル	変更の可能性あり

ip_multidata_outbound

説明	ネットワークスタックが、転送時、一度に複数のパケットをネットワークデバイスドライバ宛てに送信できるようにします。 このパラメータを有効にすると、ホスト CPU の利用率またはネットワークスループット (あるいはこの両方) が向上し、パケットあたりの処理コストが減少します。 現在このパラメータは、IP フラグメントを転送するための複数データ送信 (MDT) 機能を制御します。たとえば、リンク MTU より大きい UDP ペイロードを送信する場合があります。このチューニング可能パラメータが有効になっている場合、UDP などの特定の上位レベルプロトコルの IP フラグメ
----	---

ントがバッチ内でネットワークデバイスドライバに送信されます。この機能を無効にすると、ネットワークスタック内の TCP および IP 断片化ロジックの両方がもとに戻り、一度に1つのパケットをドライバに送信するようになります。

MDT 機能を利用できるのは、この機能をサポートするデバイスドライバのみです。

181 ページの「`tcp_mdt_max_pbufs`」も参照してください。

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	このパラメータをデバッグやその他の目的で有効にする必要がない場合は、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、217 ページの「 <code>ip_multidata_outbound</code> (Solaris 10 リリース)」を参照してください。

ip_queue_fanout

説明	queue と TCP/IP 接続 を関連付けるモードを判定します。
	値 0 の場合、新しい TCP/IP 接続と、この接続を作成した CPU が関連付けられます。値 1 の場合、異なる CPU に属する複数の queue との接続が関連付けられます。接続をファンアウトするために使用される queue の数は、167 ページの「 <code>ip_soft_rings_cnt</code> 」に基づいています。
デフォルト	0
範囲	0 または 1
動的か	はい
どのような場合に変更するか	特定の条件下で、すべての CPU に負荷を分散したい場合、このパラメータの値を 1 に設定します。たとえば、CPU 数が NIC 数を上回り、単一の NIC の

ゾーン構成	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。
コミットレベル	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。
変更履歴	変更の可能性あり
	詳細は、217 ページの「 ip_squeue_fanout (Solaris 10 11/06 リリース) 」を参照してください。

ip_soft_rings_cnt

説明	受信 TCP/IP 接続を展開するために使用される squeue の数を決定します。
	<hr/> <p>注-着信トラフィックは、いずれかのリングに置かれます。リングが過負荷になっている場合、パケットは破棄されます。パケットが破棄されるたびに、kstat dls カウンタ <code>dls_soft_ring_pkt_drop</code> が増分されます。</p> <hr/>
デフォルト	2
範囲	0 から nCPU。nCPU は、システム内の CPU の最大数です。
動的か	いいえ。このパラメータを変更する場合は、インタフェースを再び plumb します。
どのような場合に変更するか	10 Gbps の NIC および多くの CPU が搭載されたシステムでは、このパラメータを 2 より大きい値に設定することを検討してください。
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	廃止
変更履歴	詳細は、217 ページの「 ip_soft_rings_cnt (Solaris 10 11/06 リリース) 」を参照してください。

ip_pmtu_min

説明	PMTUD (Path MTU Discovery) プロトコルが動作する最小の値を決定します。 TCP プロトコルで PMTUD for TCP が実行される最小の MTU を決定するために 2 つのチューニング可能パラメータ (tcps_mss_min と ip_pmtu_min) の最大値が使用されている場合を除いて、このパラメータにはその他のチューニング可能パラメータとの直接の相互作用はありません。
デフォルト	576
範囲	68 から 65,535
動的か	いいえ
どのような場合に変更するか	このパラメータを調整すると、PMTUD の動作をオーバーライドできますが、このパラメータは 68 (バイト) よりも小さく調整しないでください。このパラメータを変更するときは、次の点を考慮してください。 <ul style="list-style-type: none">▪ PMTUD が動作している環境で、最小の MTU サイズを一致させます。▪ ip_pmtu_min がこのしきい値を下回ると、PMTUD がオフになります。▪ Solaris 10 8/11 よりも前のリリースが動作しているシステム、または 144488-12 または 144489-12 よりも前のパッチレベルが適用されたシステムで、以前の PMTUD 動作に戻すには、このパラメータを 68 に設定します。
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。

ip_ire_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	10 分
範囲	5 秒から 277 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。
コミットレベル	変更の可能性あり

ip_icmp_return_data_bytes と ip6_icmp_return_data_bytes

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	64 バイト (ip_icmp_return_data_bytes) 1280 バイト (ip6_icmp_return_data_bytes)
範囲	8 から 65,536 バイト
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

TCP チューニング可能パラメータ

tcp_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

tcp_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。

- 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、217 ページの「[tcp_local_dack_interval \(Solaris 10 リリース\)](#)」を参照してください。

tcp_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

tcp_local_dacks_max

説明

肯定応答 (ACK) が生成される前に宛先 (直接接続している) から受け取られる TCP セグメントの最大数を指定します。TCP セグメントは、個々の接続の最大セグメントサイズ (MMS) 単位で表されます。このパラメータに 0 か 1 を設定すると、すべて

のセグメントが 1 MSS の長さであるとみなされ、遅延 ACK は使用されなくなります。実際の数は、接続ごとに動的に計算されます。この値はデフォルトの最大値です。

デフォルト	8
範囲	0 から 16
動的か	はい
どのような場合に変更するか	この値は変更しないでください。ただし、遅延 ACK の影響でネットワークトラフィックが著しく混雑するような状況では、この値を減らします。ただし、2 より小さくしないでください。
コミットレベル	変更の可能性あり

tcp_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

tcp_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

tcp_xmit_hiwat

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 203 ページ の「ルート別のメトリック」を参照してください。 174 ページ の「tcp_max_buf」も参照してください。
デフォルト	49,152
範囲	4096 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

tcp_recv_hiwat

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203ページの「ルート別のメトリック」を参照してください。174ページの「tcp_max_buf」と189ページの「tcp_recv_hiwat_minmss」も参照してください。
デフォルト	49,152
範囲	2048 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

tcp_max_buf

説明	最大バッファサイズをバイト数で指定します。このパラメータは、 <code>setsockopt(3XNET)</code> を使用するアプリケーションによって設定される送信バッファサイズと受信バッファサイズを制御します。
デフォルト	1,048,576
範囲	8192 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

tcp_cwnd_max

説明	TCP 輻輳ウィンドウ (cwnd) の最大値をバイト数で指定します。
----	-------------------------------------

	TCP 輻輳ウィンドウについては、RFC 1122 と 2581 を参照してください。
デフォルト	1,048,576
範囲	128 から 1,073,741,824
動的か	はい
どのような場合に変更するか	アプリケーションが <code>setsockopt(3XNET)</code> を使用してウィンドウサイズに <code>tcp_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>tcp_cwnd_max</code> を超えることはありません。したがって、 <code>tcp_max_buf</code> は <code>tcp_cwnd_max</code> より大きくすべきです。
コミットレベル	変更の可能性あり

tcp_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	4
範囲	1 から 4
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

tcp_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) 後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、175 ページの「 <code>tcp_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

tcp_sack_permitted

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

tcp_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

tcp_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	60,000 (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

tcp_ecn_permitted

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せん。

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非標準拠です。

これらのデバイスを考慮し、このパラメータのデフォルト値は1に設定されています。ただし、まれに、受動的に有効にした場合でも問題が生じる場合があります。必要がある場合以外は、パラメータを0に設定しないでください。

コミットレベル

変更の可能性あり

tcp_conn_req_max_q

説明

`accept(3SOCKET)`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。179ページの「`tcp_conn_req_max_q0`」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	<p>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。</p> <p>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。</p> <p><code>tcp_conn_req_max_q</code> を増やしても、アプリケーションでそれだけの数の保留状態の TCP 接続を持てるとは限りません。アプリケーションでは、<code>listen(3SOCKET)</code> を使用して、保留状態の TCP 接続の最大数をソケットごとに変更できます。このパラメータは、アプリケーションが <code>listen()</code> を使用して設定できる最大値を表します。つまり、このパラメータ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、<code>listen()</code> に指定された値によっては <code>tcp_conn_req_max_q</code> よりもはるかに少ないことがあります。</p>
コミットレベル	変更の可能性あり

tcp_conn_req_max_q0

説明	<p>単一の TCP リスナーが持つことができる、不完全な (3 段階ハンドシェイクがまだ終わっていない) 保留状態の TCP 接続のデフォルトの最大数を指定します。</p> <p>TCP の 3 段階ハンドシェイクについては、RFC 793 を参照してください。178 ページの「<code>tcp_conn_req_max_q</code>」も参照してください。</p>
デフォルト	1024

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>tcp_conn_req_max_q0</code> と、各ソケットについて保留状態にある接続の最大数との関係は、次のとおりです。</p> <p>接続要求を受信すると、TCP はまず、受け付けられるのを待っている保留状態の TCP 接続 (3 段階ハンドシェイクが終わっている) の数が、そのリスナーに対する最大数 (N) を超えていないかをチェックします。接続数が超えていると、その要求は拒否されます。超えていなければ、TCP 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

tcp_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 1024
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

tcp_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、nddパラメータtcp_rst_sent_rate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

tcp_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

tcp_mdt_max_pbufs

説明	TCPによって生成される、1つのM_MULTIDATAメッセージで伝達できるペイロードバッファの数
----	---

	を指定します。165 ページの「 ip_multidata_outbound 」も参照してください。
デフォルト	16
範囲	1 から 16
動的か	はい
どのような場合に変更するか	このパラメータを減少させると、TCP によって生成される M_MULTIDATA メッセージごとのペイロードバッファの量を制限し、デバイスドライバ開発のデバッグに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

tcp_naglim_def

説明	このパラメータは、Nagle アルゴリズムのしきい値を制御します。TCP は、このパラメータの最小値と接続の MSS を使用して、Nagle アルゴリズムが機能するタイミングを判断します。たとえば、新しいデータの量が 1 MSS を超える場合、そのデータはこのパラメータの値に関係なく送出されます。このパラメータが 1 に設定されている場合、すべての TCP 接続で Nagle は無効になります。
デフォルト	4,096
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	データを遅延なしに送信する必要があるリアルタイムのアプリケーションでは、高速伝送を必要とするソケットに対して、tcp_naglim_def パラメータを設定するのではなく、setsockopt() を使って TCP_NODELAY を 1 に設定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
変更履歴	詳細は、218 ページの「 tcp_naglim_def (Solaris 10 リリース) 」を参照してください。

tcp_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することができます。
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、 217 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。

tcp_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することができます。
単位	ポート番号
デフォルト	65,535
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、217 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。

/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_sizes=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_squeue_worker_wait

説明	TCP/IP パケットを <code>squeue</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>squeue</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
変更履歴	詳細は、217 ページの「 ip_squeue_worker_wait (Solaris 10 11/06 リリース) 」を参照してください。

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。

tcp_keepalive_interval

説明	この <code>ndd</code> パラメータは、システム全体で TCP 接続がアイドル状態になってから最初にプローブが送信されるまでの間隔を設定します。 Oracle Solaris は、RFC 1122 に説明されている TCP キープアライブメカニズムをサポートしています。このメカニズムは、TCP ソケットで <code>SO_KEEPALIVE</code> ソケットオプションを設定することで有効になります。
----	---

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。

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

tcp_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 TCP 接続で、TCP が <code>tcp_ip_abort_interval</code> の間再転送を行なっても、この間に相手側のエンドポイントから肯定応答をまったく受け取らないと、この接続は閉じられます。
	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。187 ページの「 <code>tcp_rexmit_interval_max</code> 」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、187 ページの「 <code>tcp_rexmit_interval_max</code> 」を参照してください。
コミットレベル	変更の可能性あり

tcp_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 203 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	3 秒
範囲	1 ミリ秒から 20 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

tcp_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。 186 ページ の「 tcp_ip_abort_interval 」も参照してください。
デフォルト	60 秒
範囲	1 ミリ秒から 2 時間
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>tcp_ip_abort_interval</code> パラメータも変更する必要があります。 <code>tcp_ip_abort_interval</code> に は、 <code>tcp_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

tcp_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。187 ページ の「 <code>tcp_rexmit_interval_max</code> 」も参照してください。
デフォルト	400 ミリ秒
範囲	1 ミリ秒から 20 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 <code>tcp_rexmit_interval_max</code> パラメータも変更する必要があります。 <code>tcp_rexmit_interval_max</code> には、 <code>tcp_rexmit_interval_min</code> の 8 倍以上の値を指定します。
コミットレベル	変更の可能性あり

tcp_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 2 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。 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

tcp_tstamp_if_wscale

説明	このパラメータに 1 が設定され、かつ、ある接続に対するウィンドウスケールオプションが有効になっていると、TCP は、その接続の <code>timestamp</code> オプションも有効にします。
----	---

デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に変更するか	この値は変更しないでください。一般に、TCP を高速ネットワークで使用する場合は、シーケンス番号のラップアラウンドに対する保護が必要になります。この場合、timestamp オプションが必要になります。
コミットレベル	変更の可能性あり

tcp_recv_hiwat_minmss

説明	デフォルトの最小受信ウィンドウサイズを制御します。最小値は、tcp_recv_hiwat_minmss に、接続の最大セグメントサイズ (MSS) を掛けた値です。
デフォルト	4
範囲	1 から 65,536
動的か	はい
どのような場合に変更するか	この値は変更しないでください。この値を変更する必要がある場合は、4 より小さい値にしないでください。
コミットレベル	変更の可能性あり

UDP チューニング可能パラメータ

udp_xmit_hiwat

説明	デフォルトの最大 UDP ソケットデータグラムサイズをバイト数で指定します。詳細は、 192 ページ の「udp_max_buf」を参照してください。
デフォルト	57,344 バイト
範囲	1,024 から 1,073,741,824 バイト
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

udp_rcv_hiwat

説明	デフォルトの最大 UDP ソケット受信バッファサイズをバイト数で指定します。詳細は、 192 ページ の「 <code>udp_max_buf</code> 」を参照してください。
デフォルト	57,344 バイト
範囲	128 から 1,073,741,824 バイト
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

udp_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することができます。
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、 217 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。

udp_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することができます。
単位	ポート番号
デフォルト	65,535
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、 217 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。

udp_do_checksum

説明	このパラメータは、送信される UDP/IPv4 パケットのチェックサムを計算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	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
変更履歴	詳細は、 218 ページの「udp_do_checksum (Solaris 10 リリース)」 を参照してください。

特別な注意を要する UDP パラメータ

次のパラメータの変更は非推奨です。

udp_max_buf

説明	UDP ソケット用の送信バッファと受信バッファのサイズを制御します。
デフォルト 範囲	2,097,152 バイト 65,536 から 1,073,741,824 バイト
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。このパラメータの値が大きすぎると、UDP ソケットアプリケーションがメモリーを過剰に使用するおそれがあります。
コミットレベル	変更の可能性あり

IPQoS チューニング可能パラメータ

ip_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

sctp_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 193 ページの「sctp_pa_max_retr」に依存します。sctp_max_init_retr が sctp_pa_max_retr 以下であれば理想的です。
コミットレベル	変更の可能性あり

sctp_pa_max_retr

説明	SCTP 接続のすべてのパスを経由する最大再送回数を制御します。この値を超過すると、SCTP 接続は終了します。
デフォルト	10
範囲	1 から 128

動的か	はい
どのような場合に変更するか	すべてのパスを経由する最大再送回数は、パスの数と各パスの最大再送回数によって決定されます。 sctp_pa_max_retr は、使用可能なすべてのパスの 194 ページの「sctp_pp_max_retr」 の合計に設定すべきです。たとえば、宛先までのパス数が 3 で、これらのパスの最大再送回数がそれぞれ 5 回である場合、sctp_pa_max_retr には 15 以下の値を設定すべきです (RFC 2960、Section 8.2 の Note を参照)。
コミットレベル	変更の可能性あり

sctp_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

sctp_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 sctp_cwnd_max より大きい値を設定しようとしても、使用される実際のウィンドウが sctp_cwnd_max を超えることはありません。したがって、 198 ページの「sctp_max_buf」 は sctp_cwnd_max 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

sctp_ipv4_ttl

説明 SCTP 接続上で、アウトバウンド IP バージョン 4 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。宛先パスが 64 ホップを超過しそうな場合は、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

sctp_heartbeat_interval

説明 HEARTBEAT チャンクからハードビートに対応したアイドル状態の宛先までの間隔を計算します。

SCTP 終端は、相手側のアイドル状態の宛先転送アドレスまでの到達性をモニターするため、定期的に HEARTBEAT チャンクを送信します。

デフォルト 30 秒
範囲 0 から 86,400 秒

動的か はい

どのような場合に変更するか RFC 2960 のセクション 8.3 を参照してください。

コミットレベル 変更の可能性あり

sctp_new_secret_interval

説明 新しいシークレットを生成するタイミングを判定します。生成されたシークレットから、Cookie の MAC を計算できます。

デフォルト 2 分
範囲 0 から 1,440 分

動的か	はい
どのような場合に変更するか	RFC 2960 のセクション 5.1.3 を参照してください。
コミットレベル	変更の可能性あり

sctp_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

sctp_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 範囲	100 ミリ秒 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

sctp_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 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	バスにおける 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。バス 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

sctp_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	102,400
範囲	1024 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

sctp_xmit_hiwat

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。198 ページの「 sctp_max_buf 」も参照してください。
デフォルト	102,400
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	アプリケーションは、 getsockopt(3SOCKET) <code>SO_SNDBUF</code> を使って、個々の接続の送信バッファを変更できます。
コミットレベル	変更の可能性あり

sctp_xmit_lowat

説明	送信ウィンドウサイズの下限值を制御します。
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、 197 ページ の「 <code>sctp_xmit_hiwat</code> 」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

sctp_rcv_hiwat

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。 198 ページ の「 <code>sctp_max_buf</code> 」も参照してください。
デフォルト	102,400
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	アプリケーションは、 <code>getsockopt(3SOCKET)</code> <code>SO_RCVBUF</code> を使って、個々の接続の受信バッファを変更できます。
コミットレベル	変更の可能性あり

sctp_max_buf

説明	最大バッファサイズ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	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	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

sctp_ipv6_hoplimit

説明	SCTP 接続上で、アウトバウンド IP バージョン 6 パケットの IP バージョン 6 ヘッダのホップ制限値を設定します。
デフォルト 範囲	60 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。宛先パスが 60 ホップを超過しそうな場合は、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

sctp_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 範囲	1,000 500 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

sctp_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 範囲	60,000 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

sctp_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

sctp_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、 199 ページ の「 <code>sctp_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

sctp_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

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

sctp_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 コミットレベル	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、 199 ページ の「 <code>sctp_rto_max</code> 」に合わせて変更できます。 変更の可能性あり

sctp_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 コミットレベル	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。 変更の可能性あり

sctp_addip_enabled

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

sctp_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

sctp_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することができます。
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

sctp_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することができます。
単位	ポート番号
デフォルト	65,535
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

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。tcp_recv_hiwat のシステムデフォルト値は 49,152 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。

tcp_recv_hiwat 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 -recvpipes x
```

その結果、ge0 リンク上にある 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 49,152 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 -recvpipes 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。

- 206 ページの「autofs」
- 206 ページの「cron」
- 206 ページの「devfsadm」
- 206 ページの「dhcpgent」
- 207 ページの「fs」
- 207 ページの「ftp」
- 207 ページの「inetinit」
- 207 ページの「init」
- 207 ページの「ipsec」
- 207 ページの「kbd」
- 208 ページの「keyserv」
- 208 ページの「login」
- 208 ページの「lu」
- 208 ページの「mpathd」
- 209 ページの「nfs」
- 209 ページの「nfslogd」
- 209 ページの「nss」
- 209 ページの「passwd」
- 209 ページの「power」
- 209 ページの「rpc.nisd」
- 209 ページの「su」
- 209 ページの「syslog」
- 210 ページの「sys-suspend」
- 210 ページの「tar」
- 210 ページの「telnetd」
- 210 ページの「utmpd」
- 211 ページの「yppasswdd」

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。各機能について1つのファイルに保存された値は、`/etc/default` ディレクトリに置かれます。ただし、すべてのシステム機能がこのディレクトリにファイルを持っているわけではありません。

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 の管理: 基本管理](#)』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ついては、『[System Administration Guide: Advanced Administration](#)』の第3章「[Managing Serial Ports With the Service Access Facility \(Tasks\)](#)」を参照してください。

autofs

この機能を利用して、自動タイムアウト、ステータスメッセージの表示やログイン、`autofs` マウントポイントのブラウズ、トレースなどの各種 `autofs` パラメータを構成できます。詳細は、[autofs\(4\)](#) を参照してください。

cron

この機能を利用して、`cron` ログインの有効/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。`ifconfig` は、DHCP から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を識別する際に、このクライアントデーモンを起動してそのインタフェースを管理します。

詳細は、[dhcpgent\(1M\)](#) の「FILES」セクションにある `/etc/default/dhcpgent` 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

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

このファイルのすべての値は、シングルユーザーのブートリクエストに応じて `init` が呼び出すシェル環境に置かれます。さらに、`init` プロセスは、自身が `/etc/inittab` ファイルから起動または再起動するすべてのコマンドにこれらの値を渡します。

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

詳細は、`kbd(1)` の「Extended Description」セクションを参照してください。

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。

lu

このファイルには、Oracle Solaris Live Upgrade 機能のデフォルト設定が含まれていません。

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

この機能を利用して、NFS デーモン構成パラメータを設定できます。

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。

power

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

rpc.nisd

詳細は、[rpc.nisd\(1M\)](#) の「FILES」セクションにある `/etc/default/rpc.nisd` 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。

sys-suspend

詳細は、sys-suspend(1M) の「FILES」セクションにある /etc/default/sys-suspend 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) をモニターし、pututxline(3C) で root 以外のプロセスによって挿入された utmp エントリがプロセスの終了前に確実に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、モニター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。デフォルトは 300 です。
- MAX_FDS - utmpd がモニターしようとするプロセスの最大数。デフォルト値は 4096 です。通常、この値を変更する必要はありません。

yppasswdd

この機能を利用して、ユーザーが `passwd - r nis -e` コマンドの使用時に、制限されたシェルにログインシェルを設定できるかどうかを構成することができます。

詳細は、[rpc.yppasswdd\(1M\)](#) のマニュアルページを参照してください。

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す。

- 213 ページの「カーネルパラメータ」
- 216 ページの「NFS チューニング可能パラメータ」
- 216 ページの「TCP/IP チューニング可能パラメータ」
- 218 ページの「廃止または削除されたパラメータ (Oracle Solaris 10)」

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 10)

zfs_arc_min

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

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。

noexec_user_stack (Solaris 10 リリース)

Solaris 10 の記述セクションが更新され、「および sun4m」の語句が削除され、「64 ビット SPARC および AMD64」の語句が追加されました。

lwp_default_stksize (Solaris 10 リリース)

Solaris 10 の記述セクションが更新され、AMD64 の場合のデフォルトおよび最大値が追加されました。

SPARC プラットフォームの Solaris 10 のデフォルト値が 24,576 に変更されました。

ページング関連パラメータ

fastscan

`fastscan` のデフォルト値が明確にされました。詳細は、[56 ページの「fastscan」](#) を参照してください。

プロセスのサイズのチューニング可能パラメータ

ngroups_max (Solaris 10 リリース)

このパラメータは以前の Solaris リリースで説明されていませんでした。グループのデフォルトの最大数が 1024 に増えました。詳細は、[47 ページの「ngroups_max」](#) を参照してください。

max_nprocs (Solaris 10 リリース)

Solaris 10 の記述セクションが更新され、「sun4m」が削除されました。

UFS パラメータ

ufs_delete_hiwat

Solaris 10 1/13 リリースで、このパラメータが新しく追加されました。詳細は、[83 ページの「ufs_delete_hiwat」](#) を参照してください。

一般的なドライバパラメータ

ddi_msix_alloc_limit

このパラメータは新たに追加されました。詳細は、[67 ページの「ddi_msix_alloc_limit」](#) を参照してください。

一般的な入出力のチューニング可能パラメータ

maxphys (Solaris 10 リリース)

デフォルト値が sun4v システムを含むように更新されています。詳細は、[68 ページ](#)の「[maxphys](#)」を参照してください。

fsflush とそれに関連するパラメータ

dopageflush (Solaris 10 リリース)

説明を明確にするために、「物理」メモリーページ数が調べられるという記述が追加されました。

ページング関連のチューニング可能パラメータ

maxpgio (Solaris 10 リリース)

Solaris 10 バージョンでは、値の範囲が誤って 1 から 1024 と記述されていました。実際の範囲は、システムのアーキテクチャーと入出力サブシステムによって決まります。詳細は、[60 ページ](#)の「[maxpgio](#)」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

ncsize (Solaris 10 リリース)

以前の Solaris 10 リリースでは、ncsize パラメータのデフォルト値が誤って次のように記述されていました。

$$4 \times (v.v_proc + \text{maxusers}) + 320 / 100$$

正しいデフォルト値は次のとおりです。

$$(4 \times (v.v_proc + \text{maxusers}) + 320) + (4 \times (v.v_proc + \text{maxusers}) + 320 / 100)$$

詳細は、[71 ページ](#)の「[ncsize](#)」を参照してください。

TMPFS パラメータ

tmpfs:tmpfs_maxkmem (Solaris 10 リリース)

範囲の説明が sun4v システムを含むように更新されています。詳細は、[84 ページ](#)の「[tmpfs:tmpfs_maxkmem](#)」を参照してください。

SPARC システム固有のパラメータ (Solaris 10 リリース)

Solaris 10 8/07 リリースでは、SPARC システム固有のパラメータセクションのタイトルが sun4v システムを含むように改訂されました。

default_tsb_size (Solaris 10 リリース)

デフォルトの記述が変更されました。詳細は、[98 ページ](#)の「`default_tsb_size`」を参照してください。

enable_tsb_rss_sizing (Solaris 10 リリース)

説明とデフォルト値および範囲の値が変更されました。詳細は、[99 ページ](#)の「`enable_tsb_rss_sizing`」を参照してください。

tsb_rss_factor (Solaris 10 リリース)

「どのような場合に変更するか」の記述例が次のように変更されました。

たとえば、`tsb_rss_factor` 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。

NFS チューニング可能パラメータ

nfs:nfs3_nra (Solaris 10 リリース)

以前の Solaris 10 リリースでは、デフォルト値が誤って記述されていました。デフォルト値は 4 です。

TCP/IP チューニング可能パラメータ

ip_forward_src_routed と ip6_forward_src_routed (Solaris 10 リリース)

以前の Solaris 10 リリースでは、これらのパラメータのデフォルト値が誤って記述されていました。正しいデフォルト値は無効です。詳細は、[164 ページ](#)の「`ip_forward_src_routed` と `ip6_forward_src_routed`」を参照してください。

ip_multidata_outbound (Solaris 10 リリース)

このパラメータは、Solaris 10 リリースで、IP フラグメントをバッチ内でネットワークドライバに送信するように拡張されました。詳細は、[165 ページ](#)の「`ip_multidata_outbound`」を参照してください。

ip_queue_fanout (Solaris 10 11/06 リリース)

Solaris 10 8/07 リリースで、ゾーン構成情報が追加されました。詳細は、[166 ページ](#)の「`ip_queue_fanout`」を参照してください。

ip_queue_worker_wait (Solaris 10 11/06 リリース)

Solaris 10 8/07 リリースで、ゾーン構成情報が追加されました。詳細は、[185 ページ](#)の「`ip_queue_worker_wait`」を参照してください。また、このパラメータは [184 ページ](#)の「`/etc/system` ファイルに設定する TCP/IP パラメータ」に移動されました。

ip_soft_rings_cnt (Solaris 10 11/06 リリース)

Solaris 10 8/07 リリースで、ゾーン構成情報が追加されました。詳細は、[167 ページ](#)の「`ip_soft_rings_cnt`」を参照してください。

ip_queue_write (Solaris 10 リリース)

Solaris 10 リリースで、このパラメータに関する記述に誤りがありました。これは削除されています。

tcp_local_dack_interval (Solaris 10 リリース)

以前の Solaris リリースでは、このパラメータの範囲が誤って記載されていました。正しい範囲は 10 ミリ秒から 1 分です。

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

Solaris 10 8/11 リリースでは、これらのパラメータの説明が新たに追加されました。

- [202 ページ](#)の「`sctp_smallest_anon_port`」
- [203 ページ](#)の「`sctp_largest_anon_port`」

- 183 ページの「tcp_smallest_anon_port」
- 183 ページの「tcp_largest_anon_port」
- 190 ページの「udp_smallest_anon_port」
- 191 ページの「udp_largest_anon_port」

tcp_naglim_def (Solaris 10 リリース)

Solaris 10 8/11 リリースでは、182 ページの「tcp_naglim_def」パラメータの説明が新たに追加されました。

udp_do_checksum (Solaris 10 リリース)

Solaris 10 8/11 リリースでは、191 ページの「udp_do_checksum」パラメータの説明が新たに追加されました。

廃止または削除されたパラメータ (Oracle Solaris 10)

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。

rstchown

このパラメータは Oracle Solaris 10 8/11 リリースから廃止されています。

説明

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。

- 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。

詳細は、[chown\(2\)](#) 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

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

System V メッセージキューパラメータ

msgsys:msginfo_msgmni

Solaris 10 リリースでは廃止。

説明	作成できるメッセージキューの最大数
データ型	符号付き整数
デフォルト	50
範囲	0 から MAXINT
動的か	いいえ。msginfo 構造体の msgmni フィールドにロードされます。
検証	なし
どのような場合に変更するか	msgget(2) 呼び出しが ENOSPC エラーを返す場合、またはソフトウェアベンダーが推奨する場合
コミットレベル	変更の可能性あり

msgsys:msginfo_msgtql

Solaris 10 リリースでは廃止。

説明	作成できるメッセージの最大数。msgsnd(2) 呼び出しでこの限度を超える要求が行われた場合は、メッセージヘッダーが使用可能になるまで要求
----	--

	は延期されます。あるいは、その要求で <code>IPC_NOWAIT</code> フラグがオンに設定されていると、要求はエラー <code>EGAIN</code> で失敗します。
データ型	符号付き整数
デフォルト	40
範囲	0 から <code>MAXINT</code>
動的か	いいえ。 <code>msginfo</code> 構造体の <code>msgctl</code> フィールドにロードされます。
検証	なし
どのような場合に変更するか	<code>msgsnd(2)</code> 呼び出しがブロックするか <code>EGAIN</code> エラーを返す場合、またはソフトウェアベンダーが推奨する場合
コミットレベル	変更の可能性あり

msgsys:msginfo_msgmnb

Solaris 10 リリースでは廃止。

説明	1つのメッセージキューに含めることのできる最大バイト数
データ型	符号なし long
デフォルト	4096
範囲	0 から物理メモリーの総量
単位	バイト
動的か	いいえ。 <code>msginfo</code> 構造体の <code>msgmnb</code> フィールドにロードされます。
検証	なし
どのような場合に変更するか	<code>msgsnd(2)</code> 呼び出しがブロックするか <code>EAGAIN</code> エラーを返す場合、またはソフトウェアベンダーが推奨する場合
コミットレベル	変更の可能性あり

msgsys:msginfo_msgssz

Solaris 10 リリースで削除。

説明	メッセージバッファ用の空間を管理するためにシステムが使用するチャンクサイズを指定します。
データ型	符号付き整数
デフォルト	40
範囲	0 から MAXINT
動的か	いいえ。msginfo 構造体の msgctl フィールドにロードされます。
検証	メッセージやキューのために作成されるデータ構造体が最大数存在する場合に消費される空間と、モジュールがロードされる時の使用可能なカーネルメモリーの 25% を比較します。使用される空間があまりに大きいと、メッセージキューモジュールはロードを拒否し、この機能は利用できません。この計算には、メッセージに消費される空間も含まれません。この状態は、モジュールが最初にロードされる時だけ起こります。
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。
コミットレベル	廃止

msgsys:msginfo_msgmap

Solaris 10 リリースで削除。

説明	システムがサポートするメッセージの数。
データ型	符号付き整数
デフォルト	100
範囲	0 から MAXINT
動的か	いいえ
検証	メッセージやキューのために作成されるデータ構造体が最大数存在する場合に消費される空間と、モジュールがロードされる時の使用可能なカーネルメモリーの 25% を比較します。使用される空間があまりに大きいと、メッセージキューモジュールはロードを拒否し、この機能は利用できません。この計算には、メッセージに消費される空間も含まれません。この状態は、モジュールが最初にロードされる時だけ起こります。

どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。
コミットレベル	廃止

msgsys:msginfo_msgseg

Solaris 10 リリースで削除。

説明	使用可能なメッセージメモリーのプールとしてシステムが使用する、msginfo_msgssz セグメントの数。メッセージに使用できる合計メモリーは msginfo_msgseg * msginfo_msgssz です。
データ型	符号付き short
デフォルト	1024
範囲	0 から 32,767
動的か	いいえ
検証	メッセージやキューのために作成されるデータ構造体が最大数存在する場合に消費される空間と、モジュールがロードされるときの使用可能なカーネルメモリーの 25% を比較します。使用される空間があまりに大きいと、メッセージキューモジュールはロードを拒否し、この機能は利用できません。この計算には、メッセージに消費される空間も含まれません。この状態は、モジュールが最初にロードされるときだけ起こります。
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。
コミットレベル	廃止

msgsys:msginfo_msgmax

Solaris 10 リリースで削除。

説明	System V メッセージの最大サイズ
データ型	符号なし long
デフォルト	2048
範囲	0 から物理メモリーの総量
単位	バイト

動的か	いいえ。msginfo 構造体の msgmax フィールドにロードされます。
検証	なし
どのような場合に変更するか	msgsnd(2) 呼び出しが EINVAL エラーを返す場合、またはソフトウェアベンダーが推奨する場合。
コミットレベル	変更の可能性あり

System V セマフォ어의パラメータ

semsys:seminfo_semmni

Solaris 10 リリースでは廃止。

説明	セマフォ識別子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	10
範囲	1 から 65,535
動的か	いいえ
検証	SEMA_INDEX_MAX (現在は 65,535) と比較し、それより大きい場合は、SEMA_INDEX_MAX の値にリセットされます。警告メッセージがコンソール、メッセージ、またはその両方に出力されません。
どのような場合に変更するか	デフォルトの組数では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。現在の構成値を越えるセットを作成しようとしても、エラーメッセージは表示されません。アプリケーションは semget 呼び出しから ENOSPC というリターンコードを受け取ります。 詳細は、semget(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

semsys:seminfo_semmsl

Solaris 10 リリースでは廃止。

説明	セマフォ識別子あたりの System V セマフォ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	25
範囲	1 から MAXINT
動的か	いいえ
検証	セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュールが最初にロードされるときに使用可能なカーネルメモリーの 25% と比較されます。使用される容量がこのメモリーのしきい値より大きい場合、モジュールはロードを拒否し、セマフォ機能は利用できません。
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。構成された値を越える数のセマフォを組として作成しようとしても、エラーメッセージは表示されません。 <code>semget(2)</code> 呼び出しからアプリケーションにリターンコード <code>EINVAL</code> が返されません。
コミットレベル	変更の可能性あり

semsys:seminfo_semopm

Solaris 10 リリースでは廃止。

説明	<code>semop</code> 呼び出しあたりの System V セマフォ操作の最大数を指定します。このパラメータは、 <code>semop()</code> システムコールで使用する <code>sops</code> 配列内の <code>sembufs</code> の数を参照します。詳細は、 <code>semop(2)</code> 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	10
範囲	1 から MAXINT
動的か	いいえ
検証	セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュールが最初にロードされるときに使用可能なカーネルメモ

どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。1回の <code>semop</code> 呼び出しで、現在許可されている値を超える回数のセマフォ操作を実行しようとしても、エラーメッセージは表示されません。アプリケーションは <code>semop()</code> 呼び出しから E2BIG というリターンコードを受け取ります。
コミットレベル	変更の可能性あり

semsys:seminfo_semmns

Solaris 10 リリースで削除。

説明	システム上の System V セマフォの最大数
データ型	符号付き整数
デフォルト	60
範囲	1 から MAXINT
動的か	いいえ
検証	セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュールが最初にロードされるときに使用可能なカーネルメモリーの 25% と比較されます。使用される容量がこのメモリーのしきい値より大きい場合、モジュールはロードを拒否し、セマフォ機能は利用できません。
どのような場合に変更するか	デフォルトのセマフォ数では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。構成された値を越える数のセマフォを作成しようとしても、エラーメッセージは表示されません。 <code>semget(2)</code> 呼び出しからアプリケーションにリターンコード ENOSPC が返されません。
コミットレベル	変更の可能性あり

semsys:seminfo_semmnu

Solaris 10 リリースで削除。

説明	System V セマフォシステムがサポートする取り消し構造体の総数
データ型	符号付き整数
デフォルト	30
範囲	1 から MAXINT
動的か	いいえ
検証	セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュールが最初にロードされるときに使用可能なカーネルメモリーの 25% と比較されます。使用される容量がこのメモリーのしきい値より大きい場合、モジュールはロードを拒否し、セマフォ機能は利用できません。
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。構成された値を越える回数の取り消し操作を実行しようとしても、エラーメッセージは表示されません。システムが取り消し構造体を使い果たすと、 semop(2) 呼び出しからアプリケーションに戻り値 ENOSPC が返されます。
コミットレベル	変更の可能性あり

semsys:seminfo_semume

Solaris 10 リリースで削除。

説明	1つのプロセスで使用できる System V セマフォ取り消し構造体の最大数
データ型	符号付き整数
デフォルト	10
範囲	1 から MAXINT
動的か	いいえ
検証	セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュールが最初にロードされるときに使用可能なカーネルメモ

どのような場合に変更するか	リーの25%と比較されます。使用される容量がこのメモリーのしきい値より大きい場合、モジュールはロードを拒否し、セマフォ機能は利用できません。
デフォルト値	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。構成された値を越える回数の取り消し操作を実行しようとしても、エラーメッセージは表示されません。 <code>semop(2)</code> 呼び出しからアプリケーションにリターンコード <code>EINVAL</code> が返されます。
コミットレベル	変更の可能性あり

semsys:seminfo_semvmx

Solaris 10 リリースで削除。

説明	セマフォを設定できる最大値
データ型	符号なし short
デフォルト	32,767
範囲	1 から 65,535
動的か	いいえ
検証	なし
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。この最大値を越えても、エラーメッセージは表示されません。 <code>semop(2)</code> 呼び出しからアプリケーションにリターンコード <code>ERANGE</code> が返されません。
コミットレベル	変更の可能性あり

semsys:seminfo_semaem

Solaris 10 リリースで削除。

説明	取り消し構造体のセマフォ値を設定できる最大値
データ型	符号なし short
デフォルト	16,384
範囲	1 から 65,535

動的か	いいえ
検証	なし
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。構成された値を越える回数の取り消し操作を実行しようとしても、エラーメッセージは表示されません。 <code>semop(2)</code> 呼び出しからアプリケーションにリターンコード <code>EINVAL</code> が返されます。
コミットレベル	変更の可能性あり

System V 共有メモリーパラメータ

shmsys:shminfo_shmmni

Solaris 10 リリースでは廃止。

説明	作成できる共有メモリーセグメント数に関するシステム全体の制限。
データ型	符号付き整数
デフォルト	100
範囲	0 から MAXINT
動的か	いいえ。shminfo 構造体の shmmni フィールドにロードされます。
検証	System V 共有メモリーに関係するデータ構造体が最大数存在する場合に消費され得る空間量を、モジュールをロードしたときに使用可能なカーネルメモリーの 25% と比較します。消費されるメモリーが大きすぎると、モジュールをロードする試みは失敗します。
どのような場合に変更するか	システムの限度が小さすぎる場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。
コミットレベル	変更の可能性あり

shmsys:shminfo_shmmax

Solaris 10 リリースでは廃止。

説明	<p>作成できるSystem V 共有メモリーセグメントの最大サイズ。このパラメータは、アプリケーションが要求されたメモリーセグメントを作成するために必要な物理リソースが実際にあるか調べる前に検査される上限値です。</p> <p>サイズが0、または指定した値より大きいサイズの共有メモリーセクションを作成しようとする、EINVAL エラーとなります。</p> <p>このパラメータは、共有メモリーセグメントのサイズとしてオペレーティングシステムが受け付けることができる最大の値だけを指定します。セグメントを作成できるかどうかは、システム上で使用可能なスワップ空間の量に完全に依存します。32 ビットプロセスの場合は、接続されるセグメントのためにプロセスのアドレス空間に十分な容量があるかどうかにも影響します。</p>
データ型	符号なし long
デフォルト	8,388,608
範囲	32 ビットシステムでは0から MAXUINT32、64 ビットシステムでは0から MAXUINT64
単位	バイト
動的か	いいえ。shminfo 構造体の shmmax フィールドにロードされます。
検証	なし
どのような場合に変更するか	デフォルト値では足りない場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。ただし、共有メモリーセグメントのサイズを制限する必要がある場合以外は、このパラメータに利用できる最大値を設定しても副次的影響はありません。
コミットレベル	変更の可能性あり

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。

- 231 ページの「最新バージョン: *Oracle Solaris 10 1/13* リリース」
- 231 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 10 1/13* リリース

このマニュアルの最新バージョンは、*Oracle Solaris 10 1/13* 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 10 1/13:** *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。
- **Solaris 10 8/11:** `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 10 8/11:** このリリースには `ngroups_max` パラメータの説明が含まれています。詳細は、47 ページの「`ngroups_max`」を参照してください。
- **Solaris 10 10/09:** 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、108 ページの「`zfs_arc_min`」および 109 ページの「`zfs_arc_max`」を参照してください。
- **Solaris 10 10/09:** 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、67 ページの「`ddi_msix_alloc_limit`」を参照してください。

- **Solaris 10 10/09:** 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、[100 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。
- **Solaris 10 5/08:** [95 ページの「SPARC システム固有のパラメータ」](#) セクションの変換格納バッファのパラメータがより適切な情報を提供するように改訂されました。このリリースでは、次のパラメータが変更されています。
 - [98 ページの「default_tsb_size」](#) - デフォルトの文が明確にされました。
 - [99 ページの「enable_tsb_rss_sizing」](#) - デフォルトの文が誤っていたため、訂正されています。
 - [99 ページの「tsb_rss_factor」](#) - 例に関するセクションが、適切なパラメータ単位ではなくパーセンテージで記述されていました。この問題は解決されています。
- **Solaris 10 8/07:** 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。
 - [68 ページの「maxphys」](#)
 - [84 ページの「tmpfs:tmpfs_maxkmem」](#)
 - [95 ページの「SPARC システム固有のパラメータ」](#)

索引

A

autofs, 206
autoup, 39

B

bufhwm, 75
bufhwm_pct, 75

C

consistent_coloring, 95
cron, 206

D

ddi_msix_alloc_limit パラメータ, 67
default_stksize, 33
default_tsb_size, 98
desfree, 50
dhcpageant, 206
dnlc_dir_enable, 73
dnlc_dir_max_size, 74
dnlc_dir_min_size, 73
doiflush, 41
dopageflush, 41,215

E

enable_tsb_rss_sizing, 99

F

fastscan, 56
freebehind, 81
fs, 207
fsflush, 38
ftp, 207

H

handspreadpages, 58
hires_tick, 94

I

inetinit, 207
init, 207
ip_addr_per_if, 164
ip_forward_src_routed, 164
ip_icmp_err_burst, 162
ip_icmp_err_interval, 162
ip_icmp_return_data_bytes, 169
ip_ire_pathmtu_interval, 169
ip_multidata_outbound, 166
ip_pmtu_min, 168
ip_policy_mask, 192
ip_respond_to_echo_broadcast, 163

ip_send_redirects, 163
ip_soft_rings_cnt, 167
ip_squeue_fanout, 166
ip_squeue_worker_wait, 185
ip_strict_dst_multihoming, 165
ip6_forward_src_routed, 164
ip6_icmp_return_data_bytes, 169
ip6_respond_to_echo_multicast, 163
ip6_send_redirects, 163
ip6_strict_dst_multihoming, 165
ipcl_conn_hash_size, 184
ipsec, 207

K

kbd, 207
keyserv, 208
kmem_flags, 63

L

lgrp_mem_pset_aware, 102
logevent_max_q_sz, 35
login, 208
lotsfree, 49
lpg_alloc_prefer, 100
lu, 208
lwp_default_stksize, 34

M

max_nprocs, 46,214
maxpgio, 60,215
maxphys, 68,215
maxpid, 45
maxuprc, 46
maxusers, 43
md_mirror:md_resync_bufsz, 103
md:mirrored_root_flag, 103
min_percent_cpu, 58
minfree, 52
moddebug, 65

mpathd, 208
msgsys:msginfo_msgmax, 222
msgsys:msginfo_msgmnb, 220
msgsys:msginfo_msgmni, 219
msgsys:msginfo_msgseg, 222
msgsys:msginfo_msgssz, 221
msgsys:msginfo_msgtql, 219

N

ncsize, 71
nnd, 162
ndquot, 77
nfs_max_threads, 133
nfs:nacache, 150
nfs:nfs_allow_preepoch_time, 124
nfs:nfs_async_clusters, 146
nfs:nfs_async_timeout, 149
nfs:nfs_cots_timeo, 124
nfs:nfs_disable_rddir_cache, 143
nfs:nfs_do_symlink_cache, 127
nfs:nfs_dynamic, 129
nfs:nfs_lookup_neg_cache, 130
nfs:nfs_nra, 136
nfs:nfs_shrinkreaddir, 140
nfs:nfs_write_error_interval, 142
nfs:nfs_write_error_to_cons_only, 142
nfs:nfs3_async_clusters, 147
nfs:nfs3_bsize, 144
nfs:nfs3_cots_timeo, 125
nfs:nfs3_do_symlink_cache, 127
nfs:nfs3_dynamic, 129
nfs:nfs3_jukebox_delay, 151
nfs:nfs3_lookup_neg_cache, 131
nfs:nfs3_max_threads, 134
nfs:nfs3_max_transfer_size, 152
nfs:nfs3_max_transfer_size_clts, 154
nfs:nfs3_max_transfer_size_cots, 155
nfs:nfs3_nra, 137,216
nfs:nfs3_pathconf_disable_cache, 122
nfs:nfs3_shrinkreaddir, 141
nfs:nfs4_async_clusters, 148
nfs:nfs4_bsize, 145
nfs:nfs4_cots_timeo, 126

nfs:nfs4_do_symlink_cache, 128
 nfs:nfs4_lookup_neg_cache, 132
 nfs:nfs4_max_threads, 135
 nfs:nfs4_max_transfer_size, 153
 nfs:nfs4_nra, 137
 nfs:nfs4_pathconf_disable_cache, 122
 nfs:nrnode, 139
 nfslogd, 209
 ngroups_max, 47,214
 noexec_user_stack, 37,213
 nss, 209
 nstrpush, 88

O

Oracle データベースチューニング, ZFS ファイルシステム, 116

P

pageout_reserve, 54
 pages_before_pager, 59
 pages_pp_maximum, 55
 passwd, 209
 physmem, 32
 pidmax, 45
 power, 209
 pr_segdisable, 92
 primarycache, ZFS ファイルシステムプロパティ, 115
 pt_cnt, 86
 pt_max_pty, 87
 pt_pctofmem, 87

R

rechoose_interval, 94
 recordsize, ZFS ファイルシステムプロパティ, 115
 reserved_procs, 44
 rlim_fd_cur, 70
 rlim_fd_max, 69

routeadm, 22
 rpc.nisd, 209
 rpcmod:clnt_idle_timeout, 156
 rpcmod:clnt_max_conns, 155
 rpcmod:cotsmaxdupreqs, 159
 rpcmod:maxdupreqs, 158
 rpcmod:svc_default_stksize, 157
 rpcmod:svc_idle_timeout, 157
 rstchown, 72,218

S

sctp_addip_enabled, 202
 sctp_cookie_life, 200
 sctp_cwnd_max, 194
 sctp_deferred_ack_interval, 196
 sctp_heartbeat_interval, 195
 sctp_ignore_path_mtu, 196
 sctp_initial_mtu, 196
 sctp_initial_out_streams, 201
 sctp_initial_ssthresh, 197
 sctp_ipv4_ttl, 195
 sctp_ipv6_hoplimit, 199
 sctp_largest_anon_port, 203
 sctp_max_buf, 198
 sctp_max_in_streams, 200
 sctp_max_init_retr, 193
 sctp_maxburst, 201
 sctp_new_secret_interval, 195
 sctp_pp_max_retr, 194
 sctp_prsctp_enabled, 202
 sctp_recv_hiwat, 198
 sctp_rto_max, 199,200
 sctp_rto_min, 199
 sctp_shutack_wait_bound, 201
 sctp_smallest_anon_port, 202
 sctp_xmit_hiwat, 197
 sctp_xmit_lowat, 197
 secondarycache, ZFS ファイルシステムプロパティ, 115
 segspt_minfree, 91
 semsys:seminfo_semaem, 227
 semsys:seminfo_semmni, 223
 semsys:seminfo_semmns, 225

semsys:seminfo_semmnu, 226
semsys:seminfo_semmsl, 224
semsys:seminfo_semopm, 224
semsys:seminfo_semume, 226
semsys:seminfo_semvmx, 227
shmsys:shminfo_shmmax, 229
shmsys:shminfo_shmmni, 228
slowscan, 57
smallfile, 82
strmsgsz, 88,89
su, 209
sun4u, 216
sun4v, 96,216
swapfs_minfree, 62
swapfs_reserve, 61
sys-suspend, 210
syslog, 209

T

tar, 210
tcp_conn_req_max_q, 178
tcp_conn_req_max_q0, 179
tcp_conn_req_min, 180
tcp_cwnd_max, 174
tcp_deferred_ack_interval, 170
tcp_deferred_acks_max, 171
tcp_ecn_permitted, 178
tcp_ip_abort_interval, 186
tcp_keepalive_interval, 186
tcp_largest_anon_port, 183
tcp_local_dack_interval, 170,217
tcp_local_dacks_max, 171
tcp_max_buf, 174
tcp_mdt_max_pbufs, 182
tcp_naglim_def, 182
tcp_rcv_hiwat, 174
tcp_rcv_hiwat_minmss, 189
tcp_rev_src_routes, 177
tcp_rexmit_interval_extra, 188
tcp_rexmit_interval_initial, 187
tcp_rexmit_interval_max, 187
tcp_rexmit_interval_min, 187
tcp_rst_sent_rate, 181

tcp_rst_sent_rate_enabled, 181
tcp_sack_permitted, 176
tcp_slow_start_after_idle, 175
tcp_slow_start_initial, 175
tcp_smallest_anon_port, 183
tcp_time_wait_interval, 177
tcp_tstamp_always, 173
tcp_tstamp_if_wscale, 188
tcp_wscale_always, 172
tcp_xmit_hiwat, 173
throttlefree, 53
timer_max, 95
tmpfs_maxkmem, 84
tmpfs_minfree, 84
tmpfs:tmpfs_maxkmem, 215
tsb_alloc_hiwat, 97
tsb_rss_factor, 99
tune_t_fsflushr, 39
tune_t_minarmem, 56

U

udp_do_checksum, 191
udp_largest_anon_port, 191
udp_max_buf, 192
udp_rcv_hiwat, 190
udp_smallest_anon_port, 190
udp_xmit_hiwat, 189
ufs_delete_hiwat, 83
ufs_HW, 80
ufs_LW, 80
ufs_ninode, 77
ufs:ufs_WRITES, 80
utmpd, 210

Y

yppasswdd, 211

Z

zfs_arc_max, 109

zfs_arc_min, 108
zfs_mdcomp_disable, 114
zfs_nocacheflush, 113
zfs_prefetch_disable, 110
ZFS ファイルシステム, Oracle データベース用の
チューニング, 116
ZFS ファイルシステムプロパティ
primarycache, 115
recordsize, 115
secondarycache, 115

