

# Oracle9i for Linuxを使いこなす技 ~ チューニング編 ~

ミラクル・リナックス株式会社

# アジェンダ

- チューニングの考え方
- Linuxをチューニングする
- Monitoring the Oracle
- Monitoring the Server
- ケーススタディ

# チューニングの考え方

# チューニングの種類

- チューニングといっても意味は広い
  - パフォーマンスチューニング  
vs トラブルを防止するためのチューニング
  - 事前のチューニング vs 事後のチューニング
  - OSのチューニング vs Oracleのチューニング
  - インスタンスのチューニング vs SQLのチューニング  
vs 物理構造のチューニング
- 目的次第でアプローチは異なる

# パフォーマンスチューニングの目的

- 限られたハードウェアリソースの範囲で、より高速なパフォーマンス
- 安定した稼働環境の構築
- 現在使用しているハードウェアの限界点の見極め

# チューニングにおける基本指針

- なるべく早い段階に行なう
  - 開発フェーズの早い段階ほど効果は高く、コストも低い。逆に、遅くなるほど効果は低く、コストも高い。
- 目標を決める
  - チューニングにはたくさんの項目があるので、具体的な数字目標を事前に決めてから行なうこと。漠然とやっていると、いつまでたっても終わらない。

# パフォーマンスチューニングの 基本的な考え方

- コンピュータの3大要素
  - メモリ、CPU、ディスク装置
- それぞれのアクセススピードを考える
  - この中で圧倒的に遅いのがディスク装置
- パフォーマンスチューニングの目標
  - メモリを含めたI/Oを少なくすること
  - その中でもディスクI/Oを少なくすること

# 各フェーズにおけるチューニング

## STEP1:業務規則のチューニング

- システム化にあたり、現在の業務規則や業務フローが適切であるかを確認する。

## STEP2:DB設計のチューニング

- DBの概念設計や論理設計のチューニングを実施する。

## STEP3:アプリケーション設計のチューニング

- 優れたパフォーマンスが得られるようにアプリケーションを設計する。

## STEP4:インデックスの作成など、DBの論理構造のチューニング

- インデックスの追加、削除の検討。適切なスキーマオブジェクト(パーティション、索引構成表、etc)の検討。



設計

調整

# 各フェーズにおけるチューニング

## STEP5:SQLのチューニング

- 最適な検索経路を選択するようにSQL文を調整する。

## STEP6:メモリ割り当てのチューニング

- Oracleに適切な量のメモリが割り当てられるように調整する。

## STEP7:ディスク/I/Oおよび物理構造のチューニング

- ディスク/I/Oの分散。表や索引に適切な領域の割り当て。

## STEP8:複数ユーザーが同時に使用したときの競合のチューニング

- ロックやディスクの競合が低減するように調整する。

## STEP9:OSのチューニング

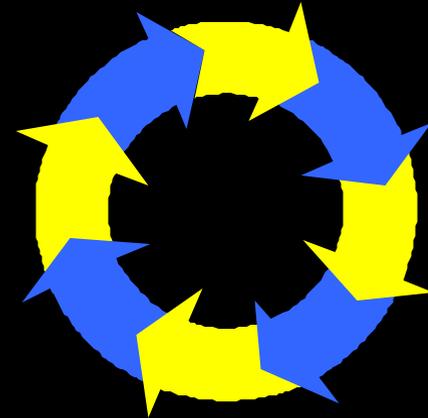
- レジストリやカーネルパラメータの変更。事後のチューニングでは、最初にやる。

↑  
プログラミング

↓  
テスト・運用

# チューニングのアプローチ

- 1.調査(現状の調査)
- 2.解析(問題点の究明)
- 3.適用(解決策の適用)
- 4.確認(効果の確認)



- チューニングは、ロジカルに行なうもの。経験と勘をたよりに気合でチューニングすることを否定はしないが、論理的な思考は重要。

# 開発フェーズ後半もしくは 運用フェーズでのチューニング

- 事後のチューニングで大変なのは、原因を発見すること。
- たくさんある要素の中から、現象をもとに仮説を立て、原因を絞り込まなければならない。
  - SQLに問題がある場合が一番多い。
  - しかし問題を見つけても、開発からしばらくたっている場合や、複雑なアプリケーションでは、修正は困難。

# チューニングの実施方法

- インスタンスチューニングでは、OracleとOSの双方で統計情報を収集する
  - DB: STATSPACK
  - OS: vmstat, iostat
- なぜ両面の情報を取得するのか？
  - メモリの空き容量やスワップの頻度、システム全体のI/Oなどは、Oracleのツールでは分からない。両面のデータを分析することによって、原因の仮説を立てられる。
- メモリ割り当ての基本方針
  - 20% くらいは余裕を持ってメモリを割り当てる。
  - 過度のメモリ割り当ては、システムの不安定を招く。

**Linuxをチューニングする**

# Linuxをチューニングする

- カーネルをアップデートする
- 共有メモリの設定
- ファイルシステムの選択
- ファイルシステムのチューニング
- 最終アクセスタイムを記録しない

# カーネルをアップデートする

- 新しいカーネルの方が、パフォーマンス面で改善されていることが多い。
- 例えばMIRACLE LINUX V2.0の場合、最新カーネルは、従来よりもI/Oパフォーマンスが向上している。
- 各ディストリビューションの状況
  - MIRACLE LINUX V2.0
    - CD-ROM :kernel 2.4.7 Webサイト:kernel-2.4.9
  - Red Hat Linux 7.1
    - CD-ROM :kernel 2.4.2 Webサイト:kernel-2.4.9

# 共有メモリの設定 (1)

- Oracleは、SGAを共有メモリ上に確保する。
- SGAの最大サイズ
  - Oracle9i Database R1(9.0.1) 約1.8GB
  - Oracle9i Database R2(9.2) :約62GB
    - 4GB以上のメモリを使用するときにはエンタープライズカーネルを使用する必要あり。

## 共有メモリの設定 (2)

- 共有メモリに関するカーネルパラメータ
  - SHMMAXは、共有メモリセグメント1つの大きさ。確保できる共有メモリの上限は $SHMMAX \times SHMSEG$
  - OracleのSHMMAXの推奨は、搭載メモリの1/2。
  - しかし経験則上、SGAが1つの共有メモリセグメントに収まるようにしたほうが良い ( $SGA < SHMMAX$ )。
- 変更はkparamやsysctlで行なう
- 共有メモリの状態を確認するには、ipcsコマンドを使用する。またOracleのsysresvコマンドも利用できる。

# kparam

- kparamは、カーネルを再構築せずにカーネルパラメータを変更するMIRACLE LINUX独自の拡張機能。
- MIRACLE LINUXでは、インストール時に搭載メモリの半分の値を設定。メモリ増設時には変更を検討する。

# kparam

- /etc/sysconfig/kparam を変更

```
# vi /etc/sysconfig/kparam
/proc/sys/kernel/semopm 100
/proc/sys/kernel/shmmax 262934528
/proc/sys/kernel/sysrq 0
/proc/sys/kernel/panic 10
/proc/sys/kernel/sem 250 32000 100 128
```

- kparam を再起動

```
# service kparam restart
```

# ファイルシステムの選択

- Oracleがサポートしているファイルシステム
  - OSベンダーがサポートしていて、同期書込みをサポートしているファイルシステム
  - 例えばMIRACLE LINUX V2.0/2.1だと、ext2, ext3, ReiserFS, RAW I/O (RAWデバイス)
- 推奨は、ext2 もしくは ext3。
- HAクラスタを使用するときには、ext3を強く推奨。
- MIRACLE LINUX V2.0/V2.1では、ルートファイルシステム以外はext3に変更可能。ルートファイルシステムもext3に変更できるが、カーネルアップデート時にmkinitrdを手動で実行する必要あり。

# ファイルシステム作成時のポイント

- ファイルシステムを作成するとき、ブロックサイズや i-node 数を指定できる。-j オプションは、ext3。
  - # mke2fs [-j] -b ブロックサイズ -T i-node数 デバイス名
- Oracleのブロックサイズは、OSのブロックサイズの倍数になるようにする。
  - 例 OSのブロックサイズ: 4K  
Oracleのブロックサイズ 4K, 8K, 16K
- 現在のファイルシステムの情報を表示する。
  - # tune2fs -l デバイス名

# ファイルシステム作成時のポイント

- ファイルシステムを作成する
  - # mke2fs -b ブロックサイズ -T i-node数 デバイス名
  - ブロックサイズ
    - 1024,2048,4098
  - i-node数
    - News 4KBあたり1node
    - Largefile 1MBあたり1node
    - Largefile4 4MBあたり1node
- データファイルを配置するだけのファイルシステム
  - # mke2fs -j -b 4098 -T largefile4 /dev/sdb1

# ファイルの最終アクセス時刻を記録しないようにする

- UNIX/Linuxでは、ファイルに関して次の3つの時間を記録している
  - 最終アクセス時刻 (atime )
    - ファイルにアクセスした最後の時間
  - 最終更新時刻 (mtime )
    - ファイルの内容を変更した最後の時間
  - 最終変更時刻 (ctime )
    - ファイルの内容もしくは属性を変更した最後の時間
- atimeは、もっとも更新されるのにもかかわらず、使い道がほとんど無い。atimeを記録しないようにすると、検索パフォーマンスが向上する。ただし効果にはバラつきがある。

# atimeを記録しない

- 設定方法は、次の2つのいずれか
  - マウントのオプションに指定する
    - /etc/fstabにnoatimeを指定する

```
/dev/sdb2          /home  ext2  defaults,noatime 1 2
```

- ファイルの属性を変更する
  - \$ chattr +A \*
  - \$ cd <データファイルのあるディレクトリ>
- statコマンドで参照可
  - \$ stat ファイル名

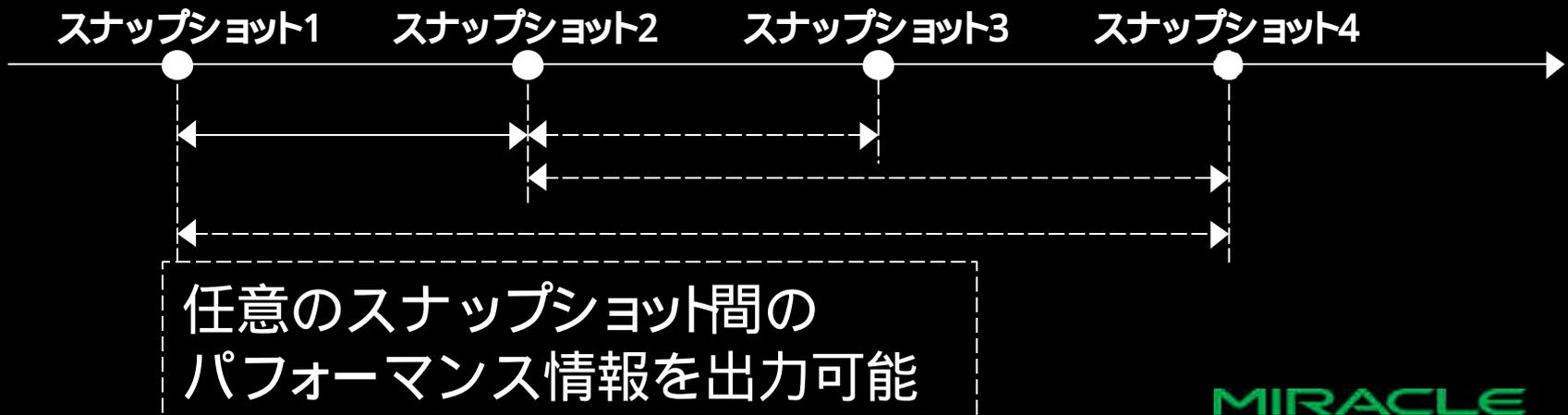
# Monitoring the Oracle

# ボトルネックを探す

- パフォーマンスの遅いSQLに出合ったら、実行計画を調べる習慣をつける。
- 実行計画やパフォーマンス統計を調べることによって、原因を推測できる。
- Oracleにはさまざまな解析ツールがある。
  - SQLトレース/tkprof
  - EXPLAIN PLAN
  - SQL\*Plusのオートトレース
  - BSTAT/ESTAT
  - STATSPACK
  - Oracle Enterprise Manager Diagnostics Pack/Tuning Pack
  - V\$ビュー

# STATSPACK

- Oracle8i R8.1.6から導入された、パフォーマンスデータの収集や分析を行なうためのツール。インスタンス全体のチューニングに使用する。BSTAT/ESTATの後継。
- 次の図のように、任意の2点間のデータを取得可能。



# STATSPACKの使い方 (1)

- STATSPACK用の表領域を作成する

```
$ sqlplus "/ as sysdba"  
SQL> @?/rdbms/admin/spcreate
```

- 失敗したときには一度削除する

```
SQL> @?/rdbms/admin/spdrop  
SQL> @?/rdbms/admin/spcreate
```

- データを収集する (2回以上実行する)

```
SQL> connect perfstat/pertstat  
SQL> execute statspack.snap          # デフォルト  
SQL> execute statspack.snap(i_snap_level=>5)    # レベル 5
```

# STATSPACKの使い方 (2)

- レポートを作成する (範囲を指定する)

```
SQL> connect perfstat/pertstat
SQL> @?/rdbms/admin/spreport
```

- 出力結果

Top 5 Wait Events

Event	Waits	Wait Time (s)	% Total Wt Time
log file parallel write	125	2	24.50
SQL*Net more data from dblink	39,020	2	23.03
SQL*Net message from dblink	234	1	20.15
sort segment request	1	1	13.69

# Monitoring the Server

# なぜサーバーのモニタリングが 必要なのか？

- システムにおける代表的なボトルネック
  - I/Oネック、CPUネック、ネットワークネック、メモリネック
- パフォーマンスチューニングでは、パフォーマンス劣化の原因となっているボトルネックを探し、それぞれにあわせた対応策を施さなければならない
  - 原因と対策
    - I/Oネック                      ディスクの増設、I/Oの分散
    - CPUネック                      CPUの増設
    - ネットワークネック          高速なネットワークの導入
    - メモリネック                    メモリの追加
- Oracleで取得できる情報だけでは、ハードウェアに起因するボトルネックの検出は困難

# Linuxでよく使われる モニタリングツール

- Linux上のモニタリングツール

  - vmstat メモリ、ディスクなどの監視

  - sysstat 総合的な監視ツール

    - sar, iostat, mpstat, isag

  - netstat ネットワークの情報を表示

  - ps aux プロセスの情報を表示

  - ipcs 共有メモリやセマフォに関する情報を表示

  - free 空きメモリや空きスワップなどを表示

  - uptime システムの稼動期間などを表示

- Linux上のモニタリングツールは、/proc ファイルシステムのデータを参照している。

# 重要な測定項目

CPU使用率	vmstat cpu(usr, sys, idl)列 sar -U
Run Queue/Wait Queue	vmstat r/b列
メモリ使用量	vmstat memory (free,swpd,buff,cache)列 sar -r
Disk I/O	vmstat bi/bo列 sar -B sar -b
スワップ領域へのI/O	vmstat si/so列 sar -W
平均待ち時間	iostat -x await列
DiskQueue長	iostat -x avgqu-sz列

# vmstatとは

- vmstatは、プロセスやメモリ、ページング、ブロックIO、トラップ、CPU稼働状況などの情報を報告する。第1行目はOSを起動してからの平均。
- CPUやメモリ、ディスクI/Oなど、幅広い情報が簡潔に表示されるので、Excelに取り込むときに便利。
  - vmstat 間隔[s] 回数 (連続のときは指定しない)

```
$ vmstat 10 20
```

procs				memory				swap		io		system			cpu		
r	b	w	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id		
0	0	0	32460	7512	15556	144024	0	0	7	20	52	82	9	4	87		
0	0	0	32460	7512	15556	144024	0	0	0	10	103	14	0	0	100		

# vmstatの使い方

- 2秒間隔で無制限に実行。CTRL+Cで停止
  - `$ vmstat 2`
- 2秒間隔で10回実行。「-n」オプションでヘッダーは1度だけ表示。teeを使用して、出力は標準出力と、ファイルの両方に出力。Excelなどに取り込むときには、この方法が便利
  - `$ vmstat -n 2 | tee output_vmstat`

# vmstatの重要な項目

- Process
  - r: CPUを割り当て中もしくは割り当て可能なプロセスの数
  - b: 割り込みを禁止しているプロセスの数。I/O待ちなどで、割り込み不可能なときに発生。
  - rは、CPUの個数以下、bはゼロであることが望ましい。
- Memory
  - swpd: 使用している仮想メモリの量 (KB)。
  - free: 空きメモリの量 (KB)。
  - buff: バッファキャッシュに使用されているメモリ量 (KB)。
  - cache: ページキャッシュに使用されているメモリ量 (KB)。

# vmstatの重要な項目

- Swap
  - si: ディスクからスワップインされているメモリの量 (kb/s)。
  - so: ディスクにスワップアウトしているメモリの量 (kb/s)。

# vmstatの重要な項目

- IO
  - bi: ブロックデバイスに送られたブロック (blocks/s)。
  - bo: ブロックデバイスから受け取ったブロック (blocks/s)。
- CPU
  - CPU の総時間に対するパーセンテージ
  - us: ユーザー時間
  - sy: システム時間
  - id: アイドル時間

# iostatとは

1行目は今までの  
平均もしくは累積

- I/O統計やCPU統計を表示するコマンド
  - iostat 間隔[s] 回数 (連続のときは指定しない)

```
$ iostat -dt 2
```

```
Linux 2.4.7-2.24mlsmp (host.your.domain.name) 03/01/02
```

```
Time: 07:35:06
```

Device:	tps	Blk_read/s	Blk_wrtn/s	Blk_read	Blk_wrtn
dev8-0	2.25	19.51	39.10	4324622	8665724
dev8-1	1.28	12.12	38.43	2685560	8516872

```
Time: 07:35:08
```

Device:	tps	Blk_read/s	Blk_wrtn/s	Blk_read	Blk_wrtn
dev8-0	1.50	0.00	24.00	0	48
dev8-1	0.50	0.00	4.00	0	8

# iostat使用例 2

- iostatは、こちらが重要
  - \$ iostat -d -x デバイス名 間隔

```
$ iostat -d -x sdb 2
```

```
Linux 2.4.7-2.24mlsmp (dhcp-0198.miraclelinux.com) 03/01/02
```

```
Device: rrqm/s wrqm/s r/s w/s rsec/s wsec/s avgrq-sz avgqu-sz await svctm %util
sdb      2.09  3.28  0.39  0.88  19.86  33.30  41.95  161.22 1357.58 660 9.96 83.76
```

```
Device: rrqm/s wrqm/s r/s w/s rsec/s wsec/s avgrq-sz avgqu-sz await svctm %util
sdb      0.00  0.00  0.00  0.00  0.00  0.00  0.00  10.00  0.00  0.00 100.00
```

# iostatの重要な項目

- avgqu-sz
  - デバイスごとの、I/Oリクエストの平均キュー長
  - これが大きいとI/O待ちが発生している可能性が高い
- await
  - I/Oリクエストを発行してから、それが実行されるまでの平均待ち時間[m sec]
- %util
  - I/Oリクエストを発行しているときのCPU使用率

# ケーススタディ

# シナリオ

- 問題点
  - 2台のマシン間でリードオンリースナップショットを作成するパフォーマンスが遅い。
- 目標
  - 夜間のメンテナンス時間内に終わるようにしたい。
- 実施した作業
  - 実際の処理 (スナップショット作成) を実行し、その間 STATSPACK, iostat, vmstat でデータを収集する。

# STATSPACKの出力結果

- STATSPACKの出力は、A4で20ページ以上あるが、一番大切なのは最初の1ページ目。

測定期間中におけるデータベースの動作傾向

## Load Profile

~~~~~	Per Second	Per Transaction
	-----	-----
Redo size:	443,895.17	121,577,953.78
Logical reads:	405.38	111,030.44
Block changes:	331.01	90,658.67
Physical reads:	35.54	9,733.11
Physical writes:	121.03	33,148.89

# STATSPACKの出力結果

- 共有プールに関する情報1

Instance Efficiency Percentages (Target 100%)

```
~~~~~
```

Buffer Nowait %:	99.99	
Redo NoWait %:		99.68
Buffer Hit %:	99.93	
In-memory Sort %:		99.78
Library Hit %:	92.46	

共有プールの使用率

- 共有プールに関する情報2

Shared Pool Statistics	Begin	End
	-----	-----
Memory Usage %:	86.27	53.81
% SQL with executions>1:	62.89	54.14
% Memory for SQL w/exec>1:	37.60	29.86

共有プールにある解析済みSQLのなかで、2回以上実行している割合

# STATSPACKの出力結果

- Top5 Wait Events **これが一番大事**

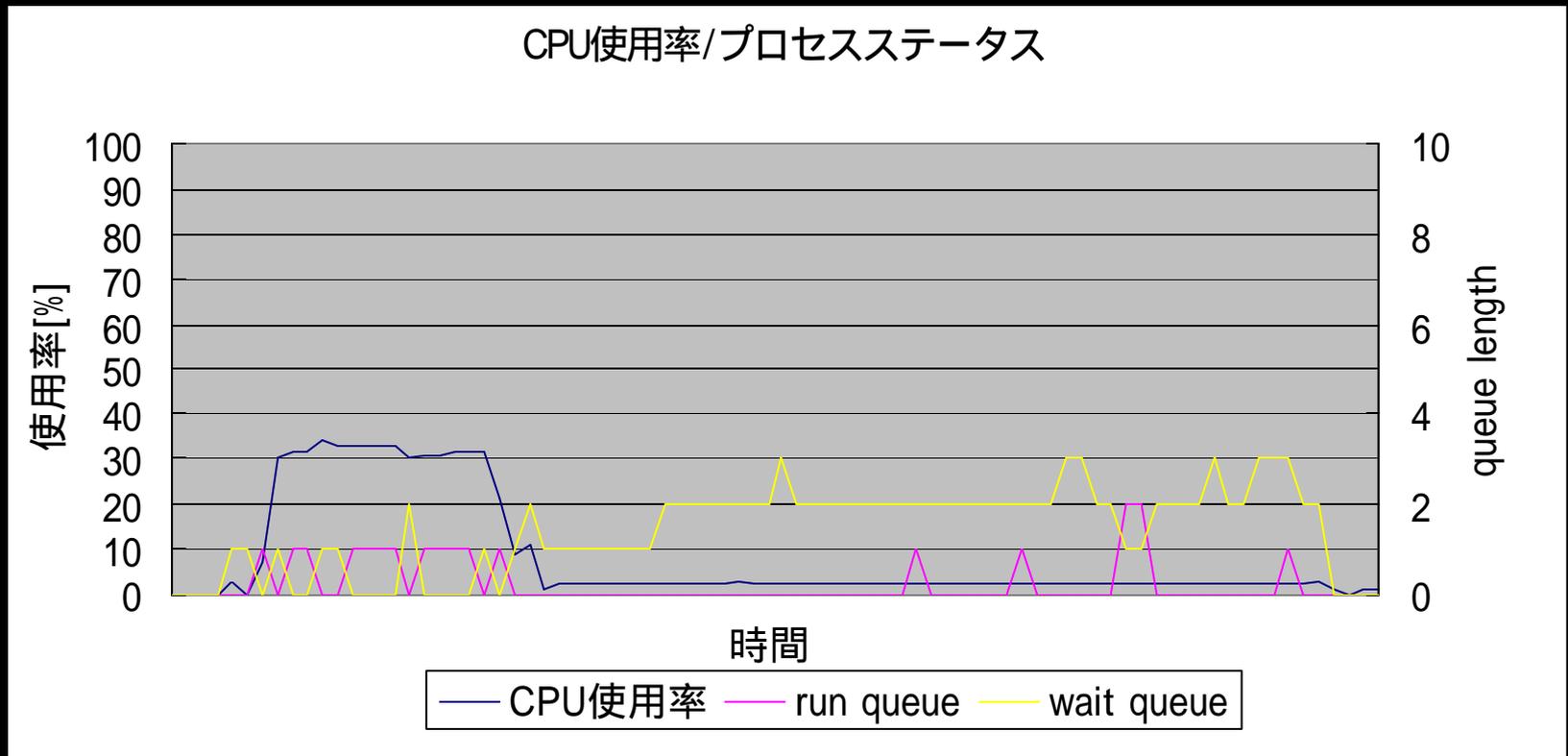
## Top 5 Wait Events

Event	Waits	Wait Time (s)	%Total Wt Time
log file switch (checkpoint incomplete)	1,182	1,131	51.80
log file parallel write	4,868	331	15.17
log buffer space	3,809	249	11.39
control file parallel write	1,745	131	5.99
log file switch completion	113	106	4.88

**全体の約80%のWaitがREDO  
ログ関連で発生している**

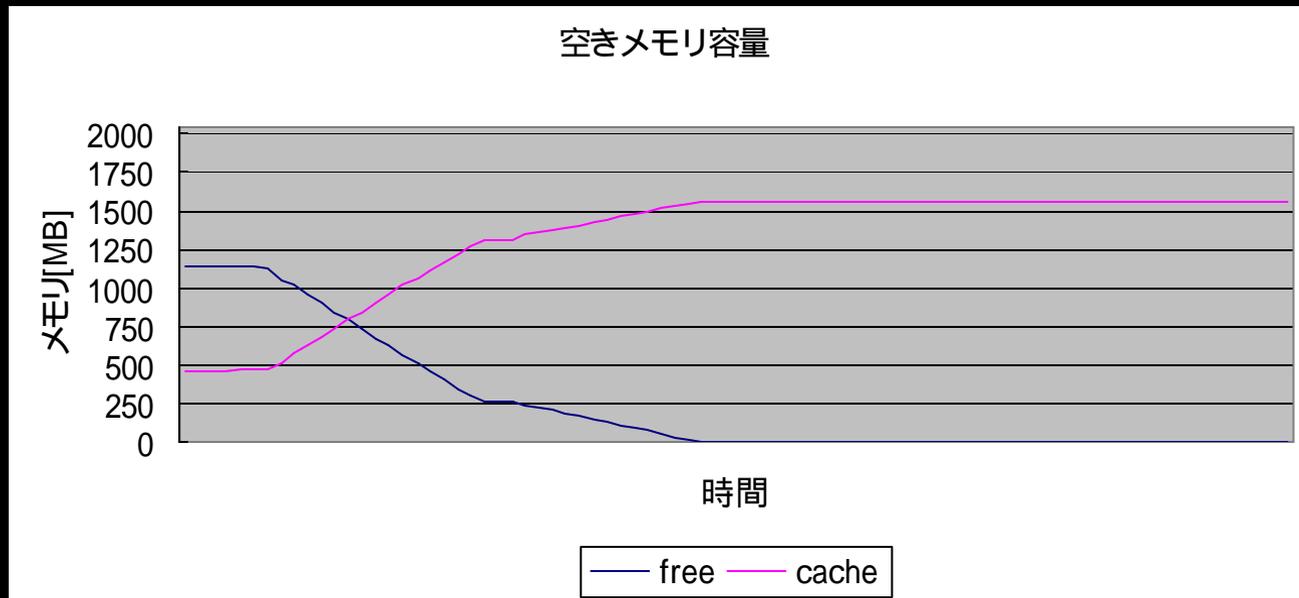
# vmstatの結果(CPU, Process)

- Wait queueがつねに1以上なので、どこかでウェイトが発生している。



# vmstatの結果(メモリ)

- 空きは十分。freeが時間とともに少なくなっているのは、キャッシュに割り当てられているため。



# 原因と対策

- STATSPACKの結果から、チェックポイントの多発によるパフォーマンス低下の可能性が高い。
- CPUのwait queueが1以上になっているのも、チェックポイントのI/O待ちのため。
- 対応策
  - REDO ログファイル拡大によるチェックポイント数の削減
  - REDO ログバッファ拡大によるREDO ログファイルへの書き込み数削減
- REDO ログファイル:10MB → 1GB、REDO ログバッファ:500KB → 5MBにしたところ、2倍のパフォーマンスが得られた。

# Appendix: 目次

- sar/isagの使い方
- freeの読み方
- Webサイト
- チューニングに関するお勧め書籍

# sysstatパッケージ

- sarやiostatなどを含んだシステム監視ツール
  - sar システムの動作統計情報を表示
  - sadc sarのラッパー
  - isag sarやsadcで取得したデータをグラフ化
  - iostat I/O統計情報を表示
  - mpstat CPUの使用率や割り込み数を表示

# sarとは

- UNIX/Linuxでよく使われている、システム情報を表示するコマンド。
- オプションを指定することによって、CPUやディスクなどのさまざまな統計データを表示。
- 実行間隔や実行回数を指定可能。
- 「o」オプションで、出力先ファイルを指定し、バイナリ形式でデータを保存可能。
- 保存したバイナリ形式のデータを「f」オプションで参照可能。

# sarの使用例 (1)

- I/O統計情報を2秒間隔で10回実行

```
$ sar -b 2 10
Linux 2.4.7-2.14mlsmp (host.your.domain.name)    10/15/01
16:52:30      tps    rtps    wtps  bread/s  bwrtn/s
16:52:32      0.00    0.00    0.00    0.00    0.00
```

- スワップ領域に対する統計情報を2秒間隔で無制限に出力させる

```
$ sar -W 2 0
Linux 2.4.7-2.14mlsmp (host.your.domain.name)    10/15/01
16:55:17  pswpin/s pswpout/s
16:55:19      0.00    0.00
16:55:21      0.00    12.00
```

- 「B」のpgpgin/sのpgpgout/sは、ディスクI/Oなので注意

## sarの使用例 (2)

- 2秒間隔で100回実行し、outfileに保存
  - `$ sar -A 2 100 -o outfile`
- 保存したoutfileの中からI/O統計情報を表示
  - `$ sar -b -f ourfile`
- オプションを指定しないときには、/var/lib/sa 配下に格納されているファイルの内容を表示
  - `$ sar`

# デフォルトの自動収集

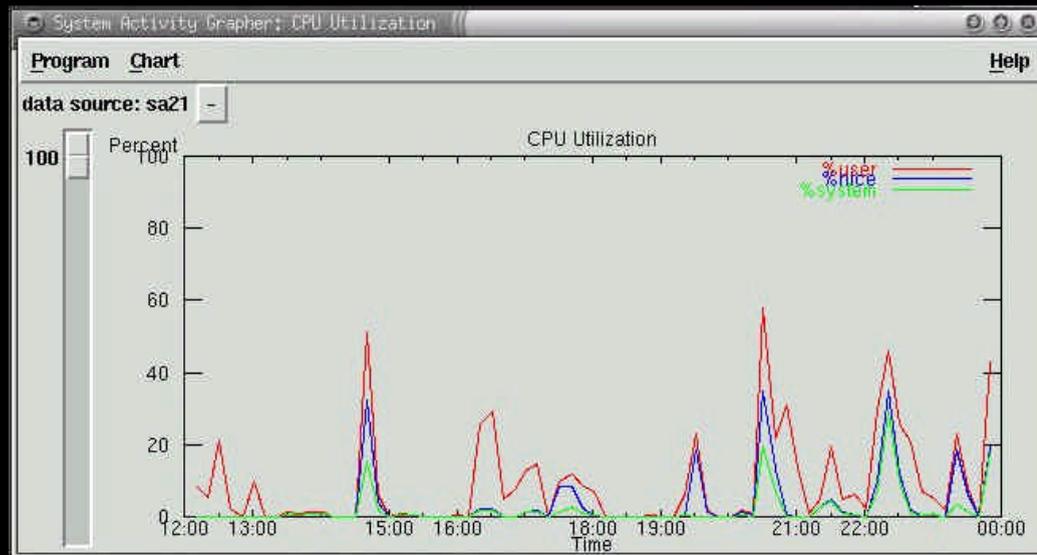
- MIRACLE LINUX V2.0 のデフォルトの設定では、sadc(sar)を使って10分に一度サーバーのアクティビティを収集している。
  - /etc/cron.hourly/sysstat, /etc/cron.daily/sysstat
- 自動的に取得したデータの格納先
  - /var/lib/sa ディレクトリ

```
$ ls /var/log/sa
sa21 sa23 sa25 sa27 sar21 sar23 sar25 sar27
sa22 sa24 sa26 sa28 sar22 sar24 sar26 sar28
```

- saNN NN日に収集したデータ(バイナリ形式)
- sarNN NN日に収集したデータ(テキスト形式)

# 収集したデータのグラフ表示

- isag(Interactive System Activity Grapher)
  - sar/sadcで取得したバイナリ形式のデータをグラフ化
  - I/O転送レート
  - CPU使用率
  - メモリ使用量
  - etc



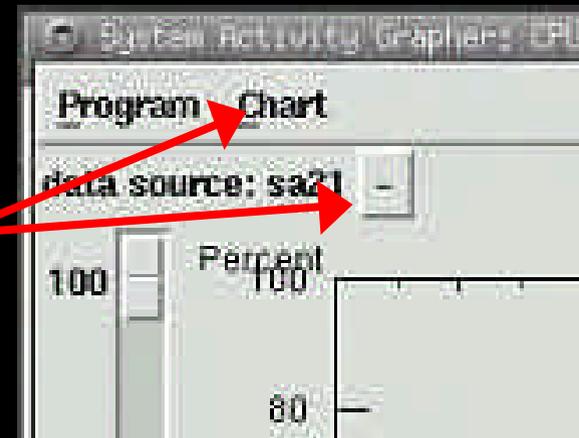
# isagの使用方法

## 使用方法

- isag [-p datafile\_path]
- datafile\_pathのデフォルトは/var/lib/sa
- カレントディレクトリのデータファイルを読み込み可能な状態で起動

```
$ isag -p .
```

- リストからデータファイルを選択  
「-」ボタンをクリック
- 表示するグラフの内容を選択  
「Chart」をクリック



# freeの読み方

\$ free

	total	used	free	shared	buffers	cached
Mem:	512120	501632	10488	102484	6156	149616
-/+ buffers/cache:		345860	166260			
Swap:	1052248	24032	1028216			

## – Mem

- Total 搭載メモリサイズ
- Used 使用しているメモリサイズ
- Free 空きメモリサイズ

計算式 = 搭載メモリ

- アプリケーションや OS が使用しているメモリ
- ページキャッシュ(cached)
- バッファキャッシュ(buffers)

- Shared 共有メモリ

一般的にOSとアプリ  
が使用している  
メモリサイズ

# freeの読み方

## – -/+ buffers/cache

- Used 使用しているメモリからページキャッシュとバッファキャッシュを引いた値

– Mem:Used - buffers - cached

- Free 空きメモリにページキャッシュとバッファキャッシュを足した値

– Mem:Free + buffers + cached

## – Swap

- Total swapとして割り当てたディスクサイズ
- Used 使用中のサイズ
- Free 使用していないサイズ

- <http://www.linux.or.jp/JF/JFdocs/The-Linux-Kernel-4.html>

# Webサイト(1)

- ミラクル・リナックス株式会社
  - <http://www.miraclelinux.com>
- MIRACLE LINUX メーリングリスト
  - <http://www.miraclelinux.com/products/ml/index.html>
- MIRACLE LINUX 対応ハードウェア
  - <http://www.miraclelinux.com/support/hardware20.html>
- MIRACLE LINUX パッチ提供
  - <http://www.miraclelinux.com/support/update/list.php3>
- MIRACLE LINUX サポート情報
  - [http://www.miraclelinux.com/products/product\\_info/support.html](http://www.miraclelinux.com/products/product_info/support.html)

# Webサイト(2)

- 日本オラクル株式会社
  - <http://www.oracle.co.jp>
- Oracle Technology Network Japan
  - <http://technet.oracle.co.jp>
  - 技術情報 : 「テクノロジー」 「Linux & Windows」
  - 会議室 : 「フォーラム」 「Linuxの部屋」
  - 日本語オンラインマニュアル: 「ドキュメント」
  - トライアル版 : 「ソフトウェア」
- OTN (Oracle Technology Network)
  - <http://otn.oracle.com>

# Webサイト(3)

- How to Get Maximum Performance from Linux and Oracle
  - <http://www.linux.com/newsitem.phtml?sid=93&aid=12536>
- Linux Performance Tuning
  - <http://linuxperf.nl.linux.org/links.html>
- Oracle FAQ
  - <http://www.orafaq.com/faq2.htm>
- sysstat home page
  - <http://perso.wanadoo.fr/sebastien.godard/>

# チューニングに関するお勧め書籍

- 「Oracle9i データベース・パフォーマンス・ガイドおよびリファレンス」
- 「Oracle9i データベース・パフォーマンス・メソッド」
  - 製品マニュアル。やはりこれが基本。
- 「Oracle データベース管理を極める13章」翔泳社
  - チューニング本ではないけど、チューニングに役立つこと満載。
- 「Oracle パフォーマンスチューニング第2版」
  - Mark Gurry, Peter Corriganオーム社
  - Oracleチューニングのバイブル。しかしOracle8ベースのため、古さも目立つ。

# チューニングに関するお勧め書籍

- 『Oracle SQL Tuning Pocket Reference 』
  - Mark Gurry, O'REILLY社
  - ポケットリファレンスといいながらも、Oracle9iにも対応し、エッセンスが凝縮されている。お勧め。
- Oracle9i for Windows 2000 Tips & Techniques
  - Scott Jesse, Matthew Hart, OSBORNE社
  - Oracle9iの新機能やWindows 2000の情報が満載。Windows 2000ユーザーはもちろんのこと、Oracle9iユーザーであれば読む価値あり。
- ORACLE High-Performance SQL Tuning
  - Donald K. Burleson, OSBORNE社
  - SQLチューニングに特化した、数少ない本。MIRACLE

# チューニングに関するお勧め書籍

- Oracle SQL High-Performance Tuning 2nd
  - Guy Harrison, Prentice Hall社
  - この第1版が翻訳されている
- Oracle DBA on Unix and Linux
  - Michael Wessler, SAMS社
  - Linux/UNIXで書かれている数少ない本。おすすめ。