

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

Part No: 819-0376-18
2011 年 8 月

ORACLE®

Copyright © 2000, 2011, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

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

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

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

目次

はじめに	13
1 Oracle Solaris システムのチューニングの概要	17
Oracle Solaris システムチューニングの新機能	17
Solaris 10 リリースでの Oracle Solaris システムのチューニング	19
デフォルトのスタックサイズ	19
System V IPC 構成	19
NFSv4 パラメータ	21
追加または変更された TCP/IP パラメータ	21
SPARC: 変換記憶バッファー (TSB) パラメータ	23
SCTP チューニング可能パラメータ	24
Oracle Solaris システムのチューニング	24
チューニング可能パラメータの説明形式	25
Oracle Solaris カーネルのチューニング	27
/etc/system ファイル	27
kmdb コマンド	28
mdb コマンド	28
Oracle Solaris の特殊な tune および var 構造体	29
Oracle Solaris システム構成情報の表示	30
sysdef コマンド	30
kstat ユーティリティー	31
2 Oracle Solaris カーネルチューニング可能パラメータ	33
チューニング可能パラメータ情報の記載箇所	34
カーネルとメモリーの一般的なパラメータ	34
physmem	34
zfs_arc_min	35

zfs_arc_max	35
default_stksize	36
lwp_default_stksize	37
logevent_max_q_sz	39
segkpsize	39
noexec_user_stack	40
fsflush とそれに関連するパラメータ	41
fsflush	41
tune_t_fsflushr	42
autoup	43
dopageflush	44
doiflush	45
プロセス規模調整パラメータ	46
maxusers	46
reserved_procs	47
pidmax	48
max_nprocs	49
maxuprc	50
ngroups_max	50
ページング関連パラメータ	51
lotsfree	52
desfree	53
minfree	55
throttlefree	56
pageout_reserve	57
pages_pp_maximum	58
tune_t_minarmem	59
fastscan	59
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

カーネルメモリーアロケータ	65
kmem_flags	66
一般的なドライバパラメータ	68
moddebug	68
ddi_msix_alloc_limit	69
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	75
dnlc_dir_max_size	76
segmap_percent	77
UFS パラメータ	77
bufhwm および bufhwm_pct	77
ndquot	79
ufs_ninode	80
ufs_WRITES	82
ufs_LW および ufs_HW	83
freebehind	84
smallfile	84
TMPFS パラメータ	85
tmpfs:tmpfs_maxmem	85
tmpfs:tmpfs_minfree	86
仮想端末	87
pt_cnt	88
pt_pctofmem	88
pt_max_pty	89
STREAMS パラメータ	90
nstrpush	90
strmsgs	90
strctlpsz	91
System V メッセージキュー	91

System V セマフォー	92
System V 共有メモリー	92
segspc_minfree	93
スケジューリング	93
rechoose_interval	93
タイマー	94
hires_tick	94
timer_max	95
SPARC システム固有のパラメータ	95
consistent_coloring	95
tsb_alloc_hiwater_factor	97
default_tsb_size	98
enable_tsb_rss_sizing	98
tsb_rss_factor	99
近傍性グループのパラメータ	100
lpg_alloc_prefer	100
lgrp_mem_default_policy	101
lgrp_mem_pset_aware	102
Solaris ボリュームマネージャーのパラメータ	103
md_mirror:md_resync_bufsz	103
md:mirrored_root_flag	104
 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	128
nfs:nfs_disable_rddir_cache	129
nfs:nfs3_bsize	130
nfs:nfs4_bsize	131
nfs:nfs_async_clusters	132
nfs:nfs3_async_clusters	133
nfs:nfs4_async_clusters	134
nfs:nfs_async_timeout	135
nfs:nacache	136
nfs:nfs3_jukebox_delay	137
nfs:nfs3_max_transfer_size	138
nfs:nfs4_max_transfer_size	139
nfs:nfs3_max_transfer_size_clts	140
nfs:nfs3_max_transfer_size_cots	141
nfssrv モジュールのパラメータ	142
nfssrv:nfs_portmon	142
nfssrv:rfs_write_async	143
rpcmod モジュールのパラメータ	144
rpcmod:clnt_max_conns	144
rpcmod:clnt_idle_timeout	144
rpcmod:svc_idle_timeout	145

rpcmod:svc_default_stksize	146
rpcmod:svc_default_max_same_xprt	146
rpcmod:maxdupreqs	147
rpcmod:cotsmaxdupreqs	148
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	151
チューニング可能パラメータ情報の記載箇所	151
IPパラメータのチューニングの概要	151
IPパラメータの妥当性検査	152
RFC (Internet Request for Comments)	152
IPチューニング可能パラメータ	152
ip_icmp_err_interval と ip_icmp_err_burst	152
ip_respond_to_echo_broadcast と ip6_respond_to_echo_multicast	153
ip_send_redirects と ip6_send_redirects	153
ip_forward_src_routed と ip6_forward_src_routed	154
ip_addrs_per_if	154
ip_strict_dst_multihoming と ip6_strict_dst_multihoming	155
ip_multidata_outbound	155
ip_sqeuee_fanout	156
ip_soft_rings_cnt	157
特別な注意を要するIPチューニング可能パラメータ	158
TCPチューニング可能パラメータ	159
tcp_deferred_ack_interval	159
tcp_local_dack_interval	159
tcp_deferred_acks_max	160
tcp_local_dacks_max	160
tcp_wscale_always	161
tcp_tstamp_always	162
tcp_xmit_hiwat	162
tcp_recv_hiwat	163
tcp_max_buf	163
tcp_cwnd_max	163
tcp_slow_start_initial	164
tcp_slow_start_after_idle	164
tcp_sack_permitted	165

tcp_rev_src_routes	166
tcp_time_wait_interval	166
tcp_ecn_permitted	166
tcp_conn_req_max_q	167
tcp_conn_req_max_q0	168
tcp_conn_req_min	169
tcp_RST_sent_rate_enabled	170
tcp_RST_sent_rate	170
tcp_mdt_max_pbufs	170
tcp_naglim_def	171
tcp_smallest_anon_port	172
tcp_largest_anon_port	172
/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	178
udp_xmit_hiwat	178
udp_recv_hiwat	179
udp_smallest_anon_port	179
udp_largest_anon_port	180
udp_do_checksum	180
特別な注意を要する UDP パラメータ	181
IPQoS チューニング可能パラメータ	181
ip_policy_mask	181
SCTP チューニング可能パラメータ	182
sctp_max_init_retr	182
sctp_pa_max_retr	183
sctp_pp_max_retr	183
sctp_cwnd_max	183
sctp_ipv4_ttl	184
sctp_heartbeat_interval	184
sctp_new_secret_interval	185
sctp_initial_mtu	185
sctp_deferred_ack_interval	185
sctp_ignore_path_mtu	186
sctp_initial_ssthresh	186
sctp_xmit_hiwat	186

sctp_xmit_lowat	187
sctp_recv_hiwat	187
sctp_max_buf	187
sctp_ipv6_hoplimit	188
sctp_rto_min	188
sctp_rto_max	188
sctp_rto_initial	189
sctp_cookie_life	189
sctp_max_in_streams	189
sctp_initial_out_streams	190
sctp_shutack_wait_bound	190
sctp_maxburst	190
sctp_addip_enabled	191
sctp_prsctp_enabled	191
sctp_smallest_anon_port	191
sctp_largest_anon_port	192
ルート別のメトリック	192
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	195
チューニング可能パラメータ情報の記載箇所	195
NCA パラメータのチューニング	196
nca:nca_conn_hash_size	196
nca:nca_conn_req_max_q	196
nca:nca_conn_req_max_q0	197
nca:nca_ppmax	197
nca:nca_vpmax	198
NCA のための一般的なシステムチューニング	199
sq_max_size	199
ge:ge_intr_mode	199
6 システム機能のパラメータ	201
システムのデフォルトのパラメータ	202
autofs	202
cron	202
devfsadm	202

dhcpagent	202
fs	202
ftp	203
inetinit	203
init	203
ipsec	203
kbd	203
keyserv	203
login	203
lu	204
mpathd	204
nfs	204
nfslogd	204
nss	204
passwd	204
power	204
rpc.nisd	204
su	205
syslog	205
sys-suspend	205
tar	205
telnetd	205
utmpd	205
yppasswdd	206
A チューニング可能パラメータの変更履歴	207
カーネルパラメータ	207
プロセスのサイズのチューニング可能パラメータ	207
一般的なドライバパラメータ	207
一般的な入出力のチューニング可能パラメータ	208
カーネルとメモリーの一般的なパラメータ	208
fsflush とそれに関連するパラメータ	208
ページング関連のチューニング可能パラメータ	209
一般的なファイルシステムパラメータ	209
TMPFS パラメータ	209

SPARC システム固有のパラメータ (Solaris 10 リリース)	209
NFS チューニング可能パラメータ	210
nfs:nfs3_nra (Solaris 10 リリース)	210
TCP/IP チューニング可能パラメータ	210
ip_forward_src_routed と ip6_forward_src_routed (Solaris 10 リリース)	210
ip_multidata_outbound (Solaris 10 リリース)	210
ip_squeue_fanout (Solaris 10 11/06 リリース)	210
ip_squeue_worker_wait (Solaris 10 11/06 リリース)	211
ip_soft_rings_cnt (Solaris 10 11/06 リリース)	211
ip_squeue_write (Solaris 10 リリース)	211
tcp_local_dack_interval (Solaris 10 リリース)	211
[tcp,sctp,udp]_smallest_anon_port と [tcp,sctp,udp]_largest_anon_port (Solaris 10 リリース)	211
tcp_naglim_def (Solaris 10 リリース)	211
udp_do_checksum (Solaris 10 リリース)	212
廃止または削除されたパラメータ	212
System V メッセージキューパラメータ	212
System V セマフォーのパラメータ	216
System V 共有メモリーパラメータ	221
 B このマニュアルの改訂履歴	223
最新バージョン: <i>Oracle Solaris 10 8/11 リリース</i>	223
Oracle Solaris リリースで追加または変更されたパラメータ	223
 索引	225

はじめに

『Oracle Solaris カーネルのチューンアップ・リファレンスマニュアル』では、Oracle Solaris のカーネルとネットワークのチューニング可能パラメータに関する参照情報を提供します。このマニュアルは、デスクトップシステムや Java 環境に関するチューニング可能パラメータの情報は提供しません。

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

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

本書の x86 に関する用語については、以下を参照してください。

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

対象読者

このドキュメントは、状況によってはカーネルのチューニング可能パラメータを変更する必要がある、熟練した Oracle Solaris システム管理者を対象としています。Oracle Solaris チューニング可能パラメータの変更に関するガイドラインについては、24 ページの「Oracle Solaris システムのチューニング」を参照してください。

内容の紹介

このドキュメントは、次の各章と付録から構成されています。

章	説明
第1章 「Oracle Solaris システムのチューニングの概要」	Oracle Solaris システムのチューニングの概要。このドキュメントで使用するカーネルのチューニング可能パラメータの書式の説明も含む
第2章 「Oracle Solaris カーネルチューニング可能パラメータ」	Oracle Solaris カーネルのチューニング可能パラメータの説明(カーネルメモリー、ファイルシステム、プロセスサイズ、ページングのパラメータなど)
第3章 「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、/etc/default ディレクトリ内のファイルを編集する
付録 A 「チューニング可能パラメータの変更履歴」	変更または廃止されたパラメータの履歴
付録 B 「このマニュアルの改訂履歴」	このマニュアルの改訂履歴(Oracle Solaris 最新リリースを含む)

Oracle Solaris のチューニングに関するほかの情報源

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

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

Oracle サポートへのアクセス

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

表記上の規則

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:
AaBbCc123	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。
\	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。	sun% grep '^#define \' XV_VERSION_STRING'

Oracle Solaris OS に含まれるシェルで使用する、UNIX のデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solaris のリリースによって異なります。

- 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]
```

- Bash シェル、Korn シェル、および Bourne シェル

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

- Bash シェル、Korn シェル、および Bourne 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。

Oracle 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、Oracle Solaris システムの別のチューニング方法についても説明します。

- 17 ページの「Oracle Solaris システムチューニングの新機能」
- 19 ページの「Solaris 10 リリースでの Oracle Solaris システムのチューニング」
- 24 ページの「Oracle Solaris システムのチューニング」
- 25 ページの「チューニング可能パラメータの説明形式」
- 27 ページの「Oracle Solaris カーネルのチューニング」
- 29 ページの「Oracle Solaris の特殊な tune および var 構造体」
- 30 ページの「Oracle Solaris システム構成情報の表示」
- 31 ページの「kstat ユーティリティー」

Oracle Solaris システムチューニングの新機能

この節では、Oracle Solaris 10 リリースで追加または変更されたパラメータについて説明します。

- **Oracle Solaris 10 8/11:** このリリースには ngroups_max パラメータの説明が含まれています。詳細は、50 ページの「ngroups_max」を参照してください。
- **Solaris 10 10/09:** このリリースには、zfs_arc_min および zfs_arc_max パラメータに関する記述が含まれます。詳細は、35 ページの「zfs_arc_min」および 35 ページの「zfs_arc_max」を参照してください。

ZFS ファイルシステムのチューニングについての詳細は、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 10/09:** このリリースでは、メモリー近傍性グループのパラメータが提供されています。これらのパラメータについての詳細は、100 ページの「近傍性グループのパラメータ」を参照してください。

- **Solaris 10 5/08:** 95 ページの「SPARC システム固有のパラメータ」節の変換格納バッファーのパラメータがより適切な情報を提供するように改訂されました。このリリースでは、次のパラメータが変更されています。
 - 98 ページの「`default_tsb_size`」
 - 98 ページの「`enable_tsb_rss_sizing`」
 - 99 ページの「`tsb_rss_factor`」
- **Solaris 10 8/07:** パラメータの情報が sun4v システムを含むように更新されています。詳細は、次のマニュアルページを参照してください。
 - 70 ページの「`maxphys`」
 - 85 ページの「`tmpfs:tmpfs_maxkmem`」
 - 95 ページの「SPARC システム固有のパラメータ」
- **Solaris 10 8/07:** IP インスタンスプロジェクトによって、ゾーンを排他的 IP ゾーンとして構成し、一部の LAN または VLAN からそのゾーンへの排他的アクセスを割り当てることができます。

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

 - 156 ページの「`ip_sqeue_fanout`」
 - 173 ページの「`ip_sqeue_worker_wait`」

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

Solaris ゾーンでの IP インスタンスの使用に関する詳細は、『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ: 資源管理と Oracle Solaris ゾーン)』を参照してください。

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

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

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

デフォルトのスタックサイズ

新しいパラメータ `default_stksize` では、あらゆるスレッド、カーネル、またはユーザーのデフォルトスタックサイズを指定します。`lwp_default_stksize` パラメータも引き続き利用できますが、すべてのカーネルスタックに作用するわけではありません。`default_stksize` が設定されている場合、それは `lwp_default_stksize` に優先します。詳細は、[36 ページの「`default_stksize`」](#) を参照してください。

System V IPC 構成

Oracle Solaris 10 リリースでは、すべての System V IPC 機能を自動構成することも、または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、およびセマフォーです。

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リブート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、現在は廃止された IPC チューニング可能パラメータと、代わりに使用できる可能性のある資源制御を識別しています。廃止された IPC チューニング可能

パラメータと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	廃止されたチューニング可能パラメータ	旧デフォルト値	最大値	新デフォルト値
process.max-msg-qbytes	msgsys:msginfo_msgrnb	4096	ULONG_MAX	65536
process.max-msg-messages	msgsys:msginfo_msgrql	40	UINT_MAX	8192
process.max-sem-ops	semsys:seminfo_semopm	10	INT_MAX	512
process.max-sem-nsems	semsys:seminfo_semmsl	25	SHRT_MAX	512
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_msgrni	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、『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ: 資源管理と Oracle Solaris ゾーン)』の「使用可能な資源制御」を参照してください。

廃止されたパラメータは、Oracle Solaris システムの `/etc/system` ファイルに引き続き含めることができます。その場合、これらのパラメータは、Oracle Solaris の以前のリリースの場合と同様に、デフォルトの資源制御値の初期化に使用されます。詳細は、212 ページの「廃止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。Oracle Solaris システムの `/etc/system` ファイルにこれらのパラメータが含まれている場合は、コメントになります。

semsys:seminfo_semmns	semsys:seminfo_semvmx
semsys:seminfo_semmnu	semsys:seminfo_semaem
semsys:seminfo_semume	semsys:seminfo_semusz
semsys:seminfo_semmap	shmsys:shminfo_shmseg
shmsys:shminfo_shmmmin	msgsys:msginfo_msgmap

msgsys:msginfo_msgseg	msgsys:msginfo_msgsz
msgsys:msginfo_msgmax	

現在使用できる資源制御の一覧については、[rcctladm\(1M\)](#) 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『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ: 資源管理と Oracle 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

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

- 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」
- 131 ページの 「nfs:nfs4_bsize」
- 134 ページの 「nfs:nfs4_async_clusters」
- 139 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

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

- 173 ページの 「ip_squeue_worker_wait」
- 156 ページの 「ip_squeue_fanout」
- 173 ページの 「ipcl_conn_hash_size」

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

- 170 ページの 「tcp_RST_SENT_RATE_ENABLED」
- 170 ページの 「tcp_RST_SENT_RATE」
- 170 ページの 「tcp_MDT_MAX_PBUFS」

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

- `ipc_tcp_conn_hash_size`
- `tcp_compression_enabled`

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

IP 転送に関する変更

この Oracle Solaris リリースでは、`ndd` コマンドで次のチューニング可能パラメータを設定する代わりに、`routeadm` 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。

- すべての設定値がリブート後も維持されます。
- 新しい `ifconfig router` および `-router` コマンドを `/etc/hostname.interface` ファイルに、インター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ーフェース上で 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ーフェース上で 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ーフェース上で 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ーフェース上で 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、インター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、インター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、特定のインターフェースで 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、特定のインターフェースで 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: 変換記憶バッファー (TSB) パラメータ

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

SCTP チューニング可能パラメータ

この Oracle Solaris リリースには、TCP と同様のサービスを提供する、信頼性の高いトランスポートプロトコルであるストリーム制御伝送プロトコル(SCTP)が用意されています。SCTP チューニング可能パラメータについては、[182 ページの「SCTP チューニング可能パラメータ」](#) を参照してください。

Oracle Solaris システムのチューニング

Oracle Solaris OS は、SPARC および x86 プロセッサで動作する、マルチスレッドでスケーラブルな UNIX オペレーティングシステムです。Solaris は、システムの負荷に自動的に対応するため、最小限のチューニングしか必要ありません。それでも、場合によってはチューニングが必要になることもあります。このドキュメントでは、Oracle Solaris OS で利用可能な、公式にサポートされているカーネルのチューニングオプションについて詳しく説明します。

Solaris カーネルは、常にロードされているコア部分と、参照が発生するとロードされるロード可能モジュールから構成されています。このドキュメント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。このドキュメントに記載された変数の値を 1 つ変更しただけで、システムパフォーマンスに意味のある影響が現れることは、ごく限られた場合にしか起きません。

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

Oracle Solaris の新しいリリースに移行する場合は、空の `/etc/system` ファイルで開始することをお勧めします。最初のステップとしては、自社製またはサードパーティー製のアプリケーションで必要とされるチューニング可能パラメータだけを追加してください。Oracle Solaris 10 リリースでは、System V IPC(セマフォー、共有メモリー、およびメッセージキュー)に関するすべてのチューニング可能パラメータが変更されているため、使用環境で変更するようにしてください。詳細は、[19 ページの「System V IPC 構成」](#) を参照してください。基準検査の確立後に、システムパフォーマンスを評価して、チューニング可能パラメータの追加設定が必要かどうかを決定します。



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

チューニング可能パラメータの説明形式

各チューニング可能パラメータの説明形式は、次のとおりです。

- パラメータ名
- 説明
- データ型
- デフォルト
- 範囲
- 単位
- 動的か
- 検査
- 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

説明

パラメータが何を行うのか、何を制御するのかという簡単な説明。

データ型

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

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

	<ul style="list-style-type: none">■ 64 ビットのカーネルが動作しているシステムでは、long 整数のビット幅は整数の 2 倍です。例：符号なし整数 = 32 ビット、符号なし long 整数 = 64 ビット
データ型	符号付きまたは符号なし short 整数または long 整数を指定。long 整数のビット幅は整数の 2 倍です。例：符号なし整数 = 32 ビット、符号なし long 整数 = 64 ビット
単位	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。 <ul style="list-style-type: none">■ MAXINT - 符号付き整数の最大値 (2,147,483,647) を表します。■ MAXUINT - 符号なし整数の最大値 (4,294,967,295) を表します。
動的か	動作中のシステムで mdb、または kmdb デバッガを使用してパラメータを変更できる場合は「はい」です。パラメータがブート時の初期化だけの場合には「いいえ」です。
検査	システムが、/etc/system ファイルに指定されたとおりの値とデフォルトの値のいずれを変数の値に適用するか調べます。また、検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	インターフェースの安定性を表します。このマニュアルで記述するパラメータの多くは「発展中 (Evolving)」または「変更の可能性あり

(Unstable)」のいずれかに分類されます。詳細は、[attributes\(5\)](#) のマニュアルページを参照してください。

変更履歴

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。

Oracle Solaris カーネルのチューニング

次の表では、パラメータに適用可能なチューニングの方法を示します。

チューニング可能パラメータの適用方法	参照先
/etc/system ファイルの変更	27 ページの「/etc/system ファイル」
カーネルデバッガ (kmdb) の使用	28 ページの「kmdb コマンド」
モジューラデバッガ (mdb) の使用	28 ページの「mdb コマンド」
ndd コマンドによる TCP/IP パラメータの設定	第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	196 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。

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

例—/etc/system を使用したパラメータの設定

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

```
set zfs:zfs_arc_max = 0x780000000
```

適正でない値からの復元

値を変更する前に `/etc/system` ファイルのコピーを作成しておけば、不正な値を簡単に元の値に戻せます。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` ファイルの名前かまたは `/dev/null` を入力します。

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

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

システム回復については、『Solaris のシステム管理(基本編)』を参照してください。

kmdb コマンド

`kmdb` 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。

`kmdb` 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ガするためには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、[kmdb\(1\)](#) のマニュアルページを参照してください。

mdb コマンド

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

システムのメモリー使用量の概要を表すビューを表示します。次に例を示します。

```
# mdb -k
Loading modules: [ unix genunix specfs dtrace zfs sd pcisch sockfs ip hook neti sctp arp
usba fcp fctl md lofs cpc random crypto fcip nca logindmux ptm ufs sppp nfs ]
> ::memstat
Page Summary          Pages        MB %Tot
-----
Kernel                95193        743  37%
ZFS File Data         96308        752  38%
Anon                  28132        219  11%
Exec and libs          1870         14   1%
Page cache             1465         11   1%
Free (cachelist)       4242         33   2%
Free (freelist)        28719        224  11%
Total                 255929       1999
Physical              254495       1988
> $q
```

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

kmdb 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。

Oracle Solaris の特殊な tune および var 構造体

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

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

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

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

Oracle Solaris システム構成情報の表示

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

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

sysdef コマンド

`sysdef` コマンドは、メモリーとプロセスのリソース制限の値、および `tune` 構造体と `v` 構造体の一部を提供します。たとえば、16G バイトのメモリーを備えた SPARC システムの `sysdef` 「チューニング可能パラメータ」セクションは、次のとおりです。

20840448	maximum memory allowed in buffer cache (bufhwm)
15898	maximum number of processes (v.v_proc)
99	maximum global priority in sys class (MAXCLSYSPRI)
15893	maximum processes per user id (v.v_maxup)
30	auto update time limit in seconds (NAUTOUP)
25	page stealing low water mark (GPGSLO)
1	fsflush run rate (FSFLUSHR)
25	minimum resident memory for avoiding deadlock (MINARMEM)
25	minimum swapable memory for avoiding deadlock (MINASMEM)

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

kstat ユーティリティー

kstat データ構造体群は、さまざまなカーネルのサブシステムやドライバによって維持されています。この構造体群は、カーネル内のデータをユーザープログラムに提供する機構を提供します。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、スーパーアダミー権限を持つ必要はありません。詳細は、[kstat\(1M\)](#) または [kstat\(3KSTAT\)](#) のマニュアルページを参照してください。

Oracle Solaris カーネルチューニング可能 パラメータ

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

- 34 ページの「カーネルとメモリーの一般的なパラメータ」
- 41 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「カーネルメモリークロケータ」
- 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」
- 77 ページの「UFS パラメータ」
- 85 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」
- 91 ページの「System V メッセージキュー」
- 92 ページの「System V セマフォー」
- 92 ページの「System V 共有メモリー」
- 93 ページの「スケジューリング」
- 94 ページの「タイマー」
- 95 ページの「SPARC システム固有のパラメータ」
- 100 ページの「近傍性グループのパラメータ」
- 103 ページの「Solaris ボリュームマネージャーのパラメータ」

チューニング可能パラメータ情報の記載箇所

チューニング可能パラメータ	参照先
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

説明	Oracle Solaris OS とファームウェアが把握されたあとで、メモリーの物理ページ数に関するシステム構成を変更します。
データ型	符号なし long
デフォルト	そのシステムで使用できる物理メモリーのページ数。これには、コアカーネルとそのデータが格納されているメモリーは含まれません。
範囲	1 からシステムの物理メモリーの総量まで
単位	ページ
動的か	いいえ
検査	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、 <code>physmem</code> 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

zfs_arc_min

説明

ZFS 調整可能置換キャッシュ (ARC) の最小サイズを決定します。関連項目:[35 ページ](#) の「[zfs_arc_max](#)」

データ型

符号なし整数 (64 ビット)

デフォルト

物理メモリーの 1/32 または 64M バイトのどちらか大きい方の値

範囲

64M バイトから [zfs_arc_max](#)

単位

バイト

動的か

いいえ

検査

はい、範囲が検査されます。

どのような場合に変更するか

システムのメモリー負荷要求が変動するとき、ZFS ARC は要求の少ない時期はデータをキャッシュし、要求の多い時期は縮小します。ただし、ZFS は [zfs_arc_min](#) の値を下回って縮小することはできません。[zfs_arc_min](#) のデフォルト値は多くのメモリーを搭載したシステムでメモリー容量の 12% であるため、かなりのメモリー容量を使用する可能性があります。最大負荷時に要求されるメモリー使用量がシステムメモリーの 88% を超える場合は、このパラメータを調整する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

変更履歴

詳細は、[208 ページ](#) の「[zfs_arc_min \(Solaris 10 リース\)](#)」を参照してください。

zfs_arc_max

説明

ZFS 調整可能置換キャッシュ (ARC) の最大サイズを決定します。関連項目:[35 ページ](#) の「[zfs_arc_min](#)」。

データ型

符号なし整数 (64 ビット)

デフォルト	搭載メモリーが4G バイト未満のシステムでは、メモリーの4分の3
	搭載メモリーが4G バイトを超えるシステムでは、 <code>physmem</code> から 1G バイトを引いた値
範囲	64M バイトから <code>physmem</code>
単位	バイト
動的か	いいえ
検査	はい、範囲が検査されます。
どのような場合に変更するか	将来のメモリー要求が非常に大きく、十分に定義されている場合、ARCがメモリー要求と競合しないように、このパラメータの値を減らしてARCを制限することを検討できます。たとえば、将来の負荷がメモリーの20%を要求することがわかっている場合、残りの80%より多くのメモリーを消費しないようにARCを制限することは意味があります。
コミットレベル	変更の可能性あり
変更履歴	詳細は、208 ページの「 zfs_arc_max (Solaris 10 リース) 」を参照してください。

default_stksize

説明	すべてのスレッドのデフォルトスタックサイズを指定します。 <code>default_stksize</code> 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。 <code>default_stksize</code> が設定されている場合、それは <code>lwp_default_stksize</code> に優先します。 37 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	<ul style="list-style-type: none"> ■ 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	<p>最小値はデフォルト値です。</p> <ul style="list-style-type: none"> ■ SPARCシステムでは PAGESIZE の3倍。 ■ x86 システムでは PAGESIZE の2倍。

	最大値はデフォルト値の 32 倍です。
単位	<code>getpagesize</code> 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。 <code>Illegal stack size, Using N</code>
どのような場合に変更するか	<code>N</code> の値は、 <code>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。
コミットレベル	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。
	変更の可能性あり

lwp_default_stksize

説明	カーネルスレッドの作成時に呼び出しルーチンが明示的に使用サイズを提供しなかった場合に使用する、スタックの大きさのデフォルト値を指定します。
データ型	整数

デフォルト

- 8192 (x86 プラットフォーム)
- 24,576 (SPARC プラットフォーム)
- 20,480 (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`

N の値は、`lwp_default_stksize` のデフォルト値です。

どのような場合に変更するか

スタック容量が足りないためにシステムがパニックになる場合。この問題を解決する最もよい方法は、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、208ページの「[lwp_default_stksize \(Solaris 10 リリース\)](#)」を参照してください。

logevent_max_q_sz

説明

キューに格納して `syseventd` デーモンへの配信を待機させることのできる、システムイベントの最大数です。システムイベントキューのサイズがこの制限に達すると、他のシステムイベントをキューに入れることはできません。

データ型

整数

デフォルト

5000

範囲

0 から MAXINT

単位

システムイベント

動的か

はい

検査

`ddi_log_sysevent` と `sysevent_post_event` によってシステムイベントが生成されるたびに、システムイベントフレームワークはこの値をチェックします。

どのような場合に変更するか

詳細は、[ddi_log_sysevent\(9F\)](#) および[sysevent_post_event\(3SYSEVENT\)](#) のマニュアルページを参照してください。

システムイベントのログ、生成、または送信が失敗したことをエラーログメッセージが示す場合。

コミットレベル

変更の可能性あり

segkpsize

説明

利用できるページング可能なカーネルのメモリー量を指定します。このメモリーは主にカーネルスレッドのスタックに使用されます。この値を増やすと、スレッドの数を増やさないのであれば個々のスレッドにより大きなスタックが使用でき、あるいはより多くのスレッドを使用できるようになります。このパラメータを設定できるのは、64ビットカーネルが動作しているシステムに限られます。64

	ビットカーネルが動作しているシステムは、デフォルトで、24Kバイトのスタックサイズを使用します。
データ型	符号なし long
デフォルト	64 ビットカーネル、2G バイト
	32 ビットカーネル、512M バイト
範囲	64 ビットカーネル、512M バイトから 24G バイト
単位	8K バイトページ
動的か	いいえ
検査	値が、最小および最大サイズ(64 ビットシステムでは 512M バイトおよび 24G バイト)と比較されます。最小値に満たないか、または最大値を超えている場合は 2G バイトにリセットされます。その作用に関するメッセージが表示されます。
どのような場合に変更するか	キャッシュの作成で実際に使用されるサイズは、検査後、segkpsize に指定されている値か、物理メモリーの 50% のうち、小さい方です。
コミットレベル	システム上で多数のプロセスをサポートしなければならない場合。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。

noexec_user_stack

説明	スタックを実行不能として指定できるので、バッファー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 Oracle Solaris システムでは、すべての 64 ビットアプリケーション

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

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

データ型	符号付き整数
デフォルト	0(無効)
範囲	0(無効)、1(有効)
単位	切り替え(オン/オフ)
動的か	はい。ただし、すでに実行中のプロセスに対しては無効です。値が設定された後に起動されたプロセスに対してのみ有効です。
検査	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、 mprotect(2) のマニュアルページを参照してください。
コミットレベル	変更の可能性あり
変更履歴	詳細は、 208 ページの「noexec_user_stack (Solaris 10 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` は定期的に実行され、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クингストアに書き出します。ページは、変更されており、かつ次の条件のどれにも該当しない場合に書き込まれます。
 - ページはカーネルページです
 - ページは使用されていません
 - ページがロックされています
 - ページにスワップデバイスが対応づけられています
 - ページが入出力操作に現在関与しています

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

ページはバックинг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と、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クинг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、[42 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 時間を使用しそぎることがあります。

tune_t_fsflushr

説明 `fsflush` の呼び出し間隔を秒数で指定します。

データ型 符号付き整数

デフォルト 1

範囲	1 から MAXINT
単位	秒
動的か	いいえ
検査	値がゼロ以下の場合は、値は 1 にリセットされ、警告メッセージが出力されます。この確認が行われるのはブート時だけです。
どのような場合に変更するか	autoup パラメータを参照してください。
コミットレベル	変更の可能性あり

autoup

説明

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

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

データ型	符号付き整数
デフォルト	30
範囲	1 から MAXINT
単位	秒
動的か	いいえ
検査	<code>autoup</code> 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	<code>autoup</code> は <code>tune_t_fsflushr</code> の整数倍でなければなりません。最小でも <code>autoup</code> は <code>tune_t_fsflushr</code> 値の 6 倍以上でなければなりません。そうでないと、 <code>fsflush</code> が呼び出されるたびに余計なメモリーが走査されます。

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

どのような場合に変更するか

`autoup` または `tune_t_fsflushr` (あるいはその両方) の変更が必要になる状況はいくつかあります。

- 大きなメモリーをもつシステム - この場合には、`autoup` を増やすと、`fsflush` の個々の呼び出しで走査されるメモリー量が少なくなります。
- メモリーの要求量が最小限のシステム - `autoup` と `tune_t_fsflushr` を両方とも増やすと、走査の回数が減ります。 `autoup` 対 `tune_t_fsflushr` の現在の比率を維持するには `autoup` も増やす必要があります。
- 一時ファイルの数が多いシステム (メールサーバーやソフトウェアビルドマシンなど) - 多数のファイルが作成されて削除された時、`fsflush` によって、これらのファイルのデータページがディスクに不必要に書き込まれるおそれがあります。

コミットレベル

変更の可能性あり

dopageflush

説明

`fsflush` の呼び出し時に、変更されたページの有無についてメモリーを検証するかどうかを制御します。`fsflush` を呼び出すたびに、システムの物理メモリーページ数が判別されます。この値は動的再構成動作によって変更されている可能性があります。呼び出しのたびに、次のアルゴリズムを使用して走査が実行されます。 ページ総数 x `tune_t_fsflushr / autoup` ページ

データ型

符号付き整数

デフォルト

1(有効)

範囲

0(無効)、1(有効)

単位

切り替え(オン/オフ)

動的か

はい

検査	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 (<code>vmstat</code> 出力の <code>sr</code> 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、 208 ページの「dopageflush (Solaris 10 リース)」 を参照してください。

doiflush

説明	<code>fsflush</code> 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、 <code>fsflush</code> の N 回目の呼び出しごとに行われます。ここで N は (<code>autoup / tune_t_fsflushr</code>) 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 N 以上であるかどうかをコードがチェックするので、同期化は <code>fsflush</code> が呼び出されるたびに行われます。 N は <code>fsflush</code> を実行するときに 1 度だけ計算されることに注意してください。その後で <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 (/tmp など) に置いた方がいいでしょう。システム上の i ノードトラフィックを減らすには、 <code>mount -noatime</code> オプションを使用します。このオプションを使うと、ファイルがアクセスされた時に i ノードの更新が行われません。

リアルタイム処理を行うシステムでは、このオプションを無効にし、アプリケーションによってファイルの同期化を明示的に行い、一貫性を保つことを望むこともあるでしょう。

コミットレベル

変更の可能性あり

プロセス規模調整パラメータ

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

maxusers

説明

`maxusers` は、当初、システムがサポートできるログインユーザーの数を指定するものでした。カーネルの生成時に、この設定値に基づいて各種テーブルの大きさが決定されました。Oracle Solaris 最新リースでは、そのサイジングの大半をシステム上のメモリー容量に基づいて行います。したがって、`maxusers` の使い方がこれまでとは大きく変わりました。引き続き、`maxusers` に基づいて決定されるサブシステムには次のものがあります。

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

データ型

符号付き整数

デフォルト

M バイト単位のメモリー容量または 2048 のどちらか小さい方

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 <code>maxusers</code> 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。 <code>out of processes</code>
	次の状況のように、デフォルトのプロセス数が多すぎる場合に、このパラメータを変更するかもしれません。
	<ul style="list-style-type: none"> ■ 多量のメモリーがあり、動作しているプロセスの数が比較的小ないデータベースサーバーでは、<code>maxusers</code> のデフォルト値を少なくすることによってシステムメモリーを節約できます。 ■ 多量のメモリーがあり、動作しているプロセスがほとんどないファイルサーバーでは、この値を減らせる場合があります。しかし、その場合、DNLCのサイズを明示的に設定する必要があります。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

説明	UIDがroot(0)のプロセス用に、プロセステーブルで確保するシステムプロセススロット数を指定します。たとえば、 <code>fsflush</code> にはroot(0)のUIDが与えられます。
データ型	符号付き整数
デフォルト	5
範囲	5 から MAXINT
単位	プロセス数

動的か	いいえ。最初のパラメータ計算の後は使用されません。
検査	/etc/system のどの設定も受け入れられます。
コミットレベル	変更の可能性あり
どのような場合に変更するか	たとえば、システムの UID 0 (root) のプロセスの数を、通常の値から 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

説明 使用可能な最大プロセス ID の値を指定します。

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。
暗默的制約	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

説明

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

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

- ディレクトリ名検索キャッシュのサイズを決めるとき (ncsize が指定されていない場合)
- UFS のディスク割り当て構造体を割り当てるとき (ndquot が指定されていない場合)
- 構成されたシステム V セマフォー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

 $10 + (16 \times \text{maxusers})$

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

詳細は、[207 ページの「max_nprocs \(Solaris 10 リース\)」](#)を参照してください。

maxuprc

説明

個々のユーザーがシステム上に作成できるプロセスの最大数を指定します。

データ型

符号付き整数

デフォルト

`max_nprocs - reserved_procs`

範囲

1 から `max_nprocs - reserved_procs`

単位

プロセス数

動的か

いいえ

検査

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

ngroups_max

説明

プロセスごとの追加グループの最大数を指定します。

データ型

符号付き整数

デフォルト

16

範囲

0 から 1024

単位

グループ

動的か

いいえ

検査

いいえ

どのような場合に変更するか

グループの最大数を増やす場合。

ある特定のユーザーに 17 以上のグループが割り当てられている場合は、NFS 環境で AUTH_SYS 資格に関する問題が発生する可能性があることに留意してください。

コミットレベル

変更の可能性あり

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ると、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、[59 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` の間のメモリ量の一部を調べ、次の条件のどれかに当てはまると走査を終了します。

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

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

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



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

CPU とメモリーの動的再構成 (DR) がサポートされています。システムでメモリーの追加や削除を伴う DR 操作があると、該当のパラメータが `/etc/system` に明示的に設定されていなければ、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かぎり、`/etc/system` に指定された値が使用されます。この場合は、値がリセットされます。

lotsfree

説明

システムのページングを開始する最初のきっかけになります。ページ数がこのしきい値に達すると、ページスキャナが立ち上がり、再利用するメモリーページを探します。

データ型

符号なし long

デフォルト

物理メモリーの 1/64 または 512K バイトのどちらか大きい方

範囲

最小値は、512K バイトまたは物理メモリーの 1/64 のどちらか大きい方であり、`getpagesize` によって返されるページサイズに基づくページ数で表されます。詳細は、[getpagesize\(3C\)](#) を参照してください。

	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。
コミットレベル	経験則によると、このパラメータは、システムが2-3秒で割り当てる必要がある量の2倍にします。このパラメータの適正値は負荷によって異なります。DBMSサーバーはデフォルトの設定で支障がないはずです。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性があるかもしれません。
	負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。許容される最小値は512Kバイトであり、 <code>getpagesize</code> によって返されるページサイズに基づくページ数で表されます。
	変更の可能性あり

desfree

説明	システム上で常時解放しておくべきメモリー容量を指定します。
データ型	符号なし整数
デフォルト	<code>lotsfree / 2</code>

範囲	最小値は、256K バイトまたは物理メモリーの 1/128 のどちらか大きい方であり、 <code>getpagesize</code> によって返されるページサイズに基づくページ数で表されます。
	最大値は物理メモリーのページ数です。この最大値は物理メモリーの 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、次の現象が生じことがあります。 <ul style="list-style-type: none"> ■ 使用可能なメモリーが <code>desfree</code> を超えない限り、非同期の入出力要求が処理されません。したがって、<code>desfree</code> の値を増やすと、増やす前なら処理されたであろう要求が拒否されることがあります。 ■ NFS の非同期書き込みが、同期書き込みとして実行されます。 ■ スワッパーが本来より早く立ち上がり、そのスワッパーの動作が、積極的な動作をする方向に傾きます。 ■ システムに前もって読み込む実行可能ページの数が本来よりも少なくなることがあります。この副次的な影響の結果、アプリケーションの動作が本来よりも遅くなる可能性があります。

どのような場合に変更するか

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

コミットレベル

変更の可能性あり

minfree

説明

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

データ型

符号なし整数

デフォルト

`desfree / 2`

範囲

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

単位

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。

動的か

ページ

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

検査

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

暗黙的制約

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

どのような場合に変更するか

コミットレベル

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

変更の可能性あり

throttlefree

説明

要求を満たせるだけのメモリーがある場合でも、メモリー割り当て要求ブロッキングをスリープ状態にするメモリー レベルを指定します。

データ型

符号なし整数

デフォルト

`minfree`

範囲

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

最大値は物理メモリーのページ数です。この最大値は物理メモリーの4%以内であるべきです。システムは、「検査」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、`/etc/system` ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。

検査

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

暗黙的制約

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

どのような場合に変更するか

一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げます。許容される最小値は128Kバイトで

コミットレベル	あり、 <code>getpagesize</code> によって返されるページサイズに基づくページ数で表されます。詳細は、 getpagesize(3C) を参照してください。
	変更の可能性あり

pageout_reserve

説明

ページアウトスレッドまたはスケジューラスレッドが独占使用できるように確保するページ数を指定します。使用可能なメモリーがこの値を下回ると、ページアウトやスケジューラ以外のプロセスに対するブロックしない割り当ては拒否されます。ページアウトには専用の小さなメモリープールが必要です。ページアウトは、ページをバッキングストアに書き込む入出力に必要なデータ構造体をここから割り当てます。

データ型

符号なし整数

デフォルト

`throttlefree / 2`

範囲

最小値は、64Kバイトまたは物理メモリーの1/512のどちらか大きい方であり、`getpagesize(3C)`によって返されるページサイズに基づくページ数で表されます。

単位

ページ

動的か

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

検査

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

暗黙的制約

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

どのような場合に変更するか

一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を引き下げる。許容される最小値は64Kバイトであり、`getpagesize`によって返されるページサイズに基づくページ数で表されます。

コミットレベル

変更の可能性あり

pages_pp_maximum

説明

ロック解除されなければならないページ数を指定します。ページのロック要求によって使用可能なメモリーがこの値を下回る場合は、その要求は拒否されます。

データ型

符号なし long

デフォルト

`tune_t_minarmem + 100` と、ブート時に使用可能なメモリーの 4% + 4M バイトのどちらか大きい方

範囲

システムが強制する最小値は `tune_t_minarmem + 100` です。最大値については、システムは強制しません。

単位

ページ

動的か

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

検査

`/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 `tune_t_minarmem + 100` よりも小さい場合、この値は `tune_t_minarmem + 100` へリセットされます。

`/etc/system` ファイルからの値が増やしても、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な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、`SHARE_MMU` フラグを指定した共有メモリーセグメントへの接続が失敗したが、使用可能なメモリーが十分ありそうな場合。

コミットレベル	大きすぎる値が原因で、メモリーのロック要求(<code>mlock</code> 、 <code>mlockall</code> 、および <code>memcntl</code>)が不必要に失敗する場合。詳細は、 mlock(3C) 、 mlockall(3C) 、および memcntl(2) のマニュアルページを参照してください。
	変更の可能性あり

tune_t_minarmem

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

説明	メモリー要求が大きいときにシステムが調べる、最大ページ数/秒を指定します。
データ型	符号付き整数
デフォルト	システムのブート後に、 <code>fastscan</code> は 64M バイトに設定されます。その後、この値は、スキャナが CPU

	の 10% を使用して 1 秒間に走査できるページ数に自動的にリセットされます。この派生した値がシステムの物理メモリーの半分を超えた場合、デフォルト値はシステムの物理メモリーの半分に制限されます。
範囲	64M バイトからシステムの物理メモリーの半分
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点での値は、/etc/system ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検査	最大値は、64M バイトと物理メモリーの 1/2 のどちらか小さい方です。
どのような場合に変更するか	メモリー不足のときにメモリーの走査を優先させたい場合。特に、システムでメモリーの要求が急激に多くなることがある場合や、多数のファイル入出力が行われることがある場合。
コミットレベル	変更の可能性あり

slowscan

説明	メモリーの再要求時にシステムが調べる、最小ページ数 / 秒を指定します。
データ型	符号付き整数
デフォルト	物理メモリーの 1/20 (ページ数) か 100 (小さい方)
範囲	1 から fastscan / 2
単位	ページ
動的か	はい。ただしメモリーの追加や削除を伴う動的再構成が行われると、動的に変更した値は無効になります。その時点での値は、/etc/system ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検査	slowscan が fastscan / 2 より大きい場合、slowscan は fastscan / 2 にリセットされます。メッセージは表示されません。

どのような場合に変更するか

コミットレベル

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。

変更の可能性あり

min_percent_cpu

説明

`pageout` が最低限消費できる CPU の割合を指定します。このパラメータは、ページスキャナで使用できる最大時間を判定するための開始点として使用されます。

データ型

符号付き整数

デフォルト

4

範囲

1 から 80

単位

%

動的か

はい

検査

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

説明

Oracle Solaris OS は双針クロックアルゴリズムを使用して、メモリー不足のときに再利用の候補となるページを探します。最初の針はメモリーに使用されていないという印を付けていきます。次の針は、最初の針の少し後から、そのページに依然として使用されていないという印が付けられているかを調べます。そうであれば、そのページが再利用の対象になります。最初の針と次の針の間隔が `handspreadpages` です。

データ型

符号なし long

デフォルト

`fastscan`

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

説明	再利用に備えてページを保管する代わりに、入出力の完了後ただちにページを解放する、システムしきい値の部分を指定します。このしきい値は <code>lotsfree + pages_before_pager</code> です。さらに、NFS 環境も、メモリーが不足するとこのしきい値を使用して非同期の活動を減らします。
データ型	符号付き整数
デフォルト	200
範囲	1 から物理メモリーのページ数
単位	ページ
動的か	いいえ
検査	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できます。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

説明

ページングシステムがキューに入れることのできるページ入出力要求の最大数を指定します。ページングシステムは、実際に使用する最大数を計算するために、この数字を 4 で割ります。このパラメータは、要求の数を制限する他に、プロセスのスワッピングを制御するためにも使用されます。

データ型

符号付き整数

デフォルト

40

範囲

最小値は 1 です。最大値はシステムアーキテクチャーによって決まります。主に、コントローラやディスクの数、そしてディスクのスワップサイズなどの入出力サブシステムによります。

単位

入出力

動的か

いいえ

検査

なし

暗黙的制約

ページヤからの入出力要求の最大数は、要求バッファー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は増やします。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「 maxpgio (Solaris 10 リース) 」を参照してください。

スワッピング関連パラメータ

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

swapfs_reserve

説明	システム (UID=0) プロセス用に予約するシステムメモリー容量を指定します。
データ型	符号なし long
デフォルト	4M バイトと物理メモリーの 1/16 のどちらか小さい方
範囲	最小値は、4M バイトまたは物理メモリーの 1/16 のどちらか小さい方であり、getpagesize によって返されるページサイズに基づくページ数で表されます。
	最大値は物理メモリーのページ数です。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つたり、スワップ空間が取得できないためにシステムプロセスが終了してしまう

場合だけ変更します。しかし、それより良い解決策は、物理メモリーかスワップデバイスをシステムに追加することです。

コミットレベル

変更の可能性あり

swapfs_minfree

説明

システムの他の部分のために、解放しておくべき物理メモリーの容量を指定します。プロセスのスワップ空間としてメモリーを予約しようするとき、それによって使用可能なメモリーがこの値を下回るおそれがあるとシステムが判断する場合、この要求は拒否されます。この方法で予約されたページは、カーネルやユーザーレベルプロセスによってロックダウンされた割り当てに対してのみ使用できます。

データ型

符号なし long

デフォルト

2M バイトと物理メモリーの 1/8 のどちらか大きい方

範囲

1 から物理メモリーのページ数

単位

ページ

動的か

いいえ

検査

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

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

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

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

kmem_flags

説明

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

次に、サポートされる5つのフラグの設定について説明します。

フラグ	設定	説明
AUDIT	0x1	クロケータは、自身の活動の最近の履歴が入ったログを維持します。ログされる項目の数は、CONTENTSも設定されているかどうかによって異なります。このログは固定の大きさです。領域を使い果たすと、古い記録から再利用されます。
TEST	0x2	ク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。

フラグ	設定	説明
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も後ろに書き込まれていないか調べます。上書きが検知されると、カーネルがパニックになります。
CONTENTS	0x8	アロケータは、バッファーが解放されると、バッファーの内容を256バイトまでログします。このフラグを使用するには、AUDITも設定する必要があります。 これらのフラグの数値は、論理的に合算し、/etc/system ファイルによって設定できます。
LITE	0x100	バッファー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ーが再び解放されていないこと、解放されるバッファーのサイズが割り当てられたものと同じであることをチェックします。このフラグは他のフラグと併用しないでください。

データ型	符号付き整数
デフォルト	0(無効)
範囲	0(無効)、1 - 15、256(0x100)
動的か	はい。実行時の変更は、新しいカーネルメモリーキャッシュだけに有効です。システムの初期設定後に新しいキャッシュを作成することはまれです。
検査	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

説明

このパラメータが有効なとき、モジュールのロードプロセスの各種ステップについてのメッセージが表示されます。

データ型

符号付き整数

デフォルト

0(メッセージを表示しない)

範囲

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。

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

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

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

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

動的か	これらの値は足し合わせて指定できます。
検査	はい
どのような場合に変更するか	なし
コミットレベル	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。 <code>0x4000000</code> 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。
	変更の可能性あり

ddi_msix_alloc_limit

説明	x86のみ: このパラメータは、デバイスインスタンスで割り当てる事のできる MSI-X(拡張メッセージ信号割り込み)の数を制御します。既存のシステムの制限により、デフォルト値は2です。このパラメータの値を大きくすることによって、デバイスインスタンスが割り当てる事のできる MSI-X 割り込みの数を増やすことができます。このパラメータを設定するには、 <code>/etc/system</code> ファイルを編集するか、またはデバイスドライバの接続が発生する前に <code>mdb</code> を使用してパラメータを設定します。
データ型	符号付き整数
デフォルト	2
範囲	1 から 16
動的か	はい
検査	なし
どのような場合に変更するか	デバイスインスタンスが割り当てる事のできる MSI-X 割り込みの数を増やすため。ただし、デバイスインスタンスが割り当てる事のできる MSI-X 割り込みの数を増やすと、割り込み数が不足してすべての割り当て要求を満足できなくなる可能性があります。この状況が起きた場合、一部のデバイスが機能を停止したり、システムが起動に失敗したりす

る可能性があります。そのような場合は、パラメータの値を小さくするか、またはパラメータを削除してください。

コミットレベル

変更の可能性あり

変更履歴

詳細は、[207 ページの「ddi_msix_alloc_limit \(Solaris 10 リリース\)」](#) を参照してください。

一般的な入出力パラメータ

maxphys

説明

物理入出力要求の最大サイズを指定します。要求がこのサイズより大きいと、ドライバはこの要求を `maxphys` サイズのチャンクに分割します。個々のファイルシステムは独立して制限値を持つことが可能で、実際に独立した制限値を持ちます。

データ型

符号付き整数

デフォルト

131,072 (sun4u または sun4v) または 57,344 (x86)。ワード転送をサポートする `sd` ドライバは 1,048,576 を使用します。`ssd` ドライバはデフォルトで 1,048,576 を使用します。

範囲

マシン固有のページサイズから `MAXINT`

単位

バイト

動的か

はい。しかし、多くのファイルシステムでは、ファイルシステムがマウントされるときに、この値がマウントポイントごとのデータ構造体に設定されます。ドライバによっては、デバイスがドライバ固有のデータ構造体に設定されるときに、この値が設定されます。

検査

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チャンクで行う場合。OLTP 操作を伴う DBMS では小さいサイズの入出力が頻繁に行われることに留意してください。その場合、`maxphys` を変更してもパフォーマンスの向上は望めません。

UFS ファイルシステムとの間で入出力を行う際、常に大量(64K バイト超)のデータの読み取り/書き込みが行われる場合も、このパラメータの変更を検討することをお勧めします。ファイルシステムは、連続性が向上するように最適化する必要があります。たとえば、シリングループのサイズを増やし、シリングループあたりの i ノード数を減らします。UFS では、転送する最大の入出力サイズに対して 1M バイトの内部制限が課せられています。

コミットレベル

変更の可能性あり

変更履歴

詳細は、[208 ページの「maxphys \(Solaris 10 リリース\)」](#) を参照してください。

rlim_fd_max

説明

1つのプロセスが開くことのできるファイル記述子に対して、強い限度を指定します。この制限を変更するには、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。32 ビットアプリケーション

コードは、より大きな `fd_set` サイズ (65,536 以下) で再コンパイルできます。64 ビットアプリケーションの `fd_set` サイズは 65,536 で、変更することはできません。

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

コミットレベル

変更の可能性あり

rlim_fd_cur

説明

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

データ型

符号付き整数

デフォルト

256

範囲

1 から `MAXINT`

単位

ファイル記述子

動的か

いいえ

検査

`rlim_fd_max` と比較します。`rlim_fd_cur` が `rlim_fd_max` より大きい場合、`rlim_fd_cur` は `rlim_fd_max` にリセットされます。

どのような場合に変更するか

1 プロセス当たりのデフォルトのオープンファイル数が十分でない場合。この値を増やす意味は、プログラムで `setrlimit` を使用して自身で使用できる最大ファイル記述子数を増やす必要がなくなる点にあります。

コミットレベル

変更の可能性あり

一般的なファイルシステムパラメータ

ncsize

説明

ディレクトリ名検索キャッシュ(DNLC)のエントリ数を指定します。このパラメータ

は、UFS、NFS、およびZFSが、解決されたパス名の要素をキャッシュするときに使用します。

DNLCは、否定的な検索情報もキャッシュします。これは、キャッシュ内で見つからない名前がキャッシュされることを意味します。

データ型

符号付き整数

デフォルト

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

範囲

0 から MAXINT

単位

DNLC のエントリ

動的か

いいえ

検査

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

`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:ninode` 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、209ページの「 ncsize (Solaris 10 リース) 」を参照してください。

rstchown

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

dnlc_dir_enable

説明

大きなディレクトリのキャッシングを有効にします。

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

データ型

符号なし整数

デフォルト

1(有効)

範囲

0(無効)、1(有効)

動的か

はい。しかし動的には変更しないでください。このパラメータは、元々無効だった場合に有効にできます。または、元々有効だった場合に、無効にできます。しかし、有効にし、無効にし、再び有効にすると、ディレクトリキャッシュが最新の状態を表さないことがあります。

検査

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、`dnlc_dir_enable` を 0 に設定してキャッシングを無効にしてください。

コミットレベル

変更の可能性あり

dnlc_dir_min_size

説明

1つのディレクトリでキャッシングする最小エントリ数を指定します。

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

データ型

符号なし整数

デフォルト

40

範囲

0 から MAXUINT(無制限)

単位	エントリ
動的か	はい、このパラメータはいつでも変更できます。
検査	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

説明 1つのディレクトリでキャッシュできるエントリの最大数を指定します。

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

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

説明	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ー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 physmem 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

説明	入出力バッファーのキャッシングに使用するメモリーの最大量を指定します。バッファーは、ファイルシステムのメタデータ(スーパー ブロック、i ノード、間接 ブロック、ディレクトリ)の書き込みに使用されます。割り当てられるメモリー量(K バイト 単位)が bufhwm を超えるまで、必要に応じてバッファー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ーが回復されるまで、バッファーキャ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、bufhwm には ufs: 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 2T バイトのどちらか小さい方。つまり、bufhwm_pct は 1 から 20 にできます。

単位	<code>bufhwm</code> : K バイト
動的か	<code>bufhwm_pct</code> : 物理メモリーの %。 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、バッファー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して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%、2T バイト、またはカーネルヒープの最大量の 1/4 のいずれか小さい方)を超える場合は、上限にリセットされます。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 출력されます。 <code>"binit: bufhwm (value attempted) out of range (range start..range end). Using N as default."</code> 「value attempted」は、 <code>/etc/system</code> ファイルに指定された値、またはカーネルデバッガ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、 <code>bufhwm_pct</code> 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 <code>/var/adm/messages</code> ファイルに output されます。 <code>"binit: bufhwm_pct(value attempted) out of range(0..20). Using 2 as default."</code>
どのような場合に変更するか	<code>bufhwm</code> と <code>bufhwm_pct</code> がどちらもゼロ以外の値に設定されている場合は、 <code>bufhwm</code> が優先されます。
	バッファー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、バッファー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ー当たり 52 バイト、64 ビットカーネルでは想定されるバッファー当たり 96 バイトを消費します。
	512M バイトの 64 ビットカーネルでは、ハッシュチェーン数は $10316 / 32 = 322$ であり、次の 2 の累乗の 512 まで拡大されま

す。したがって、ハッシュヘッダーは 512 x 96 バイト、すなわち 48K バイトを消費します。ハッシュヘッダー割り当てでは、バッファーが 32K バイトであることが前提です。

バッファーポ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、カーネルデバッガを使用して、カーネルの `bfreeclist` 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 ]
> bfreeclist::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ーサイズは通常、1K バイトより大きいので、ヘッダー構造体の数を判断することはできません。しかし、一部の領域は、このシステムに割り当てられた制御構造体からうまく回収することができます。

512M バイトのシステムでは、同じ構造体は、10144K バイトのうち 4K バイトだけが割り当てられていないことを示します。また、`kstat -n biostats` で `biostats` の `kstat` を調べると、このシステムでは、`buffer_cache_hits` と `buffer_cache_lookups` の割合も適切であることが分かったとします。これらの情報は、このシステムのデフォルト設定であることを示します。

コミットレベル	変更の可能性あり
---------	----------

ndquot

説明

割り当て対象となる UFS ファイルシステム用の割り当て構造体の数を指定します。このパラメータは、1 つまたは複数の UFS ファイルで割り当てが有効になっているときだけ適用されます。歴史的経緯により、`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。
コミットレベル	<code>dquot table full</code> 変更の可能性あり

ufs_ninode

説明 メモリー内で維持すべき 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られます。参照されていたページはメモリーに残ります。

i ノードをアイドリングする場合、カーネルはこのアイド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られると、キューの占有率が下位境界値と比較され、占有率が下位境界値を超えていると、そのキューに関連するスレッドが起こされます。キューが起こされるると、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	<code>ufs_ninode</code> が 0 以下の場合、この値は <code>ncsize</code> 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。 <code>kstat -n inode_cache</code> で報告される <code>maxsize reached</code> フィールドが <code>kstat</code> の <code>maxsize</code> フィールドより大きい場合は、 <code>ufs_ninode</code> の値が小さすぎる可能性があります。i ノードのアイドリングが多すぎる場合も、問題になることがあります。
コミットレベル	i ノードのアイドリングが多すぎるかどうかは、 <code>kstat -n inode_cache</code> を使用して、 <code>inode_cache</code> <code>kstat</code> を調べることで判断できます。 <code>thread idles</code> はバックグラウンドスレッドがアイドリングした i ノード数を、 <code>vget idles</code> 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。

ufs_WRITESEN

説明	<code>ufs_WRITESEN</code> 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。 <code>ufs_HW</code> を参照し、書き込みを行うべきか、未処理のバイト数が <code>ufs_LW</code> 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、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

コミットレベル	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。
	変更の可能性あり

ufs_LW および ufs_HW

説明

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 $8 \times 1024 \times 1024$ 、ufs_HW の場合は $16 \times 1024 \times 1024$

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、カーネルデバッガを使用した場合だけです。

コミットレベル

変更の可能性あり

freebehind

説明

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

データ型

ブール型

デフォルト

1(有効)

範囲

0(無効)、1(有効)

動的か

はい

検査

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

説明

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、深刻なメモリー要求を行わずに、10M バイトのファイルを何千もキャッシュするのに十分なメモリーがあります。しかし、この状況はあくまでもアプリケーションに強く依存します。

`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	0
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 バイトのファイルです。
コミットレベル	変更の可能性あり

TMPFS パラメータ

tmpfs:tmpfs_maxkmem

説明	TMPFS がデータ構造体 (tmp ノードとディレクトリエンタリ) に使用できるカーネルメモリーの最大量を指定します。
データ型	符号なし long
デフォルト	1 ページまたは物理メモリーの 4% (どちらか大きい方)。
範囲	1 ページのバイト数 (sun4u か sun4v システムの場合は 8192、その他のシステムの場合は 4096) か

ら、 TMPFS が最初に使用されたときに存在していたカーネルメモリーの 25%。

単位	バイト
動的か	はい
検査	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_malloc: 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一量は、 <code>tmp_kmemspace</code> フィールドにあります。カーネルデバッガを使用すると、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

説明	TMPFS がシステムの他の部分のために残しておくスワップ空間の最小量を指定します。
データ型	符号付き long
デフォルト	256
範囲	0 からスワップ空間サイズの最大値
単位	ページ
動的か	はい
検査	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示しています。 <code>fs-name: File system full, swap space limit exceeded</code>
コミットレベル	変更の可能性あり

仮想端末

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

- telnet、rlogin、または rsh コマンドを使用した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ーフタウンドウを作成するときに使用するインターフェースを提供する。

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

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

構成処理では、次の 3つの関連する変数が使用されます。

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

pt_cnt のデフォルト値はゼロで、pt_max_pty が設定されていない限り、システムは pt_pctofmem に指定されたメモリー量に基づいてログインを制限します。pt_cnt がゼロでない場合は、この制限に達するまで pty が割り当てられます。この制限に達すると、システムは pt_max_pty を参照します。pt_max_pty の値がゼロ以外の場合、pt_cnt と比較されます。pt_cnt が pt_max_pty より小さい場合は、pty を割り当てるることができます。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_pctofmem に基づいた制限値が有効となるのは、pt_pctofmem 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のサポートに使用されるまで実際に割り当てられません。しかし、メモリーがいったん割り当てられると、解放されません。

pt_cnt

説明

使用できる /dev/pts エントリの数は、システム上で使用できる物理メモリー容量によって決まる限度の範囲内で動的です。pt_cnt は、システムがサポートできるログイン数の最小値を決める3つの変数のうちの1つです。システムがサポートできる /dev/pts デバイスのデフォルトの最大数は、ブート時に、指定されたシステムメモリーの割合(pt_pctofmem を参照)に適合する pty 構造体の数を計算することによって決められます。pt_cnt がゼロの場合、システムはこの最大数まで割り当てます。pt_cnt がゼロでない場合は、システムは pt_cnt かデフォルトの最大数のうち大きい方まで割り当てます。

データ型

符号なし整数

デフォルト

0

範囲

0 から maxpid

単位

ログイン / ウィンドウ

動的か

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

説明

データ構造体が /dev/pts エントリをサポートするために消費できる物理メモリーの最大の割合を指定します。64ビットカーネルのシステムでは /dev/pts エントリ当たり 176 バイトを消費しま

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

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

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

説明	1 つのシステム呼び出しで STREAM に渡し、メッセージのデータ部分に格納できる最大バイト数を指定します。 このサイズを超える <code>write</code> は、複数のメッセージに分割されます。 詳細は、 write(2) のマニュアルページを参照してください。
データ型	符号付き整数
デフォルト	65,536
範囲	0 から 262,144
単位	バイト
動的か	はい
検査	なし

どのような場合に変更するか コミットレベル	<code>putmsg</code> 呼び出しから <code>ERANGE</code> が返された場合。詳細は、 putmsg(2) のマニュアルページを参照してください。
	変更の可能性あり

strctlsz

説明	1つのシステム呼び出しで STREAM に渡し、メッセージの制御部分に格納できる最大バイト数を指定します。
データ型	符号付き整数
デフォルト	1024
範囲	0 から MAXINT
単位	バイト
動的か	はい
検査	なし
どのような場合に変更するか コミットレベル	ソフトウェアベンダーの指定がある場合。この限度を超えると、 putmsg(2) 呼び出しから <code>ERANGE</code> が返されます。
	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ーフェースを提供します。Oracle Solaris 環境には、メッセージをキューに入れたりキューから取り出したりするためのインターフェースが用意されています。メッセージは、自身の型を持つことができます。キューに入れる場合、メッセージはキューの終わりに置かれます。キューを解除する場合は、指定された型の最初のメッセージがキューから削除されます。型が指定されていない場合は、最初のメッセージが削除されます。

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

これらのシステムリソースをチューニングする方法については、『[Oracle Solaris のシステム管理 \(Oracle Solaris コンテナ: 資源管理と Oracle Solaris ゾーン\)](#)』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、212 ページの「[廃止または削除されたパラメータ](#)」を参照してください。

System V セマフォー

System V セマフォーは Oracle Solaris OS で計数型セマフォーを提供します。「セマフォー」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 セマフォーでは、セマフォーの標準的な設定/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 セマフォーによって、1組のセマフォーに同時に操作を実行したり、プロセスが停止した場合にそのプロセスによる最後の操作を取り消したりすることもできます。

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

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

廃止された System V セマフォーパラメータの詳細は、212 ページの「[廃止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には shmsys: 接頭辞が含まれている必要があります。

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

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

Oracle Solaris 10 リリースでの新しい資源制御の使用方法の詳細は、『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ: 資源管理と Oracle Solaris ゾーン)』の第6章「資源制御 (概要)」を参照してください。

廃止された System V 共有メモリーパラメータの詳細は、212 ページの「廃止または削除されたパラメータ」を参照してください。

segspt_minfree

説明	ISM 共有メモリーに割り当てる事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、ほぼ間違いなく、最大値 128M バイト (0x4000) で十分です。
コミットレベル	変更の可能性あり

スケジューリング

rechoose_interval

説明	プロセスが最後に実行していた CPU に対するすべての親和性を失ったとみなされるまでの、クロック刻みの数。この期間が過ぎると、すべての CPU は
----	---

スレッドスケジューリングの候補と見なされます。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。
コミットレベル	このパラメータを変更する前に、プロセッサセットの機能またはプロセッサバインディングの使用を検討してください。詳細は、 psrset(1M) または pbind(1M) のマニュアルページを参照してください。
	変更の可能性あり

タイマー

hires_tick

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

検査	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

説明	使用できるPOSIX タイマーの数を指定します。
データ型	符号付き整数
デフォルト	32
範囲	0 から MAXINT
動的か	いいえ。値を増やすと、システムクラッシュを起こす可能性があります。
検査	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、 <code>EAGAIN</code> エラーを受け取ります。
コミットレベル	変更の可能性あり

SPARC システム固有のパラメータ

consistent_coloring

説明	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利用できます。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。デフォルタ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起こると、ページ配置コードは 1 つの区画から 1 つのページを割り当てます。選択されるページは、次の 3 つのアルゴリズムのどれが使用されているかによって異なります。

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

動的か

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

説明

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

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

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

この係数を使用して `tsb_alloc_hiwater` の値を増やす場合は、注意が必要です。システム停止を防止するには、高位境界値が `swapfs_minfree` と `segsp_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 に割り当てるここと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

説明	すべてのプロセスに割り当てる初期変換記憶バッファー(TSB)のサイズを選択します。
データ型	整数
デフォルト	デフォルト値は0(8K バイト)で、これは512エントリに対応します。
範囲	指定可能な値は、次のとおりです。

値	説明
0	8K バイト
1	16K バイト
3	32K バイト
4	128K バイト
5	256K バイト
6	512K バイト
7	1M バイト

動的か	はい
検査	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、209 ページの「 default_tsb_size (Solaris 10 リリース) 」を参照してください。

enable_tsb_rss_sizing

説明	TSB 発見的容量調整に基づく常駐セットサイズ(RSS)を有効にします。
----	--------------------------------------

データ型	ブール型
デフォルト	1 (TSB のサイズ変更が可能)
範囲	0 (TSB は <code>tsb_default_size</code> のまま) または 1 (TSB のサイズ変更が可能)
動的か	0 に設定した場合、 <code>tsb_rss_factor</code> は無視されます。
検査	はい
どのような場合に変更するか	はい
コミットレベル	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。
変更履歴	変更の可能性あり 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

説明	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ぬ 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することができますが、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。

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) を使用するなどの SPARC または x86 システムにも適用できる、汎用的なメモリーチューニング可能パラメータについて説明します。

lpg_alloc_prefer

説明

大規模なメモリーページの割り当てを行う際に、要求されたページサイズがローカルのメモリーグループ内ではすぐに利用できないが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、Oracle 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 検査	いいえ 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。
コミットレベル	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方がより重要となります。 TLB の誤動作は、 <code>trapstat -T</code> コマンドを使用して監視できます。

lgrp_mem_default_policy

説明	この変数は、Oracle Solaris OS が使用するデフォルトのメモリ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るが、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 の割り当てる次
2	LGRP_MEM_POLICY_RANDOM_PSET	プロセスにわたってランダム
3	LGRP_MEM_POLICY_RANDOM_PSET	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

不確実

lgrp_mem_pset_aware

説明

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

		ロセッサセット内のプロセッサで構成されている lgroup のみから選択されるのかが決定されます。
		プロセッサセットの作成についての詳細は、 psrset(1M) を参照してください。
データ型	ブール型	
デフォルト	0。	Oracle Solaris OS はシステム内のすべての lgroup からメモリーを選択します
範囲		<ul style="list-style-type: none"> ■ 0。 Oracle Solaris OS はシステム内のすべての lgroup からメモリーを選択します(デフォルト)。 ■ 1。 プロセッサセット内のプロセッサで構成されている lgroup のみからメモリーを選択しようと試みます。最初の試みが失敗した場合は、任意の lgroup 内のメモリーを割り当てることができます。
動的か	いいえ	
検査	なし	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。
どのような場合に変更するか		
コミットレベル	不確実	

Solaris ボリュームマネージャーのパラメータ

md_mirror:md_resync_bufsz

説明	RAID 1 ボリューム(ミラー)の再同期に使用するバッファーのサイズを 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。全体的なシステムパフォーマンスに対し十分なメモリーがある場合は、この値を増加させても他のパフォーマンス上の問題は発生しません。
コミットレベル	ミラー再同期の速度を上げる場合は、充分なパフォーマンスとなるまで、このパラメータの値を (128 ブロックずつ) 増加させてください。かなり大きなシステムや新しいシステムでは、この値を 2048 とするのが適当と考えられます。旧式のシステムでこの値を高く設定するとシステムがハングアップする場合があります。
	変更の可能性あり

md:mirrored_root_flag

説明	複製定数に関する Solaris ボリュームマネージャー要件を変更し、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。
	<ul style="list-style-type: none"> ■ root (/) またはその他のシステムに不可欠なファイルシステムがミラー化されている場合 ■ 使用できるディスクまたはコントローラが2つだけである ■ システムの無人リブートが必要である
	このパラメータを有効にすると、システムの状態(どちら側のミラーが良好または「保守(Maintenance)」状態などのなどを含む)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。
コミットレベル	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。
	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」
- 144 ページの「rpcmod モジュールのパラメータ」

チューニング可能パラメータ情報の記載箇所

チューニング可能パラメータ	参照先
Oracle Solaris カーネルのチューニング可能パラメータ	第 2 章 「Oracle Solaris カーネルチューニング可能パラメータ」
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、24 ページの「Oracle Solaris システムのチューニング」を参照してください。



注意-パラメータ名や、それが存在するモジュール、デフォルト値は、リリースによって変わることがあります。変更を行ったり、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するパラメータについて説明します。

nfs:nfs3_pathconf_disable_cache

説明	NFS バージョン 3 でマウントされたファイルシステムの pathconf 情報をキャッシングするかどうかを制御します。
データ型	整数(32 ビット)
デフォルト	0(キャッシングを有効にする)
範囲	0(キャッシングを有効にする) または 1(キャッシングを無効にする)
単位	ブール値
動的か	はい
検査	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ングされます。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ングエントリを検証する方法はありません。
コミットレベル	変更の可能性あり

nfs:nfs4_pathconf_disable_cache

説明	NFS バージョン 4 でマウントされたファイルシステムの pathconf 情報をキャッシングするかどうかを制御します。
データ型	整数(32 ビット)

デフォルト	0(キャッシングを有効にする)
範囲	0(キャッシングを有効にする)または1(キャッシングを無効にする)
単位	ブル値
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ングされます。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ング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_preeepoch_time

説明

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」)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 日よりも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

説明	トランSPORTプロトコルとして TCP などの接続型トランSPORT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。
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。
コミットレベル	クライアントが無用にタイムアウトしてしまうことを防ぐには、このパラメータを増やします。ただし、値の範囲が非常に広いため、この値を大きくすぎると、再転送が長い間検出されないおそれがあります。
	変更の可能性あり

nfs:nfs3_cots_timeo

説明	トランSPORTプロトコルとして TCP などの接続型トランSPORT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
どのような場合に変更するか	TCP では、要求と応答が適切に転送されるよう機能します。しかし、特に遅いネットワークにおいて往復時間が非常に長くなると、NFS バージョン 3 のクライアントがタイムアウトになってしまう可能性があります。

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

コミットレベル

変更の可能性あり

nfs:nfs4_cots_timeo

説明

トランSPORTプロトコルとしてTCPなどの接続型トランSPORT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ため、この値を大きくしそうだと、再転送が長い間検出されないおそれがあります。

コミットレベル

変更の可能性あり

nfs:nfs_do_symlink_cache

説明

NFSバージョン2でマウントされたファイルシステムに対して、シンボリックリンクファイルの内容をキャッシュするかどうかを制御します。

データ型

整数(32ビット)

デフォルト

1(キャッシングを有効にする)

範囲

0(キャッシングを無効にする)または1(キャッシングを有効にする)

単位

ブール値

動的か

はい

検査

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

説明

NFSバージョン3でマウントされたファイルシステムに対して、シンボリックリンクファイルの内容をキャッシュするかどうかを制御します。

データ型

整数(32ビット)

デフォルト	1(キャッシングを有効にする)
範囲	0(キャッシングを無効にする) または 1(キャッシングを有効にする)
単位	ブール値
動的か	はい
検査	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

説明	NFS バージョン 4 でマウントされたファイルシステムに対して、シンボリックリンクファイルの内容をキャッシ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	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

コミットレベル	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。
	変更の可能性あり

nfs:nfs_dynamic

説明

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

データ型

整数(32ビット)

デフォルト

1(有効)

範囲

0(無効)、1(有効)

動的か

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

検査

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

説明

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

	ウトと読み取り / 書き込みの転送サイズを調整することによって、再転送の回数を減らそうとするものです。
データ型	整数(32ビット)
デフォルト	0(無効)
範囲	0(無効)、1(有効)
単位	ブール値
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	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

説明	NFS バージョン 2 でマウントされたファイルシステムに対して、ネガティブ名前キャッシュを使用するかどうかを制御します。このネガティブ名前キャッシュを使用すると、ロックアップされたファイル名が存在しない場合には、そのファイル名が記録されます。このキャッシュは、存在しないことがすでにわかっているファイル名のロックアップ要求が、ネットワークを介して行われないようにする目的で使用されます。
データ型	整数(32ビット)
デフォルト	1(有効)
範囲	0(無効)、1(有効)
単位	ブール値
動的か	はい
検査	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

説明

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

このキャッシング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ング機構になります。

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ングを無効にします。

`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

説明

NFS バージョン 4 でマウントされたファイルシステムに対して、ネガティブ名前キャッシング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ングを使用すると、ロックアップされたファイル名が存在しない場合には、そのファイル名が記録されます。このキャッシングは、存在しないことがすでにわかっているファイル名のロックアップ要求が、ネットワークを介して行われないようにする目的で使用されます。

データ型

整数(32 ビット)

デフォルト

1(有効)

範囲

0(無効)、1(有効)

単位

ブール値

動的か

はい

検査	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。
読み取り専用でマウントされているファイルシステムがサーバーで変更されたら、その変更をクライアントでただちに見る必要がある場合は、このパラメータを使用してネガティブキャッシュを無効にします。	
コミットレベル	nfs:nfs_disable_raddir_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_raddir_cache」を参照してください。
	変更の可能性あり

nfs:nfs_max_threads

説明	NFS バージョン 2 クライアントの非同期入出力を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。
データ型	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/サーバーリソースをいっそう有効に活用できるようになります。
コミットレベル	変更の可能性あり

nfs:nfs3_max_threads

説明	NFS バージョン 3 クライアントの非同期入出力を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/サーバーリソースをいっそ有効に活用できるようになります。
コミットレベル	変更の可能性あり

nfs:nfs4_max_threads

説明	NFS バージョン 4 クライアントの非同期入出力を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/サーバーリソースをいっそう有効に活用できるようになります。
コミットレベル	変更の可能性あり

nfs:nfs_nra

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

コミットレベル

変更の可能性あり

nfs:nfs3_nra

説明

ファイルへの順次アクセスが検出されたときに NFS バージョン 3 クライアントによってキューに入れられる先読み操作の数を制御します。これらの先読み操作では、並行性が高まり、読み取りのスループットが向上します。個々の先読み要求は、通常、ファイルデータの 1 論理ブロックに対するものです。

データ型

整数(32 ビット)

デフォルト

4

範囲

0 から $2^{31} - 1$

単位

論理ブロック(130 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

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:nfs3_nra (Solaris 10 リース) 」を参照してください。

nfs:nfs4_nra

説明	ファイルへの順次アクセスが検出されたときに NFS バージョン 4 クライアントによってキューに入れられる先読み操作の数を制御します。これらの先読み操作では、並行性が高まり、読み取りのスループットが向上します。個々の先読み要求は、通常、ファイルデータの 1 論理ブロックに対するものです。
データ型	整数(32 ビット)
デフォルト	4
範囲	0 から $2^{31} - 1$
単位	論理ブロック(131 ページの「nfs:nfs4_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:nrnnode

説明

NFS クライアントの rnode キャッシュのサイズを制御します。

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

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

データ型

整数(32 ビット)

デフォルト

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

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。

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、`nrnnode` 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、`rnnode` 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、`rnnode` 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、`rnnode` 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

nfs:nfs_shrinkreaddir

説明

以前の一部の NFS サーバーでは、NFS バージョン 2 の REaddir 要求で 1024 バイトより大きいディレクトリ情報を求めるものが正しく処理されないことがありました。これは、サーバーの実装にバグがあったためです。このパラメータは、NFS バージョン 2 クライアントでの対処方法を含んでいます。

このパラメータが有効化されると、クライアントは、1024 バイトよりも大きいディレクトリ情報を求める REaddir 要求を生成しなくなります。このパラメータを無効にすると、送信されるサイズは、`getdents` システム呼び出しを使用するか、または `NFS_MAXDATA` (8192 バイト) を使用して渡されるサイズのどちらか小さい方に設定されます。詳細は、[getdents\(2\)](#) のマニュアルページを参照してください。

データ型

整数 (32 ビット)

デフォルト

0 (無効)

範囲	0(無効)、1(有効)
単位	ブール値
動的か	はい
検査	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

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

説明	NFS クライアントが受け取った ENOSPC および EDQUOT 書き込みエラーのロギング間隔を制御します。このパラメータは、NFS バージョン 2、3、および 4 のクライアントに影響を与えます。
データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	5 秒
範囲	32 ビットプラットフォームでは 0 から $2^{31}-1$ 64 ビットプラットフォームでは 0 から $2^{63}-1$
単位	秒
動的か	はい
検査	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

nfs:nfs_write_error_to_cons_only

説明	NFS 書き込みエラーをシステムコンソールと syslog に記録するか、それともシステムコンソールだけに記録するかを制御します。このパラメータは、NFS
----	---

データ型	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。
デフォルト	整数(32ビット)
範囲	0(システムコンソールとsyslog) または1(システムコンソール)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	syslogd デーモンによってロギングされるメッセージを含むファイルシステムがいっぱいになるのを防ぐには、このパラメータの値を調べます。このパラメータを有効にすると、メッセージはシステムコンソールに出力されるだけで、syslog メッセージファイルにはコピーされません。
コミットレベル	変更の可能性あり

nfs:nfs_disable_rddir_cache

説明	REaddir 要求と REaddirplus 要求に対する応答を格納するために、キャッシングを使用するかどうかを制御します。このキャッシングを使用すると、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つたり、ディレクトリからファイルを削除しても古い名前が削除されない場合は、この問題があります。

このパラメータは、NFS バージョン 2、3、および 4 でマウントされたファイルシステムのキャッシングに適用されます。このパラメータは NFS でマウントされたすべてのファイルシステムに適用されるため、キャッシングをファイルシステムごとに有効にしたり、無効にしたりすることはできません。

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

- 116 ページの「`nfs:nfs_lookup_neg_cache`」
- 117 ページの「`nfs:nfs3_lookup_neg_cache`」
- 118 ページの「`nfs:nfs4_lookup_neg_cache`」

コミットレベル

変更の可能性あり

nfs:nfs3_bsize

説明

NFS バージョン 3 のクライアントが使用する論理ブロックサイズを制御します。このブロックサイズは、クライアントが入出力を行うときにサーバーに対する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

どのような場合に変更するか	なることがあります。このパラメータには、そのプラットフォームの PAGESIZE より小さい値を設定しないでください。さらに、このパラメータの値が大きすぎると、メモリー割り当てが許可されるまで待つ間に、システムがハンギングすることがあります。
コミットレベル	データ転送サイズの最大値を変更したい場合は、このパラメータの値を調べてください。このパラメータは、nfs:nfs3_max_transfer_size パラメータと連携して変更してください。転送サイズを増やしたい場合は両方のパラメータを増やします。転送サイズを減らしたい場合は、通常、このパラメータを減らすだけで十分です。

変更の可能性あり

nfs:nfs4_bsize

説明

データ型

デフォルト

範囲

単位

動的か

検査

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。

符号なし整数(32 ビット)

32,768 (32K バイト)

0 から $2^{31} - 1$

バイト

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。

ありません。このパラメータの設定が小さすぎたり、大きすぎたりすると、システムの動作が異常になることがあります。このパラメータには、そのプラットフォームの PAGESIZE より小さい値を設定しないでください。さらに、このパラメータの値が大きすぎると、メモリー割り当てが許可されるまで待つ間に、システムがハンギングすることがあります。

どのような場合に変更するか

データ転送サイズの最大値を変更したい場合は、このパラメータの値を調べてください。このパラメータは、`nfs:nfs4_max_transfer_size` パラメータとセットで変更してください。転送サイズを増やしたい場合は両方のパラメータを増やします。転送サイズを減らしたい場合は、通常、このパラメータを減らすだけで十分です。

コミットレベル

変更の可能性あり

nfs:nfs_async_clusters

説明

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できないようにします。

しかし、書き込みの一括化(書き込みをまとめ)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。キューから要求を一度に 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。
どのような場合に変更するか	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。
コミットレベル	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。

nfs:nfs3_async_clusters

説明	NFS バージョン3 クライアントによって生成される非同期要求の組み合わせを制御します。非同期要求には、read-ahead、putpage、pageio、readdir-ahead、commitという5つのタイプがあります。クライアントは、これらのタイプをラウンドロビンに処理し、特定の要求タイプだけを優遇することがないようにします。
データ型	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
デフォルト	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。
範囲	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。

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。この値によって、アルゴリズムの公平性の部分が実質的に無効にされます。
どのような場合に変更するか	あるタイプについて次のタイプへ移行する前に生成される非同期操作の数を増やす場合。これによって、クライアントからの操作のクラスタに依存するサーバーの機能が生かされる場合があります。
コミットレベル	変更の可能性あり

nfs:nfs4_async_clusters

説明	<p>NFS バージョン 4 クライアントによって生成される非同期要求の組み合わせを制御します。非同期要求には、read-ahead、putpage、pageio、readdir-ahead、commit、および inactive という 6 つのタイプがあります。クライアントは、これらのタイプをラウンドロビンに処理し、特定の要求タイプだけを優遇することがないようにします。</p> <p>しかし、書き込みの一括化(書き込みをまとめ)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。キューから要求を一度に 1 つずつ取り出したのでは、クライアントのパフォーマンスを高めるために設けられたこのサーバー機能が生かされません。</p> <p>そこで、このパラメータを使用して、タイプが変更されるまでに送信される、各要求タイプの要求数を制御します。</p>
----	--

データ型	符号なし整数(32ビット)
デフォルト	1
範囲	0から $2^{31}-1$
単位	非同期要求
動的か	はい。ただし、ファイルシステムに対するクラスタ設定は、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	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

説明	非同期入出力要求を実行するスレッドが終了するまで、なにもしない休眠状態を続けることのできる時間の長さを制御します。実行する要求がないと各スレッドは休眠状態になります。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。 <ul style="list-style-type: none"> ■ スレッドの終了までの時間を短くして、カーネルリソースの解放を早くする。 ■ スレッドの終了までの時間を長くして、スレッドの作成や破棄にかかるオーバーヘッドを減らす。
コミットレベル	変更の可能性あり

nfs:nocache

説明	NFS クライアント上のファイルアクセスキャッシュにアクセスするハッシュキューの数を調整します。ファイルアクセスキャッシュは、ユーザーがアクセスしようとするファイルに関する、ユーザーの持つファイルアクセス権を格納します。キャッシュそのものは動的に割り当てられます。しかし、キャッシュに対するインデックスを作成するためのハッシュキューは、静的に割り当てられます。このアルゴリズムでは、アクティブファイルごとに1つのアクセスキャッシュエントリが、ハッシュバケットごとにこれらの4つのアクセスキャッシュエントリがあるものとみなします。したがって、このパラメータの値には、デフォルトで nrnode パラメータの値が設定されます。
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 ncache の値に nrnode パラメータの値が設定されるべきであることを示しています。
範囲	1 から $2^{31} - 1$

単位	アクセスキャッシュエントリ
動的か	いいえ。この値は、/etc/system ファイルにパラメータを追加するか、パラメータを変更し、その後システムをリブートすることによってのみ変更できます。
検査	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ーをシステムに割り当てようとし、その間におそらくハングします。
どのような場合に変更するか	1つのファイルごとに1つのアクセスキャッシュエントリがあるという基本的な前提が損われるおそれがある場合は、このパラメータの値を検討します。複数のユーザーが同じファイルにほぼ同時にアクセスするタイムシェアリングモードのシステムでは、この前提が損なわれる可能性があります。このような場合には、予想されるアクセスキャッシュのサイズを増やすことが、キャッシュへのハッシュアクセスの効率性を保つ上で役立つことがあります。
コミットレベル	変更の可能性あり

nfs:nfs3_jukebox_delay

説明	NFS バージョン 3 クライアントが前回の要求で NFS3ERR_JUKEBOX エラーを受け取ってから、新しい要求を送信するまでに待機する時間の長さを制御します。NFS3ERR_JUKEBOX エラーは、通常、何らかの理由でファイルが一時的に使用できないときにサーバーから返されます。このエラーは、通常、階層型記憶装置、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

説明	NFS バージョン 3 の READ、WRITE、REaddir、または REaddirplus 要求のデータ部分の最大サイズを制御します。このパラメータは、サーバーが返す要求の最大サイズとクライアントが生成する要求の最大サイズの両方を制御します。
データ型	整数(32 ビット)
デフォルト	1,048,576(1M バイト)
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	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 には、1 データグラム当たり 64K バイトという厳しい制限値があります。この 64K バイトには、要求のデータ部分のほかに、RPC ヘッダーやその他の NFS 情報も含まれている必要があります。

どのような場合に変更するか

す。この制限値が大きすぎると、UDP エラーのためにクライアントとサーバーの通信に問題が発生することがあります。

コミットレベル

ネットワークを介して送信するデータのサイズをチューニングする場合。通常、`nfs:nfs3_bsize` パラメータもこのパラメータの変更が反映されるように更新すべきです。

たとえば、転送サイズを 32K バイトより大きい値に増やす場合は、その増加した値が反映されるように `nfs:nfs3_bsize` を更新します。そうしないと、ネットワーク上で送信される要求のサイズは変わりません。詳細は、[130 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。

変更の可能性あり

nfs:nfs4_max_transfer_size

説明

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

データ型

整数(32 ビット)

デフォルト

32,768(32 K バイト)

範囲

0 から $2^{31} - 1$

単位

バイト

動的か

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

検査

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SPORT 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、[138 ページ](#) の「`nfs:nfs3_max_transfer_size`」を参照してください。

どのような場合に変更するか

ネットワークを介して送信するデータのサイズをチューニングする場合。通常、`nfs:nfs4_bsize` パラメータもこのパラメータの変更が反映されるように更新すべきです。

たとえば、転送サイズを 32K バイトより大きい値に増やす場合は、その増加した値が反映されるように `nfs:nfs4_bsize` を更新します。そうしないと、ネットワーク上で送信される要求のサイズは変わりません。詳細は、[131 ページ](#) の「`nfs:nfs4_bsize`」を参照してください。

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

コミットレベル

変更の可能性あり

nfs:nfs3_max_transfer_size_clts

説明

NFS バージョン 3 の UDP を介した READ、WRITE、REaddir、または REaddirplus 要求のデータ部分の最大サイズを制御します。このパラメータは、サーバーが返す要求の最大サイズとクライアントが生成する要求の最大サイズの両方を制御します。

データ型

整数(32 ビット)

デフォルト

32,768 (32 K バイト)

範囲

0 から $2^{31} - 1$

単位

バイト

動的か

はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを

検査	変更してからそのファイルシステムをアンマウントし、再びマウントします。
どのような場合に変更するか	ありません。しかし、サーバー側の最大転送サイズに0を設定すると、クライアントはおそらく、誤作動するか、単にサーバーに要求を送信しないかのどちらかになる可能性があります。
コミットレベル	このパラメータは変更しないでください。 変更の可能性あり

nfs:nfs3_max_transfer_size_cots

説明	NFS バージョン 3 の TCP を介した READ、WRITE、REaddir、または REaddirplus 要求のデータ部分の最大サイズを制御します。このパラメータは、サーバーが返す要求の最大サイズとクライアントが生成する要求の最大サイズの両方を制御します。
データ型	整数(32 ビット)
デフォルト	1,048,576 バイト
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	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

説明

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ー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。このセキュリティーチェックでは、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(セキュリティーチェックを無効にする)

範囲

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

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

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、セキュリティー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

説明

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

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:cint_max_conn

説明

個々の NFS サーバーと通信するときに、NFS クライアントが使用する TCP 接続の数を制御します。1 つの接続で RPC を多重化できるように、カーネル RPC が構築されます。しかし、必要な場合には複数の接続を使用できます。

データ型

整数(32 ビット)

デフォルト

1

範囲

1 から $2^{31} - 1$

単位

接続

動的か

はい

検査

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きることができます。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できます。

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:cint_idle_timeout

説明

クライアントとサーバー間の接続が終了するまでにアイドル状態を維持できる、クライアント側の時間の長さを制御します。

データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	300,000 ミリ秒 (5 分)
範囲	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

説明	クライアントとサーバー間の接続が終了するまでにアイドル状態を維持できる、サーバー側の時間の長さを制御します。
データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	360,000 ミリ秒 (6 分)
範囲	32 ビットプラットフォームでは 0 から $2^{31} - 1$ 64 ビットプラットフォームでは 0 から $2^{63} - 1$
単位	ミリ秒
動的か	はい
検査	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

説明

カーネル RPC サービス スレッドに対するカーネル
スタックのサイズを設定します。

データ型

整数(32 ビット)

デフォルト

デフォルト値は 0 です。この場合、スタックサイズ
はシステムデフォルトに設定されます。

範囲

0 から $2^{31} - 1$

単位

バイト

動的か

はい。新しく割り当てられるすべてのスレッドに適
用されます。スタックサイズはスレッドの作成時に
設定されます。したがって、このパラメータの変更
は、既存のスレッドには適用されず、新しく割り当
てられるすべてのスレッドに適用されます。

検査

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SPORT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合せ
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

説明

各トランSPORT終端の要求を最大でいくつ処理し
たら、次のトランSPORT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SPORT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
SPORT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ときに設定されます。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

説明

コネクションレストラנס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 <code>MKDIR</code> 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。一般に、コネクションレストランsport のクライアントのタイムアウトは比較的短く、1 秒 から 20 秒くらいです。
コミットレベル	変更の可能性あり

rpcmod:cotsmaxdupreqs

説明	接続型トランsport 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。
どのような場合に変更するか	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。
コミットレベル	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、情報を保持していかなければなり 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。
	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。

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

チューニング可能パラメータ情報の記載箇所

チューニング可能パラメータ	参照先
Oracle Solaris カーネルのチューニング可能パラメータ	第2章 「Oracle Solaris カーネルチューニング可能パラメータ」
NFS チューニング可能パラメータ	第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、21 ページの「追加または変更された TCP/IP パラメータ」を参照してください。

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

- 173 ページの「ipcl_conn_hash_size」

- 173 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。

`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

説明

IP で IPv4 または IPv6 ICMP エラーメッセージを生成する頻度を制御します。IP は、`ip_icmp_err_interval` の間に最大で `ip_icmp_err_burst` の IPv4 または IPv6 ICMP エラーメッセージを生成します。

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

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

ip_respond_to_echo_broadcast と ip6_respond_to_echo_multicast

説明	IPv4 や IPv6 が、ブロードキャスト ICMPv4 エコー要求またはマルチキャスト ICMPv6 エコー要求に応答するかどうかを制御します。
デフォルト	1(有効)
範囲	0(無効)、1(有効)
動的か	はい
どのような場合に変更するか	セキュリティー上の理由でこの動作を行いたくない場合、無効にします
コミットレベル	変更の可能性あり

ip_send_redirects と ip6_send_redirects

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

コミットレベル

変更の可能性あり

ip_forward_src_routed と ip6_forward_src_routed

説明

IPv4 または IPv6 が、パケットをソース IPv4 ルーティングオプションを指定して転送するか、IPv6 ルーティングヘッダーを指定して転送するかを制御します。

デフォルト

0(無効)

範囲

0(無効)、1(有効)

動的か

はい

どのような場合に変更するか

サービス妨害攻撃を防ぐためにこのパラメータは無効のままにします。

コミットレベル

変更の可能性あり

変更履歴

詳細は、210 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_addrs_per_if

説明

実インターフェースに対応する論理インター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。論理インター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

説明

非転送インターフェースに到着したパケットを、そのインター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、該当するインターフェースに対し `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)を通過するインターフェースがマシンにある場合は、このパラメータに 1 を設定します。

コミットレベル

変更の可能性あり

ip multidata outbound

説明

ネットワークスタックが、転送時、一度に複数のパケットをネットワークデバイスドライバ宛てに送信できるようにします。

このパラメータを有効にすると、ホスト CPU の利用率またはネットワークスループット(あるいはこの両方)が向上し、パケットあたりの処理コストが減少します。

現在このパラメータは、IP フラグメントを転送するための複数データ送信(MDT)機能を制御します。たとえば、リンク MTU より大きい UDP ペイロードを送信する場合があります。このチューニング可能パラメータが有効になっている場合、UDP などの特定の上位レベルプロトコルの IP フラグメ

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

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

[170 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、 210 ページの「ip_multidata_outbound (Solaris 10 リリース)」 を参照してください。

ip_squeue_fanout

説明	squeue と TCP/IP 接続を関連付けるモードを判定します。
	値 0 の場合、新しい TCP/IP 接続と、この接続を作成した CPU が関連付けられます。値 1 の場合、異なる CPU に属する複数の squeue との接続が関連付けられます。接続をファンアウトするために使用される squeue の数は、 157 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、 210 ページの「ip_squeue_fanout (Solaris 10 11/06 リリース) 」を参照してください。

ip_soft_rings_cnt

説明	受信 TCP/IP 接続を展開するために使用される squeue の数を決定します。
<hr/>	
デフォルト	注-着信トラフィックは、いずれかのリングに置かれます。リングが過負荷になっている場合、パケットは破棄されます。パケットが破棄されるたびに、kstat dls カウンタ <code>dls_soft_ring_pkt_drop</code> が増分されます。
範囲	2
動的か	0 から nCPU。nCPU は、システム内の CPU の最大数です。 いいえ。このパラメータを変更する場合は、インターフェースを再び plumb します。
どのような場合に変更するか	10 Gbps の NIC および多くの CPU が搭載されたシステムでは、このパラメータを 2 より大きい値に設定することを検討してください。
ゾーン構成	このパラメータを設定できるのは、大域ゾーン内だけです。
コミットレベル	廃止
変更履歴	詳細は、 211 ページの「ip_soft_rings_cnt (Solaris 10 11/06 リリース) 」を参照してください。

特別な注意を要する IP チューニング可能パラメータ

次のパラメータの変更は非推奨です。

ip_ire_pathmtu_interval

説明

IP がパス最大転送単位 (PMTU) 検出情報をフラッシュしてから PMTU を再び検出開始するまでの間隔をミリ秒単位で指定します。

PMTU の検出については、RFC 1191 を参照してください。

デフォルト

10 分

範囲

5 秒から 277 時間

動的か

はい

どのような場合に変更するか

この値は変更しないでください。

コミットレベル

変更の可能性あり

ip_icmp_return_data_bytes と ip6_icmp_return_data_bytes

説明

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

デフォルト

64 バイト (ip_icmp_return_data_bytes)

1280 バイト (ip6_icmp_return_data_bytes)

範囲

8 から 65,536 バイト

動的か

はい

どのような場合に変更するか

この値は変更しないでください。ただし、ICMP エラーメッセージに含む情報を増やすとネットワークの問題を診断する上で役立つことがあります。この機能が必要な場合は、値を増やします。

コミットレベル

変更の可能性あり

TCP チューニング可能パラメータ

tcp_deferred_ack_interval

説明

直接接続していないホストに対する TCP 遅延肯定応答(ACK) タイマーのタイムアウト値を指定します。

RFC 1122 の 4.2.3.2 を参照してください。

デフォルト

100 ミリ秒

範囲

1 ミリ秒から 1 分

動的か

はい

どのような場合に変更するか

このパラメータには、500 ミリ秒を超える値を設定しないでください。

次の場合は、値を増やします。

- ネットワークリンクが遅く (57.6 Kbps 未満)、最大セグメントサイズ (MSS) が 512 バイトを超える
- この間隔が複数の TCP セグメントを受信するには短すぎる

コミットレベル

変更の可能性あり

tcp_local_dack_interval

説明

直接接続しているホストに対する TCP 遅延肯定応答(ACK) タイマーのタイムアウト値を指定します。

RFC 1122 の 4.2.3.2 を参照してください。

デフォルト

50 ミリ秒

範囲

10 から 500 ミリ秒

動的か

はい

どのような場合に変更するか

このパラメータには、500 ミリ秒を超える値を設定しないでください。

次の場合は、値を増やします。

- ネットワーククリンクが遅く (57.6 Kbps 未満)、最大セグメントサイズ (MSS) が 512 バイトを超える
- この間隔が複数の TCP セグメントを受信するには短すぎる

コミットレベル

変更の可能性あり

変更履歴

詳細は、211 ページの「[tcp_local_dack_interval \(Solaris 10 リリース\)](#)」を参照してください。

tcp_deferred_acks_max

説明

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

デフォルト

2

範囲

0 から 16

動的か

はい

どのような場合に変更するか

この値は変更しないでください。ただし、遅延 ACK の影響でネットワークトラフィックが著しく混雑するような状況では、この値を減らします。ただし、2 より小さくしないでください。

コミットレベル

変更の可能性あり

tcp_local_dacks_max

説明

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

デフォルト	のセグメントが 1 MSS の長さであるとみなされ、遅延 ACK は使用されなくなります。実際の数は、接続ごとに動的に計算されます。この値はデフォルトの最大値です。
範囲	8
動的か	0 から 16
どのような場合に変更するか	はい
コミットレベル	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。
	変更の可能性あり

tcp_wscale_always

説明

このパラメータが有効になっていると(デフォルトの設定)、ウィンドウスケールオプションの値が0の場合でも、TCP は常にウィンドウスケールオプションを指定して SYN セグメントを送信します。ウィンドウスケールオプションの指定された SYN セグメントを受信すると、パラメータが無効になっている場合でも、TCP は、ウィンドウスケールオプションを指定して SYN セグメントに応答します。オプションの値は受信ウィンドウサイズに従って設定されます。

ウィンドウスケールオプションについては、RFC 1323 を参照してください。

デフォルト	1(有効)
範囲	0(無効)、1(有効)
動的か	はい
どのような場合に変更するか	ウィンドウスケールオプションをサポートしていない古い TCP スタックとの相互運用性の問題がある場合は、このパラメータを無効にしてください。
コミットレベル	変更の可能性あり

tcp_tstamp_always

説明	1 が設定されていると、TCP は常にタイムスタンプオプションを指定して SYN セグメントを送信します。TCP は、タイムスタンプオプションの指定された(0 の場合もある)SYN セグメントを受信すると、タイムスタンプオプションを指定して SYN セグメントに応答します。
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。
コミットレベル	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。

tcp_xmit_hiwat

説明	デフォルトの送信ウィンドウサイズをバイト数で指定します。ルートごとに異なる値を設定する方法については、後述の 192 ページ の「ルート別」のメトリック」を参照してください。 163 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 setsockopt(3XNET) SO_SNDBUF を使って、送信バッファーを接続ごとに変更できます。
コミットレベル	変更の可能性あり

tcp_recv_hiwat

説明

デフォルトの受信ウインドウサイズをバイト数で指定します。ルートごとに異なる値を設定する方法については、後述の[192 ページ](#)の「ルート別のメトリック」を参照してください。[163 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、受信バッファーを接続ごとに変更できます。

コミットレベル

変更の可能性あり

tcp_max_buf

説明

最大バッファー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ーサイズと受信バッファー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、ネットワーククリンクの速度に合わせて値を増やします。

コミットレベル

変更の可能性あり

tcp_cwnd_max

説明

TCP 幅轍ウインドウ (cwnd) の最大値をバイト数で指定します。

TCP 幅轍ウィンドウについては、RFC 1122 と 2581 を参照してください。

デフォルト	1,048,576
範囲	128 から 1,073,741,824
動的か	はい
どのような場合に変更するか	アプリケーションが setsockopt(3XNET) を使用して ウィンドウサイズに <code>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 るべきです。
コミットレベル	変更の可能性あり

tcp_slow_start_initial

説明	幅轍ウィンドウ (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。
	幅轍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 の幅轍を招く場合は、この値を減らします。
コミットレベル	変更の可能性あり

tcp_slow_start_after_idle

説明	幅轍ウィンドウが 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) あとの幅轍ウィンドウのサイズを TCP 接続 の MSS 単位で指定します。
----	--

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、 164 ページ の「 <code>tcp_slow_start_initial</code> 」を参照してください。
コミットレベル	変更の可能性あり

tcp_sack_permitted

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

tcp_rev_src_routes

説明	0が設定されていると、TCPは、セキュリティー上の理由により、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

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

tcp_ecn_permitted

説明	ECN (Explicit Congestion Notification、明示的輻輳通知) のサポートを制御します。
	このパラメータが0に設定されていると、TCPは、ECN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もあります。混乱するデバイスは IETF 非準拠です。

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

コミットレベル

変更の可能性あり

tcp_conn_req_max_q

説明

[accept\(3SOCKET\)](#) によって受け付けられるのを待っている TCP リスナーの、保留状態の TCP 接続のデフォルトの最大数を指定します。[168 ページ](#) の「[tcp_conn_req_max_q0](#)」も参照してください。

デフォルト	128
範囲	1 から 4,294,967,295
動的か	はい
どのような場合に変更するか	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 <code>tcp_conn_req_max_q</code> 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

tcp_conn_req_max_q0

説明	单一の TCP リスナーが持つことができる、不完全な(3段階ハンドシェークがまだ終わっていない)保留状態の TCP 接続のデフォルトの最大数を指定します。
	TCP の 3段階ハンドシェークについては、RFC 793 を参照してください。 167 ページ の「 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 4,294,967,295
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。そうでなければ、その要求は受け付けられます。それ以外の場合、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

tcp_conn_req_min

説明	受け付けられるのを待っている、单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 listen(3SOCKET) 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、 listen(3SOCKET) を使用するアプリケーションが保留状態の TCP 接続の最大数を過度に小さく設定するのを防ぐことができます。この値は、着信接続要求の頻度に応じて増やすことができます。
コミットレベル	変更の可能性あり

tcp_rst_sent_rate_enabled

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

tcp_rst_sent_rate

説明	TCP が 1 秒間に送信できる最大 RST セグメント数を設定します。
デフォルト	40
範囲	0 から 4,294,967,295
動的か	はい
どのような場合に変更するか	TCP 環境では、正当な理由により、デフォルト値より多くの RST が生成される場合があります。このような場合は、このパラメータのデフォルト値を引き上げます。
コミットレベル	変更の可能性あり

tcp_mdt_max_pbufs

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

	を指定します。 155 ページ の「 ip_multidata_outbound 」も参照してください。
デフォルト	16
範囲	1 から 16
動的か	はい
どのような場合に変更するか	このパラメータを減少させると、TCP によって生成される M_MULTIDATA メッセージごとのペイロードバッファーの量を制限し、デバイスドライバ開発のデバッグに役立つことがあります。
コミットレベル	変更の可能性あり

tcp_naglim_def

説明	このパラメータは、Nagle アルゴリズムのしきい値を制御します。TCP は、このパラメータの最小値と接続の MSS を使用して、Nagle アルゴリズムが機能するタイミングを判断します。たとえば、新しいデータの量が 1 MSS を超える場合、そのデータはこのパラメータの値に関係なく送出されます。このパラメータが 1 に設定されている場合、すべての TCP 接続で Nagle は無効になります。
デフォルト	4,096
範囲	1 から 65,535
動的か	はい
どのような場合に変更するか	データを遅延なしに送信する必要があるリアルタイムのアプリケーションでは、高速伝送を必要とするソケットに対して、 <code>tcp_naglim_def</code> パラメータを設定するのではなく、 <code>setsockopt()</code> を使って <code>TCP_NODELAY</code> を 1 に設定するようにしてください。
コミットレベル	変更の可能性あり
変更履歴	詳細は、 211 ページ の「 tcp_naglim_def (Solaris 10 リース) 」を参照してください。

tcp_smallest_anon_port

説明

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

単位

ポート番号

デフォルト

32,768

範囲

1,024 から 65,534

動的か

はい

どのような場合に変更するか

より広範囲の一時的なポートが必要な場合。

コミットレベル

変更の可能性あり

変更履歴

詳細は、211 ページ

の「[tcp,sctp,udp]_smallest_anon_port と
[tcp,sctp,udp]_largest_anon_port (Solaris 10 リリース)」を参照してください。

tcp_largest_anon_port

説明

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

単位

ポート番号

デフォルト

65,535

範囲

1,024 から 65,535

動的か

はい

どのような場合に変更するか

より広範囲の一時的なポートが必要な場合。

コミットレベル	変更の可能性あり
変更履歴	詳細は、 211 ページ の「[tcp,sctp,udp]_smallest_anon_port と [tcp,sctp,udp]_largest_anon_port (Solaris 10 リ ース)」を参照してください。

/etc/system ファイルに設定する TCP/IP パラメータ

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

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

```
set ip:ipcl_conn_hash_sizes=value
```

ipcl_conn_hash_size

説明

IP によって使用される接続ハッシュテーブルのサ
イズを制御します。デフォルト値 0 の場合、シス
テムは、使用可能なメモリー容量に基づいて、ブート
時にこのパラメータの適切な値を自動的に決定しま
す。

データ型

符号なし整数

デフォルト

0

範囲

0 から 82,500

動的か

いいえ。このパラメータはブート時にのみ変更でき
ます。

どのような場合に変更するか

常に非常に大量の TCP 接続が確立されているシ
ステムでは、それに応じてこの値を増やすことが可
れます。ハッシュテーブルサイズを大きくすると、よ
り多くのメモリーが消費され、ユーザーアプリ
ケーション用として提供できるメモリーの量が減り
ます。

コミットレベル

変更の可能性あり

ip_squeue_worker_wait

説明

TCP/IP パケットを squeue に格納して処理する
ワーカスレッドが起動するまでの最大遅延時間を制

御します。*squeue* 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、211 ページの「 ip_squeue_worker_wait (Solaris 10 11/06 リリース) 」を参照してください。

特別な注意を要する TCP パラメータ

次のパラメータの変更は非推奨です。

tcp_keepalive_interval

説明

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

Oracle Solaris は、RFC 1122 に説明されている TCP キープアライブメカニズムをサポートしています。この機構は、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るべき、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、不必要的ネットワークトラフィックが生じる可能性があるとともに、ネットワークの一時的な問題のために未完了のまま接続が終了してしまう可能性も高くなります。
コミットレベル	変更の可能性あり

tcp_ip_abort_interval

説明

TCP 接続に対するデフォルトの合計再送タイムアウト値をミリ秒で指定します。ある TCP 接続で、TCP が `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「tcp_rexmit_interval_max」](#) を参照してください。

コミットレベル

変更の可能性あり

tcp_rexmit_interval_initial

説明

TCP 接続に対するデフォルトの初期再送タイムアウト値をミリ秒で指定します。ルートごとに異なる値を設定する方法については、後述の[192 ページの「ルート別のメトリック」](#) を参照してください。

デフォルト

3 秒

範囲

1 ミリ秒から 20 秒

動的か

はい

どのような場合に変更するか

この値は変更しないでください。この値を小さくすると、不要な再転送が行われるおそれがあります。

コミットレベル

変更の可能性あり

tcp_rexmit_interval_max

説明

デフォルトの最大再送タイムアウト値 (RTO) をミリ秒で指定します。すべての TCP 接続に対して、計算された RTO がこの値を超えることはできません。[175 ページの「tcp_ip_abort_interval」](#) も参照してください。

デフォルト

60 秒

範囲

1 ミリ秒から 2 時間

動的か

はい

どのような場合に変更するか

通常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。

单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 `tcp_ip_abort_interval` パラメータも変更するべきです。`tcp_ip_abort_interval` には、`tcp_rexmit_interval_max` の 4 倍以上の値を指定します。

コミットレベル

変更の可能性あり

tcp_rexmit_interval_min

説明

デフォルトの最小再送タイムアウト値 (RTO) をミリ秒で指定します。すべての TCP 接続に対して、計算された RTO がこの値を下回ることはできません。[176 ページ](#) の「[tcp_rexmit_interval_max](#)」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 [tcp_rexmit_interval_max](#) パラメータも変更するべきです。[tcp_rexmit_interval_max](#) には、[tcp_rexmit_interval_min](#) の 8 倍以上の値を指定します。

コミットレベル

変更の可能性あり

tcp_rexmit_interval_extra

説明

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

デフォルト

0 ミリ秒

範囲

0 から 2 時間

動的か

はい

どのような場合に変更するか

この値は変更しないでください。

計算された RTO が接続に対して適切でない場合は、不要な再転送を避けるためにこの値を変更することができます。

コミットレベル

変更の可能性あり

tcp_tstamp_if_wscale

説明

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

デフォルト

1(有効)

範囲

0(無効)、1(有効)

動的か

はい

どのような場合に変更するか

この値は変更しないでください。一般に、TCP を高速ネットワークで使用する場合は、シーケンス番号のラップアラウンドに対する保護が必要になります。この場合、timestamp オプションが必要になります。

コミットレベル

変更の可能性あり

tcp_recv_hiwat_minmss

説明

デフォルトの最小受信ウィンドウサイズを制御します。最小値は、tcp_recv_hiwat_minmss に、接続の最大セグメントサイズ (MSS) を掛けた値です。

デフォルト

4

範囲

1 から 65,536

動的か

はい

どのような場合に変更するか

この値は変更しないでください。この値を変更する必要がある場合は、4 より小さい値にしないでください。

コミットレベル

変更の可能性あり

UDP チューニング可能パラメータ**udp_xmit_hiwat**

説明

デフォルトの最大 UDP ソケットデータグラムサイズをバイト数で指定します。詳細は、[181 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) <code>SO_SNDBUF</code> を使用してソケットごとに変更できます。一般には、デフォルト値を変更する必要はありません。
コミットレベル	変更の可能性あり

udp_recv_hiwat

説明	デフォルトの最大 UDP ソケット受信バッファーサイズをバイト数で指定します。詳細は、 181 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) <code>SO_RCVBUF</code> を使用してソケットごとに変更できます。一般には、デフォルト値を変更する必要はありません。
コミットレベル	変更の可能性あり

udp_smallest_anon_port

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

範囲	1,024 から 65,534
動的か	はい
どのような場合に変更するか	より広範囲の一時的なポートが必要な場合。
コミットレベル	変更の可能性あり
変更履歴	詳細は、211 ページ の「[tcp,sctp,udp]_smallest_anon_port と [tcp,sctp,udp]_largest_anon_port (Solaris 10 リ リース)」を参照してください。

udp_largest_anon_port

説明	このパラメータは、UDP が一時的なポートとして選択できる最大ポート番号を制御します。アプリケーションでは、指定されたプロトコルを使って接続を作成する際に、ポート番号を指定しないで一時的なポートを使用することができます。一時的なポートは、特定のアプリケーションに関連付けられていません。接続が閉じられると、ポート番号を別のアプリケーションで再利用することができます。
単位	ポート番号
デフォルト	65,535
範囲	1,024 から 65,535
動的か	はい
どのような場合に変更するか	より広範囲の一時的なポートが必要な場合。
コミットレベル	変更の可能性あり
変更履歴	詳細は、211 ページ の「[tcp,sctp,udp]_smallest_anon_port と [tcp,sctp,udp]_largest_anon_port (Solaris 10 リ リース)」を参照してください。

udp_do_checksum

説明	このパラメータは、送信される UDP/IPv4 パケットのチェックサムを計算するかどうかを制御します。
デフォルト	1(有効)

範囲	0(無効)、1(有効)
動的か	はい
どのような場合に変更するか	このパラメータは変更しないでください。
コミットレベル	変更の可能性あり
変更履歴	詳細は、 212 ページの「udp_do_checksum (Solaris 10 リリース)」 を参照してください。

特別な注意を要する UDP パラメータ

次のパラメータの変更は非推奨です。

udp_max_buf

説明	UDP ソケット用の送信バッファーと受信バッファーのサイズを制御します。
デフォルト	2,097,152 バイト
範囲	65,536 から 1,073,741,824 バイト
動的か	はい
どのような場合に変更するか	この値は変更しないでください。このパラメータの値が大きすぎると、UDP ソケットアプリケーションがメモリーを過剰に使用するおそれがあります。
コミットレベル	変更の可能性あり

IPQoS チューニング可能パラメータ

ip_policy_mask

説明	IPQoS 処理を有効または無効にします。有効または無効にするコールアウト位置は、次のとおりです。転送送信、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 处理をマスク、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 处理を無効にします。

デフォルト	0(すべてのコールアウト位置で IPQoS 处理が有効)
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 处理が無効
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 处理を有効または無効にしたい場合
コミットレベル	変更の可能性あり

SCTP チューニング可能パラメータ

sctp_max_init_retr

説明	SCTP 終端が INIT チャンクの再送信位置で行う最大試行回数を制御します。SCTP 終端は、SCTP 設定構造で、この値を上書きできます。
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 183 ページの「 sctp_pa_max_retr 」に依存します。 <code>sctp_max_init_retr</code> が <code>sctp_pa_max_retr</code> 以下であれば理想的です。
コミットレベル	変更の可能性あり

sctp_pa_max_retr

説明

SCTP 接続のすべてのパスを経由する最大再送回数を制御します。この値を超過すると、SCTP 接続は終了します。

デフォルト

10

範囲

1 から 128

動的か

はい

どのような場合に変更するか

すべてのパスを経由する最大再送回数は、パスの数と各パスの最大再送回数によって決定されます。[sctp_pa_max_retr](#) は、使用可能なすべてのパスの 183 ページの「[sctp_pp_max_retr](#)」の合計に設定するべきです。たとえば、宛先までのパス数が 3 で、これらのパスの最大再送回数がそれぞれ 5 回である場合、[sctp_pa_max_retr](#) には 15 以下の値を設定するべきです (RFC 2960、Section 8.2 の Note を参照)。

コミットレベル

変更の可能性あり

sctp_pp_max_retr

説明

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

デフォルト

5

範囲

1 から 128

動的か

はい

どのような場合に変更するか

この値を 5 より小さい値に変更しないでください。

コミットレベル

変更の可能性あり

sctp_cwnd_max

説明

SCTP 接続の輻輳ウィンドウの最大値を制御します。

デフォルト

1,048,576

範囲

128 から 1,073,741,824

動的か	はい
どのような場合に変更するか	アプリケーションが setsockopt(3XNET) 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、187 ページの「 sctp_max_buf 」は <code>sctp_cwnd_max</code> より大きくするべきです。
コミットレベル	変更の可能性あり

sctp_ipv4_ttl

説明	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	64
範囲	1 から 255
動的か	はい
どのような場合に変更するか	通常、この値を変更する必要はありません。宛先バ スが 64 ホップを超過しそうな場合は、このパラ メータの値を大きくします。
コミットレベル	変更の可能性あり

sctp_heartbeat_interval

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

sctp_new_secret_interval

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

sctp_initial_mtu

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

sctp_deferred_ack_interval

説明	SCTP 遅延肯定応答 (ACK) タイマーのタイムアウト値をミリ秒で設定します。
デフォルト	100 ミリ秒
範囲	1 から 60,000 ミリ秒
動的か	はい
どのような場合に変更するか	RFC 2960 の 6.2 を参照してください。
コミットレベル	変更の可能性あり

sctp_ignore_path_mtu

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

sctp_initial_ssthresh

説明	相手側の宛先アドレスの初期スロースタートしきい値を設定します。
デフォルト	102,400
範囲	1024 から 4,294,967,295
動的か	はい
どのような場合に変更するか	RFC 2960 の 7.2.1 を参照してください。
コミットレベル	変更の可能性あり

sctp_xmit_hiwat

説明	デフォルトの送信ウィンドウサイズをバイト数で指定します。187 ページの「 sctp_max_buf 」も参照してください。
デフォルト	102,400
範囲	8,192 から 1,073,741,824
動的か	はい
どのような場合に変更するか	アプリケーションは、 getsockopt(3SOCKET) <code>SO_SNDBUF</code> を使って、個々の接続の送信バッファーを変更できます。
コミットレベル	変更の可能性あり

sctp_xmit_lowat

説明	送信ウィンドウサイズの下限値を制御します。
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ーを書き込み可能にするために必要な最小限のサイズを設定します。必要に応じて、 186 ページ の「 sctp_xmit_hiwat 」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_recv_hiwat

説明	デフォルトの受信ウィンドウサイズをバイト数で指定します。 187 ページ の「 sctp_max_buf 」も参照してください。
デフォルト	102,400
範囲	8,192 から 1,073,741,824
動的か	はい
どのような場合に変更するか	アプリケーションは、 getsockopt(3SOCKET) <code>SO_RCVBUF</code> を使って、個々の接続の受信バッファーを変更できます。
コミットレベル	変更の可能性あり

sctp_max_buf

説明	最大バッファー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 getsockopt(3SOCKET) を使用して設定する送信バッファーおよび受信バッファー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、ネットワーククリンク速度に合わせて、このパラメータの値を大きくします。

コミットレベル

変更の可能性あり

sctp_ipv6_hoplimit

説明

SCTP 接続上で、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 ホップを超過しそうな場合は、このパラメータの値を大きくします。

コミットレベル

変更の可能性あり

sctp_rto_min

説明

相手側のすべての宛先アドレスの再送タイムアウト (RTO) の下限値をミリ秒で設定します。

デフォルト

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

説明

相手側のすべての宛先アドレスの再送タイムアウト (RTO) の上限値をミリ秒で設定します。

デフォルト

60,000

範囲

1,000 から 60,000,000

動的か

はい

どのような場合に変更するか コミットレベル	RFC 2960 の 6.3.1 を参照してください。 変更の可能性あり
--------------------------	---

sctp_rto_initial

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

sctp_cookie_life

説明	Cookie の寿命をミリ秒で設定します。
デフォルト	60,000
範囲	10 から 60,000,000
動的か	はい
どのような場合に変更するか コミットレベル	通常、この値を変更する必要はありません。このパラメータは、 188 ページの「sctp_rto_max」 に合わせて変更できます。 変更の可能性あり

sctp_max_in_streams

説明	SCTP 接続 1 個あたりに許可された最大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 コミットレベル	RFC 2960 の 5.1.1 を参照してください。 変更の可能性あり

sctp_initial_out_streams

説明	SCTP 接続 1 個あたりに許可された最大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

説明	SHUTDOWN チャンクの送信後、SHUTDOWN ACK を待機する最大待ち時間をミリ秒で制御します。
デフォルト	60,000
範囲	0 から 300,000
動的か	はい
どのような場合に変更するか	通常、この値を変更する必要はありません。このパラメータは、 188 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_maxburst

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

sctp_addip_enabled

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

sctp_prsctp_enabled

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

sctp_smallest_anon_port

説明	このパラメータは、SCTP が一時的なポートとして選択できる最小ポート番号を制御します。アプリケーションでは、指定されたプロトコルを使って接続を作成する際に、ポート番号を指定しないで一時的なポートを使用することができます。一時的なポートは、特定のアプリケーションに関連付けられていません。接続が閉じられると、ポート番号を別のアプリケーションで再利用することができます。
単位	ポート番号
デフォルト	32,768
範囲	1,024 から 65,534

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

sctp_largest_anon_port

説明	このパラメータは、SCTPが一時的なポートとして選択できる最大ポート番号を制御します。アプリケーションでは、指定されたプロトコルを使って接続を作成する際に、ポート番号を指定しないで一時的なポートを使用することができます。一時的なポートは、特定のアプリケーションに関連付けられていません。接続が閉じられると、ポート番号を別のアプリケーションで再利用することができます。
単位	ポート番号
デフォルト	32,768
範囲	1,024 から 65,534
動的か	はい
どのような場合に変更するか	より広範囲の一時的なポートが必要な場合。
コミットレベル	変更の可能性あり

ルート別のメトリック

ルート別のメトリックを使用して、一定のプロパティーを 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ーフェースがあるとします。`tcp_recv_hiwat` のシステムデフォルト値は 49,152 バイトです。このデフォルト値は Fast Ethernet インタフェースには十分ですが、Gigabit Ethernet インタフェースには不十分な可能性があります。

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

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

192.123.123.0	192.123.123.4	U	1	4	hme0
192.123.124.0	192.123.124.4	U	1	4	ge0
default	192.123.123.1	UG	1	8	

この例では、次の処理が行われます。

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

その結果、*ge0* リンク上にある 192.123.124.0 ネットワークへのすべての接続には、デフォルトの受信ウィンドウサイズ 49,152 の代わりに、受信バッファーサイズ *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ーサイズ *y* を使用します。複数のインターフェースがある場合は、-ifp 引数を使用して、使用するインター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ー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。

- 196 ページの「nca:nca_conn_hash_size」
- 196 ページの「nca:nca_conn_req_max_q」
- 197 ページの「nca:nca_conn_req_max_q0」
- 197 ページの「nca:nca_ppmax」
- 198 ページの「nca:nca_vpmax」
- 199 ページの「sq_max_size」
- 199 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
Oracle Solaris カーネルのチューニング可能パラメータ	第 2 章 「Oracle Solaris カーネルチューニング可能パラメータ」
NFS チューニング可能パラメータ	第 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、[27 ページの「Oracle 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

説明	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、システムパフォーマンス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

説明	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

説明

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

説明

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容量(ページ数)を指定します。この値は、合計メモリ容量の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

512Mバイトを超えるメモリーを搭載したシステムでNCAを使用する場合。システムの未使用物理メモリ容量が多い場合は、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

キャッシングに格納できます。その結果、システムパフォーマンスが向上します。

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` と併せて(と連動して)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

説明

NCA がページをキャッシング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

512M バイトを超えるメモリーを搭載したシステムで 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ングに格納できます。その結果、システムパフォーマンスが向上します。

このパラメータの値は `nca_ppmax` に合わせて大きくなります。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 のパフォーマンス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、インターフェースを割り込みモードにするとパフォーマンスが向上します。

たとえば、64 ビットカーネルでブートされる 4G バイトのメモリーを搭載したシステムでは、/etc/system ファイル内に次のパラメータを設定するようにします。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

説明	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。
コミットレベル	変更の可能性あり

ge:ge_intr_mode

説明	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。

- 202 ページの「autofs」
- 202 ページの「cron」
- 202 ページの「devfsadm」
- 202 ページの「dhcpagent」
- 202 ページの「fs」
- 203 ページの「ftp」
- 203 ページの「inetinit」
- 203 ページの「init」
- 203 ページの「ipsec」
- 203 ページの「kbd」
- 203 ページの「keyserv」
- 203 ページの「login」
- 204 ページの「lu」
- 204 ページの「mpathd」
- 204 ページの「nfs」
- 204 ページの「nfslogd」
- 204 ページの「nss」
- 204 ページの「passwd」
- 204 ページの「power」
- 204 ページの「rpc.nisd」
- 205 ページの「su」
- 205 ページの「syslog」
- 205 ページの「sys-suspend」
- 205 ページの「tar」
- 205 ページの「telnetd」
- 205 ページの「utmpd」
- 206 ページの「yppasswdd」

システムのデフォルトのパラメータ

さまざまなシステム機能の動作は、その機能が起動時に読み込む一連の値によって制御されます。各機能について1つのファイルに保存された値は、`/etc/default`ディレクトリに置かれます。ただし、すべてのシステム機能がこのディレクトリにファイルを持っているわけではありません。

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。

dhcpagent

DHCP のクライアント使用率は、dhcpagent 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ー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ーフェースを管理します。

詳細は、[dhcpagent\(1M\)](#) の「FILES」セクションにある `/etc/default/dhcpagent` の情報をお読みください。

fs

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

ftp

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

詳細は、[ftp\(4\)](#) を参照してください。

inetinit

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

init

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

このファイルのすべての値は、シングルユーザーのブート要求に応じて `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。

ipsec

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

kbd

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

keyserv

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

login

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

lu

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

mpathd

この機能を利用して、[in.mpathd](#)構成パラメータを設定できます。

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

nfs

この機能を利用して、NFS デーモン構成パラメータを設定できます。

詳細は、[nfs\(4\)](#) のマニュアルページを参照してください。

nfslogd

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

nss

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

詳細は、[nss\(4\)](#) のマニュアルページを参照してください。

passwd

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

power

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

rpc.nisd

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

su

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

syslog

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

sys-suspend

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

tar

-f 関数修飾子の詳細は、[tar\(1\)](#) のマニュアルページを参照してください。

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

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。

telnetd

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

utmpd

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

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

- SCAN_PERIOD - utmpd が、監視するプロセスがまだ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 - r nis -e コマンドの使用時に、制限されたシェルにログインシェルを設定できるかどうかを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

この章には、特定のチューニング可能パラメータの変更履歴を記載します。この項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。

- [207 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」](#)
- [210 ページの「TCP/IP チューニング可能パラメータ」](#)
- [212 ページの「廃止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 10 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。

一般的なドライバパラメータ

ddi_msix_alloc_limit (Solaris 10 リリース)

Solaris 10 10/09 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、[69 ページの「ddi_msix_alloc_limit」](#)を参照してください。

一般的な入出力のチューニング可能パラメータ

maxphys (Solaris 10 リリース)

デフォルト値が sun4v システムを含むように更新されています。詳細は、[70 ページ](#) の「[maxphys](#)」を参照してください。

カーネルとメモリーの一般的なパラメータ

zfs_arc_min (Solaris 10 リリース)

このパラメータに関する記述は Solaris 10 10/09 リリースで新しく追加されました。詳細は、[35 ページ](#) の「[zfs_arc_min](#)」を参照してください。

zfs_arc_max (Solaris 10 リリース)

このパラメータに関する記述は Solaris 10 10/09 リリースで新しく追加されました。詳細は、[35 ページ](#) の「[zfs_arc_max](#)」を参照してください。

noexec_user_stack (Solaris 10 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 10 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。

fsflush とそれに関連するパラメータ

dopageflush (Solaris 10 リリース)

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

ページング関連のチューニング可能パラメータ

maxpgio (Solaris 10 リリース)

Solaris 10 バージョンでは、値の範囲が誤って 1 から 1024 と記述されていました。実際の範囲は、システムのアーキテクチャーと入出力サブシステムによって決まります。詳細は、[63 ページの「maxpgio」](#) を参照してください。

一般的なファイルシステムパラメータ

ncsize (Solaris 10 リリース)

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

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

正しいデフォルト値は次のとおりです。

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

詳細は、[73 ページの「ncsize」](#) を参照してください。

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

SPARC システム固有のパラメータ (Solaris 10 リリース)

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

default_tsb_size (Solaris 10 リリース)

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

enable_tsb_rss_sizing (Solaris 10 リリース)

説明とデフォルト値および範囲の値が変更されました。詳細は、[98 ページ](#) の「[enable_tsb_rss_sizing](#)」を参照してください。

tsb_rss_factor (Solaris 10 リリース)

「どのような場合に変更するか」の記述例が次のように変更されました。

たとえば、`tsb_rss_factor` を 384(事実上は 75%) ではなく 256(事実上は 50%) に変更すると、状況によっては、TSB における仮想アドレスの衝突を排除できることがあります、特に負荷の大きいシステムでは、カーネルメモリーの使用量が増えます。

NFS チューニング可能パラメータ

nfs:nfs3_nra (Solaris 10 リリース)

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

TCP/IP チューニング可能パラメータ

ip_forward_src_routed と ip6_forward_src_routed (Solaris 10 リリース)

以前の Solaris 10 リリースでは、これらのパラメータのデフォルト値が誤って記述されていました。正しいデフォルト値は無効です。詳細は、[154 ページ](#) の「[ip_forward_src_routed と ip6_forward_src_routed](#)」を参照してください。

ip_multidata_outbound (Solaris 10 リリース)

このパラメータは、Solaris 10 リリースで、IP フラグメントをバッチ内でネットワークドライバに送信するように拡張されました。詳細は、[155 ページ](#) の「[ip_multidata_outbound](#)」を参照してください。

ip_sqeue_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、[156 ページ](#) の「[ip_sqeue_fanout](#)」を参照してください。

ip_squeue_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、[173 ページ](#)の「[ip_squeue_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、[157 ページ](#)の「[ip_soft_rings_cnt](#)」を参照してください。

ip_squeue_write (Solaris 10 リリース)

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

tcp_local_dack_interval (Solaris 10 リリース)

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

[tcp,sctp,udp]_smallest_anon_port と [tcp,sctp,udp]_largest_anon_port (Solaris 10 リリース)

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

- [191 ページ](#)の「[sctp_smallest_anon_port](#)」
- [192 ページ](#)の「[sctp_largest_anon_port](#)」
- [172 ページ](#)の「[tcp_smallest_anon_port](#)」
- [172 ページ](#)の「[tcp_largest_anon_port](#)」
- [179 ページ](#)の「[udp_smallest_anon_port](#)」
- [180 ページ](#)の「[udp_largest_anon_port](#)」

tcp_naglim_def (Solaris 10 リリース)

Solaris 10 8/11 リリースでは、[171 ページ](#)の「[tcp_naglim_def](#)」パラメータの説明が新たに追加されました。

udp_do_checksum (Solaris 10 リリース)

Solaris 10 8/11 リリースでは、180 ページの「[udp_do_checksum](#)」パラメータの説明が新たに追加されました。

廃止または削除されたパラメータ

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

System V メッセージキュー パラメータ

msgsys:msginfo_msgmni

Solaris 10 リリースでは廃止。

説明	作成できるメッセージキューの最大数
データ型	符号付き整数
デフォルト	50
範囲	0 から MAXINT
動的か	いいえ。msginfo 構造体の msgmni フィールドにロードされます。
検査	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_msqtql

Solaris 10 リリースでは廃止。

説明	作成できるメッセージの最大数。msgsnd(2) 呼び出しでこの限度を超える要求が行われた場合は、メッセージヘッダーが使用可能になるまで要求は延期されます。あるいは、その要求で 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 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_msgmnb

Solaris 10 リリースでは廃止。

説明	1 つのメッセージキューに含めることのできる最大バイト数
データ型	符号なし long
デフォルト	4096
範囲	0 から物理メモリーの総量
単位	バイト
動的か	いいえ。msginfo 構造体の msgmnb 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_msgssz

Solaris 10 リリースで削除。

説明	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	廃止

msgsys:msginfo_msgmap

Solaris 10 リリースで削除。

説明	システムがサポートするメッセージの数。
データ型	符号付き整数
デフォルト	100
範囲	0 から MAXINT
動的か	いいえ
検査	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	廃止

msgsys:msginfo_msgseg

Solaris 10 リリースで削除。

説明

使用可能なメッセージメモリーのプールとしてシステムが使用する、`msginfo_msgssz` セグメントの数。メッセージに使用できる合計メモリーは `msginfo_msgseg * msginfo_msgssz` です。

データ型

符号付き short

デフォルト

1024

範囲

0 から 32,767

動的か

いいえ

検査

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ります。

どのような場合に変更するか

デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。

コミットレベル

廃止

msgsys:msginfo_msgmax

Solaris 10 リリースで削除。

説明

System V メッセージの最大サイズ

データ型

符号なし long

デフォルト

2048

範囲

0 から物理メモリーの総量

単位

バイト

動的か

いいえ。`msginfo` 構造体の `msgmax` フィールドにロードされます。

検査

なし

どのような場合に変更するか コミットレベル	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。
	変更の可能性あり

System V セマフォーのパラメータ

semsys:seminfo_semmni

Solaris 10 リリースでは廃止。

説明	セマフォー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、またはその両方に表示されます。

どのような場合に変更するか コミットレベル	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/oracle/system/vsemmsl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。

semsys:seminfo_semmsl

Solaris 10 リリースでは廃止。

説明	セマフォー識別子あたりの System V セマフォー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	セマフォー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、セマフォー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ー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 <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_semopm

Solaris 10 リリースでは廃止。

説明	<code>semop</code> 呼び出しあたりの System V セマフォー操作の最大数を指定します。このパラメータは、 <code>semop()</code> システムコールで使用する <code>sops</code> 配列内の <code>sembufs</code> の数を参照します。詳細は、 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	セマフォー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、セマフォー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 `semop` 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ー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 `semop()` 呼び出しから `E2BIG` というリターンコードを受け取ります。

コミットレベル

変更の可能性あり

semsys:seminfo_semmns

Solaris 10 リリースで削除。

説明

システム上の System V セマフォーの最大数

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

セマフォー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、セマフォー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ー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ー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 10 リリースで削除。

説明

System V セマフォーシステムがサポートする取り消し構造体の総数

データ型	符号付き整数
デフォルト	30
範囲	1 から MAXINT
動的か	いいえ
検査	セマフォー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、セマフォー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_semume

説明	Solaris 10 リリースで削除。
データ型	1 つのプロセスで使用できる System V セマフォー取り消し構造体の最大数
デフォルト	符号付き整数
範囲	10
動的か	1 から MAXINT
検査	いいえ
	セマフォー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、セマフォー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。[semop\(2\)](#)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_semvmx

Solaris 10 リリースで削除。

説明

セマフォーを設定できる最大値

データ型

符号なし short

デフォルト

32,767

範囲

1 から 65,535

動的か

いいえ

検査

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。[semop\(2\)](#)呼び出しからアプリケーションにリターンコード `ERANGE` 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 10 リリースで削除。

説明

取り消し構造体のセマフォー値を設定できる最大値

データ型

符号なし short

デフォルト

16,384

範囲

1 から 65,535

動的か

いいえ

検査

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。[semop\(2\)](#)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

System V 共有メモリーパラメータ

shmsys:shminfo_shmmni

Solaris 10 リリースでは廃止。

説明	作成できる共有メモリーセグメント数に関するシステム全体の制限。
データ型	符号付き整数
デフォルト	100
範囲	0 から MAXINT
動的か	いいえ。 <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_shmmmax

Solaris 10 リリースでは廃止。

説明	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と、`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。 <code>shminfo</code> 構造体の <code>shmmmax</code> 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

このマニュアルの改訂履歴

この項では、このマニュアルの改訂履歴について説明しています。

- [223 ページの「最新バージョン: Oracle Solaris 10 8/11 リリース」](#)
- [223 ページの「Oracle Solaris リリースで追加または変更されたパラメータ」](#)

最新バージョン: Oracle Solaris 10 8/11 リリース

このマニュアルの最新バージョンは、Oracle Solaris 10 8/11 リリースに該当します。

Oracle Solaris リリースで追加または変更されたパラメータ

次に、追加、変更、または廃止されたカーネルチューニング可能パラメータについて説明します。

- **Oracle Solaris 10 8/11:** このリリースには `ngroups_max` パラメータの説明が含まれています。詳細は、[50 ページの「`ngroups_max`」](#) を参照してください。
- **Solaris 10 10/09:** このリリースには、`zfs_arc_min` および `zfs_arc_max` パラメータに関する記述が含まれます。詳細は、[35 ページの「`zfs_arc_min`」](#) および [35 ページの「`zfs_arc_max`」](#) を参照してください。
- **Solaris 10 10/09:** このリリースには `ddi_msix_alloc_limit` パラメータが含まれます。このパラメータは、デバイスインスタンスが割り当てることのできる MSI-X 割り込みの数を増やすために使用できます。詳細は、[69 ページの「`ddi_msix_alloc_limit`」](#) を参照してください。
- **Solaris 10 10/09:** このリリースでは、メモリー近傍性グループのパラメータが提供されます。これらのパラメータについての詳細は、[100 ページの「近傍性グループのパラメータ」](#) を参照してください。

- **Solaris 10 5/08:** 95 ページの「SPARC システム固有のパラメータ」節の変換格納バッファーのパラメータがより適切な情報を提供するように改訂されました。このリリースでは、次のパラメータが変更されています。
 - 98 ページの「`default_tsb_size`」 - デフォルトの文が明確にされました。
 - 98 ページの「`enable_tsb_rss_sizing`」 - デフォルトの文が誤っていたため、訂正されています。
 - 99 ページの「`tsb_rss_factor`」 - 例に関する部分が、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 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`」
 - 85 ページの「`tmpfs:tmpfs_maxkmem`」
 - 95 ページの「SPARC システム固有のパラメータ」

索引

数字・記号

, 35

dopageflush, 44, 208

A

autofs, 202

autooup, 43

E

enable_tsb_rss_sizing, 98

B

bufhwm, 77

bufhwm_pct, 77

F

fastscan, 59

freebehind, 84

fs, 202

fsflush, 41

ftp, 203

C

consistent_coloring, 95

cron, 202

G

ge_intr_mode, 199

D

ddi_msix_alloc_limit パラメータ, 69

default_stksize, 36

default_tsb_size, 98

desfree, 53

dhcpagent, 202

dnlc_dir_enable, 75

dnlc_dir_max_size, 76

dnlc_dir_min_size, 75

doiflush, 45

H

handspreadpages, 61

hires_tick, 94

I

inetinit, 203

init, 203

ip_addrs_per_if, 154

ip_forward_src_routed, 154
 ip_icmp_err_burst, 152
 ip_icmp_err_interval, 152
 ip_icmp_return_data_bytes, 158
 ip_ire_pathmtu_interval, 158
 ip_multidata_outbound, 156
 ip_policy_mask, 181
 ip_respond_to_echo_broadcast, 153
 ip_send_redirects, 153
 ip_soft_rings_cnt, 157
 ip_squeue_fanout, 156
 ip_squeue_worker_wait, 173
 ip_strict_dst_multihoming, 155
 ip6_forward_src_routed, 154
 ip6_icmp_return_data_bytes, 158
 ip6_respond_to_echo_multicast, 153
 ip6_send_redirects, 153
 ip6_strict_dst_multihoming, 155
 ipcl_conn_hash_size, 173
 ipsec, 203

K

kbd, 203
 keyserv, 203
 kmem_flags, 66

L

lgrp_mem_pset_aware, 102
 logevent_max_q_sz, 39
 login, 203
 lotsfree, 52
 lpg_alloc_prefer, 100
 lpg_mem_default_policy, 101
 lu, 204
 lwp_default_stksize, 37

M

max_nprocs, 49, 207, 208
 maxpio, 63, 209

maxphys, 70
 maxpid, 48
 maxuprc, 50
 maxusers, 46
 md_mirror:md_resync_bufsz, 103
 md:mirrored_root_flag, 104
 min_percent_cpu, 61
 minfree, 55
 moddebug, 68
 mpathd, 204
 msgsys:msginfo_msgmax, 215
 msgsys:msginfo_msgrnb, 213
 msgsys:msginfo_msgrnbi, 212
 msgsys:msginfo_msgrnseg, 215
 msgsys:msginfo_msgrnssz, 213
 msgsys:msginfo_msgrnsql, 212

N

nca_conn_hash_size, 196
 nca_conn_req_max_q, 196
 nca_conn_req_max_q0, 197
 nca_ppmax, 197
 nca_vpmax, 198
 ncsizes, 73
 ndd, 152
 ndquot, 79
 nfs_max_threads, 119
 nfs:nocache, 136
 nfs:nfs_allow_preeepoch_time, 110
 nfs:nfs_async_clusters, 132
 nfs:nfs_async_timeout, 135
 nfs:nfs_cots_timeo, 110
 nfs:nfs_disable_rddir_cache, 129
 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
 nfs:nfs_write_error_to_cons_only, 128
 nfs:nfs3_async_clusters, 133
 nfs:nfs3_bsize, 130
 nfs:nfs3_cots_timeo, 111

nfs:nfs3_do_symlink_cache, 113
 nfs:nfs3_dynamic, 115
 nfs:nfs3_jukebox_delay, 137
 nfs:nfs3_lookup_neg_cache, 117
 nfs:nfs3_max_threads, 120
 nfs:nfs3_max_transfer_size, 138
 nfs:nfs3_max_transfer_size_clts, 140
 nfs:nfs3_max_transfer_size_cots, 141
 nfs:nfs3_nra, 123, 210
 nfs:nfs3_pathconf_disable_cache, 108
 nfs:nfs3_shrinkreaddir, 127
 nfs:nfs4_async_clusters, 134
 nfs:nfs4_bsize, 131
 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, 139
 nfs:nfs4_nra, 124
 nfs:nfs4_pathconf_disable_cache, 108
 nfs:nrnode, 125
 nfslogd, 204
 nfssrv:nfs_portmon, 142
 nfssrv:rfs_write_async, 143
 ngroups_max, 50
 noexec_user_stack, 40, 208
 nss, 204
 nstrpush, 90

P

pageout_reserve, 57
 pages_before_pager, 62
 pages_pp_maximum, 58
 passwd, 204
 physmem, 34
 pidmax, 48
 power, 204
 pt_cnt, 88
 pt_max_pty, 89
 pt_pctofmem, 88

R

rechoose_interval, 93
 reserved_procs, 47
 rlim_fd_cur, 72
 rlim_fd_max, 71
 routeadm, 22
 rpc.nisd, 204
 rpcmod:clnt_idle_timeout, 144
 rpcmod:clnt_max_conns, 144
 rpcmod:cotsmaxdupreqs, 148
 rpcmod:maxdupreqs, 147
 rpcmod:svc_default_stksize, 146
 rpcmod:svc_idle_timeout, 145
 rstatcown, 74

S

sctp_addip_enabled, 191
 sctp_cookie_life, 189
 sctp_cwnd_max, 183
 sctp_deferred_ack_interval, 185
 sctp_heartbeat_interval, 184
 sctp_ignore_path_mtu, 186
 sctp_initial_mtu, 185
 sctp_initial_out_streams, 190
 sctp_initial_ssthresh, 186
 sctp_ipv4_ttl, 184
 sctp_ipv6_hoplimit, 188
 sctp_largest_anon_port, 192
 sctp_max_buf, 187
 sctp_max_in_streams, 189
 sctp_max_init_retr, 182
 sctp_maxburst, 190
 sctp_new_secret_interval, 185
 sctp_pp_max_retr, 183
 sctp_prsctp_enabled, 191
 sctp_recv_hiwat, 187
 sctp_rto_max, 188, 189
 sctp_rto_min, 188
 sctp_shutack_wait_bound, 190
 sctp_smallest_anon_port, 191
 sctp_xmit_hiwat, 186
 sctp_xmit_lowat, 187
 segmap_percent, 77

segspt_minfree, 93
 semsys:seminfo_semaem, 220
 semsys:seminfo_semmni, 216
 semsys:seminfo_semmns, 218
 semsys:seminfo_semmnu, 218
 semsys:seminfo_semmsl, 216
 semsys:seminfo_semopm, 217
 semsys:seminfo_semume, 219
 semsys:seminfo_semvmx, 220
 shmsys:shminfo_shmmax, 221
 shmsys:shminfo_shmmni, 221
 slowscan, 60
 smallfile, 84
 sq_max_size, 199
 strmsgsz, 90, 91
 su, 205
 sun4u, 96, 209
 sun4v, 96, 209
 swapfs_minfree, 65
 swapfs_reserve, 64
 sys-suspend, 205
 syslog, 205

tcp_rexmit_interval_extra, 177
 tcp_rexmit_interval_max, 176
 tcp_rexmit_interval_min, 177
 tcp_RST_SENT_RATE, 170
 tcp_RST_SENT_RATE_ENABLED, 170
 tcp_SACK_PERMITTED, 165
 tcp_SLOW_START_AFTER_IDLE, 164
 tcp_SLOW_START_INITIAL, 164
 tcp_SMALLEST_ANON_PORT, 172
 tcp_TIME_WAIT_INTERVAL, 166
 tcp_TSTAMP_ALWAYS, 162
 tcp_TSTAMP_IF_WSCALE, 178
 tcp_WSCALE_ALWAYS, 161
 tcp_XMIT_HIWAT, 162
 throttlefree, 56
 timer_max, 95
 tmpfs_maxkmem, 85
 tmpfs_minfree, 86
 tmpfs:tmpfs_maxkmem, 209
 tsb_alloc_hiwater, 97
 tsb_rss_size, 99
 tune_t_fsflushr, 42
 tune_t_minarmem, 59

T

tar, 205
 tcp_conn_req_max_q, 167
 tcp_conn_req_max_q0, 168
 tcp_conn_req_min, 169
 tcp_cwnd_max, 163
 tcp_deferred_ack_interval, 159
 tcp_deferred_acks_max, 160
 tcp_ecn_permitted, 167
 tcp_ip_abort_interval, 175, 176
 tcp_keepalive_interval, 175
 tcp_largest_anon_port, 172
 tcp_local_dack_interval, 159, 211
 tcp_local_dacks_max, 160
 tcp_max_buf, 163
 tcp_mdt_max_pbufs, 171
 tcp_naglim_def, 171
 tcp_recv_hiwat, 163
 tcp_recv_hiwat_minmss, 178
 tcp_rev_src_routes, 166

U

udp_do_checksum, 180
 udp_largest_anon_port, 180
 udp_max_buf, 181
 udp_recv_hiwat, 179
 udp_smallest_anon_port, 179
 udp_xmit_hiwat, 178
 ufs_HW, 83
 ufs_LW, 83
 ufs_ninode, 80
 ufs:ufs_WRITES, 82
 utmpd, 205

Y

ypasswdd, 206

Z

`zfs_arc_max`, 35, 208
`zfs_arc_min`, 208

