



Solaris カーネルのチューン アップ・リファレンスマ ニュアル



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-0376-16
2008 年 10 月

Sun Microsystems, Inc. (以下 米国 Sun Microsystems 社とします) は、本書に記述されている製品に含まれる技術に関連する知的財産権を所有します。特に、この知的財産権はひとつかそれ以上の米国における特許、あるいは米国およびその他の国において申請中の特許を含んでいることがあります。それらに限定されるものではありません。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

U.S. Government Rights Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者によって開発された素材を含んでいることがあります。

本製品に含まれる HG-MinchoL、HG-MinchoL-Sun、HG-PMinchoL-Sun、HG-GothicB、HG-GothicB-Sun、および HG-PGothicB-Sun は、株式会社リコーがリョービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェースマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、Sun のロゴマーク、Solaris のロゴマーク、Java Coffee Cup のロゴマーク、docs.sun.com、NFS、SunOS、UNIX、Ultra、UltraSPARC、Java および Solaris は、米国およびその他の国における米国 Sun Microsystems 社またはその子会社の商標、登録商標もしくは、サービスマークです。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn8 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。Copyright(C) OMRON Co., Ltd. 1995-2006. All Rights Reserved. Copyright(C) OMRON SOFTWARE Co., Ltd. 1995-2006 All Rights Reserved.

「ATOK for Solaris」は、株式会社ジャストシステムの著作物であり、「ATOK for Solaris」にかかる著作権、その他の権利は株式会社ジャストシステムおよび各権利者に帰属します。

「ATOK」および「推測変換」は、株式会社ジャストシステムの登録商標です。

「ATOK for Solaris」に添付するフェイスマーク辞書は、株式会社ブレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド』に添付のものを使用しています。

「ATOK for Solaris」に含まれる郵便番号辞書 (7 桁/5 桁) は日本郵政公社が公開したデータを元に制作された物です (一部データの加工を行なっています)。

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは、OPEN LOOK のグラフィカル・ユーザインタフェースを実装するか、またはその他の方法で米国 Sun Microsystems 社との書面によるライセンス契約を遵守する、米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

本書で言及されている製品や含まれている情報は、米国輸出規制法で規制されるものであり、その他の国の輸出入に関する法律の対象となることがあります。核、ミサイル、化学あるいは生物兵器、原子力の海洋輸送手段への使用は、直接および間接を問わず厳しく禁止されています。米国が禁輸の対象としている国や、限定はされませんが、取引禁止顧客や特別指定国民のリストを含む米国輸出排除リストで指定されているものの輸出および再輸出は厳しく禁止されています。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われないものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: Solaris Tunable Parameters Reference Manual

Part No: 817-0404-16

Revision A

目次

はじめに	15
1 Solaris システムのチューニングの概要	19
Solaris システムチューニングの新機能	19
Solaris 10 リリースにおける Solaris システムチューニングの新機能	21
デフォルトのスタックサイズ	22
System V IPC 構成	22
NFSv4 パラメータ	24
追加または変更された TCP/IP パラメータ	24
SPARC: 変換記憶バッファ (TSB) パラメータ	26
SCTP チューニング可能パラメータ	26
Solaris システムのチューニング	27
チューニング可能パラメータの説明形式	28
Solaris カーネルのチューニング	30
/etc/system ファイル	30
kldb コマンド	31
mdb コマンド	32
Solaris の特殊な tune および var 構造体	33
Solaris システム構成情報の閲覧	33
sysdef コマンド	33
kstat ユーティリティ	34
2 Solaris カーネルチューニング可能パラメータ	35
チューニング可能パラメータ情報の記載箇所	36
カーネルとメモリーの一般的なパラメータ	36
physmem	36
default_stksize	37

lwp_default_stksize	38
logevent_max_q_sz	39
segkpsize	40
noexec_user_stack	41
fsflush とそれに関連するパラメータ	42
fsflush	42
tune_t_fsflushr	43
autoup	43
dopageflush	45
doiflush	45
プロセス規模調整パラメータ	46
maxusers	47
reserved_procs	48
pidmax	49
max_nprocs	49
maxuprc	50
ページング関連パラメータ	51
lotsfree	53
desfree	54
minfree	55
throttlefree	56
pageout_reserve	57
pages_pp_maximum	58
tune_t_minarmem	59
fastscan	60
slowscan	60
min_percent_cpu	61
handspreadpages	61
pages_before_pager	62
maxpgio	63
スワッピング関連パラメータ	64
swapfs_reserve	64
swapfs_minfree	65
カーネルメモリアロケータ	65
kmem_flags	66
一般的なドライバパラメータ	68

moddebug	68
一般的な入出力パラメータ	70
maxphys	70
rlim_fd_max	71
rlim_fd_cur	72
一般的なファイルシステムパラメータ	73
ncsize	73
rstchown	74
dnlc_dir_enable	75
dnlc_dir_min_size	76
dnlc_dir_max_size	76
segmap_percent	77
UFS パラメータ	78
bufhwm および bufhwm_pct	78
ndquot	80
ufs_ninode	80
ufs_WRITES	83
ufs_LW および ufs_HW	83
freebehind	84
smallfile	85
TMPFS パラメータ	86
tmpfs:tmpfs_maxmem	86
tmpfs:tmpfs_minfree	87
仮想端末	87
pt_cnt	88
pt_pctofmem	89
pt_max_pty	90
STREAMS パラメータ	90
nstrpush	90
strmsgsz	91
strctlsz	91
System V メッセージキュー	92
System V セマフォ	92
System V 共有メモリー	93
segspt_minfree	93
スケジューリング	94

rechoose_interval	94
タイマー	95
hires_tick	95
timer_max	95
sun4u または sun4v 固有のパラメータ	96
consistent_coloring	96
tsb_alloc_hiwater_factor	97
default_tsb_size	98
enable_tsb_rss_sizing	99
tsb_rss_factor	100
近傍性グループのパラメータ	101
lpg_alloc_prefer	101
lgrp_mem_default_policy	102
lgrp_mem_pset_aware	103
Solaris ボリュームマネージャーのパラメータ	104
md_mirror:md_resync_bufsz	104
md:mirrored_root_flag	105
3 NFS 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
NFS 環境のチューニング	107
NFS モジュールのパラメータ	108
nfs:nfs3_pathconf_disable_cache	108
nfs:nfs4_pathconf_disable_cache	108
nfs:nfs_allow_preepoch_time	109
nfs:nfs_cots_timeo	110
nfs:nfs3_cots_timeo	111
nfs:nfs4_cots_timeo	112
nfs:nfs_do_symlink_cache	113
nfs:nfs3_do_symlink_cache	113
nfs:nfs4_do_symlink_cache	114
nfs:nfs_dynamic	115
nfs:nfs3_dynamic	115
nfs:nfs_lookup_neg_cache	116
nfs:nfs3_lookup_neg_cache	117

nfs:nfs4_lookup_neg_cache	118
nfs:nfs_max_threads	119
nfs:nfs3_max_threads	120
nfs:nfs4_max_threads	121
nfs:nfs_nra	122
nfs:nfs3_nra	123
nfs:nfs4_nra	124
nfs:nrnode	125
nfs:nfs_shrinkreaddir	126
nfs:nfs3_shrinkreaddir	127
nfs:nfs_write_error_interval	128
nfs:nfs_write_error_to_cons_only	129
nfs:nfs_disable_rddir_cache	129
nfs:nfs_bsize	130
nfs:nfs3_bsize	131
nfs:nfs4_bsize	132
nfs:nfs_async_clusters	133
nfs:nfs3_async_clusters	134
nfs:nfs4_async_clusters	135
nfs:nfs_async_timeout	136
nfs:nacache	137
nfs:nfs3_jukebox_delay	138
nfs:nfs3_max_transfer_size	139
nfs:nfs4_max_transfer_size	140
nfs:nfs3_max_transfer_size_clts	141
nfs:nfs3_max_transfer_size_cots	142
nfssrv モジュールのパラメータ	142
nfssrv:nfs_portmon	143
nfssrv:rfs_write_async	143
nfssrv:nfsauth_ch_cache_max	145
nfssrv:exi_cache_time	145
rpcmod モジュールのパラメータ	146
rpcmod:clnt_max_conns	146
rpcmod:clnt_idle_timeout	147
rpcmod:svc_idle_timeout	147
rpcmod:svc_default_stksize	148

rpcmod:svc_default_max_same_xprt	149
rpcmod:maxdupreqs	150
rpcmod:cotsmaxdupreqs	151
4 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	153
チューニング可能パラメータ情報の記載箇所	153
IP パラメータのチューニングの概要	153
IP パラメータの妥当性検査	154
RFC (Internet Request for Comments)	154
IP チューニング可能パラメータ	154
ip_icmp_err_interval と ip_icmp_err_burst	154
ip_respond_to_echo_broadcast と ip6_respond_to_echo_multicast	155
ip_send_redirects と ip6_send_redirects	155
ip_forward_src_routed と ip6_forward_src_routed	156
ip_addrs_per_if	156
ip_strict_dst_multihoming と ip6_strict_dst_multihoming	157
ip_multidata_outbound	157
ip_queue_fanout	158
ip_soft_rings_cnt	159
特別な注意を要する IP チューニング可能パラメータ	160
TCP チューニング可能パラメータ	161
tcp_deferred_ack_interval	161
tcp_local_dack_interval	161
tcp_deferred_acks_max	162
tcp_local_dacks_max	162
tcp_wscale_always	163
tcp_tstamp_always	164
tcp_xmit_hiwat	164
tcp_rcv_hiwat	165
tcp_max_buf	165
tcp_cwnd_max	165
tcp_slow_start_initial	166
tcp_slow_start_after_idle	166
tcp_sack_permitted	167
tcp_rev_src_routes	168

tcp_time_wait_interval	168
tcp_ecn_permitted	168
tcp_conn_req_max_q	169
tcp_conn_req_max_q0	170
tcp_conn_req_min	171
tcp_rst_sent_rate_enabled	172
tcp_rst_sent_rate	172
tcp_mdt_max_pbufs	173
/etc/system ファイルに設定する TCP/IP パラメータ	173
特別な注意を要する TCP パラメータ	174
UDP チューニング可能パラメータ	179
udp_xmit_hiwat	179
udp_rcv_hiwat	179
特別な注意を要する UDP パラメータ	180
IPQoS チューニング可能パラメータ	180
ip_policy_mask	180
SCTP チューニング可能パラメータ	181
sctp_max_init_retr	181
sctp_pa_max_retr	181
sctp_pp_max_retr	182
sctp_cwnd_max	182
sctp_ipv4_ttl	183
sctp_heartbeat_interval	183
sctp_new_secret_interval	183
sctp_initial_mtu	184
sctp_deferred_ack_interval	184
sctp_ignore_path_mtu	184
sctp_initial_ssthresh	185
sctp_xmit_hiwat	185
sctp_xmit_lowat	185
sctp_rcv_hiwat	186
sctp_max_buf	186
sctp_ipv6_hoplimit	186
sctp_rto_min	187
sctp_rto_max	187
sctp_rto_initial	187

sctp_cookie_life	188
sctp_max_in_streams	188
sctp_initial_out_streams	188
sctp_shutack_wait_bound	189
sctp_maxburst	189
sctp_addip_enabled	189
sctp_prsctp_enabled	190
ルート別のメトリック	190
5 ネットワークキャッシュとアクセラレータのチューニング可能パラメータ	193
チューニング可能パラメータ情報の記載箇所	193
NCA パラメータのチューニング	194
nca:nca_conn_hash_size	194
nca:nca_conn_req_max_q	194
nca:nca_conn_req_max_q0	195
nca:nca_ppmax	195
nca:nca_vpmax	196
NCA のための一般的なシステムチューニング	197
sq_max_size	197
ge:ge_intr_mode	198
6 システム機能のパラメータ	199
システムのデフォルトのパラメータ	200
autofs	200
cron	200
devfsadm	200
dhcpgent	200
fs	200
ftp	201
inetinit	201
init	201
keyserv	201
kbd	201
login	201
mpathd	201

nfs	202
nfslogd	202
nss	202
passwd	202
power	202
rpc.nisd	202
su	202
syslog	202
sys-suspend	203
tar	203
utmpd	203
yppasswdd	203
A チューニング可能パラメータの変更履歴	205
カーネルパラメータ	205
プロセスのサイズのチューニング可能パラメータ	205
一般的な入出力のチューニング可能パラメータ	205
カーネルとメモリーの一般的なパラメータ	206
fsflush とそれに関連するパラメータ	206
ページング関連のチューニング可能パラメータ	207
一般的なファイルシステムパラメータ	207
ncsize (Solaris 10 リリース)	207
UFS チューニング可能パラメータ	207
TMPFS パラメータ	209
sun4u または sun4v 固有のパラメータ (Solaris 10 リリース)	210
NFS チューニング可能パラメータ	210
nfs:nfs3_nra (Solaris 10)	210
nfs:nrnode (Solaris 9 8/03)	210
nfs:nfs_write_error_interval (Solaris 9 8/03)	211
nfs:nfs_write_error_to_cons_only (Solaris 9 8/03)	211
nfs:nfs_disable_rddir_cache (Solaris 9 8/03)	211
nfs:nfs3_max_transfer_size (Solaris 9 8/03)	211
TCP/IP チューニング可能パラメータ	211
ip_forward_src_routed と ip6_forward_src_routed (Solaris 10 リリース)	211
ip_multidata_outbound (Solaris 10 リリース)	212

ip_multidata_outbound (Solaris 9 リリース)	212
ip_queue_fanout (Solaris 10 11/06 リリース)	214
ip_queue_worker_wait (Solaris 10 11/06 リリース)	214
ip_soft_rings_cnt (Solaris 10 11/06 リリース)	214
ip_queue_write (Solaris 10 リリース)	214
tcp_conn_hash_size (Solaris 9 リリース)	214
tcp_wscale_always (Solaris 9 リリース)	215
ipc_tcp_conn_hash_size (Solaris 9 リリース)	215
tcp_compression_enabled (Solaris 9 リリース)	216
ip_forwarding および ip6_forwarding (Solaris 9 リリース)	217
xxx:ip_forwarding (Solaris 9 リリース)	217
tcp_conn_req_max_q0 (Solaris 8 リリース)	217
UDP チューニング可能パラメータ	219
udp_xmit_hiwat (Solaris 9 リリース)	219
udp_recv_hiwat (Solaris 9 リリース)	219
udp_max_buf (Solaris 9 リリース)	219
ネットワークキャッシュとアクセラレータ (NCA) のチューニング可能パラメータ	219
sq_max_size (Solaris 9 12/02 リリース)	219
旧式または削除されたパラメータ	220
ページング関連のチューニング可能パラメータ	220
System V メッセージキューパラメータ	221
System V セマフォのパラメータ	225
System V 共有メモリーパラメータ	230
NFS モジュールのパラメータ	232
B このマニュアルの改訂履歴	233
最新バージョン: Solaris 10 10/08 リリース	233
Solaris リリースで追加または変更されたパラメータ	233
Solaris 10 リリースで追加または変更されたパラメータ	235
Solaris カーネルチューニング可能パラメータ	235
TSB パラメータ	236
NFS パラメータ	237
TCP/IP パラメータ	237
SCTP チューニング可能パラメータ	238

システム機能のパラメータ	238
sun4m サポートの終了	239
Solaris 9 リリースで追加または変更されたパラメータ	239
ip_policy_mask	239
sun4d サポートの終了	239
サポートされなくなったまたは旧式のパラメータ	239
Solaris 8 リリースで追加されたパラメータ	240
logevent_max_q_sz	240
索引	241

はじめに

『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、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、CDE、GNOME、または 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 (UltraSPARC®, SPARC64, AMD64, Pentium, Xeon EM64T) を使用するシステムをサポートします。サポートされるシステムについては、Solaris 10 Hardware Compatibility List (<http://www.sun.com/bigadmin/hcl>) を参照してください。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、AMD64 または EM64T システムに関する 64 ビット特有の情報を指します。
- 「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。

サポートされるシステムについては、Solaris 10 Hardware Compatibility List (<http://www.sun.com/bigadmin/hcl>) を参照してください。

対象読者

Solaris システムの管理者は、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しなければならない場合があります。このマニュアルは、このような習熟した Solaris システム管理者を対象としています。Solaris チューニング可能パラメータを変更する場合のガイドラインについては、[27 ページ](#)の「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

このマニュアルは、次の各章と付録から構成されています。

章	説明
第1章「Solaris システムのチューニングの概要」	Solaris システムのチューニングの概要。このマニュアル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「Solaris カーネルチューニング可能パラメータ」	Solaris カーネルのチューニング可能パラメータの説明 (カーネルメモリーやファイルシステム、プロセスサイズ、ページングのパラメータなど)
第3章「NFS チューニング可能パラメータ」	NFS チューニング可能パラメータの説明 (シンボリックリンクのキャッシュや、動的再転送、RPC セキュリティーのパラメータなど)
第4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 転送やソールスルーティング、バッファーサイジングのパラメータなど)
第5章「ネットワークキャッシュとアクセラレータのチューニング可能パラメータ」	ネットワークキャッシュとアクセラレータ (NCA) のチューニング可能パラメータの説明
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」	変更や旧式となっ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 (現行の Solaris リリースの情報を含む)

Solaris のチューニングに関するほかの情報源

Solaris のチューニングに関するほかの情報源として次のものがあります。

チューニングに関する情報源	参照先
パフォーマンスチューニングクラス	http://suned.sun.com
オンラインパフォーマンスチューニング情報	http://www.solarisinternals.com/si/index.php

チューニングに関する情報源

参照先

Sun Microsystems Press から発行されているパフォーマンスチューニング関連文書の注文 <http://www.sun.com/books/blueprints.series.html>

マニュアル、サポート、およびトレーニング

Sun の Web サイトでは、次のサービスに関する情報も提供しています。

- マニュアル (<http://jp.sun.com/documentation/>)
- サポート (<http://jp.sun.com/support/>)
- トレーニング (<http://jp.sun.com/training/>)

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-1 表記上の規則

字体または記号	意味	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。 system%
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。	system% su password:
<i>AaBbCc123</i>	変数を示します。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。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第5章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。

表 P-1 表記上の規則 (続き)

字体または記号	意味	例
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	<code>sun% grep '^#define \ XV_VERSION_STRING'</code>

コード例は次のように表示されます。

■ C シェル

```
machine_name% command y|n [filename]
```

■ C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

■ Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

■ Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

Solaris システムのチューニングの概要

この章では、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、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。

- 19 ページの「Solaris システムチューニングの新機能」
- 21 ページの「Solaris 10 リリースにおける Solaris システムチューニングの新機能」
- 27 ページの「Solaris システムのチューニング」
- 28 ページの「チューニング可能パラメータの説明形式」
- 30 ページの「Solaris カーネルのチューニング」
- 33 ページの「Solaris の特殊な tune および var 構造体」
- 33 ページの「Solaris システム構成情報の閲覧」
- 34 ページの「kstat ユーティリティ」

Solaris システムチューニングの新機能

この節では、この Solaris 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。

- **Solaris 10 10/08:** このマニュアルの Solaris 10 バージョンでは、誤って `nfs4_shrinkreaddir` パラメータの情報が含まれていました。このパラメータは使用できません。
- **Solaris 10 10/08:** ZFS ファイルシステムのチューニングについては、次のサイトを参照してください。
http://www.solarisinternals.com/wiki/index.php/ZFS_Evil_Tuning_Guide
- **Solaris 10 5/08:** メモリーの近傍性グループのパラメータは、Solaris 10 5/08 カーネルのパッチで提供されます。これらのパラメータについての詳細は、[101 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:** このマニュアルの Solaris 10 バージョンでは、誤って `nfs4_dynamic` パラメータの情報が含まれていました。このパラメータは使用できません。

- **Solaris 10 5/08:** 96 ページの「[sun4u または sun4v 固有のパラメータ](#)」節の変換記憶バッファのパラメータが、より適切な情報に更新されています。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](#)」
 - 100 ページの「[tsb_rss_factor](#)」
- **Solaris 10 5/08:** このマニュアルの Solaris 10 バージョンでは、誤って `tcp_keepalive_abort_interval` パラメータの情報が含まれていました。このパラメータは、Solaris Express リリースでしか使用できません。さらに、`tcp_keepalive_interval` パラメータの情報が [160 ページの「特別な注意を要する IP チューニング可能パラメータ」](#) に移動されました。
- **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。
 - 70 ページの「[maxphys](#)」
 - 86 ページの「[tmpfs:tmpfs_maxkmem](#)」
 - 96 ページの「[sun4u または sun4v 固有のパラメータ](#)」
- **Solaris 10 8/07:** 以前にこのマニュアルに記載されていた `maxpgio` パラメータの値の範囲に関する情報には誤りがありました。詳細は、[63 ページの「maxpgio」](#) を参照してください。
- **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。

- 158 ページの「[ip_queue_fanout](#)」
- 174 ページの「[ip_queue_worker_wait](#)」
- 159 ページの「[ip_soft_rings_cnt](#)」

その他の TCP、IP、UDP、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、UDP、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に関する詳細は、『[Solaris のシステム管理 \(Solaris コンテナ: 資源管理と Solaris ゾーン\)](#)』を参照してください。

- **Solaris 10 8/07:** 以前にこのマニュアルに記載されていた `ip_squeue_write` パラメータの情報は不正確であったため、削除されました。
- **Solaris 10 11/06:** Solaris 10 リリースで `ncsize` パラメータのデフォルト値に関する記述に誤りがありました。詳細は、[73 ページの「ncsize」](#) を参照してください。
- **Solaris 10 11/06:** Solaris 10 リリースで `nfs:nfs3_nra` パラメータのデフォルト値に関する記述に誤りがありました。デフォルト値は 4 です。詳細は、[123 ページの「nfs:nfs3_nra」](#) を参照してください。
- **Solaris 10 6/06:** `ip_squeue_fanout` パラメータが変更されました。詳細は、[158 ページの「ip_squeue_fanout」](#) および [159 ページの「ip_soft_rings_cnt」](#) を参照してください。
- **Solaris 10 6/06:** `ip_multidata_outbound` パラメータが拡張されました。詳細は、[157 ページの「ip_multidata_outbound」](#) を参照してください。
- **Solaris 10 6/06:** `ip_forward_src_routed` パラメータおよび `ip6_forward_src_routed` パラメータが修正されました。Solaris 9 以降、このパラメータのデフォルト値は有効ではなく無効です。詳細は、[156 ページの「ip_forward_src_routed と ip6_forward_src_routed」](#) を参照してください。
- **Solaris 10 6/06:** UDP パラメータが修正されました。Solaris 10 リリースでこれらのパラメータのデフォルト値が変更されましたが、以前は新しいデフォルト値に関する記述がありませんでした。詳細は、[179 ページの「UDP チューニング可能パラメータ」](#) を参照してください。
- **Solaris 10 6/06:** Solaris 10 リリースで `sq_max_size` パラメータのデフォルト値に関する記述に誤りがありました。詳細は、[197 ページの「sq_max_size」](#) を参照してください。

Solaris 10 リリースにおける Solaris システムチューニングの新機能

この節では、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。

- [22 ページの「デフォルトのスタックサイズ」](#)
- [22 ページの「System V IPC 構成」](#)
- [24 ページの「NFSv4 パラメータ」](#)
- [24 ページの「追加または変更された TCP/IP パラメータ」](#)
- [26 ページの「SPARC: 変換記憶バッファ \(TSB\) パラメータ」](#)
- [26 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` に優先します。詳細は、[37 ページの「default_stksize」](#) を参照してください。

System V IPC 構成

この Solaris 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、または資源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、およびセマフォです。

資源制御によって、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再起動しなければなりませんでした。

しかし、資源制御で 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資源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、現在は旧式となっている 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資源制御を識別しています。旧式の IPC チューニング可能パラメータと資源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、資源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。

資源制御	旧式の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	<code>ULONG_MAX</code>	65536
<code>process.max-msg-messages</code>	<code>msgsys:msginfo_msgtql</code>	40	<code>UINT_MAX</code>	8192
<code>process.max-sem-ops</code>	<code>semsys:seminfo_semopm</code>	10	<code>INT_MAX</code>	512
<code>process.max-sem-nsems</code>	<code>semsys:seminfo_semmsl</code>	25	<code>SHRT_MAX</code>	512

資源制御	旧式の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
project.max-shm-memory	shmsys:shminfo_shmmax*	0x800000	UINT64_MAX	物理メモリーの 1/4
project.max-shm-ids	shmsys:shminfo_shmmni	100	2 ²⁴	128
project.max-msg-ids	msgsys:msginfo_msgmni	50	2 ²⁴	128
project.max-sem-ids	semsys:seminfo_semmni	10	2 ²⁴	128

* `project.max-shm-memory` 資源制御は 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。

資源制御についての詳細は、『[Solaris のシステム管理 \(Solaris コンテナ: 資源管理と Solaris ゾーン\)](#)』の「[使用可能な資源制御](#)」を参照してください。

旧式のパラメータは、Solaris システム上の `/etc/system` ファイルに引き続き指定できます。その場合、これらのパラメータは、以前の Solaris リリースの場合と同様に、デフォルトの資源制御値の初期化に使用されます。詳細は、[220 ページの「旧式または削除されたパラメータ」](#)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。これらのパラメータが Solaris システム上の `/etc/system` ファイルに含まれていた場合は、コメント扱いになります。

<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>
<code>shmsys:shminfo_shmmin</code>	<code>msgsys:msginfo_msgmap</code>
<code>msgsys:msginfo_msgseg</code>	<code>msgsys:msginfo_msgssz</code>
<code>msgsys:msginfo_msgmax</code>	

現在使用できる資源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。資源制御の構成方法については、`project(4)` のマニュアルページ、および『[Solaris のシステム管理 \(Solaris コンテナ: 資源管理と Solaris ゾーン\)](#)』の第 6 章「[資源制御 \(概要\)](#)」を参照してください。

NFSv4 パラメータ

このリリースでは、NFSv4 プロトコルの次のパラメータが新しく追加されました。

- 108 ページの「[nfs:nfs4_pathconf_disable_cache](#)」
- 112 ページの「[nfs:nfs4_cots_timeo](#)」
- 114 ページの「[nfs:nfs4_do_symlink_cache](#)」
- 118 ページの「[nfs:nfs4_lookup_neg_cache](#)」
- 121 ページの「[nfs:nfs4_max_threads](#)」
- 124 ページの「[nfs:nfs4_nra](#)」
- 132 ページの「[nfs:nfs4_bsize](#)」
- 135 ページの「[nfs:nfs4_async_clusters](#)」
- 140 ページの「[nfs:nfs4_max_transfer_size](#)」

NFSv4 パラメータについては、108 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

この Solaris リリースでは、次の IP パラメータが追加されました。

- 174 ページの「[ip_queue_worker_wait](#)」
- 158 ページの「[ip_queue_fanout](#)」
- 173 ページの「[ipcl_conn_hash_size](#)」

この Solaris リリースでは、次の TCP パラメータが追加されました。

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

この Solaris リリースでは、次の TCP/IP パラメータは旧式となりました。

- [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 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、`routadm` 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 `routeadm` 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。

- すべての設定値が再起動後も維持されます。
- 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設定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。

```
# routeadm -e ipv4-forwarding
```

```
# routeadm -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。

```
# routeadm -d ipv4-forwarding
```

```
# routeadm -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

このリリースでは、Translation Storage Buffer (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、[96 ページの「sun4u または sun4v 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 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のストリーム制御転送プロトコル (SCTP) が含まれています。SCTP チューニング可能パラメータについては、[181 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。

Solaris システムのチューニング

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。このマニュアルでは、Solaris OS で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の詳細を記述します。

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。このマニュアル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。最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。このマニュアル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。このマニュアル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。

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。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、[22 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 - このマニュアルで説明するチューニング可能パラメータは、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
- 検査
- 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す。名前にコロンが含まれている場合、コロンの左側の文字列はロード可能モジュールの名前を表し、コロンの右側の文字列はモジュール内の変数名を表します。コロンの右側の文字列はモジュール内のパラメータ名を示します。次に例を示します。

module_name:variable

備考欄

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。
- 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

単位	単位の種類を表します。この項目は省略されることがあります。
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	システムの検査で取り得る範囲や、データ型の上下限を表す範囲です。 ■ 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 <code>mdb</code> 、または <code>kmdb</code> 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。
検査	システムが、 <code>/etc/system</code> ファイルに指定されたとおりの値とデフォルトの値のいずれを変数の値に適用するか調べます。また、検査がいつ適用されるかも示します。
暗黙的制約	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。この項目は省略されることがあります。
どのような場合に変更するか	この値を変更したくなる理由について説明します。エラーメッセージまたは戻りコードが含まれます。
ゾーン構成	パラメータを排他的 IP ゾーン内で設定できるか、または大域ゾーン内で設定する必要があるかを識別します。共有 IP ゾーン内で設定できるパラメータはありません。
コミットレベル	インタフェースの安定性を表します。このマニュアルで記述するパラメータの多くは「発展中 (Evolving)」または「変更の可能性あり (Unstable)」のいずれかに分類されます。詳細は、 <code>attributes(5)</code> のマニュアルページを参照してください。
変更履歴	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。この項目は省略されることがあります。

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	30 ページの「/etc/system ファイル」
カーネルデバッグ (kmdb) の使用	31 ページの「kmdb コマンド」
モジュールデバッグ (mdb) の使用	32 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 4 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	194 ページの「NCA パラメータのチューニング」

/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機構を提供します。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。

Solaris 8 より前のリリースでは、パラメータの値を設定する /etc/system のエントリは、次の 2 つの処理段階に分けて適用されていました。

- 最初の処理段階では、さまざまなブートストラップパラメータ (maxusers など) を取得し、基本となるシステムパラメータを初期設定します。
- 次の処理段階では、ブートストラップパラメータを使用して基本構成を計算し、/etc/system ファイルに指定されているすべての値を適用します。ブートストラップパラメータに関しては、この処理段階で計算された値やリセットされた値は、/etc/system ファイルに指定されている値で置き換えられます。

2 つめの処理段階では、許容されないように思われる値にパラメータを設定したり、初期構成で値が上書きされるパラメータ (max_nprocs など) に値を割り当てたりして、ユーザーや管理者にとって混乱を招く場合があります。

Solaris 8 リリースから、構成パラメータを計算する前にすべての値を設定する 1 つの処理段階が設けられました。

例 — /etc/system を使用したパラメータの設定

次の /etc/system エントリでは、NFS バージョン 2 ソフトウェアを使用してマウントされたファイルシステムに対し先読みするブロックの数を指定します。

```
set nfs:nfs_nra=4
```

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。次に例を示します。

```
# 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 `/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。

システム回復については、『[Solaris のシステム管理 \(基本編\)](#)』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

Solaris 8 リリースから組み込まれたモジュールデバッガ `mdb` は Solaris のデバッガの中でも独特で、簡単に拡張できます。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処理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` を使用した値の変更

整数パラメータ `maxusers` の値を 495 から 512 に変更するには、次のようにします。

```
# mdb -kw
Loading modules: [ unix krtld genunix ip logindmux ptm nfs ipc lofs ]
> maxusers/D
maxusers:
maxusers:      495
> maxusers/W 200
maxusers:      0x1ef      =      0x200
> $q
```

実際に変更する場合は、`maxusers` を変更したいパラメータのアドレスに、値を設定したい値に置き換えて、このコマンドを実行します。

モジュールデバッガの使用についての詳細は、『Solaris モジュールデバッガ』を参照してください。

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

たとえば、UFS モジュールがロードされている場合、各デバッガは `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。

Solaris の特殊な tune および var 構造体

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。tune 構造体 (/usr/include/sys/tuneable.h ファイルで定義されている) 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。

以前のバージョンの『Solaris のシステム管理 (第2巻)』をはじめ、さまざまなマニュアルには、tune 構造体のパラメータを設定する適切な方法として tune:field-name (field-name には前述の実際のパラメータ名を指定する) という構文が指定されていました。しかし、この設定は何のメッセージもなく失敗します。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。

Solaris システム構成情報の閲覧

システム構成情報を調べるツールはいくつかあります。ツールによっては、スーパーユーザー権限が必要です。それ以外のツールは、一般ユーザーの権限で実行できます。動作中のシステム上で mdb を使うか、あるいは kadb でブートし、カーネルデバッガですべての構造体やデータアイテムを調べることができます。

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

sysdef コマンド

sysdef コマンドは、System V IPC の設定値、STREAMS チューニング可能パラメータ、プロセス資源制限、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、512M バイトの Sun™ Ultra™ 80 システム上で sysdef を実行すると、「チューニング可能パラメータ」セクションは次のように表示されます。

```
10387456      maximum memory allowed in buffer cache (bufhwm)
      7930      maximum number of processes (v.v_proc)
      99        maximum global priority in sys class (MAXCLSPRI)
```

7925	maximum processes per user id (v.v_maxup)
30	auto update time limit in seconds (NAUTOUP)
25	page stealing low water mark (GPGSLO)
5	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機構を提供します。この機構を利用する場合、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。

注 - unix モジュールの kstat データ構造体を `system_pages` という名前で使用しても、`cachefree` の統計情報は得られません。`cachefree` は Solaris 9 リリースからサポートされなくなりました。

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 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。

- 36 ページの「カーネルとメモリーの一般的なパラメータ」
- 42 ページの「`fsflush` とそれに関連するパラメータ」
- 46 ページの「プロセス規模調整パラメータ」
- 51 ページの「ページング関連パラメータ」
- 64 ページの「スワッピング関連パラメータ」
- 65 ページの「カーネルメモリアロケータ」
- 68 ページの「一般的なドライバパラメータ」
- 70 ページの「一般的な入出力パラメータ」
- 73 ページの「一般的なファイルシステムパラメータ」
- 78 ページの「UFS パラメータ」
- 86 ページの「TMPFS パラメータ」
- 87 ページの「仮想端末」
- 90 ページの「STREAMS パラメータ」
- 92 ページの「System V メッセージキュー」
- 92 ページの「System V セマフォ」
- 93 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」
- 95 ページの「タイマー」
- 96 ページの「`sun4u` または `sun4v` 固有のパラメータ」
- 101 ページの「近傍性グループのパラメータ」
- 104 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
NFS チューニング可能パラメータ	第3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4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」
ネットワークキャッシュとアクセラレータ (NCA) のチューニング可能パラメータ	第5章「ネットワークキャッシュとアクセラレータ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

この節では、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。

physmem

備考欄	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	そのシステムで利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
検査	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

備考欄

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 に優先します。[38 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。
- AMD64 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。
- AMD64 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。

検査

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最もよい方法は、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

備考欄

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 (AMD64 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。
- AMD64 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。

検査

8192 以上、262,144 (256 x 1024) 以下にする必要があります。また、システムページサイズの倍数でなければなりません。これらの条件が満たされないと、次のメッセージが表示されます。

Illegal stack size, Using N

どのような場合に変更するか	N の値は、<code>lwp_default_stksize</code> のデフォルト値です。 スタック容量が足りないためにシステムがパニックになる場合。この問題を解決する最もよい方法は、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、 206 ページの「<code>lwp_default_stksize</code> (Solaris 9 リリース)」 を参照してください。

logevent_max_q_sz

備考欄	キューに格納して <code>syseventd</code> デモンへの配信を待機させることのできる、システムイベントの最大数です。システムイベントキューのサイズがこの制限に達すると、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
検査	<code>ddi_log_sysevent</code> と <code>sysevent_post_event</code> によってシステムイベントが生成されるたびに、システムイベントフレームワークはこの値をチェックします。

	詳細は、 <code>ddi_log_sysevent(9F)</code> および <code>sysevent_post_event(3SYSEVENT)</code> のマニュアルページを参照してください。
どのような場合に変更するか	システムイベントのログ、生成、または送信が失敗したことをエラーログメッセージが示す場合。
コミットレベル	変更の可能性あり

segkpsize

備考欄

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 ビットカーネルでは 2G バイト 32 ビットカーネルでは 512M バイト
範囲	64 ビットカーネルでは 512M バイトから 24G バイト
単位	8K バイトのページ
動的か	いいえ
検査	最小サイズと最大サイズ (64 ビットシステムではそれぞれ 512 バイトと 24G バイト) に対して値を比較します。最小サイズに満たないか、または最大サイズ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。 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。
どのような場合に変更するか	システム上で多数のプロセスをサポートしなければならない場合。デフォルトサイズの 2G バイトは、

少なくとも 1G バイトの物理メモリーがあるという前提です。このデフォルトサイズによって、87,000 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。64 ビットカーネルのスタックサイズは、プロセスが 32 ビットプロセスでも 64 ビットプロセスでも同じです。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 `segkpsize` を増やすことができます。

コミットレベル

変更の可能性あり

変更履歴

[206 ページの「segkpsize \(Solaris 9 12/02 リリース\)」](#)を参照してください。

noexec_user_stack

備考欄

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っそう困難になります。

64 ビットカーネルが動作している 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 Solaris 2.6、7、8、9、または 10 リリースが動作しているすべてのシステム上に存在しますが、有効になるのは 64 ビットの SPARC と AMD64 のアーキテクチャーの場合に限られます。

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。
検査	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	206 ページの「 <code>noexec_user_stack</code> (Solaris 9 リリース)」を参照してください。

fsflush とそれに関連するパラメータ

この節では、`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つの作業を行います。

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、メモリーの一部分を検証し、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ある
 - ページは使用されていない
 - ページがロックされている
 - ページにスワップデバイスが対応づけられている
 - ページが入出力操作に現在関与している

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

ページはバッキングストアにフラッシュされますが、それを使用しているプロセスとの接続は保たれます。フラッシュしておく、システムのメモリーが不足したときのページの再利用が簡単になります。これは、フラッシュ後にそのページ

が変更されていなければ、ページを回収する前にそのページをバッキングストアに書き出す必要がなくなり、遅延を避けられるからです。

3. fsflush はファイルシステムのメタデータをディスクに書き込みます。この書き込みは n 回目の呼び出しごとに行われます。 n はさまざまな構成変数から計算されます。詳細は、43 ページの「`tune_t_fsflushr`」と 43 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

備考欄	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
検査	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

備考欄	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、autoup の値は、空リストからバッファーを書き出すかどうかの制御にも使用されます。 B_DELWRI フラグが付いているバッファー (変更されているファイルコンテンツページを示す) は、空リストに置かれている時間が autoup 秒を超えると書き出されます。autoup の値を増やすと、バッファーがメモリーに置かれている時間が長くなります。
データ型	符号付き整数
デフォルト	30
範囲	1 から MAXINT
単位	秒
動的か	いいえ
検査	autoup がゼロ以下の場合は、30 に再設定され、警告メッセージが出力されます。この確認が行われるのはブート時だけです。
暗黙的制約	autoup は tune_t_fsflushr の整数倍でなければなりません。最小でも autoup は tune_t_fsflushr 値の 6 倍以上でなければなりません。そうでないと、fsflush が呼び出されるたびに余計なメモリーが走査されます。 dopageflush がゼロでない場合にメモリーを検査するには、全体のシステムページ数に tune_t_fsflushr を掛け合わせた値が autoup 以上でなければなりません。
どのような場合に変更するか	autoup または tune_t_fsflushr (あるいはその両方) の変更が必要になる状況はいくつかあります。 ■ 大きなメモリーをもつシステム – この場合には、autoup を増やすと、fsflush の個々の呼び出しで走査されるメモリー量が少なくなります。 ■ メモリーの要求量が最小限のシステム – autoup と tune_t_fsflushr を両方とも増やすと、走査の回数が減ります。autoup 対 tune_t_fsflushr の現在の比率を維持するには autoup も増やす必要があります。

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

コミットレベル

変更の可能性あり

dopageflush

備考欄

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 $\times$ tune_t_fsflushr / autoup ページ

データ型

符号付き整数

デフォルト

1 (有効)

範囲

0 (無効)、1 (有効)

単位

切り替え (オン/オフ)

動的か

はい

検査

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

doiflush

備考欄

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 (autoup / 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、<code>tune_t_fsflushr</code> が <code>autoup</code> より大きいと、反復カウンタが <code>N</code> 以上であるかどうかをコードがチェックするので、同期化は <code>fsflush</code> が呼び出されるたびに行われます。<code>N</code> は <code>fsflush</code> を実行するときに 1 度だけ計算されることに注意してください。その後で <code>tune_t_fsflushr</code> や <code>autoup</code> を変更しても、同期化操作の頻度に影響はありません。
データ型	符号付き整数
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
単位	切り替え (オン/オフ)
動的か	はい
検査	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 (<code>/tmp</code> など) に置いた方がいいでしょう。Solaris 7 以降、<code>mount</code> の <code>-noatime</code> オプションを使用することによって、システム上の <code>i</code> ノードトラフィックを減らすことができます。このオプションを使うと、ファイルがアクセスされた時に <code>i</code> ノードの更新が行われません。 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

備考欄

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。現在の Solaris リリースでは、規模調節の大半をシステム上のメモリー容量に基づいて行います。したがって、maxusers の使い方がこれまでとは大きく変わりました。引き続き、maxusers に基づいて決定されるサブシステムには次のものがあります。

- システムで利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総量か 2048 (小さい方)

範囲

/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。

/etc/system ファイルに設定されている場合は、1 から 4096。

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。

検査

なし

どのような場合に変更するか

システムによって計算されたデフォルトのユーザープロセス数が小さすぎる場合。このような状況は、システムコンソールに表示される次のメッセージでわかります。

out of processes

次の状況のように、デフォルトのプロセス数が多すぎる場合に、このパラメータを変更するかもしれません。

- 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節約できます。
- 多量のメモリーがあり、動作しているプロセスがほとんどないファイルサーバーでは、この値を減らせる場合があります。しかし、その場合、DNLCのサイズを明示的に設定する必要があります。73 ページの「`ncsize`」を参照してください。
- 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計算サーバーでは、この値を減らせる場合があります。

コミットレベル

変更の可能性あり

reserved_procs

備考欄

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。

検査

Solaris 8 以降のリリースでは、どのような `/etc/system` 設定でも有効です。

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、通常の値から 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

備考欄

使用可能な最大プロセス ID の値を指定します。これは Solaris 8 以降のリリースで有効です。

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、妥当性検査を行うために使用されます。

/etc/system ファイルに maxpid エントリを追加して設定しようとしても、効果はありません。

データ型

符号付き整数

デフォルト

30,000

範囲

266 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。

検査

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。

コミットレベル

変更の可能性あり

max_nprocs

備考欄

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

この値は、他のいくつかのシステムデータ構造体のサイズを決定する場合にも使用されます。このパラメータが作用する他のデータ構造体は、次のとおりです。

- 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

10 + (16 x maxusers)

範囲

266 から maxpid の値

動的か

いいえ

検査

はい。値は maxpid と比較され、それより大きい場合は maxpid に設定されます。x86 プラットフォームでは、さらにプラットフォーム固有の値と比較されます。max_nprocs は、max_nprocs、maxpid、プラットフォーム値のうち最も小さい値に設定されます。SPARC プラットフォームと x86 プラットフォームはどちらもプラットフォーム値として 65,534 を使用します。

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

[205 ページの「max_nprocs \(Solaris 9 リリース\)」](#)を参照してください。

maxuprc

備考欄

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
検査	はい。この値は max_nprocs - reserved_procs と比較され、2つの値のうちの小さい方に設定されます。
どのような場合に変更するか	1 ユーザーが作成できるプロセスの数を強く制限するために、デフォルト値より小さい値を指定したい場合 (システムが作成できるプロセスの数が多くても)。この限度を超えると、次の警告メッセージがコンソールかメッセージファイルに出力されます。
	out of per-user processes for uid N
コミットレベル	変更の可能性あり

ページング関連パラメータ

Solaris OS では、必要に応じてページングされる仮想メモリーシステムを使用します。システム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、[60 ページの「fastscan」](#) および [60 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す。メモリーへの負荷が大きくなると、それに比例してページアウト操作をサポートするために消費されるCPU時間が増加し、最大で1つのCPUの80%が消費されます。このアルゴリズムは、`slowscan` と `fastscan` の間のメモリー量の一部を調べ、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った。
- 予定のページ数を調べた。
- 長すぎる時間が経過した。

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

Solaris 9 以降のリリースでは、ページングサブシステムの構成機構が変更されています。システムは `fastscan`、`slowscan`、および `handspreadpages` の事前定義された値を使用せずに、ブート時にこれらのパラメータへ適切な値を割り当てます。`/etc/system` ファイル内のこれらのパラメータを設定すると、システムが最適でない数値を使用する場合があります。



注意 - `/etc/system` ファイルから、VM システムのチューニングをすべて削除してください。まずデフォルトの設定値で実行してから、これらのパラメータの調整が必要かどうかを判定してください。また、`cachefree` および `priority_paging` を設定しないでください。これらは Solaris 9 以降のリリースでは削除されています。

Solaris 7 5/99 リリースから、CPU とメモリーに対する動的再構成 (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ければ、そのパラメータ値が再計算されます。明示的に設定されている場合は、変数の値に対する制約に反しない限り、`/etc/system` に指定された値が使用されます。違反する場合は、値がリセットされます。

lotsfree

備考欄	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 か 512K バイト (大きい方)
範囲	最小値は、512K バイトか物理メモリーの 1/64 の大きい方です。この値は、<code>getpagesize</code> から返されるページサイズに基づくページ数で表わされます。詳細は、<code>getpagesize(3C)</code> を参照してください。 最大値は物理メモリーのページ数です。この最大値は、物理メモリーの 30% 以内であるべきです。システムは、「検査」の項目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。
検査	<code>lotsfree</code> が物理メモリーの総量より大きい場合、値はデフォルトにリセットされます。
暗黙的制約	<code>lotsfree</code> が <code>desfree</code> よりも大きく、 <code>desfree</code> が <code>minfree</code> よりも大きいという関係が常に維持されるようにする必要があります。
どのような場合に変更するか	ページ要求が急激に増えるような場合には、メモリーアルゴリズムが要求に対応できないことがあります。これを回避するには、早期にメモリーの回収を開始するのも 1 つの方法です。これは、ページングシステムにいくらか余裕を与えることになります。 経験則によると、このパラメータは、システムが 2 ～ 3 秒で割り当てる必要がある量の 2 倍にします。このパラメータの適正值は負荷によって異なります。DBMS サーバーはデフォルトの設定で支障がないはずですが、しかし、ファイルシステムの入出力負荷が非常に大きい場合は、このパラメータを調整する必要があるかもしれません。

コミットレベル	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。指定可能な最小値は512K バイトで、この値は <code>getpagesize</code> から返されるページサイズに基づくページ数で表わされます。 変更の可能性あり
---------	---

desfree

備考欄	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	<code>lotsfree / 2</code>
範囲	最小値は256K バイトか物理メモリーの1/128 の大きい方です。この値は <code>getpagesize</code> から返されるページサイズに基づくページ数で表わされます。 最大値は物理メモリーのページ数です。この最大値は物理メモリーの15% 以内であるべきです。システムは、「検査」の項目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。
検査	<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。

	■ 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増えることがあります。この副次的な影響の結果、アプリケーションの動作が本来よりも遅くなる可能性があります。
どのような場合に変更するか	負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。指定可能な最小値は 256K バイトで、この値は <code>getpagesize</code> から返されるページサイズに基づくページ数で表わされます。
コミットレベル	変更の可能性あり

minfree

備考欄

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 バイトか物理メモリーの 1/256 の大きい方です。この値は `getpagesize` から返されるページサイズに基づくページ数で表わされます。

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、「検査」の項目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。
検査	<code>minfree</code> が <code>desfree</code> より大きい場合、 <code>minfree</code> は <code>desfree/2</code> に設定されます。メッセージは表示されません。
暗黙的制約	<code>lotsfree</code> が <code>desfree</code> よりも大きく、 <code>desfree</code> が <code>minfree</code> よりも大きいという関係が常に維持されるようにする必要があります。
どのような場合に変更するか	一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。指定可能な最小値は 128K バイトで、この値は <code>getpagesize</code> から返されるページサイズに基づくページ数で表わされます。
コミットレベル	変更の可能性あり

throttlefree

備考欄	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	<code>minfree</code>
範囲	最小値は 128K バイトか物理メモリーの 1/256 の大きい方です。この値は <code>getpagesize</code> から返されるページサイズに基づくページ数で表わされます。 最大値は物理メモリーのページ数です。この最大値は物理メモリーの 4% 以内であるべきです。システムは、「検査」の項目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。
検査	<code>throttlefree</code> が <code>desfree</code> よりも大きい場合、 <code>throttlefree</code> は <code>minfree</code> に設定されます。メッセージは表示されません。
暗黙的制約	<code>lotsfree</code> が <code>desfree</code> よりも大きく、 <code>desfree</code> が <code>minfree</code> よりも大きいという関係が常に維持されるようにする必要があります。
どのような場合に変更するか	一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。指定可能な最小値は 128K バイトで、この値は <code>getpagesize</code> から返されるページサイズに基づくページ数で表わされます。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

備考欄	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。この変数は、メモリー不足が極めて深刻な場合でもページアウト操作を行えるように、Solaris 2.6 リリースから導入されました。
データ型	符号なし整数
デフォルト	<code>throttlefree / 2</code>
範囲	最小値は 64K バイトか物理メモリーの 1/512 の大きい方です。この値は <code>getpagesize(3C)</code> から返されるページサイズに基づくページ数で表わされます。 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、「検査」の項目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。
検査	<code>pageout_reserve</code> が <code>throttlefree / 2</code> より大きい場合、 <code>pageout_reserve</code> は <code>throttlefree / 2</code> に設定されます。メッセージは表示されません。
暗黙的制約	<code>lotsfree</code> が <code>desfree</code> よりも大きく、 <code>desfree</code> が <code>minfree</code> よりも大きいという関係が常に維持されるようにする必要があります。
どのような場合に変更するか	一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。指定可能な最小値は 64K バイトで、この値は <code>getpagesize</code> から返されるページサイズに基づくページ数で表わされます。
コミットレベル	変更の可能性あり

pages_pp_maximum

備考欄	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	<code>tune_t_minarmem + 100</code> か、ブート時に使用可能なメモリーの 4% + 4M バイト (大きい方)
範囲	システムが強制する最小値は <code>tune_t_minarmem + 100</code> です。最大値については、システムは強制しません。
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点でこの値は、 <code>/etc/system</code> ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。

検査	/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 <code>tune_t_minarmem + 100</code> よりも小さい場合、この値は <code>tune_t_minarmem + 100</code> へリセットされます。 /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。検査は、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。
どのような場合に変更するか	メモリーのロック要求や、<code>SHARE_MMU</code> フラグを指定した共有メモリーセグメントへの接続が失敗したが、使用可能なメモリーが十分ありそうな場合。 大きすぎる値が原因で、メモリーのロック要求 (<code>mlock</code>、<code>mlockall</code>、および <code>memcntl</code>) が不必要に失敗する場合。詳細は、<code>mlock(3C)</code>、<code>mlockall(3C)</code>、および <code>memcntl(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	207 ページの「 pages_pp_maximum (Solaris 9 リリース以前の Solaris リリース) 」を参照してください。

tune_t_minarmem

備考欄	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
検査	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

備考欄	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	64M バイトか物理メモリーの 1/2 (小さい方)
範囲	1 から物理メモリーの 1/2
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。
検査	最大値は、64M バイトと物理メモリーの 1/2 のうちで小さい方です。
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

備考欄	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。
検査	<code>slowscan</code> が <code>fastscan/2</code> より大きい場合、 <code>slowscan</code> は <code>fastscan/2</code> 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

備考欄	<code>pageout</code> 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
検査	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

備考欄	Solaris OS は双針クロック アルゴリズムを使用し、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。使用されていないければ、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。
検査	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

備考欄	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

検査	なし
どのような場合に変更するか	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。
	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。
コミットレベル	変更の可能性あり
maxpgio	
備考欄	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
検査	なし
暗黙的制約	ページャからの入出力要求の最大数は、要求バッファのリストのサイズによって制限されます。現在のサイズは256です。
どのような場合に変更するか	このパラメータはメモリ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ことがあります。既存の

入出力サブシステムは、追加される入出力の負荷に対処できる必要があります。また、スワップパーティションとアプリケーションファイルが同じディスク上にある場合、スワップ入出力の増加はアプリケーションの入出力のパフォーマンスを低下させることがあります。

コミットレベル

変更の可能性あり

変更履歴

詳細は、[207 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

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`

備考欄

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 バイトか物理メモリーの 1/16 (小さい方)

範囲

最小値は 4M バイトか物理メモリーの 1/16 の小さい方です。この値は `getpagesize` から返されるページサイズに基づくページ数で表わされます。

最大値は物理メモリーのページ数です。最大値は、物理メモリーの 10% 以内であるべきです。システムは、「検査」の項目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

検査

なし

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

swapfs_minfree

備考欄	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 バイトか物理メモリーの 1/8 (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
検査	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
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

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断片を配分します。アロケータは、そのクライアントが使用するさまざまなサイズのキャッシュを作成します。一方、クライアントは、特定サイズの構造体の割り当てのためなど、クライアントが使用するキャッシュの作成をアロケータに要求できます。アロケータが管理する各キャッシュに関する統計は、`kstat -c kmem_cache` コマンドで表示できます。

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

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

kmem_flags

備考欄

Solaris カーネルメモリアロケータには、デバッグとテストのための各種オプションがあり、Solaris OS の内部開発サイクルを通じて、これらの機能が盛んに利用されました。Solaris 2.5 リリース以降、このようなオプションのサブセットが利用できるようになりました。これらは `kmem_flags` 変数によって制御されていましたが、この変数はカーネルデバッグで設定され、その後システムをリブートしていました。カーネルメモリアロケータのインスタンス化と `/etc/system` ファイルの解析のタイミングの問題のため、これらのフラグは Solaris 8 より前のリリースでは `/etc/system` ファイルに設定できませんでした。

次に、サポートされる 5 つのフラグの設定について説明します。

フラグ	設定	説明
AUDIT	0x1	アロケータは、自身の活動の最近の履歴が入ったログを維持します。ログされる項目の数は、 <code>CONTENTS</code> も設定されているかどうかによって異なります。このログは固定の大きさです。領域を使い果たすと、古い記録から再利用されます。

フラグ	設定	説明
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、論理和をとることができ、Solaris 8 リリースでは /etc/system ファイルで設定し、それより前のリリースでは kadb をブートして、カーネルを起動する前にこれらのフラグを設定します。

	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、Solaris 7 3/99 リリースから使用できるようになりました。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。
検査			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

備考欄	モジュールの読み込み処理の各段階に関するメッセージを表示する場合に使用します。
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	最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。

```
Nov 5 16:12:28 sys genunix: [ID 943528 kern.notice]
load 'sched/TS_DPTBL' id 9 loaded @ 0x10126438/
0x10438dd8 size 132/2064
Nov 5 16:12:28 sys genunix: [ID 131579 kern.notice]
installing TS_DPTBL, module id 9.
```

- 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。

```
Nov 5 16:16:50 sys krtld: [ID 284770 kern.notice]
kobj_open: can't open /platform/SUNW,Ultra-80/kernel/
sched/TS_DPTBL
Nov 5 16:16:50 sys krtld: [ID 284770 kern.notice]
kobj_open: can't open /platform/sun4u/kernel/sched/
TS_DPTBL
Nov 5 16:16:50 sys krtld: [ID 797908 kern.notice]
kobj_open: '/kernel/sch...
Nov 5 16:16:50 sys krtld: [ID 605504 kern.notice]
descr = 0x2a
Nov 5 16:16:50 sys krtld: [ID 642728 kern.notice]
kobj_read_file: size=34,
Nov 5 16:16:50 sys krtld: [ID 217760 kern.notice]
offset=0
Nov 5 16:16:50 sys krtld: [ID 136382 kern.notice]
kobj_read: req 8192 bytes,
Nov 5 16:16:50 sys krtld: [ID 295989 kern.notice]
got 4224
Nov 5 16:16:50 sys krtld: [ID 426732 kern.notice]
read 1080 bytes
Nov 5 16:16:50 sys krtld: [ID 720464 kern.notice]
copying 34 bytes
Nov 5 16:16:50 sys krtld: [ID 234587 kern.notice]
count = 34
[33 lines elided]
Nov 5 16:16:50 sys genunix: [ID 943528 kern.notice]
load 'sched/TS_DPTBL' id 9 loaded @ 0x10126438/
0x10438dd8 size 132/2064
Nov 5 16:16:50 sys genunix: [ID 131579 kern.notice]
installing TS_DPTBL, module id 9.
Nov 5 16:16:50 sys genunix: [ID 324367 kern.notice]
init 'sched/TS_DPTBL' id 9 loaded @ 0x10126438/
0x10438dd8 size 132/2064
```

- 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
検査	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

一般的な入出力パラメータ

maxphys

備考欄	物理入出力要求の最大サイズを指定します。要求がこのサイズより大きいと、ドライバはこの要求を maxphys サイズの断片に分割します。個々のファイルシステムは独立して制限値を持つことが可能で、実際に独立した制限値を持ちます。
データ型	符号付き整数
デフォルト	131,072 (sun4u または sun4v) または 57,344 (x86)。ワイド転送をサポートする sd ドライバは 1,048,576 を使用します。ssd ドライバはデフォルトで 1,048,576 を使用します。
範囲	マシン固有のページサイズから MAXINT
単位	バイト
動的か	はい。しかし、多くのファイルシステムでは、ファイルシステムがマウントされるときに、この値がマウントポイントごとのデータ構造体に設定され

検査	なし
どのような場合に変更するか	raw デバイスに対する入出力を大きな断片で行う場合。OLTP 操作を伴う DBMS では小さいサイズの入出力が頻繁に行われることに留意してください。その場合、maxphys を変更しても性能の向上は望めません。
コミットレベル	変更の可能性あり
変更履歴	詳細は、 205 ページの「maxphys (Solaris 10 リリース)」 を参照してください。
rlim_fd_max	
備考欄	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。この制限を変更するには、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
検査	なし
どのような場合に変更するか	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。次に例を示します。

- 標準入出力を使用する 32 ビットプログラムでは、256 ファイル記述子に制限されます。標準入出力を使用する 64 ビットプログラムでは、20 億の記述子まで使用できます。具体的には、標準入出力は libc(3LIB) の stdio(3C) 関数を指します。
- select はデフォルトで、fd_set につき 1024 の記述子に制限されます。詳細は、select(3C) を参照してください。ただし、Solaris 7 リリース以降では、32 ビットアプリケーションコードをより大きな fd_set サイズ (65,536 以下) で再コンパイルできます。64 ビットアプリケーションの fd_set サイズは 65,536 で、変更することはできません。

システム全体に対してこれを変更する別の方法として plimit(1) コマンドがあります。plimit を使用して親となるプロセスの限度を変更すると、すべての子プロセスがその限度を継承します。この方法は inetd などのデーモンに有効です。

コミットレベル

変更の可能性あり

変更履歴

206 ページの「rlim_fd_max (Solaris 8 リリース)」を参照してください。

rlim_fd_cur

備考欄

1つのプロセスが開くことのできるファイル記述子に対して、「ソフト」限度を指定します。プロセスは、自身のファイル記述子数限度を、rlim_fd_max で指定される「強い」限度の範囲内で調整できます。それには、setrlimit() 呼び出しを使用するか、プロセスを実行するシェルで limit コマンドを実行します。「強い」限度の範囲内で値を調整する場合には、スーパーユーザー特権は必要ありません。

データ型

符号付き整数

デフォルト

256

範囲

1 から MAXINT

単位

ファイル記述子

動的か	いいえ
検査	<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> を使用して自身で利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`

備考欄	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 や NFS 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。 Solaris 8 6/00 リリース以降、DNLC は否定的な検索情報もキャッシュします。つまり、キャッシュ内で検出されない名前がキャッシュされます。
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 <code>MAXINT</code>
単位	DNLC のエントリ
動的か	いいえ
検査	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	Solaris 8 6/00 より前のリリースでは、キャッシュが小さすぎるかどうかを判定するのは困難でした。これを推定するには、 <code>kstat -n ncstats</code> が返したエン

トリ数に着目します。システムの負荷やファイルアクセスのパターンに較べこの値が大きいように思われる場合は、DNLCのサイズに原因があるかもしれません。

Solaris 8 6/00 リリース以降では、`kstat -n dnlcstats` 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。`pick_heuristic` パラメータと `pick_last` 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。

`ncsize` 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、`ncsize` 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`ncsize` 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`ncsize` に64バイトの構造体をそれぞれ割り当てます。`ufs_ninode` と `nfs:nrnode` 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、[207 ページの「`ncsize` \(Solaris 10 リリース\)」](#)を参照してください。

rstchown

備考欄

`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
検査	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	旧式

dnlc_dir_enable

備考欄	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 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。
検査	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、 <code>dnlc_dir_enable</code> 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

備考欄	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 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。
検査	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

備考欄	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 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。
検査	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

segmap_percent

備考欄	高速アクセスファイルシステムキャッシュに使用するメモリの最大量を指定します。このメモリープールは空きメモリーリストから差し引かれます。
データ型	符号なし整数
デフォルト	システム起動時の空きメモリーの 12%。
範囲	2M バイトから <code>physmem</code> の 100%。
単位	物理メモリーの %
動的か	いいえ
検査	なし
どのような場合に変更するか	活発なファイルシステム動作が予想され、なおかつ使用可能な空きメモリーが十分にある場合は、このパラメータの値を増やします。
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

備考欄	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。割り当てメモリー量 (K バイト単位) が <code>bufhwm</code> 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、バッファーク্যাッシュからメタデータが破棄されます。 歴史的経緯により、<code>bufhwm</code> には <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	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 2TB のどちらか小さい方。つまり、 <code>bufhwm_pct</code> は 1 から 20 にできます。
単位	<code>bufhwm</code>: K バイト <code>bufhwm_pct</code>: 物理メモリーの %。
動的か	いいえ。<code>bufhwm</code> および <code>bufhwm_pct</code> 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、この値が調整されます。 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。 稼働中に <code>bufhwm</code> または <code>bufhwm_pct</code> を変更しても無効です。
検査	<code>bufhwm</code> 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%、2TB、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す。無効な値を試みると、システムコンソールと <code>/var/adm/messages</code> ファイルに次のメッセージが出力されます。 <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。

512 M 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 に拡大されます。したがって、ハッシュヘッダーは 512×96 バイト、すなわち 48 K 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 32 K 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突き止め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 バイトシステム上でこの構造体が、10144K バイトのうち 4K バイトだけがまだ割り当てられていないことを示したとします。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
 変更履歴 [207 ページの「bufhwm \(Solaris 9 リリース\)」](#) を参照してください。

ndquot

備考欄 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、`ufs:` 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

検査 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。

`dquot table full`

コミットレベル 変更の可能性あり

ufs_ninode

備考欄 メモリー内で維持すべき i ノードの数を指定します。i 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。

この場合のキーとなるパラメータは `ufs_ninode` です。このパラメータを使用して、`i` ノード キャッシュの処理に関係する2つのキーとなる境界値が計算されます。高位境界値は `ufs_ninode / 2`、下位境界値は `ufs_ninode / 4` で計算されます。

システムが `i` ノードの処理を終わると、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つめの機構が使用されます。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

検査

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 を使用して、inode_cache kstat を調べることで判断できます。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

備考欄	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
検査	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

備考欄	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。 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。
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
検査	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

備考欄	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
検査	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

備考欄	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。 大容量メモリーシステムには、深刻なメモリー不足を引き起こすことなく、10M バイトのファイルを千単位でキャッシュできるだけの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。 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。
データ型	符号付き整数
デフォルト	32,768
範囲	0 から 2,147,483,647
動的か	はい
検査	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

	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 バイトのファイルを意味します。
コミットレベル	変更の可能性あり

TMPFS パラメータ

`tmpfs:tmpfs_maxkmem`

備考欄	TMPFS がデータ構造体 (<code>tmp</code> 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 (<code>sun4u</code> か <code>sun4v</code> 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
検査	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、 209 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

備考欄	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
検査	なし
どのような場合に 変更するか	TMPFS が大量に使用されるシステムで適度なスワップ空間を維持するために、この値を増やすことができます。次のメッセージがコンソールやメッセージファイルに出力された場合は、使用量がこの限度に達したことを示しています。

fs-name: File system full, swap space limit exceeded

コミットレベル	変更の可能性あり
変更履歴	209 ページの「tmpfs:tmpfs_minfree (Solaris 8 リリース)」 を参照してください。

仮想端末

仮想端末 (pty) は、Solaris ソフトウェアにおいて主に 2 つの目的で使用されます。

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

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

旧バージョンの Solaris では、pty の望ましい値をシステムに明示的に設定する手順を踏む必要がありました。Solaris 8 リリース以降、新しい方式によって、ほとんどの状況でチューニングが不要になりました。pty のデフォルト値は、現在、システムのメ

メモリ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使Ⓐできるカーネルメモリーの割合 (%)。ゼロを指定すると、リモートユーザーがシステムにログインすることはできません。
- `pt_max_pty` – `pty` 数の強い制限の最大値

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

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

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

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

このメモリーは、`pty` のサポートに使Ⓐ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

備考欄

使Ⓐできる `/dev/pts` エントリの数は、システム上で使Ⓐ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 がデフォルトの最大数のうち大きい方まで割り当てます。
データ型	符号なし整数
デフォルト	0
範囲	0 から maxpid
単位	ログイン/ウィンドウ
動的か	いいえ
検査	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

備考欄	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
検査	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

備考欄	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
-----	----

検査	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

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

strmsgsz

備考欄	1つのシステム呼び出しで STREAM に渡し、メッセージのデータ部分に格納できる最大バイト数を指定します。このサイズを超える <code>write</code> は、複数のメッセージに分割されます。詳細は、 <code>write(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	65,536
範囲	0 から 262,144
単位	バイト
動的か	はい
検査	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、 <code>putmsg(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

strctlsz

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

範囲	0 から MAXINT
単位	バイト
動的か	はい
検査	なし
どのような場合に変更するか	ソフトウェアベンダーの指定がある場合。この限度を超えると、putmsg(2) 呼び出しから ERANGE が返されます。
コミットレベル	変更の可能性あり

System V メッセージキュー

System V メッセージキューは、カーネルが作成したキューを使用してメッセージを交換する、メッセージ転送インタフェースを提供します。Solaris 環境では、メッセージをキューに入れたりキューを解除したりするためのインタフェースが提供されます。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。

Solaris 10 リリースの System V メッセージキューに関する詳細については、[22 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、『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 6 章「資源制御 (概要)」を参照してください。

旧式の System V メッセージキューの詳細は、[220 ページ](#)の「[旧式または削除されたパラメータ](#)」を参照してください。

System V セマフォ

System V セマフォは 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。

Solaris 10 リリースにおけるセマフォーリソースの変更については、[22 ページの「System V IPC 構成」](#)を参照してください。

Solaris 10 リリースでの新しい資源制御の使用方法については、『[Solaris のシステム管理 \(Solaris コンテナ: 資源管理と Solaris ゾーン\)](#)』の第 6 章「[資源制御 \(概要\)](#)」を参照してください。

旧式の System V セマフォーパラメータの詳細は、[220 ページの「旧式または削除されたパラメータ」](#)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は shmsys: 接頭辞を含んでいる必要があります。Solaris 7 リリースから、keyserd デーモンは System V 共有メモリーを使用するようになりました。

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資源は、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。

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、[22 ページの「System V IPC 構成」](#)を参照してください。

Solaris 10 リリースでの新しい資源制御の使用方法については、『[Solaris のシステム管理 \(Solaris コンテナ: 資源管理と Solaris ゾーン\)](#)』の第 6 章「[資源制御 \(概要\)](#)」を参照してください。

旧式の System V 共有メモリーパラメータの詳細は、[220 ページの「旧式または削除されたパラメータ」](#)を参照してください。

segspt_minfree

備考欄	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
検査	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、ほとんどの場合、最大値 128M バイト (0x4000) で十分です。
コミットレベル	変更の可能性あり

スケジューリング

rechoose_interval

備考欄	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ます。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
検査	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、Solaris 2.6 リリース以来利用可能なプロセッサセット機能を使用するか、またはプロセッサ結合を使用することを検討してください。詳細は、 <code>psrset(1M)</code> または <code>pbind(1M)</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

タイマー

hires_tick

備考欄	このパラメータを設定すると、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 の代わりに 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。
検査	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

備考欄	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。
検査	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 <code>timer_create</code> システムコールの実行時に、EAGAIN エラーを受け取ります。
コミットレベル	変更の可能性あり

sun4u または sun4v 固有のパラメータ

`consistent_coloring`

備考欄

Solaris 2.6 リリースから、UltraSPARC® (sun4u) 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導入されました。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にと起こると、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。Solaris 8 リリースでは、これがデフォルトのアルゴリズムです。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

はい

検査

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

備考欄

`tsb_alloc_hiwater` を初期化して、変換記憶バッファ (TSB) に割り当てることのできる物理メモリー量に、次のように上限を設けます。

`tsb_alloc_hiwater` = 物理メモリー (バイト数) /
`tsb_alloc_hiwater_factor`

TSB に割り当てられたメモリーが
`tsb_alloc_hiwater` の値と等しい場合、TSB メモリー割り当てアルゴリズムはマッピングされていないページとして TSB メモリーを再利用しようとします。

この係数を使用して `tsb_alloc_hiwater` の値を増やす場合は、注意が必要です。システム停止を防止するには、高位境界値が `swapfs_minfree` と `segspt_minfree` の値よりかなり小さくなるようにする必要があります。

データ型

整数

デフォルト

32

範囲

1 から MAXINIT

係数 1 の場合、すべての物理メモリーを TSB に割り当てることができるようになるので、システムが停止する可能性があります。また、係数が大きすぎると、TSB に割り当てることができるメモリーが残らないので、システムパフォーマンスが低下します。

動的か

はい

検査

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

備考欄

すべてのプロセスに割り当てる初期変換記憶バッファ (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) です。これは 512 エントリに対応します。

範囲	指定可能な値は、次のとおりです。																
	<table><tr><th>値</th><th>説明</th></tr><tr><td>0</td><td>8K バイト</td></tr><tr><td>1</td><td>16K バイト</td></tr><tr><td>3</td><td>32K バイト</td></tr><tr><td>4</td><td>128K バイト</td></tr><tr><td>5</td><td>256K バイト</td></tr><tr><td>6</td><td>512K バイト</td></tr><tr><td>7</td><td>1M バイト</td></tr></table>	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
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																
検査	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、 210 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

備考欄	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
検査	はい
どのような場合に変更するか	0 に設定すると、TSB の増加を防ぐことができます。ほとんどの場合、このパラメータはデフォルト設定のままにしておくべきです。
コミットレベル	変更の可能性あり
変更履歴	詳細は、 210 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

備考欄	RSS 発見的容量調整の RSS 対 TSB 期間比を制御します。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 期間の割合が出ます。
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
検査	なし
どのような場合に変更するか	TSB での仮想アドレスの衝突による場合など、システムが TSB ミスに起因する過度の数のトラップに直面している場合は、この値を 0 に減らしてもよいかもしれません。 たとえば、tsb_rss_factor を 384 (事実上は 75%) でではなく 256 (事実上は 50%) に変更すると、状況によっては、TSB における仮想アドレスの衝突を排除できることがあります。特に負荷の大きいシステムでは、カーネルメモリーの使用量が増えます。 TSB の動きは、trapstat -T コマンドで監視できます。

コミットレベル

変更の可能性あり

変更履歴

詳細は、[210 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

この節では、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`

備考欄

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遠隔メモリーグループからであれば要求を満たせるときにヒューリスティックを制御します。

デフォルトでは、ローカルの空きメモリーは断片化されているが、遠隔の空きメモリーは断片化されていない場合に、Solaris OS は遠隔大規模ページを割り当てます。このパラメータを 1 に設定した場合、大規模なメモリーページをローカルで割り当てるため、たとえばローカルのメモリーグループ内で小さなページを集めて大きなページに合体させるといった追加動作が行われます。

データ型

ブール型

デフォルト

0 (ローカルの空きメモリーが断片化されていて、遠隔の空きメモリーが断片化されていない場合は、遠隔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、遠隔の空きメモリーが断片化されていない場合は、遠隔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、遠隔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

検査	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。 ページの場所とサイズは、http://opensolaris.org で入手可能な NUMA 可観測性ツールを使用して監視できます。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	コミットされていない

lgrp_mem_default_policy

備考欄	この変数は、Solaris OS が使用するデフォルトのメモリー割り当てポリシーを反映するものです。この変数は整数であり、その値は <code>sys/lgrp.h</code> ファイルに示されているいずれかのポリシーに対応しているようにします。
データ型	整数
デフォルト	1 (LGRP_MEM_POLICY_NEXT)。これはメモリー割り当てが、デフォルトでメモリー割り当てを実行しているスレッドのホーム lgroup になることを示します。
範囲	指定可能な値は、次のとおりです。

値	説明	コメント
0	LGRP_MEM_POLICY_DEFAULT	システムのデフォルトのポリシーを使用
1	LGRP_MEM_POLICY_NEXT	スレッドのホーム lgroup の割り当ての次
2	LGRP_MEM_POLICY_RANDOM_PROC	プロセスにわたってランダム
3	LGRP_MEM_POLICY_RANDOM_PSET	プロセッサセットにわたってランダム
4	LGRP_MEM_POLICY_RANDOM	すべての lgroup にわたってランダム
5	LGRP_MEM_POLICY_ROUNDROBIN	すべての lgroup にわたってラウンドロビン
6	LGRP_MEM_POLICY_NEXT_CPU	メモリーにアクセスするための近接した次の CPU

動的か	いいえ
検査	なし
どのような場合に変更するか	使用するアプリケーションが、メモリーの待ち時間に影響を受ける場合。この待ち時間とは、NUMA を使用するシステム上でローカルメモリーの代わりに遠隔メモリーを割り当てる際に発生する時間をさします。
コミットレベル	コミットされていない

lgrp_mem_pset_aware

備考欄	プロセスがユーザープロセッサセット内で実行されている場合は、この変数によって、このプロセスのためにランダムに配置されたメモリーがシステム内のすべての lgroup から選択されるのか、またはプロセッサセット内のプロセッサで構成されている lgroup のみから選択されるのかが決定されます。
-----	---

	プロセッサセットの作成についての詳細は、 <code>psrset(1M)</code> を参照してください。
データ型	ブール型
デフォルト	0。Solaris OS はシステム内のすべての <code>lggroup</code> からメモリーを選択します
範囲	■ 0。Solaris OS はシステム内のすべての <code>lggroup</code> 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 <code>lggroup</code> 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 <code>lggroup</code> 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
検査	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	コミットされていない

Solaris ボリュームマネージャーのパラメータ

`md_mirror:md_resync_bufsz`

備考欄	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
検査	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。システム性能全体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性能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、十分な性能となるまで、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

備考欄	複製定数に関する Solaris ボリュームマネージャー要件を変更し、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
検査	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。

- 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
- 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監視してください。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。

状態データベースの複製については、[『Solaris ボリュームマネージャの管理』の第6章「状態データベース \(概要\)」](#)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 チューニング可能パラメータ

この章では、NFS のチューニング可能パラメータについて説明します。

- 107 ページの「NFS 環境のチューニング」
- 108 ページの「NFS モジュールのパラメータ」
- 142 ページの「`nfssrv` モジュールのパラメータ」
- 146 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
Solaris カーネルのチューニング可能パラメータ	第2章「Solaris カーネル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4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」
ネットワークキャッシュとアクセラレータ (NCA) のチューニング可能パラメータ	第5章「ネットワークキャッシュとアクセラレータ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、27 ページの「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マニュアルをチェックしてください。

NFS モジュールのパラメータ

ここでは、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

備考欄	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
検査	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

備考欄	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
検査	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

備考欄	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。 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。 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。 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。時刻フィールドがフルの 32 ビットの時刻を表しているのか、時刻フィールドが負の時刻、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。
-----	--

	32 ビットから 64 ビットに変換するときに、時刻に符号を付けるかどうかを決定することはできません。時刻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、時刻値がフルの 32 ビット時刻値を本当に表している場合は、時刻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 ビット時刻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
検査	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

備考欄	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

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

nfs:nfs3_cots_timeo

備考欄	トランスポートプロトコルとして TCP などの接続型トランスポートを使用している、NFS バージョン 3 でマウントされたファイルシステムのデフォルトの RPC タイムアウトを制御します。
データ型	符号付き整数 (32 ビット)
デフォルト	600 (60 秒)
範囲	0 から $2^{31} - 1$
単位	1/10 秒
動的か	はい。ただし、ファイルシステムの RPC タイムアウトは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	なし 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。
どのような場合に変更するか	

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

備考欄

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の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。

検査

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と、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

備考欄

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

プール値

動的か

はい

検査

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

備考欄

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
検査	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

備考欄	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
検査	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

備考欄

UDP などのコネクションレストランスポートを使用する、NFS バージョン 2 でマウントされたファイルシステムに対して、「動的再転送」と呼ばれる機能を有効にするかどうかを制御します。この機能は、サーバーの応答時間を監視し、RPC タイムアウトと読み取り / 書き込みの転送サイズを調整することによって、再転送の回数を減らそうとするものです。

データ型

整数 (32 ビット)

デフォルト

1 (有効)

範囲

0 (無効)、1 (有効)

動的か

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

検査

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

備考欄

UDP などのコネクションレストランスポートを使用する、NFS バージョン 3 でマウントされたファイルシステムに対して、「動的再転送」と呼ばれる機能を有効にするかどうかを制御します。この機能は、サーバーの応答時間を監視し、RPC タイムアウトと読み取り / 書き込みの転送サイズを調整することによって、再転送の回数を減らそうとするものです。

データ型	整数 (32 ビット)
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
単位	ブール値
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	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

備考欄

NFS バージョン 2 でマウントされたファイルシステムに対して、ネガティブ名前キャッシュを使用するかどうかを制御します。このネガティブ名前キャッシュを使用すると、ルックアップされたファイル名が存在しない場合には、そのファイル名が記録されます。このキャッシュは、存在しないことがすでにわかっているファイル名のルックアップ要求が、ネットワークを介して行われないようにする目的で使用されます。

データ型	整数 (32 ビット)
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
単位	ブール値
動的か	はい
検査	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。整合性機構は、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、この整合性機構が通常の属性キャッシュ機構になります。

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、[129 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`

備考欄

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、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

プール値

動的か

はい

検査

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。整合性機構は、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、この整合性機構が通常の属性キャッシュ機構になります。

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、[129 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:nfs4_lookup_neg_cache`

備考欄

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

プール値

動的か

はい

検査

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。整合性機構は、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、この整合性機構が通常の属性キャッシュ機構になります。

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、[129 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`

備考欄

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

整数 (16 ビット)

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。
検査	なし
どのような場合に変更するか	ある時点で存在する同時入出力操作の数を増やしたり、減らしたりする場合。たとえば、帯域幅が非常に狭いネットワークでは、NFS クライアントによるネットワークの過負荷を防止するためにこの値を減らすことがあるかもしれません。また、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント / サーバリソースをいっそう有効に活用できるようになります。
コミットレベル	変更の可能性あり

nfs:nfs3_max_threads

備考欄	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	整数 (16 ビット)
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。
検査	なし
どのような場合に変更するか	ある時点で存在する同時入出力操作の数を増やしたり、減らしたりする場合。たとえば、帯域幅が非常に狭いネットワークでは、NFS クライアントによるネットワークの過負荷を防止するためにこの値を減らすことがあるかもしれません。また、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント / サーバリソースをいっそう有効に活用できるようになります。
コミットレベル	変更の可能性あり

nfs:nfs4_max_threads

備考欄	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	整数 (16 ビット)
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。
検査	なし
どのような場合に変更するか	ある時点で存在する同時入出力操作の数を増やしたり、減らしたりする場合。たとえば、帯域幅が非常に狭いネットワークでは、NFS クライアントによるネットワークの過負荷を防止するためにこの値を減らすことがあるかもしれません。また、広帯域幅のネットワークにおいて、クライアントとサーバーに十分なリソースがある場合は、この値を増やすことができるかもしれません。そうすることによって、使用可能なネットワーク帯域幅とクライアント / サーバリソースをいっそう有効に活用できるようになります。
コミットレベル	変更の可能性あり

nfs:nfs_nra

備考欄	ファイルへの順次アクセスが検出されたときに NFS バージョン 2 クライアントによってキューに入れられる先読み操作の数を制御します。これらの先読み操作では、並行性が高まり、読み取りのスループットが向上します。個々の先読み要求は、通常、ファイルデータの 1 論理ブロックに対するものです。
データ型	整数 (32 ビット)
デフォルト	4
範囲	0 から $2^{31} - 1$
単位	論理ブロック (130 ページの「nfs:nfs_bsize」を参照)
動的か	はい
検査	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

備考欄

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 (131 ページの「nfs:nfs3_bsize」を参照)

動的か

はい

検査

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、 210 ページ の「 <code>nfs:nfs3_nra (Solaris 10)</code> 」を参照してください。

`nfs:nfs4_nra`

備考欄	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(132 ページ の「 <code>nfs:nfs4_bsize</code> 」を参照)
動的か	はい
検査	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

備考欄

NFS クライアントの `rnode` キャッシュのサイズを制御します。

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

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

データ型

整数 (32 ビット)

デフォルト

このパラメータのデフォルト値は 0 です。これは `nrnode` の値が `ncsize` パラメータの値に設定されるべきであることを示しています。実際、`nrnode` の値が正でないと、`nrnode` には `ncsize` が設定されません。

範囲

1 から $2^{31} - 1$

単位

`rnode`

動的か

いいえ。この値は、`/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。

検査

`rnode` 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

`rnode` 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、`nrnode` 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、`nrnode` 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、`nrnode` を小さい値に設定すると、システムメモリーでは `rnode` 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、`nrnode` の値を増やして、ファイルメタデータを格納できるように最適化すると、メタデータを要求するネットワーク呼び出しの数を減らすことができます。

推奨はできませんが、`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、[210 ページの「`nfs:nrnode \(Solaris 9 8/03\)`」](#)を参照してください。

`nfs:nfs_shrinkreaddir`

備考欄

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
検査	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

備考欄	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
検査	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

備考欄	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
検査	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
変更履歴	211 ページの「nfs:nfs_write_error_interval (Solaris 9 8/03)」 を参照してください。

nfs:nfs_write_error_to_cons_only

備考欄	NFS 書き込みエラーをシステムコンソールと <code>syslog</code> に記録するか、それともシステムコンソールだけに記録するかを制御します。このパラメータは、NFS バージョン2、3、および4のクライアントのメッセージに影響を与えます。
データ型	整数 (32 ビット)
デフォルト	0 (システムコンソールと <code>syslog</code>)
範囲	0 (システムコンソールと <code>syslog</code>) または 1 (システムコンソール)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	<code>syslogd</code> デーモンによってロギングされるメッセージを含むファイルシステムがいっぱいになるのを防ぐには、このパラメータの値を調べます。このパラメータを有効にすると、メッセージはシステムコンソールに出力されるだけで、 <code>syslog</code> メッセージファイルにはコピーされません。
コミットレベル	変更の可能性あり
変更履歴	211 ページの「nfs:nfs_write_error_to_cons_only (Solaris 9 8/03)」 を参照してください。

nfs:nfs_disable_rddir_cache

備考欄	<code>REaddir</code> 要求と <code>REaddirplus</code> 要求に対する応答を格納するために、キャッシュを使用するかどうかを制御します。このキャッシュを使用すると、ディレクトリ情報を取得するためにサーバーを繰り返し呼び出すことがなくなります。
データ型	整数 (32 ビット)
デフォルト	0 (キャッシングを有効にする)
範囲	0 (キャッシングを有効にする) または 1 (キャッシングを無効にする)
単位	ブール値

動的か	はい
検査	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刻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必要があります。 ■ 116 ページの「<code>nfs:nfs_lookup_neg_cache</code>」 ■ 117 ページの「<code>nfs:nfs3_lookup_neg_cache</code>」 ■ 118 ページの「<code>nfs:nfs4_lookup_neg_cache</code>」
コミットレベル	変更の可能性あり
変更履歴	211 ページ の「 <code>nfs:nfs_disable_rddir_cache</code> (Solaris 98/03)」を参照してください。

nfs:nfs_bsize

備考欄	NFS バージョン 2 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	8192 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。
検査	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 <code>PAGESIZE</code> より小さい値を設定しないでください。さらに、このパラメータの値が大きすぎると、メモリ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 コミットレベル	このパラメータは変更しないでください。 変更の可能性あり

`nfs:nfs3_bsize`

備考欄	NFS バージョン 3 のクライアントが使用する論理ブロックサイズを制御します。このブロックサイズは、クライアントが入出力を行うときにサーバーに対して読み取りや書き込みを行うデータ量を表します。
データ型	符号なし整数 (32 ビット)
デフォルト	32,768 (32K バイト)
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、ファイルシステムのブロックサイズは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	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 <code>PAGESIZE</code> より小さい値を設定しないでください。さらに、このパラメータの値が大きすぎると、メモリ割り当てが許可されるまで待つ間に、システムがハングすることがあります。

どのような場合に変更するか	データ転送サイズの最大値を変更したい場合は、このパラメータの値を調べてください。このパラメータは、 <code>nfs:nfs3_max_transfer_size</code> パラメータと連携して変更してください。転送サイズを増やしたい場合は両方のパラメータを増やします。転送サイズを減らしたい場合は、通常、このパラメータを減らすだけで十分です。
コミットレベル	変更の可能性あり

`nfs:nfs4_bsize`

備考欄

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。
検査	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 <code>PAGESIZE</code> より小さい値を設定しないでください。さらに、このパラメータの値が大きすぎると、メモリー割り当てが許可されるまで待つ間に、システムがハングすることがあります。
どのような場合に変更するか	データ転送サイズの最大値を変更したい場合は、このパラメータの値を調べてください。このパラメータは、 <code>nfs: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

備考欄

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。

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。

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。

検査

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

備考欄	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

検査	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

備考欄	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。
検査	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

備考欄	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
検査	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

備考欄

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す。

データ型	整数 (32 ビット)
デフォルト	このパラメータのデフォルト値は0です。この値は <code>nacache</code> の値に <code>nrnode</code> パラメータの値が設定されるべきであることを示しています。
範囲	1 から $2^{31} - 1$
単位	アクセスキャッシュエントリ
動的か	いいえ。この値は、 <code>/etc/system</code> ファイルにパラメータを追加するか、パラメータを変更し、その後システムをリブートすることによってのみ変更できます。

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

nfs:nfs3_jukebox_delay

備考欄	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、階層型記憶装置、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
検査	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

備考欄	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。
検査	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 は、データグラム当たり 64K バイトの「強い」制限があります。この 64K バイトには、要求のデータ部分の他に RPC ヘッダーやその他の NFS 情報を含まなければなりません。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、その値を反映させて `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、[131 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

変更履歴

[211 ページの「nfs:nfs3_max_transfer_size \(Solaris 9/03\)」](#) を参照してください。

`nfs:nfs4_max_transfer_size`

備考欄

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 (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、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。

検査

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、 139 ページ の「 <code>nfs:nfs3_max_transfer_size</code> 」を参照してください。
どのような場合に変更するか	ネットワークを介して送信するデータのサイズをチューニングする場合。通常、 <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。
	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、その値を反映させて <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、 132 ページ の「 <code>nfs:nfs4_bsize</code> 」を参照してください。
コミットレベル	転送サイズをデフォルト値より小さくする場合は、 <code>mount</code> コマンドの <code>-wsize</code> または <code>-rsize</code> オプションをファイルシステム単位で使用します。 変更の可能性あり

`nfs:nfs3_max_transfer_size_clts`

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

備考欄	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	1048576 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。
検査	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

nfssrv モジュールのパラメータ

この節では、nfssrv モジュールの NFS パラメータについて説明します。

nfssrv:nfs_portmon

備考欄

NFS サーバーがクライアント側の整合性を確保するために行おうとする一部のセキュリティチェックを制御します。NFS では、要求を送信したソースポートが「予約ポート」だったかどうかをチェックできます。予約ポートには 1024 未満の番号が与えられます。BSD ベースのシステムでは、これらのポートは root が実行するプロセス用に予約されています。このセキュリティチェックでは、ユーザーが独自の RPC ベースのアプリケーションを作成して、NFS クライアントが使用するアクセスチェックを破ることを防止できます。

データ型

整数 (32 ビット)

デフォルト

0 (セキュリティチェックを無効にする)

範囲

0 (セキュリティチェックを無効にする) または 1 (セキュリティチェックを有効にする)

単位

プール値

動的か

はい

検査

なし

どのような場合に変更するか

悪意のあるユーザーが、普通ならアクセス権のない NFS サーバーを使用してファイルにアクセスするのを防ごうとする場合、このパラメータを使用します。しかし、「予約ポート」は広範にサポートされている概念ではありません。したがって、このチェックにおけるセキュリティの側面は非常に弱いものです。また、すべての NFS クライアントが予約の範囲内のポート番号にトランスポート終端を結びつけるわけでもありません。したがって、セキュリティチェックを有効にすると、相互運用性に関する問題が起きることがあります。

コミットレベル

変更の可能性あり

nfssrv:rfs_write_async

備考欄

NFS バージョン 2 サーバーが WRITE 要求をどのように処理するかを制御します。NFS バージョン 2 プロトコルでは、WRITE 要求に関連するすべての変更済

みデータとメタデータが安定したストレージに格納されていないと、サーバーはクライアントに応答できません。NFS バージョン 2 の WRITE 要求は、データは 8192 バイトに制限されます。したがって、各 WRITE 要求によって、複数の小さい書き込みがストレージサブシステムに対して行われることがあります。これは、パフォーマンス低下の原因になります。

NFS バージョン 2 の WRITE 要求を高速化する方法の 1 つは、クライアントの動作を活用することです。クライアントは、複数の WRITE 要求をバッチで(一括して)送信する傾向があります。サーバーでは、この動作を利用して複数の WRITE 要求を 1 つの要求にクラスタ化し、ファイルシステムに出すことができます。こうして、要求の数を少なくし、要求のサイズを大きくして、データをストレージサブシステムに書き込むことができます。この方式によって、WRITE 要求のスループットが大幅に向上します。

データ型	整数 (32 ビット)
デフォルト	1 (クラスタ化を有効にする)
範囲	0 (クラスタ化を無効にする) または 1 (クラスタ化を有効にする)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	特に PC クライアントなど、非常に小さい NFS クライアントのなかには、複数の WRITE 要求をバッチ化しないものがあります。したがって、クライアントが要求した動作が存在しないということもあります。また、NFS バージョン 2 サーバーにおいて、クラスタ化はオーバーヘッドの増加をもたらすだけで、パフォーマンスがむしろ下がるという場合もあります。
コミットレベル	変更の可能性あり

nfssrv:nfsauth_ch_cache_max

備考欄	NFS 認証サーバーに接続するクライアントハンドルのキャッシュサイズを制御します。サーバーは、NFS クライアントを認証して、クライアントが使用しようとしているファイルハンドルへのアクセスが許可できるかどうかを判定します。
データ型	整数 (32 ビット)
デフォルト	16
範囲	0 から $2^{31} - 1$
単位	クライアントハンドル
動的か	はい
検査	なし
どのような場合に変更するか	このキャッシュは動的でないため、すべてのクライアントハンドルが使用されていると、クライアントハンドルの割り当て要求は失敗します。その結果、NFS サーバーは、この要求を認証できなかったという理由で要求を廃棄します。しかし、ほとんどの場合、NFS クライアントはタイムアウトになり、要求を再転送するため、これは問題になりません。しかし、クライアントにソフトマウントされたファイルシステムの場合、クライアントは、タイムアウトになっても要求を再試行せず、アプリケーションにエラーを返すことがあります。サーバーのキャッシュを負荷に対応できるだけの大きさにしておけば、このような状況を防止できることがあります。
コミットレベル	変更の可能性あり

nfssrv:exi_cache_time

備考欄	システムのメモリー要求によりエントリが破棄されるまでに、NFS 認証キャッシュに保持される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	3600 秒 (1 時間)

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
検査	なし
どのような場合に変更するか	NFS 認証キャッシュのサイズは、エントリがキャッシュから消去されるまでの最小限の時間を変更することによって調整することができます。キャッシュのサイズは、大きくなりすぎないように制御すべきです。そうすれば、この時間のプロセスによって開放されなかったシステムリソースを使用できます。
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

この節では、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

備考欄	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
検査	なし
どのような場合に変更するか	一般には、1 つの接続だけでネットワーク帯域幅全体を使いこなすことができます。しかし、ネットワークが提供する帯域幅を TCP が 1 つのストリームだ

けで利用できない場合は、複数の接続を使えば、クライアントとサーバー間のスループットが向上することがあります。

接続数の増加にはそれなりの影響があります。接続数が増えると、各接続を維持するために必要なカーネルリソースの使用量も増えます。

コミットレベル

変更の可能性あり

rpcmod:clnt_idle_timeout

備考欄

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

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

検査

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

備考欄

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
検査	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

備考欄	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。
検査	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 `pagesize` 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:svc_default_max_same_xprt

備考欄

各トランスポート終端の要求を最大でいくつ処理したら、次のトランスポート終端に進むかを制御します。カーネル RPC では、サービススレッドのプールとトランスポート終端のプールが使用されます。個々のサービススレッドは、どのトランスポート終端からの要求でも処理できます。ただし、パフォーマンス上の理由により、次のトランスポート終端に進む前に各トランスポート終端の複数の要求が処理されます。このアプローチにより、不足を避け、パフォーマンス上の利点を得ることができます。

データ型

整数 (32 ビット)

デフォルト

8

範囲

0 から $2^{31} - 1$

単位

要求

動的か

はい。ただし、トランスポート終端を切り替える前に要求を最大でいくつ処理するかは、トランスポート終端がカーネル RPC サブシステムに構成される時に設定されます。このパラメータへの変更は、新しいトランスポート終端だけに適用されます。つまり、既存のトランスポート終端には無効です。

検査

なし

どのような場合に変更するか

サービスが、NFS バージョン 2 の `WRITE` 要求を高速化するクラスタ化などのクライアントの動作を利用できるようにこのパラメータをチューニングすることができます。このパラメータの値を増やすことにより、サーバー側でクライアントの動作の利点をいっそう活用できる可能性があります。

コミットレベル

変更の可能性あり

rpcmod:maxdupreqs

備考欄

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す。キャッシュのサイズを著しく大きくすると、キャッシュ内のエントリの検索に長い時間がかかることがあります。

このパラメータの値を 0 に設定しないでください。0 に設定すると、NFS サーバーが非べき等の要求を処理できなくなります。

検査

なし

どのような場合に変更するか

NFS クライアントで不正な障害エラーが検出された場合は、このパラメータの値を調べます。たとえば、ディレクトリの作成が失敗したのに、実際にはディレクトリが作成されている場合は、再転送された MKDIR 要求をサーバーが検出しなかった可能性があります。

キャッシュのサイズは、サーバーの負荷に見合ったものでなければなりません。キャッシュには非べき等の要求が格納されるため、キャッシュでは、要求全体の一部だけしか管理する必要がありません。キャッシュは、クライアントによる再転送を検出できるだけの間、情報を保持していなければなりません。

	ん。一般に、コネクションレストランスポートのクライアントのタイムアウトは比較的短く、1 秒から 20 秒くらいです。
コミットレベル	変更の可能性あり

rpcmod:cotsmaxdupreqs

備考欄	接続型トランスポートにおける RPC レベルの再転送を検出する、重複要求キャッシュのサイズを制御します。このキャッシュは、クライアントネットワークアドレス、RPC の手順番号、プログラム番号、バージョン番号、および、トランザクション ID でインデックス化されています。このキャッシュにより、非べき等であるかもしれない再転送要求の処理が防止されます。
データ型	整数 (32 ビット)
デフォルト	1024
範囲	1 から $2^{31} - 1$
単位	要求
動的か	はい
検査	キャッシュのサイズは動的に決められますが、キャッシュへの高速アクセスを可能にするハッシュキューのサイズは静的に決められます。キャッシュのサイズを著しく大きくすると、キャッシュ内のエントリの検索に長い時間がかかることがあります。 このパラメータの値を 0 に設定しないでください。0 に設定すると、NFS サーバーが非べき等の要求を処理できなくなります。
どのような場合に変更するか	NFS クライアントで不正な障害エラーが検出された場合は、このパラメータの値を調べます。たとえば、ディレクトリの作成が失敗したのに、実際にはディレクトリが作成されている場合は、再転送された MKDIR 要求をサーバーが検出しなかった可能性があります。 キャッシュのサイズは、サーバーの負荷に見合ったものでなければなりません。キャッシュには非べき

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

コミットレベル

変更の可能性あり

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。

- 154 ページの「IP チューニング可能パラメータ」
- 161 ページの「TCP チューニング可能パラメータ」
- 179 ページの「UDP チューニング可能パラメータ」
- 180 ページの「IPQoS チューニング可能パラメータ」
- 181 ページの「SCTP チューニング可能パラメータ」
- 190 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
Solaris カーネルのチューニング可能パラメータ	第2章「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3章「NFS チューニング可能パラメータ」
ネットワークキャッシュとアクセラレータ (NCA) のチューニング可能パラメータ	第5章「ネットワークキャッシュとアクセラレータの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、24 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。

- 173 ページの「`ipcl_conn_hash_size`」

- 174 ページの「[ip_squeue_worker_wait](#)」

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

たとえば、`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方法については、『[Solaris のシステム管理 \(基本編\)](#)』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査

この節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 文書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`

備考欄

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

備考欄	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

備考欄	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

備考欄

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、[211 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

備考欄

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。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

備考欄	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

備考欄	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。この機能が無効にすると、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。

173 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、212 ページの「 ip_multidata_outbound (Solaris 10 リリース)」を参照してください。

ip_queue_fanout

備考欄	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。接続を展開するために使用される queue の数は、159 ページの「ip_soft_rings_cnt」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、 214 ページの「ip_squeue_fanout (Solaris 11/06 リリース)」 を参照してください。

ip_soft_rings_cnt

備考欄	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。
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 dls_soft_ring_pkt_drop が増分されます。	
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	旧式
変更履歴	詳細は、 214 ページの「ip_soft_rings_cnt (Solaris 11/06 リリース)」 を参照してください。

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

備考欄	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

備考欄	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
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

備考欄	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。 ■ 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

備考欄	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	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。

- 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_deferred_acks_max

備考欄

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

備考欄

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

	ACK は使用されなくなります。実際の数値は、接続ごとに動的に計算されます。この値はデフォルトの最大値です。
デフォルト	8
範囲	0 から 16
動的か	はい
どのような場合に変更するか	この値は変更しないでください。ただし、遅延 ACK の影響でネットワークトラフィックが著しく混雑するような状況では、この値を減らします。ただし、2 より小さくしないでください。
コミットレベル	変更の可能性あり

tcp_wscale_always

備考欄	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 セグメントに応答します。オプションの値は受信ウィンドウサイズに従って設定されます。
	ウィンドスケールオプションについては、RFC 1323 を参照してください。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に変更するか	ウィンドスケールオプションをサポートしていない古い TCP スタックとの相互運用性の問題がある場合は、このパラメータを無効にしてください。
コミットレベル	変更の可能性あり
変更履歴	詳細は、 215 ページ の「 tcp_wscale_always (Solaris 9 リリース) 」を参照してください。

tcp_tstamp_always

備考欄	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 セグメントに応答します。
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

備考欄	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 190 ページの「ルート別のメトリック」 を参照してください。 165 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

備考欄

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[190 ページ](#)の「[ルート別のメトリック](#)」を参照してください。[165 ページ](#)の「[tcp_max_buf](#)」と [178 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 `setsockopt(3XNET)` `SO_RCVBUF` 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

備考欄

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、`setsockopt(3XNET)` を使用するアプリケーションによって設定される送信バッファサイズと受信バッファサイズを制御します。

デフォルト

1,048,576

範囲

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

備考欄

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

備考欄	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

備考欄	輻輳ウィンドウが 1 再送タイムアウト (RTO) の間アイドルにされた (セグメントをまったく受信しなかった) あとの輻輳ウィンドウのサイズを TCP 接続の MSS 単位で指定します。
-----	--

	輻輳ウィンドウの初期サイズがどのように計算されるかについては、RFC 2414 を参照してください。
デフォルト	4
範囲	1 から 16,384
動的か	はい
どのような場合に変更するか	詳細は、166 ページの「 tcp_slow_start_initial 」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

備考欄	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す。
コミットレベル	変更の可能性あり

tcp_rev_src_routes

備考欄	0 が設定されていると、TCP は、セキュリティー上の理由により、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

備考欄	TCP 接続を TIME-WAIT 状態に保つ時間をミリ秒で指定します。 RFC 1122 の 4.2.2.13 を参照してください。
デフォルト	600,000 (60 秒)
範囲	1 秒から 10 分
動的か	はい
どのような場合に変更するか	この値は 60 秒より小さくしないでください。 このパラメータの変更方法については、RFC 1122 の 4.2.2.13 を参照してください。
コミットレベル	変更の可能性あり

tcp_ecn_permitted

備考欄	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 機構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機構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機構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機構をサポートしていることを接続先に通知します。

このパラメータを2に設定すると、TCPは接続を受け付けた時点でECN機構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機構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、この機構により、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

備考欄

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。[170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 4,294,967,296
動的か	はい
どのような場合に変更するか	複数の接続要求を受けることのある Web サーバーのようなアプリケーションでは、着信頻度に応じてこのデフォルト値を増やすことができます。 このパラメータに著しく大きい値を設定しないでください。保留状態の TCP 接続はメモリーを過剰に使用することがあります。さらに、保留状態の TCP 接続の数が多すぎて、アプリケーションが接続要求を適時に処理できない場合は、新しい着信要求が拒否されることがあります。 tcp_conn_req_max_q を増やしても、アプリケーションでそれだけの数の保留状態の TCP 接続を持てるとは限りません。アプリケーションでは、listen(3SOCKET) を使用して、保留状態の TCP 接続の最大数をソケットごとに変更できます。このパラメータは、アプリケーションが listen() を使用して設定できる最大値を表します。つまり、このパラメータに非常に大きな値を設定しても、あるソケットに対する実際の最大数は、listen() に指定された値によっては tcp_conn_req_max_q よりもはるかに少ないことがあります。
コミットレベル	変更の可能性あり
変更履歴	217 ページの「xxx:ip_forwarding (Solaris 9 リリース)」 を参照してください。

tcp_conn_req_max_q0

備考欄	単一の TCP リスナーが持つことができる、不完全な (3 段階ハンドシェークがまだ終わっていない) 保留状態の TCP 接続のデフォルトの最大数を指定します。 TCP の 3 段階ハンドシェークについては、RFC 793 を参照してください。169 ページの「tcp_conn_req_max_q」も参照してください。
デフォルト	1024

範囲	0 から 4,294,967,296
動的か	はい
どのような場合に変更するか	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。 <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。 接続要求を受信すると、TCP はまず、受け付けられるのを待っている保留状態の TCP 接続 (3 段階ハンドシェークが終わっている) の数が、そのリスナーに対する最大数 (N) を超えていないかをチェックします。接続数が超えていると、その要求は拒否されます。超えていなければ、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。超えていなければ、その要求は受け付けられます。超えていると、最も古い不完全な保留状態の TCP 要求がドロップされます。
コミットレベル	変更の可能性あり
変更履歴	217 ページの「 xxx:ip_forwarding (Solaris 9 リリース) 」を参照してください。

tcp_conn_req_min

備考欄	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

備考欄

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

備考欄

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 4,294,967,296

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

備考欄	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。 157 ページ の「 <code>ip_multidata_outbound</code> 」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

/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

備考欄	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せん。ハッシュテーブルサイズを大きくすると、より多くのメモリーが消費され、ユーザーアプリケーション用として提供できるメモリーの量が減ります。
コミットレベル	変更の可能性あり
ip_queue_worker_wait	
備考欄	TCP/IP パケットを <i>queue</i> 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。 <i>queue</i> 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、 214 ページの「ip_queue_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

備考欄

この nnd パラメータは、システム全体で TCP 接続がアイドル状態になってから最初にプローブが送信されるまでの間隔を設定します。

Solaris は、RFC 1122 に記述されている TCP キープアライブ機構をサポートします。この機構は、TCP ソケットで `SO_KEEPALIVE` ソケットオプションを設定することで有効になります。

ソケットで `SO_KEEPALIVE` が有効な場合、TCP 接続が 2 時間 (`tcp_keepalive_interval` パラメータのデフォルト値) アイドル状態になると最初のキープアライブプローブが送信されます。ピアがプローブに 8 分間応答しない場合、TCP 接続が終了します。

また、個々のアプリケーションに `TCP_KEEPALIVE_THRESHOLD` ソケットオプションを使用してデフォルトの間隔を置き換えれば、各アプリケーションがソケットごとに独自の間隔を持つようにできます。オプションの値は、ミリ秒単位の符号なし整数です。tcp(7P) のマニュアルページも参照してください。

デフォルト

2 時間

範囲

10 秒から 10 日

単位

符号なし整数 (ミリ秒)

動的か

はい

どのような場合に変更するか

この値は変更しないでください。この値を小さくすると、不必要なネットワークトラフィックが生じる可能性があるとともに、ネットワークの一時的な問題のために未完了のまま接続が終了してしまう可能性も高くなります。

コミットレベル

変更の可能性あり

tcp_ip_abort_interval

備考欄

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 `tcp_ip_abort_interval` の間再転送を行

なっても、この間に相手側のエンドポイントから肯定応答をまったく受け取らないと、この接続は閉じられます。

TCP 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。176 ページの「`tcp_rexmit_interval_max`」も参照してください。

デフォルト	8 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、176 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`

備考欄	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190 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`

備考欄	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。175 ページの「 <code>tcp_ip_abort_interval</code> 」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	通常のネットワーク環境では、この値を変更しないでください。 単一の接続の往復時間 (RTT) が 10 秒程度になるような特別な状況では、この値を増やすことができます。この値を変更する場合は、合わせて <code>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	
備考欄	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いけません。176 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	通常のネットワーク環境では、この値を変更しないでください。 TCP の RTO 計算は、RTT のもっとも大きい変動に対処できます。単一の接続の往復時間 (RTT) が 10 秒程度になるような特別な状況では、この値を増やすことができます。この値を変更する場合は、合わせて <code>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	
備考欄	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	
備考欄	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、 <code>timestamp</code> 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	
備考欄	デフォルトの最小受信ウィンドウサイズを制御します。最小値は、 <code>tcp_recv_hiwat_minmss</code> に、接続の最大セグメントサイズ (MSS) を掛けた値です。
デフォルト	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

備考欄

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、[180 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 `setsockopt(3XNET) SO_SNDBUF` を使用してソケットごとに変更できます。一般には、デフォルト値を変更する必要はありません。

コミットレベル

変更の可能性あり

変更履歴

詳細は、[219 ページ](#)の「`udp_xmit_hiwat (Solaris 9 リリース)`」を参照してください。

udp_recv_hiwat

備考欄

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、[180 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

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 `setsockopt(3XNET) SO_RCVBUF` を使用してソケットごとに変更できます。一般には、デフォルト値を変更する必要はありません。

コミットレベル

変更の可能性あり

変更履歴

詳細は、219 ページの「[udp_rcv_hiwat \(Solaris 9 リリース\)](#)」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	
備考欄	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
変更履歴	詳細は、219 ページの「 udp_max_buf (Solaris 9 リリース) 」を参照してください。

IPQoS チューニング可能パラメータ

ip_policy_mask

備考欄

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。転送送信、転送受信、ローカル送信、またはローカル受信。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	転送送信	転送受信	ローカル送信	ローカル受信
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、すべてのローカル受信パケットの 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

備考欄	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、この値を上書きできます。
デフォルト	8
範囲	0 から 128
動的か	はい
どのような場合に変更するか	INIT 再送回数は 181 ページの「 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

備考欄	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。 <code>sctp_pa_max_retr</code> は、使用可能なすべてのパスの182 ページの「 <code>sctp_pp_max_retr</code> 」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、 <code>sctp_pa_max_retr</code> 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`

備考欄	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`

備考欄	SCTP 接続の輻輳ウィンドウの最大値を制御します。
デフォルト	1,048,576
範囲	128 から 1,073,741,824
動的か	はい
どのような場合に変更するか	アプリケーションが <code>setsockopt(3XNET)</code> を使用してウィンドウサイズに <code>sctp_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>sctp_cwnd_max</code> を超えることはありません。したがって、186 ページの「 <code>sctp_max_buf</code> 」は <code>sctp_cwnd_max</code> より大きくする必要があります。
コミットレベル	変更の可能性あり

sctp_ipv4_ttl

備考欄	SCTP 接続上で、発信 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 範囲	64 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

備考欄	HEARTBEAT チャンクからハードビートに対応したアイドル状態の宛先までの間隔を計算します。 SCTP 終端は、相手側のアイドル状態の宛先転送アドレスまでの到達性を監視するため、定期的に HEARTBEAT チャンクを送信します。
デフォルト 範囲	30 秒 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 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

備考欄	新しいシークレットを生成するタイミングを判定します。生成されたシークレットから、Cookie の MAC を計算できます。
デフォルト 範囲	2 分 0 から 1,440 分
動的か	はい
どのような場合に変更するか	RFC 2960 の 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

備考欄	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

備考欄	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 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

備考欄	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	パスにおける MTU の変更を無視する場合、このパラメータを有効にします。パス MTU が減った場合、このパラメータを有効にすると、IP 分割が行われます。

コミットレベル	変更の可能性あり
---------	----------

sctp_initial_ssthresh

備考欄	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 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

備考欄	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。186 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) SO_SNDBUF 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

備考欄	送信ウィンドウサイズの下限値を制御します。
デフォルト	8,192
範囲	8,192 から 1,073,741,824
動的か	はい
どのような場合に変更するか	通常、この値を変更する必要はありません。このパラメータは、ソケットの送信バッファを書き込み可能にするために必要な最小限のサイズを設定しま

	す。必要に応じて、 185 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

備考欄	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。 186 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

備考欄	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_ip6_hoplimit

備考欄	SCTP 接続上で、発信 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	60
範囲	0 から 255
動的か	はい
どのような場合に変更するか	通常、この値を変更する必要はありません。宛先パスが 60 ホップを超過するような場合は、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

備考欄	相手側のすべての宛先アドレスの再送タイムアウト (RTO) の下限値をミリ秒で設定します。
デフォルト	1,000
範囲	500 から 60,000
動的か	はい
どのような場合に変更するか	RFC 2960 の 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

備考欄	相手側のすべての宛先アドレスの再送タイムアウト (RTO) の上限値をミリ秒で設定します。
デフォルト	60,000
範囲	1,000 から 60,000,000
動的か	はい
どのような場合に変更するか	RFC 2960 の 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

備考欄	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 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

備考欄	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、 187 ページ の「sctp_rto_max」に合わせて変更できます。
コミットレベル	変更の可能性あり

sctp_max_in_streams

備考欄	SCTP 接続 1 個あたりに許可された最大着信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 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

備考欄	SCTP 接続 1 個あたりに許可された最大発信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 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_shutack_wait_bound

備考欄	SHUTDOWN チャンクの送信後、SHUTDOWN ACK を待機する最大待ち時間をミリ秒で制御します。
デフォルト	60,000
範囲	0 から 300,000
動的か	はい
どのような場合に変更するか	通常、この値を変更する必要はありません。このパラメータは、 187 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_maxburst

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

sctp_addip_enabled

備考欄	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

備考欄	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

ルート別のメトリック

Solaris 8 リリースから、ルート別のメトリック (測定基準) 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 24,576 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 -recvpipe x
```

この結果、ge0 リンクの 192.123.124.0 ネットワークへのすべての接続には、デフォルトの受信ウィンドウサイズ 24567 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。次に例を示します。

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

接頭辞ルートのゲートウェイがデフォルトのルーターであることに留意してください。そのネットワークへのすべての接続は受信バッファサイズ y を使用します。複数のインタフェースがある場合は、-ifp 引数を使用して、使用するインタフェースを指定します。それによって、特定の宛先に対してどのインタフェースを使用するかを制御できます。メトリックを確認するには route(1M) get コマンドを使用します。

ネットワークキャッシュとアクセラレータのチューニング可能パラメータ

この章では、ネットワークキャッシュとアクセラレータ (NCA) のチューニング可能パラメータの一部を説明します。

- 194 ページの「`nca:nca_conn_hash_size`」
- 194 ページの「`nca:nca_conn_req_max_q`」
- 195 ページの「`nca:nca_conn_req_max_q0`」
- 195 ページの「`nca:nca_ppmax`」
- 196 ページの「`nca:nca_vpmax`」
- 197 ページの「`sq_max_size`」
- 198 ページの「`ge:ge_intr_mode`」

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
Solaris カーネルのチューニング可能パラメータ	第 2 章「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 3 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 4 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」

NCA パラメータのチューニング

これらのパラメータの設定は、Web サーバー専用のシステムに適しています。これらのパラメータの設定によりキャッシュページに割り当てるメモリーを増やすことができます。この章で説明するチューニングパラメータはすべて、`/etc/system` ファイル内で設定します。

`/etc/system` ファイルにチューニング可能パラメータを追加する方法については、[30 ページの「Solaris カーネルのチューニング」](#)を参照してください。

nca:nca_conn_hash_size

備考欄	もっとも近い素数に調整された、すべての TCP 接続の NCA モジュール内のハッシュテーブルサイズを制御します。
デフォルト	383 ハッシュテーブルエントリ
範囲	0 から 201,326,557
動的か	いいえ
どのような場合に変更するか	NCA の TCP ハッシュテーブルサイズが小さすぎて、着信 TCP 接続を追跡できない場合。この場合、多くの TCP 接続が同じハッシュテーブルエントリとしてグループ化されます。このような状況は、NCA が多くの TCP 接続を受けて、システム性能が低下しているときに発生します。
コミットレベル	変更の可能性あり

nca:nca_conn_req_max_q

備考欄	NCA が待機する保留中の TCP 接続の最大数を指定します。
デフォルト	256 接続
範囲	0 から 4,294,967,295
動的か	いいえ
どのような場合に変更するか	過剰な数の TCP 接続が確立しているために、NCA が新たに確立した接続を直後に閉じる場合。NCA が多くの TCP 接続を受けていて、これ以上の負荷を処理することはできても、これ以上の接続を受け

付けることができない場合は、このパラメータの値を大きくします。その結果、NCA は、同時により多くの TCP 接続を処理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

nca:nca_conn_req_max_q0

備考欄

NCA が待機する保留中でまだ確立されていない (3 種類のハンドシェークが未完了) TCP 接続の最大数を指定します。

デフォルト

1024 接続

範囲

0 から 4,294,967,295

動的か

いいえ

どのような場合に変更するか

すでに多数の TCP 接続が保留となっているため、NCA が追加の接続を拒否する場合。NCA が多くの TCP 接続を受けていて、これ以上の負荷を処理することはできても、これ以上の接続を受け付けることができない場合は、このパラメータの値を大きくします。その結果、NCA は、同時により多くの TCP 接続を処理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

nca:nca_ppmax

備考欄

NCA がページをキャッシュするために使用できる最大物理メモリー容量 (ページ数) を指定します。この値は、合計メモリー容量の 75 パーセント以下に収める必要があります。

デフォルト

物理メモリーの 25 パーセント

範囲

物理メモリーの 1 パーセントから最大量まで

動的か

いいえ

どのような場合に変更するか

メモリー容量が 512M バイト以上システム上の NCA を使用する場合。システムの未使用物理メモリー容量が多い場合は、このパラメータの値を大きくします。これにより、NCA は、このメモリーを使って

新しいオブジェクトを効果的にキャッシュに格納できます。その結果、システム性能が向上します。

仮想メモリーより物理メモリーが大きいシステム (4G バイトを超えるメモリーを持つ 32 ビットカーネル) の場合を除き、このパラメータの値は `nca_vpmax` に併せて大きくします。システムのページサイズを判定するには、`pagesize(1)` を使用します。

コミットレベル

変更の可能性あり

`nca:nca_vpmax`

備考欄

NCA がページをキャッシュするために使用できる最大仮想メモリー容量 (ページ数) を指定します。この値は、合計メモリー容量の 75 パーセント以下に収める必要があります。

デフォルト

仮想メモリーの 25 パーセント

範囲

仮想メモリーの 1 パーセントから最大量まで

動的か

いいえ

どのような場合に変更するか

メモリー容量が 512M バイト以上システム上の NCA を使用する場合。システムの未使用仮想メモリー容量が多い場合は、このパラメータの値を大きくします。これにより、NCA は、このメモリーを使って新しいオブジェクトを効果的にキャッシュに格納できます。その結果、システム性能が向上します。

このパラメータの値は `nca_ppmax` に合わせて大きくします。仮想メモリーよりも物理メモリーが大きなシステムの場合を除き、このパラメータは `nca_vpmax` とほぼ同じ値に設定します。

コミットレベル

変更の可能性あり

NCAのための一般的なシステムチューニング

NCA パラメータの設定以外にも、一般的なシステムのチューニングにより、NCA の性能を向上させることができます。GigabitEthernet (ge ドライバ) を使用する場合は、インタフェースを割り込みモードにすると性能が向上します。

たとえば、64 ビットカーネルでブートする 4G バイトのメモリーのあるシステムの場合、/etc/system ファイル内に次のパラメータを設定するとよいでしょう。システムのページサイズを判定するには、pagesize を使用します。

```
set sq_max_size=0
set ge:ge_intr_mode=1
set nca:nca_conn_hash_size=82500
set nca:nca_conn_req_max_q=100000
set nca:nca_conn_req_max_q0=100000
set nca:nca_ppmax=393216
set nca:nca_vpmax=393216
```

sq_max_size

備考欄

宛先 STREAMS キューが QFULL メッセージを生成する前に、syncq の深度 (メッセージ数) を設定します。

デフォルト

10000 メッセージ

範囲

0 (無制限) から MAXINT

動的か

いいえ

どのような場合に変更するか

メモリーの大きなシステム上で NCA を動作させている場合、このパラメータの値を増やしてドライバがキューに入れることのできるパケット数を増やします。サーバーの負荷が大きい場合、このパラメータの値を増やして、パケットがドロップされたりパケット処理が後回しにされることなくモジュールやドライバのデータ処理能力を向上させることができます。

Solaris 8 以降のリリース - 実稼動システムではこのパラメータを 0 に設定しないでください。このパラメータの変更が必要な場合は、この値を徐々に増やしてシステムを監視します。

Solaris 7 以前のリリース - 実稼動システムではこのパラメータを 0 に設定しないでください。このパラ

メータの変更が必要な場合は、この値を最大 100 ま
で徐々に増やしてシステムを監視します。

コミットレベル

変更の可能性あり

変更履歴

219 ページの「[sq_max_size \(Solaris 9 12/02 リリース\)](#)」を参照してください。

ge:ge_intr_mode

備考欄

ge ドライバが、パケットをキューに入れず、上位
通信層に直接送信できるようにします。

デフォルト

0 (パケットを上位層のキューに入れる)

範囲

0 (有効) または 1 (無効)

動的か

いいえ

どのような場合に変更するか

NCA を有効にしている場合は、このパラメータを 1
に設定し、パケットを割り込みモードで NCA へ配
信し、よりすばやく処理できるようにします。

コミットレベル

変更の可能性あり

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。

- 200 ページの「autofs」
- 200 ページの「cron」
- 200 ページの「devfsadm」
- 200 ページの「dhcpagent」
- 200 ページの「fs」
- 201 ページの「ftp」
- 201 ページの「inetinit」
- 201 ページの「init」
- 201 ページの「keyserv」
- 201 ページの「kbd」
- 201 ページの「login」
- 201 ページの「mpathd」
- 202 ページの「nfs」
- 202 ページの「nfslogd」
- 202 ページの「nss」
- 202 ページの「passwd」
- 202 ページの「power」
- 202 ページの「rpc.nisd」
- 202 ページの「su」
- 202 ページの「syslog」
- 203 ページの「sys-suspend」
- 203 ページの「tar」
- 203 ページの「utmpd」
- 203 ページの「ypasswdd」

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。

autofs

この機能を利用して、自動タイムアウト、状態メッセージの表示や記録、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) のマニュアルページを参照してください。

cron

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

devfsadm

現在、このファイルは使用されていません。

dhcpgent

DHCP のクライアント使用率は、dhcpgent デーモンによって提供されます。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) のマニュアルページの「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) のマニュアルページの「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)` のマニュアルページ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)` のマニュアルページの「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要求に応じて `init` が呼び出すシェルの環境に置かれます。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。

keyserv

詳細は、`keyserv(1M)` のマニュアルページの「FILES」セクションにある `/etc/default/keyserv` の情報を参照してください。

kbd

詳細は、`kbd(1)` のマニュアルページの「Extended Description」セクションを参照してください。

login

詳細は、`login(1)` のマニュアルページの「FILES」セクションにある `/etc/default/login` の情報を参照してください。

mpathd

この機能を利用して、`in.mpathd` 構成パラメータを設定できます。

詳細は、`in.mpathd(1M)` のマニュアルページを参照してください。

nfs

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) のマニュアルページの「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) のマニュアルページの「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) のマニュアルページの「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) のマニュアルページの「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) のマニュアルページの「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) のマニュアルページの「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)のマニュアルページの「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。

次に例を示します。

```
% 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。

utmpd

utmpd デーモンは /var/adm/utmpx (および Solaris の以前のバージョンでは /var/adm/utmp も) を監視し、pututxline(3C) で root 以外のプロセスによって挿入された utmp エントリがプロセスの終了前に確実に消去されるようにします。

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

- SCAN_PERIOD - utmpd が、監視するプロセスがまだ生きているかどうかの /proc のチェックとチェックの合間に休眠する秒数。デフォルト値は 300 です。
- MAX_FDS - utmpd が監視しようとするプロセスの最大数。デフォルト値は 4096 です。通常、この値を変更する必要はありません。

yppasswdd

この機能を利用して、ユーザーが passwd - rnis -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設定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

この付録には、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。この付録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。

- 205 ページの「カーネルパラメータ」
- 210 ページの「NFS チューニング可能パラメータ」
- 211 ページの「TCP/IP チューニング可能パラメータ」
- 219 ページの「ネットワークキャッシュとアクセラレータ (NCA) のチューニング可能パラメータ」
- 220 ページの「旧式または削除されたパラメータ」

カーネルパラメータ

プロセスのサイズのチューニング可能パラメータ

`max_nprocs` (**Solaris 9** リリース)

Solaris 10 の記述部分が更新され、「sun4m」が削除されました。

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、[70 ページ](#)の「`maxphys`」を参照してください。

maxphys (**Solaris 9** リリース)

Solaris 10 の maxphys のデフォルトの項から「126,976 (sun4m)」が削除されました。

rlim_fd_max (**Solaris 8** リリース)

Solaris 8 バージョンでは、デフォルトは1024 です。それ以降のSolaris リリースでは、デフォルトは65,536 です。

カーネルとメモリーの一般的なパラメータ

noexec_user_stack (**Solaris 9** リリース)

Solaris 10 の説明の項が更新され、「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 9** リリース)

Solaris 10 の説明の項が更新され、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。

noexec_user_stack (**Solaris 2.6、Solaris 7、Solaris 8** リリース)

Solaris 9 の説明の項が更新され、「およびsun4d」の語句が削除されました。

segkpsize (**Solaris 9 12/02** リリース)

旧 Solaris 9 リリースで、単位が8K バイトのページであるべきところで、誤ってM バイトの指定になっていました。さらに、範囲およびデフォルトの項で、Solaris 10 リリースに関して次の語句が削除されました。このパラメータは64 ビットカーネルが動作しているシステムに限って使用できるからです。

32 ビットカーネルでは512M バイト

fsflush とそれに関連するパラメータ

dopageflush (**Solaris 10** リリース)

Solaris 10 9/08 リリースでは、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、[63 ページ](#)の「[maxpgio](#)」を参照してください。

pages_pp_maximum (Solaris 9 リリース以前の Solaris リリース)

Solaris 8 バージョンで、デフォルトの記述が次のようになっています。

トリプレットの最大値(200, tune_t_minarmem + 100, [ブート時に使用可能なメモリーの10%])

一般的なファイルシステムパラメータ

ncsize (Solaris 9 およびそれ以前のリリース)

Solaris 9 およびそれ以前のリリースでは、「どのような場合に変更するか」の項で、`nfs:nrnode` パラメータが誤って `nfs:nfs_rnode` になっていました。

ncsize (Solaris 10 リリース)

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、[73 ページ](#)の「[ncsize](#)」を参照してください。

UFS チューニング可能パラメータ

bufhwm (Solaris 9 リリース)

このパラメータの情報は Solaris 10 リリースで大幅に変更されました。詳細は、[78 ページ](#)の「[bufhwm](#) および [bufhwm_pct](#)」を参照してください。

備考欄	入出力バッファをキャッシュするためのメモリーの最大量。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。バッファは、その量が <code>bufhwm</code> に達するまで必要に応じて割り当てられます。<code>bufhwm</code> に達すると、バッファを再利用して要求を満たします。 歴史的経緯により、このパラメータには <code>ufs</code>: 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	物理メモリーの 2%
範囲	80K バイトから物理メモリーの 20%
単位	K バイト
動的か	いいえ。この値は、ハッシュバケットサイズの計算に使用された後、データ構造体に格納されます。このデータ構造体は、バッファの割り当てや解放に応じて、そのフィールドの値を調整します。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。
検査	<code>bufhwm</code> が 80K バイトより小さいか、<code>bufhwm</code> が、物理メモリーの 20% もしくは現在のカーネルヒープ値の 2 倍のうち小さい方より大きい場合、<code>bufhwm</code> は、物理メモリーの 20% か現在のカーネルヒープ値の 2 倍のうち小さい方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 <code>/var/adm/messages</code> ファイルに出力されます。 <pre>"binit: bufhwm out of range (value attempted). Using N."</pre> 「value attempted」は、<code>/etc/system</code> で指定された値または <code>kadb -d</code> による値を表します。<code>N</code> 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。
どのような場合に 変更するか	バッファは必要が生じた時にのみ割り当てられるため、デフォルト設定によるオーバーヘッドは、考えられる最大のバッファ数を処理するために必要になる制御構造体が割り当てられることです。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あたり 52 バイト、64 ビットカーネルでは想定されるバッファあたり 104 バイトを消費します。512M バイトの 64 ビットカーネルでは、この構造体は 104*10144 バイト (約 1M バイト) を消費します。ヘッダーの割り当てでは、バッファサイズは 1K バイトであると想定します。ただし、ほとんどの場合、バッファサイズはこれより大きいのが普通です。

バッファー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。この構造体の `bufsize` フィールドが、残っている可能性があるメモリーのバイト数です。`mdb` を使用して `buf` マクロでこのフィールドを表示するには、次のようにします。

```
# mdb -kLoading modules: [ unix krtld genunix ip nfs ipc ]
> bfreelist$<buf
bfreelist:
[ elided ]
bfreelist + 0x78:    bufsize                [ elided ]
                  75734016
```

6G バイトのメモリーを装備したこのシステムでは、`bufhwm` は 122277 です。要求されたバッファーの実際のサイズは通常、1K バイトを超えるので、ヘッダー構造体の数を直接調べることは困難です。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 バイトシステム上でこの構造体が、10144K バイトのうち 4K バイトだけがまだ割り当てられていないことを示したとします。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

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、[86 ページ](#) の「`tmpfs:tmpfs_maxkmem`」を参照してください。

`tmpfs:tmpfs_minfree` (Solaris 8 リリース)

Solaris 8 バージョンで、「ページ」であるべき単位が誤って「バイト」になっていました。

sun4u または sun4v 固有のパラメータ (Solaris 10 リリース)

Solaris 07 4/4 リリースでは、Sun-8u 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ますが、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)

Solaris 10 リリースで、デフォルト値に関する記述に誤りがありました。デフォルト値は 4 です。

nfs:nrnode (Solaris 9 8/03)

Solaris 10 の記述が更新され、「NFS バージョン 4 クライアント」の語句が追加されました。

nfs:nfs_write_error_interval (Solaris 9 8/03)

Solaris 10 の記述が更新され、「NFS バージョン 4 クライアント」の語句が追加されました。

nfs:nfs_write_error_to_cons_only (Solaris 9 8/03)

Solaris 10 の記述が更新され、「NFS バージョン 4 クライアント」の語句が追加されました。

nfs:nfs_disable_rddir_cache (Solaris 9 8/03)

Solaris 10 の変更する場合の記述が更新され、「NFS バージョン 4 クライアント」の語句が追加されました。

nfs:nfs3_max_transfer_size (Solaris 9 8/03)

Solaris 10 のデフォルトの記述が32,768,(32K バイト) から 1,048,576 (1M バイト) に変更されました。

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)

Solaris 9 リリースおよび 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載されていました。正しいデフォルト値は無効です。

備考欄 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 1 (有効)

ip_multidata_outbound (Solaris 10 リリース)

このパラメータは、Solaris Express 12/05 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、[157 ページ](#)の「`ip_multidata_outbound`」を参照してください。

備考欄

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。

複数データ送信 (multidata transmit、MDT) 機能は、この機能をサポートするデバイスドライバでのみ有効です。

[173 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、[212 ページ](#)の「`ip_multidata_outbound` (Solaris 9 リリース)」を参照してください。

ip_multidata_outbound (Solaris 9 リリース)

このパラメータの情報は、Solaris 9 8/03 リリース以後、大幅に変更されました。詳細は、[157 ページ](#)の「`ip_multidata_outbound`」を参照してください。

備考欄

このパラメータによりネットワークスタックは、転送時にネットワークデバイスドライバに対して一度に複数のパケットを送信できます。

	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。
	複数データ送信 (multidata transmit、MDT) 機能は、この機能をサポートするデバイスドライバでのみ有効です。
	MDT パラメータを使用するには、 <code>/etc/system</code> ファイル内で次のパラメータを有効にする必要があります。
	<code>set ip:ip_use_dl_cap = 0x1</code>
デフォルト	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	この機能はシステムパフォーマンスを向上させる目的でいつでも有効にできますが、次の点に注意します。 ■ この機能を有効にすると、IP 層と DLPI プロバイダの間でパケットの形が変化することがあります。よって、<code>ifconfig</code> の <code>modinsert</code> 機能を使用して IP 層と DLPI プロバイダ間に動的に挿入されるサードパーティー製 STREAMS モジュールは、MDT STREAMS データ形式を理解しないため、動作しない可能性があります。 同様に、<code>autopush(1m)</code> 機構によって IP 層と DLPI プロバイダ間に挿入されるモジュールも、動作しない可能性があります。 ■ STREAMS モジュールが MDT を認識しない場合は、この機能を無効にしてください。たとえば、<code>ipfilter</code> や Checkpoint Firewall-1 などのパブリックドメインユーティリティは MDT を認識しません。
コミットレベル	変更の可能性あり

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、[158 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、[174 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[173 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、[159 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_conn_hash_size (Solaris 9 リリース)

このパラメータは Solaris 10 リリースで削除されました。

備考欄	すべての TCP 接続に対する TCP モジュールのハッシュテーブルサイズを制御します。
データ型	符号付き整数
デフォルト	512
範囲	512 から 1,073,741,824
暗黙的制約	この値には2のべき乗を指定すべきです。
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。
検査	パラメータの値が2のべき乗でないと、一番近い2のべき乗に丸められます。

どのような場合に変更するか	何万に上る TCP 接続が継続的にあるシステムでは、この値をそれに応じて増やします。TCP は、デフォルト値で 2～3 千のアクティブ接続までは十分にサポートできます。ハッシュテーブルのサイズを増やすと、それだけ多くのメモリーが必要になりますので、適切な値を指定するようにしてメモリーが不必要に使用されるのを防いでください。
コミットレベル	変更の可能性あり

tcp_wscale_always (Solaris 9 リリース)

このパラメータのデフォルト値は、Solaris 10 リリースで有効に変更されました。

備考欄	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 セグメントに応答します。オプションの値は受信ウィンドウサイズに従って設定されます。
	ウィンド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	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	ウィンドスケールオプションをサポートしていない古い TCP スタックとの相互運用性の問題がある場合は、このパラメータを無効にしてください。
コミットレベル	変更の可能性あり

ipc_tcp_conn_hash_size (Solaris 9 リリース)

このパラメータは Solaris 10 リリースで削除されました。

備考欄	すべてのアクティブな(「確立済み (ESTABLISHED)」状態) TCP 接続に対して、IP モジュールのハッシュテーブルサイズを制御します。
データ型	符号なし整数
デフォルト	512
範囲	512 から 2,147,483,648
暗黙的制約	この値は2のべき乗を指定すべきです。
動的か	いいえ。このパラメータはブート時のみ変更することができます。
検査	パラメータの値が2のべき乗でないと、一番近い2のべき乗に丸められます。
どのような場合に変更するか	何万に上るアクティブ TCP 接続が継続的にあるシステムでは、この値をそれに応じて増やします。TCP は、デフォルト値で2～3千までのアクティブ接続を十分にサポートできます。ハッシュテーブルのサイズを増やすと、それだけ多くのメモリーが必要になりますので、適切な値を指定するようにしてメモリーが不必要に使用されるのを防いでください。
コミットレベル	変更の可能性あり

tcp_compression_enabled (Solaris 9 リリース)

このパラメータは Solaris 10 リリースで削除されました。

備考欄	1に設定すると、TIME-WAIT 状態の TCP 接続のプロトコル制御ブロックが圧縮され、メモリー使用量が減ります。0が設定されていると、圧縮は行われません。 168 ページ の「 tcp_time_wait_interval 」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	圧縮機能はオフにしないでください。
コミットレベル	変更の可能性あり

ip_forwarding および ip6_forwarding (Solaris 9 リリース)

このパラメータは Solaris 10 リリースで旧式とされました。

備考欄	IP がインタフェース間で IPv4 転送を実行するの か、IPv6 転送を実行するのかを制御します。 217 ページの「xxx:ip_forwarding (Solaris 9 リリース)」 も参照してください。
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	IP 転送が必要な場合は、これを有効にします。
コミットレベル	変更の可能性あり

xxx:ip_forwarding (Solaris 9 リリース)

このパラメータは Solaris 10 リリースで旧式とされました。

備考欄	特定の xxx インタフェースに対して IPv4 転送を有効にします。このパラメータの正確な表記は <i>interface-name:ip_forwarding</i> です。たとえば、2つのインタフェース hme0 と hme1 があるとします。対応するパラメータ名を示します。 hme0:ip_forwarding と hme1:ip_forwarding
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	IPv4 転送が必要な場合は、このパラメータを使用してインタフェースごとに転送を有効にします。
コミットレベル	変更の可能性あり

tcp_conn_req_max_q0 (Solaris 8 リリース)

「どのような場合に変更するか」の記述が最近の Solaris リリースでは下記から変更されました。

どのような場合に変更するか

極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。

`tcp_conn_req_max_q0` 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。

接続要求を受信すると、TCP はまず、受け付けられるのを待っている保留状態の TCP 接続 (3 段階ハンドシェークが終わっている) の数 (N) が、そのリスナーに対する最大数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

`tcp_conn_req_max_q0` の合計を超えていないかをチェックします。超えていなければ、その要求は受け付けられます。超えていると、最も古い不完全な保留状態の TCP 要求がドロップされます。

次のようになりました。

どのような場合に変更するか

極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。

`tcp_conn_req_max_q0` 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。

接続要求を受信すると、TCP はまず、受け付けられるのを待っている保留状態の TCP 接続 (3 段階ハンドシェークが終わっている) の数が、そのリスナーに対する最大数 (N) を超えていないかをチェックします。接続数が超えていると、その要求は拒否されます。超えていなければ、TCP は、不完全な保留状態の TCP 接続の数が、 N と

`tcp_conn_req_max_q0` の合計を超えていないかをチェックします。超えていなければ、その要求は受け付けられます。超えていると、最も古い不完全な保留状態の TCP 要求がドロップされます。

UDP チューニング可能パラメータ

udp_xmit_hiwat (Solaris 9 リリース)

デフォルト値と値の範囲が Solaris 10 リリースで変更されました。

デフォルト 8192 バイト

範囲 4096 から 65,536 バイト

udp_recv_hiwat (Solaris 9 リリース)

デフォルト値と値の範囲が Solaris 10 リリースで変更されました。

デフォルト 8192 バイト

範囲 4096 から 65,536 バイト

udp_max_buf (Solaris 9 リリース)

デフォルト値が Solaris 10 リリースで変更されました。

デフォルト 262,144 バイト

ネットワークキャッシュとアクセラレータ (NCA) の チューニング可能パラメータ

sq_max_size (Solaris 9 12/02 リリース)

このパラメータの情報は最近の Solaris リリースで大幅に変更されました。[197 ページ](#)の「[sq_max_size](#)」を参照してください。

備考欄	宛先ストリームキューが QFULL メッセージを生成するまでの、同期の深度 (メッセージの数) です。
デフォルト	2 メッセージ
範囲	1 から 0 (無制限)
動的か	いいえ

どのような場合に変更するか	メモリーの大きなシステム上でNCAを動作させている場合、このパラメータの値を増やしてドライバがキューに入れることのできるパケット数を増やします。サーバーの負荷が大きい場合、このパラメータの値を増やして、パケットがドロップされたりパケット処理が後回しにされることなくモジュールやドライバのデータ処理能力を向上させます。
コミットレベル	変更の可能性あり

旧式または削除されたパラメータ

ここでは、最近の Solaris リリースで旧式または削除されたパラメータについて説明します。

ページング関連のチューニング可能パラメータ

cachefree (Solaris 8 リリース)

Solaris 9 以降のリリースでは旧式

備考欄	Solaris 8 リリースで、ファイルシステムページのキャッシング方法が変更されました。これらの変更には、優先度ページング機能が含まれます。
-----	---

注 - cachefree と priority_paging の両方の設定を /etc/system ファイルから削除してください。

キャッシュ方法の変更により、ファイルシステムの動作に伴って発生する仮想メモリーシステムへの負担がほとんど解消されます。統計によると、次のような新しい動作が見られます。

- ページ再利用の数が多くなります。これは、入出力が終了すると、ページが明示的に空リストに追加されるためです。
- 空きメモリーが増えます。これは、空きメモリーのカウントにファイルキャッシュの大部分が含まれるためです。
- 走査率が大幅に減ります。

コミットレベル	旧式
---------	----

priority_paging (Solaris 8 リリース)

Solaris 9 以降のリリースでは旧式

備考欄 この変数に基づき `lotsfree` の 2 倍の値が `cachefree` に設定されます。

Solaris 8 リリースで、ファイルシステムページのキャッシング方法が変更されました。これらの変更には、優先度ページング機能が含まれます。

注 - `cachefree` と `priority_paging` の両方の設定を `/etc/system` ファイルから削除してください。

コミットレベル 旧式

tune_t_gpgslo (Solaris 7 リリース)

備考欄 Solaris 8 リリースでは旧式。変数は互換性の理由で残されています。

tune_t_minasmem (Solaris 7 リリース)

備考欄 Solaris 8 リリースでは旧式。変数は互換性の理由で残されています。

System V メッセージキューパラメータ

msgsys:msginfo_msgmni (Solaris 9 リリース)

Solaris 10 リリースでは旧式

備考欄	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。
検査	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 9 リリース)

Solaris 10 リリースでは旧式

備考欄	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 IPC_NOWAIT フラグがオンに設定されていると、要求はエラー EGAIN 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 MAXINT
動的か	いいえ。msginfo 構造体の msgtql フィールドにロードされます。
検査	なし
どのような場合に変更するか	msgsnd(2) 呼び出しがブロックするか EGAIN 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_msgmb (Solaris 9 リリース)

Solaris 10 リリースでは旧式

備考欄	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。msginfo 構造体の msgmb フィールドにロードされます。
検査	なし

どのような場合に変更するか	msgsnd(2) 呼び出しがブロックするか EAGAIN 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 9 リリース)

Solaris 10 リリースで削除

備考欄	メッセージバッファー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 msgtql フィールドにロードされます。
-----	--

検査	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す。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	旧式
---------	----

msgsys:msginfo_msgmap (Solaris 9 リリース)

Solaris 10 リリースで削除

備考欄	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
-----	-----

検査	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す。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	旧式

msgsys:msginfo_msgseg (Solaris 9 リリース)

Solaris 10 リリースで削除

備考欄	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
検査	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す。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	旧式

msgsys:msginfo_msgmax (Solaris 9 リリース)

Solaris 10 リリースで削除

備考欄	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。
検査	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 セマフォのパラメータ**semsys:seminfo_semmni (Solaris 9 リリース)**

Solaris 10 リリースでは旧式

備考欄	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
検査	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す。
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。現在の設定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 9 リリース)

Solaris 10 リリースでは旧式

備考欄

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

検査

セマフォとそれらに關係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。設定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。`semget(2)` 呼び出しからアプリケーションにリターンコード `EINVAL` が返されます。

コミットレベル

変更の可能性あり

`semsys:seminfo_semopm` (Solaris 9 リリース)

Solaris 10 リリースでは旧式

備考欄

`semop` 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、`semop()` システムコールで使用する `sops` 配列内の `sembufs` の数を参照します。詳細は、`semop(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

10

範囲	1 から MAXINT
動的か	いいえ
検査	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。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 9 リリース)

Solaris 10 リリースで削除

備考欄	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
検査	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。設定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。semget(2) 呼び出しからアプリケーションにリターンコード ENOSPC が返されます。

コミットレベル 変更の可能性あり

semsys:seminfo_semmnu (Solaris 9 リリース)

Solaris 10 リリースで削除

備考欄 System V セマフォースystem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

検査 セマフォとそれらに関係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。設定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_sesume (Solaris 9 リリース)

備考欄 Solaris 10 リリースで削除

1 つのプロセスで利用できる 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
検査	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。設定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 9 リリース)

Solaris 10 リリースで削除

備考欄	セマフォを設定できる最大値
データ型	符号なし <code>short</code>
デフォルト	32,767
範囲	1 から 65,535
動的か	いいえ
検査	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> が返されます。
コミットレベル	変更の可能性あり

`semsys:seminfo_semaem` (Solaris 9 リリース)

Solaris 10 リリースで削除

備考欄	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 <code>short</code>

デフォルト	16,384
範囲	1 から 65,535
動的か	いいえ
検査	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。設定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_semmap` (Solaris 7 リリース)

Solaris 8 リリースでは旧式。変数は互換性の理由でカーネルに残されていますが、使用されません。

`semsys:seminfo_semusz` (Solaris 7 リリース)

Solaris 8 リリースでは旧式。値を入力しても無視されます。

System V 共有メモリーパラメータ

`shmsys:shminfo_shmmni` (Solaris 9 リリース)

Solaris 10 リリースでは旧式

備考欄	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。 <code>shminfo</code> 構造体の <code>shmmni</code> フィールドにロードされます。
検査	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 9 リリース)

Solaris 10 リリースでは旧式

備考欄	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。
	サイズが 0 の共有メモリーセグメントや、指定した値より大きいサイズの共有メモリーセグメントを作成しようとする、EINVAL エラーとなります。
	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。
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。
検査	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

shmsys:shminfo_shmmin (Solaris 8 リリース)

Solaris 9 リリースでは旧式。変数は互換性の理由でカーネルに残されていますが、使用されません。

shmsys:shminfo_shmseg (Solaris 8 リリース)

Solaris 9 リリースでは旧式。変数は互換性の理由でカーネルに残されていますが、使用されません。

NFS モジュールのパラメータ

nfs:nfs_32_time_ok (Solaris 7)

Solaris 8 リリースでは旧式。

このマニュアルの改訂履歴

この付録では、このマニュアルの改訂履歴について説明しています。

- 233 ページの「最新バージョン: *Solaris 10 10/08* リリース」
- 235 ページの「*Solaris 10* リリースで追加または変更されたパラメータ」
- 239 ページの「*Solaris 9* リリースで追加または変更されたパラメータ」
- 240 ページの「*Solaris 8* リリースで追加されたパラメータ」

最新バージョン: **Solaris 10 10/08** リリース

このマニュアルの最新バージョンは、*Solaris 10 10/08* リリースに該当します。

Solaris リリースで追加または変更されたパラメータ

次に、追加、変更、または旧式となっ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。

- **Solaris 10 10/08:** このマニュアルの *Solaris 10* バージョンでは、誤って `nfs4_shrinkreaddir` パラメータの情報が含まれていました。このパラメータは使用できません。
- **Solaris 10 10/08:** ZFS ファイルシステムの調整については、次のサイトを参照してください。
http://www.solarisinternals.com/wiki/index.php/ZFS_Evil_Tuning_Guide
- **Solaris 10 5/08:** メモリーの近傍性グループのパラメータは、*Solaris 10 5/08* カーネルのパッチで提供されます。これらのパラメータについての詳細は、[101 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:** このマニュアルの *Solaris 10* バージョンでは、誤って `nfs4_dynamic` パラメータの情報が含まれていました。このパラメータは使用できません。

- **Solaris 10 5/08:** 96 ページの「[sun4u または sun4v 固有のパラメータ](#)」節の変換記憶バッファのパラメータが、より適切な情報に更新されています。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。
 - 100 ページの「[tsb_rss_factor](#)」 - 例に関する部分が、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 5/08:** このマニュアルの Solaris 10 バージョンでは、誤って `tcp_keepalive_abort_interval` パラメータの情報が含まれていました。このパラメータは、Solaris Express リリースでしか使用できません。
- **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。
 - 70 ページの「[maxphys](#)」
 - 86 ページの「[tmpfs:tmpfs_maxkmem](#)」
 - 96 ページの「[sun4u または sun4v 固有のパラメータ](#)」
- **Solaris 10 8/07:** 以前にこのマニュアルに記載されていた `maxpgio` パラメータの値の範囲に関する情報には誤りがありました。詳細は、63 ページの「[maxpgio](#)」を参照してください。
- **Solaris 10 8/07:** 大域ゾーンのみで変更可能なパラメータの変更については、19 ページの「[Solaris システムチューニングの新機能](#)」を参照してください。
- **Solaris 10 8/07:** 以前にこのマニュアルに記載されていた `ip_squeue_write` パラメータの情報は不正確であったため、削除されました。
- **Solaris 10 11/06:** Solaris 10 リリースで、`ncsize` パラメータのデフォルト値に関する記述に誤りがありました。詳細は、73 ページの「[ncsize](#)」を参照してください。
- **Solaris 10 11/06:** Solaris 10 リリースで、`nfs:nfs3_nra` パラメータのデフォルト値に関する記述に誤りがありました。デフォルト値は 4 です。詳細は、123 ページの「[nfs:nfs3_nra](#)」を参照してください。
- **Solaris 10 6/06:** `ip_squeue_fanout` パラメータが変更されました。詳細は、158 ページの「[ip_squeue_fanout](#)」を参照してください。
- **Solaris 10 6/06:** Solaris 10 リリースで `sq_max_size` パラメータに関する記述に誤りがありました。デフォルト値は 10000 メッセージです。詳細は、197 ページの「[sq_max_size](#)」を参照してください。
- **Solaris 10 6/06:** `logevent_max_q_sz` パラメータのデフォルト値が 2000 イベントから 5000 イベントに変更されました。詳細は、39 ページの「[logevent_max_q_sz](#)」を参照してください。
- **Solaris 10 6/06:** Solaris 10 リリースで `lwp_default_stksize` パラメータに関する記述に誤りがありました。SPARC システムのデフォルト値は 24,576 です。詳細は、38 ページの「[lwp_default_stksize](#)」を参照してください。

- **Solaris 10 6/06:** `ip_multidata_outbound` パラメータが拡張されました。詳細は、[157 ページの「ip_multidata_outbound」](#) を参照してください。
- **Solaris 10 6/06:** Solaris 9 および Solaris 10 リリースでの `ip_forward_src_routed` パラメータおよび `ip6_forward_src_routed` パラメータのデフォルト値に関する記述に誤りがありました。正しいデフォルト値は無効です。詳細は、[156 ページの「ip_forward_src_routed と ip6_forward_src_routed」](#) を参照してください。
- **Solaris 10 6/06:** UDP パラメータのデフォルト値および範囲が Solaris 10 リリースで変更されました。以前は新しい値に関する記述がありませんでした。新しい値については、[179 ページの「UDP チューニング可能パラメータ」](#) を参照してください。
- **Solaris 10 6/06:** Solaris 10 リリースで `sq_max_size` パラメータに関する記述に誤りがありました。デフォルト値は 10000 メッセージです。詳細は、[197 ページの「sq_max_size」](#) を参照してください。

Solaris 10 リリースで追加または変更されたパラメータ

次に、追加、変更、または旧式となっ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。

Solaris カーネル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

[37 ページの「default_stksize」](#) パラメータが Solaris 10 リリースで新しく追加されました。

[38 ページの「lwp_default_stksize」](#) パラメータおよび [41 ページの「noexec_user_stack」](#) パラメータがこのリリースで変更されました。

UFS

Solaris 10 リリースでは、次のパラメータが変更されています。

- [78 ページの「bufhwm および bufhwm_pct」](#)
- [73 ページの「ncsize」](#)

一般的なファイルシステム

Solaris 10 リリースでは、次のパラメータの記述が新たに追加されています。

- [84 ページの「freebehind」](#)
- [77 ページの「segmap_percent」](#)

- [85 ページの「smallfile」](#)

System V メッセージキュー

Solaris 10 リリースでは、次のパラメータが削除されました。

- msgsys:msginfo_msgmap
- msgsys:msginfo_msgmax
- msgsys:msginfo_msgseg
- msgsys:msginfo_msgssz

Solaris 10 リリースでは、次のパラメータは旧式とされました。

- msgsys:msginfo_msgmnb
- msgsys:msginfo_msgmni
- msgsys:msginfo_msgtql

System V セマフォー

Solaris 10 リリースでは、次のパラメータが削除されました。

- semsys:seminfo_semmaem
- semsys:seminfo_semmap
- semsys:seminfo_semmns
- semsys:seminfo_semmnu
- semsys:seminfo_sevmx
- semsys:seminfo_sesume
- semsys:seminfo_semusz

System V 共有メモリー

Solaris 10 リリースから次のパラメータが削除されました。

- shmsys:shminfo_shmmin
- shmsys:shminfo_shmseg

次のパラメータは旧式となりました。

- shmsys:shminfo_shmmni
- shmsys:shminfo_shmmax

TSB パラメータ

Solaris 10 リリースでは、次の TSB パラメータが新しく追加されました。

- [97 ページの「tsb_alloc_hiwater_factor」](#)
- [98 ページの「default_tsb_size」](#)

- 99 ページの「`enable_tsb_rss_sizing`」
- 100 ページの「`tsb_rss_factor`」

NFS パラメータ

すべての NFS パラメータについて、安定度が「発展中」から「変更される可能性あり」に変更されました。

Solaris 10 リリースでは、次の NFSv4 パラメータが新しく追加されました。

- 135 ページの「`nfs:nfs4_async_clusters`」
- 132 ページの「`nfs:nfs4_bsize`」
- 112 ページの「`nfs:nfs4_cots_timeo`」
- 114 ページの「`nfs:nfs4_do_symlink_cache`」
- 118 ページの「`nfs:nfs4_lookup_neg_cache`」
- 121 ページの「`nfs:nfs4_max_threads`」
- 140 ページの「`nfs:nfs4_max_transfer_size`」
- 124 ページの「`nfs:nfs4_nra`」
- 108 ページの「`nfs:nfs4_pathconf_disable_cache`」

Solaris 10 リリースでは、次の NFS パラメータが追加または変更されました。

- 122 ページの「`nfs:nfs_nra`」
- 123 ページの「`nfs:nfs3_nra`」
- 127 ページの「`nfs:nfs3_shrinkreaddir`」

次の NFS パラメータはこれまで誤って提供されていましたが、削除されました。

- `nfsserv:nfs_shrinkreaddir`
- `nfsserv:nfs3_shrinkreaddir`

TCP/IP パラメータ

Solaris 10 リリースでは、次の IP パラメータが新しく追加されました。

- 174 ページの「`ip_squeue_worker_wait`」
- 158 ページの「`ip_squeue_fanout`」
- 173 ページの「`ipcl_conn_hash_size`」

この Solaris リリースでは、次の TCP パラメータが追加されました。

- 172 ページの「`tcp_rst_sent_rate_enabled`」
- 172 ページの「`tcp_rst_sent_rate`」
- 173 ページの「`tcp_mdt_max_pbufs`」

Solaris 10 リリースでは、次の TCP/IP パラメータが旧式とされました。

- tcp_conn_hash_size
- ipc_tcp_conn_hash_size
- tcp_compression_enabled
- ip_forwarding
- ip6_forwarding
- xxx_forwarding

SCTP チューニング可能パラメータ

Solaris 10 リリースでは、次の SCTP パラメータが新しく追加されました。

- 181 ページの「sctp_max_init_retr」
- 181 ページの「sctp_pa_max_retr」
- 182 ページの「sctp_pp_max_retr」
- 182 ページの「sctp_cwnd_max」
- 183 ページの「sctp_ipv4_ttl」
- 183 ページの「sctp_heartbeat_interval」
- 183 ページの「sctp_new_secret_interval」
- 184 ページの「sctp_initial_mtu」
- 184 ページの「sctp_deferred_ack_interval」
- 184 ページの「sctp_ignore_path_mtu」
- 185 ページの「sctp_initial_ssthresh」
- 185 ページの「sctp_xmit_hiwat」
- 185 ページの「sctp_xmit_lowat」
- 186 ページの「sctp_recv_hiwat」
- 186 ページの「sctp_max_buf」
- 186 ページの「sctp_ipv6_hoplimit」
- 187 ページの「sctp_rto_min」
- 187 ページの「sctp_rto_max」
- 187 ページの「sctp_rto_initial」
- 188 ページの「sctp_cookie_life」
- 188 ページの「sctp_max_in_streams」
- 188 ページの「sctp_initial_out_streams」
- 189 ページの「sctp_shutack_wait_bound」
- 189 ページの「sctp_maxburst」
- 189 ページの「sctp_addip_enabled」
- 190 ページの「sctp_prsctp_enabled」

システム機能のパラメータ

Solaris 10 リリースでは、次のシステム機能が新しく追加されました。

- 200 ページの「autofs」
- 201 ページの「ftp」

- [202 ページ](#)の「nfs」

Solaris 10 リリースでは、inetd システム機能が旧式とされました。

sun4m サポートの終了

Solaris 10 リリースは sun4m プラットフォームをサポートしません。sun4m のサポート終了に伴ない、次のパラメータが変更されました。

- max_nprocs
- maxphys
- noexec_user_stack

Solaris 9 リリースで追加または変更されたパラメータ

次に、Solaris 9 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。

ip_policy_mask

Solaris 9 12/02 リリースでは、このパラメータが追加されました。詳細は、[180 ページ](#)の「ip_policy_mask」を参照してください。

sun4d サポートの終了

Solaris 9 リリースは sun4d プラットフォームをサポートしません。sun4d のサポート終了に伴ない、次のパラメータが変更されました。

- max_nprocs
- maxphys
- noexec_user_stack

サポートされなくなったまたは旧式のパラメータ

priority_paging と cachefree のサポート終了

Solaris 9 リリースは、priority_paging および cachefree チューニング可能パラメータをサポートしません。これらは優先度ページングと同様の機能を常に有効にするページングポリシーを実装する拡張ファイルシステムキャッシングアーキテクチャにより置き換えられました。/etc/system ファイル内でこれらのパラメータを設定しようとすると、ブート時に次のような警告が表示されます。

```
sorry, variable 'priority_paging' is not defined in the 'kernel'  
sorry, variable 'cachefree' is not defined in the 'kernel'
```

/etc/system ファイルを含む SUNWcsr パッケージは、priority_paging や cachefree のチューニング可能パラメータを含むことを防止するように変更されました。
/etc/system ファイルに priority_paging または cachefree パラメータを含んだまま、Solaris 9 リリースへアップグレードしたり、SUNWcsr パッケージを追加すると、次のようになります。

1. /etc/system ファイル内に priority_paging または cachefree パラメータが設定されていると次のメッセージが表示されます。

```
/etc/system has been modified since it contains references to priority  
paging tunables. Please review the changed file.
```

2. /etc/system ファイル内の priority_paging および cachefree を設定するすべての行の前に次のコメントが挿入されます。たとえば priority_paging を 1 に設定している場合、priority_paging 値の行の前に、次の行が挿入されます。

*NOTE: As of Solaris 9, priority paging is unnecessary and has been removed.

* Since references to priority paging-related tunables will now result in
* boot-time warnings, the assignment below has been commented out. For more
* details, see the Solaris 9 Release Notes, or the "Solaris Tunable Parameters
* Reference Manual".

System V 共有メモリー

次のパラメータは旧式とされました。

- shmsys:shminfo_shmmin
- shmsys:shminfo_shmseg

Solaris 8 リリースで追加されたパラメータ

logevent_max_q_sz

Solaris 8 1/01 リリースでは、このパラメータが新しく追加されました。詳細は、[39 ページの「logevent_max_q_sz」](#)を参照してください。

索引

A

autofs, 200
autoup, 44

B

bufhwm, 78, 208
bufhwm_pct, 78

C

cachefree, 220, 239
consistent_coloring, 96
cron, 200

D

default_stksize, 37
default_tsb_size, 98
desfree, 54
dhcpageant, 200
dnlc_dir_enable, 75
dnlc_dir_max_size, 76
dnlc_dir_min_size, 76
doiflush, 45
dopageflush, 45, 206

E

enable_tsb_rss_sizing, 99

F

fastscan, 60
freebehind, 84
fs, 200
fsflush, 42
ftp, 201

G

ge_intr_mode, 198

H

handspreadpages, 61
hires_tick, 95

I

inetinit, 201
init, 201
ip_addr_per_if, 156
ip_forward_src_routed, 156, 211, 235
ip_forwarding, 217
ip_icmp_err_burst, 154
ip_icmp_err_interval, 154

ip_icmp_return_data_bytes, 160
ip_ire_pathmtu_interval, 160
ip_multidata_outbound, 158,212,213,235
ip_policy_mask, 180,239
ip_respond_to_echo_broadcast, 155
ip_send_redirects, 155
ip_soft_rings_cnt, 159
ip_squeue_fanout, 158,234
ip_squeue_worker_wait, 174
ip_strict_dst_multihoming, 157
ip6_forward_src_routed, 156,211,235
ip6_forwarding, 217
ip6_icmp_return_data_bytes, 160
ip6_respond_to_echo_multicast, 155
ip6_send_redirects, 155
ip6_strict_dst_multihoming, 157
ipc_tcp_conn_hash_size, 216
ipcl_conn_hash_size, 173

K

kbd, 201
keyserv, 201
kmem_flags, 66

L

lgrp_mem_pset_aware, 103
logevent_max_q_sz, 39,240
login, 201
lotsfree, 53
lpg_alloc_prefer, 101
lpg_mem_default_policy, 102
lwp_default_stksize, 38,234

M

max_nprocs, 50,205,206,239
maxpgio, 63,207
maxphys, 70,239
maxpid, 49
maxuprc, 50

maxusers, 47
md_mirror:md_resync_bufsz, 104
md:mirrored_root_flag, 105
min_percent_cpu, 61
minfree, 55
moddebug, 68
mpathd, 201
msgsys:msginfo_msgmax, 225
msgsys:msginfo_msgmnb, 222
msgsys:msginfo_msgmni, 221
msgsys:msginfo_msgseg, 224
msgsys:msginfo_msgssz, 223
msgsys:msginfo_msgtql, 222

N

nca_conn_hash_size, 194
nca_conn_req_max_q, 194
nca_conn_req_max_q0, 195
nca_ppmax, 195
nca_vpmax, 196
ncsize, 73,207
ndd, 154
ndquot, 80
nfs_32_time_ok, 232
nfs_max_threads, 119
nfs:nacache, 137
nfs:nfs_allow_preepoch_time, 110
nfs:nfs_async_clusters, 133
nfs:nfs_async_timeout, 136
nfs:nfs_bsize, 130
nfs:nfs_cots_timeo, 110
nfs:nfs_disable_rddir_cache, 129,211
nfs:nfs_do_symlink_cache, 113
nfs:nfs_dynamic, 115
nfs:nfs_lookup_neg_cache, 116
nfs:nfs_nra, 122
nfs:nfs_shrinkreaddir, 126
nfs:nfs_write_error_interval, 128,211
nfs:nfs_write_error_to_cons_only, 129,211
nfs:nfs3_async_clusters, 134
nfs:nfs3_bsize, 131
nfs:nfs3_cots_timeo, 111
nfs:nfs3_do_symlink_cache, 113

nfs:nfs3_dynamic, 115
 nfs:nfs3_jukebox_delay, 138
 nfs:nfs3_lookup_neg_cache, 117
 nfs:nfs3_max_threads, 120
 nfs:nfs3_max_transfer_size, 139,211
 nfs:nfs3_max_transfer_size_clts, 141
 nfs:nfs3_max_transfer_size_cots, 142
 nfs:nfs3_nra, 123,210
 nfs:nfs3_pathconf_disable_cache, 108
 nfs:nfs3_shrinkreaddir, 127
 nfs:nfs4_async_clusters, 135
 nfs:nfs4_bsize, 132
 nfs:nfs4_cots_timeo, 112
 nfs:nfs4_do_symlink_cache, 114
 nfs:nfs4_lookup_neg_cache, 118
 nfs:nfs4_max_threads, 121
 nfs:nfs4_max_transfer_size, 140
 nfs:nfs4_nra, 124
 nfs:nfs4_pathconf_disable_cache, 108
 nfs:nrnode, 125,210
 nfslogd, 202
 nfssrv:exi_cache_time, 145
 nfssrv:nfs_portmon, 143
 nfssrv:nfsauth_ch_cache_max, 145
 nfssrv:rfs_write_async, 144
 noexec_user_stack, 41,206,239
 nss, 202
 nstrpush, 90

P

pageout_reserve, 57
 pages_before_pager, 62
 pages_pp_maximum, 58,207
 passwd, 202
 physmem, 36
 pidmax, 49
 power, 202
 priority_paging, 221,239
 pt_cnt, 88
 pt_max_pty, 90
 pt_pctofmem, 89

R

rechoose_interval, 94
 reserved_procs, 48
 rlim_fd_cur, 72
 rlim_fd_max, 71,206
 routeadm, 25
 rpc.nisd, 202
 rpcmod:clnt_idle_timeout, 147
 rpcmod:clnt_max_conns, 146
 rpcmod:cotsmaxdupreqs, 151
 rpcmod:maxdupreqs, 150
 rpcmod:svc_default_stksize, 148
 rpcmod:svc_idle_timeout, 147
 rstchown, 74

S

sctp_addip_enabled, 189
 sctp_cookie_life, 188
 sctp_cwnd_max, 182
 sctp_deferred_ack_interval, 184
 sctp_heartbeat_interval, 183
 sctp_ignore_path_mtu, 184
 sctp_initial_mtu, 184
 sctp_initial_out_streams, 188
 sctp_initial_ssthresh, 185
 sctp_ipv4_ttl, 183
 sctp_ipv6_hoplimit, 186
 sctp_max_buf, 186
 sctp_max_in_streams, 188
 sctp_max_init_retr, 181
 sctp_maxburst, 189
 sctp_new_secret_interval, 183
 sctp_pp_max_retr, 182
 sctp_prsctp_enabled, 190
 sctp_recv_hiwat, 186
 sctp_rto_max, 187
 sctp_rto_min, 187
 sctp_shutack_wait_bound, 189
 sctp_xmit_hiwat, 185
 sctp_xmit_lowat, 185
 segkpsize, 206
 segmap_percent, 77
 segspt_minfree, 93

semsys:seminfo_semaem, 229
semsys:seminfo_semap, 230
semsys:seminfo_semmni, 225
semsys:seminfo_semmns, 227
semsys:seminfo_semmnu, 228
semsys:seminfo_semmssl, 226
semsys:seminfo_semopm, 226
semsys:seminfo_semume, 228
semsys:seminfo_semusz, 230
semsys:seminfo_sevmx, 229
shmsys:shminfo_shmmax, 231
shmsys:shminfo_shmmin, 232, 236, 240
shmsys:shminfo_shmmni, 230
shmsys:shminfo_shmseg, 232, 236, 240
slowscan, 60
smallfile, 85
sq_max_size, 197, 219, 234, 235
strmsgsz, 91
su, 202
sun4u, 96, 210
sun4v, 96, 210
swapfs_minfree, 65
swapfs_reserve, 64
sys-suspend, 203
syslog, 202

T

tar, 203
tcp_compression_enabled, 216
tcp_conn_hash_size, 214
tcp_conn_req_max_q, 169
tcp_conn_req_max_q0, 170
tcp_conn_req_min, 171
tcp_cwnd_max, 165
tcp_deferred_ack_interval, 161
tcp_deferred_acks_max, 162
tcp_ecn_permitted, 169
tcp_ip_abort_interval, 175, 176
tcp_keepalive_interval, 175
tcp_local_dack_interval, 161
tcp_local_dacks_max, 162
tcp_max_buf, 165
tcp_mdt_max_pbufs, 173

tcp_recv_hiwat, 165
tcp_recv_hiwat_minmss, 178
tcp_rev_src_routes, 168
tcp_rexmit_interval_extra, 177
tcp_rexmit_interval_max, 176
tcp_rexmit_interval_min, 177
tcp_rst_sent_rate, 172
tcp_rst_sent_rate_enabled, 172
tcp_sack_permitted, 167
tcp_slow_start_after_idle, 166
tcp_slow_start_initial, 166
tcp_time_wait_interval, 168
tcp_tstamp_always, 164
tcp_tstamp_if_wscale, 178
tcp_wscale_always, 163, 215
tcp_xmit_hiwat, 164
throttlefree, 56
timer_max, 95
tmpfs_maxkmem, 86
tmpfs_minfree, 87
tmpfs:tmpfs_maxkmem, 209
tmpfs:tmpfs_minfree, 209
tsb_alloc_hiwat, 97
tsb_rss_size, 100
tune_t_fsflushr, 43
tune_t_gpgslo, 221
tune_t_minarmem, 59
tune_t_minasmem, 221

U

udp_max_buf, 180, 219
udp_recv_hiwat, 179, 219
udp_xmit_hiwat, 179, 219
ufs_HW, 83
ufs_LW, 83
ufs_ninode, 80
ufs:ufs_WRITES, 83
utmpd, 203

X

xxx:ip_forwarding, 217

Y

yppasswdd, 203

