

MIRACLE

【A-1】

Oracle Linux
Summit2003 in Spring

Oracle on Linux startup講座
～Linuxを選択してみる～

ミラクル・リナックス株式会社
製品本部 マーケティング部
プロダクトマネジャー
佐藤 剛春

Agenda

- RDBMSって何？
 - RDBMSに求められること
- アーキテクチャ概要
 - 整合性の確保
 - 記憶領域管理
 - インスタンス
 - 内部動作
- 稼働環境
- 運用時の負荷の軽減
- Linuxを選択する理由
 - 高い信頼性・安全性
 - DB Serverとして高性能
 - File Serverとして高性能
 - スケーラビリティ向上
 - 進化するLinuxカーネル

RDBMSって何？

- Relational DataBase Management Systemの略称
- Relational Database
 - データを単純な2次元の表形式で表現
 - データ項目の列とレコードの行によって構成
 - 各表を関連付けて管理可能
- Database Management System
 - 複数のユーザによる、複数または同一データの使用を可能とする機能を提供
 - データ整合性を保全する機能(トランザクション管理)
 - データ破壊からの復旧機能(バックアップ・リカバリ)
 - 他

RDBMSに求められること

Internet

- データの種類が増加
- あらゆる情報へのアクセス要求
- より高度なサービスの要求

システムの追加？

部門での管理？

既存システムとの連携は？



RDBMSに求められること

- 情報の集約の重要性

- 目的毎に構築されたシステムでは、

- どこに最新のデータが存在するのか？
- どんな形式で格納されているのか？
- 類似データとの関連性はどうなっているのか？
- 全社的な情報活用にはどのデータを使用すればよいのか？
- システムの運用主体は利用部門？情報システム部門？

といった把握が困難となり、情報のフラグメントと重複が発生する。

整合性の確保

～ロックの違い

	Oracle	Others
行レベルロック	Yes	Yes
マルチバージョンニングのサポート	Yes	-
ページ/テーブル ロック	テーブル	両方
ロックエスカレーション	-	Yes
ロード中のデッドロック	-	Yes
分離レベルのトレードオフ	-	Yes
読み込みが書き込みをブロック	-	Yes
書き込みが読み込みをブロック	-	Yes
ダーティリード	-	Yes

行レベル・ロックとは

- あるユーザが更新中のデータに対して、別のユーザの更新を防ぐための排他制御が「ロック」です。
- 一般的なロックのレベル(単位)
 - 表
 - ページ
 - 行
- ロックの単位が大きくなると、他のデータにも不要なロックがかかってしまい、ロック解除待ちが多発するため、処理効率が大幅に低下します。

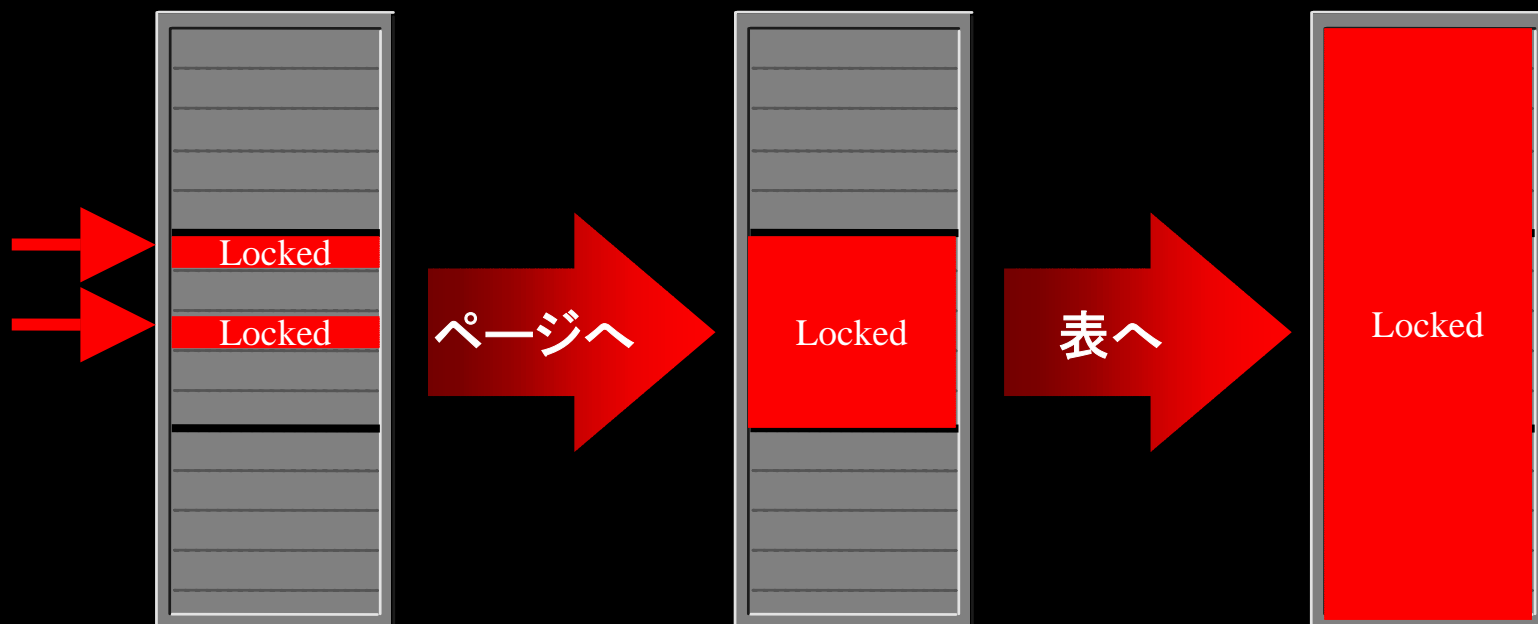
行レベル・ロック機能

データベース機能比較表

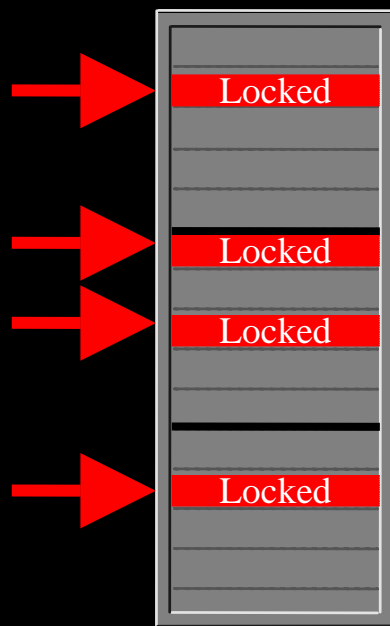
	他社製RDBMS	Oracle9i DB
行レベル・ロック	○	○

他社製RDBMS 不完全な行レベル・ロック

- ロックする行数がある一定の値を越えると、ロック・エスカレーションが発生し、ページ・レベル、表・レベルへ移行する。
- アプリケーションの同時実行性の低下



Oracleデータベース 完全な行レベル・ロック



- Oracleデータベースは常に行レベル・ロックをしています。
- Oracleの行レベル・ロックは、ロックの行数に制限がなくロックエスカレーションが発生しません。
- きめ細かい行レベル・ロックにより、ロックの競合が大幅に軽減されます。

行レベル・ロック機能

データベース機能比較表

	他社製RDBMS	Oracle9i DB
行レベル・ロック	△	◎

トランザクション分離レベル

分離レベル(ANSI/ISO SQL92)	Oracle	他社RDBMS
UNCOMMITTED READ(未コミット読取り)	—	○
READ COMMITTED(コミット読取り)	◎	◎
REPEATABLE READ(繰返し可能読取り)	—	○
SERIALIZABLE(直列可能)	○	○

デフォルトは「**READ COMMITTED(コミット読取り)**」
しかし、ベンダーによって実装方法が**違います**。

他社製RDBMS

ロックを使用した**READ COMMITTED**(コミット読み取り)

user2



SELECT

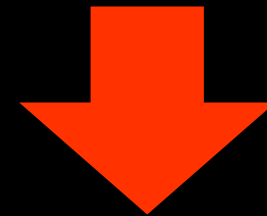
user1



UPDATE



- user1があるデータを更新。同時に行をロック。
- user2が参照開始。
- ロックされている行があると待ち状態。



同時実行性の欠如

他社製RDBMS

READ UNCOMMITTED (非コミット読み取り)

user2



SELECT

user1

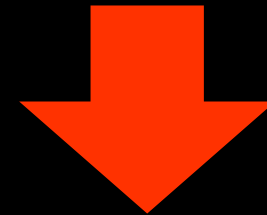


UPDATE

更新中

ダーティリード

- user1があるデータを更新。同時に行をロック。
- user2が参照開始。
- ロックされている行があると更新中のデータを読み込む



データの一貫性の欠如

Oracleデータベース

真のREAD COMMITTED(コミット読み取り)

user2

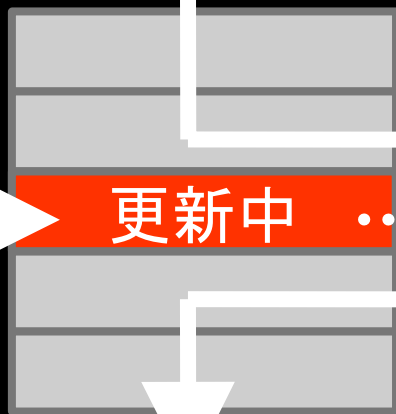


SELECT

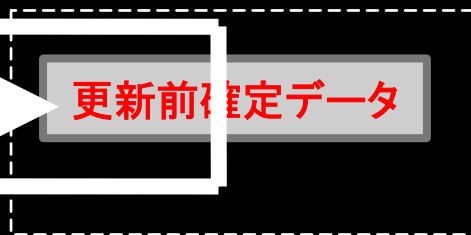
user1



UPDATE



- user1があるデータを更新。同時に行をロック。
- user2が参照開始。
- ロックされている行があるとUNDO表領域にコピーされたデータを読み込む。

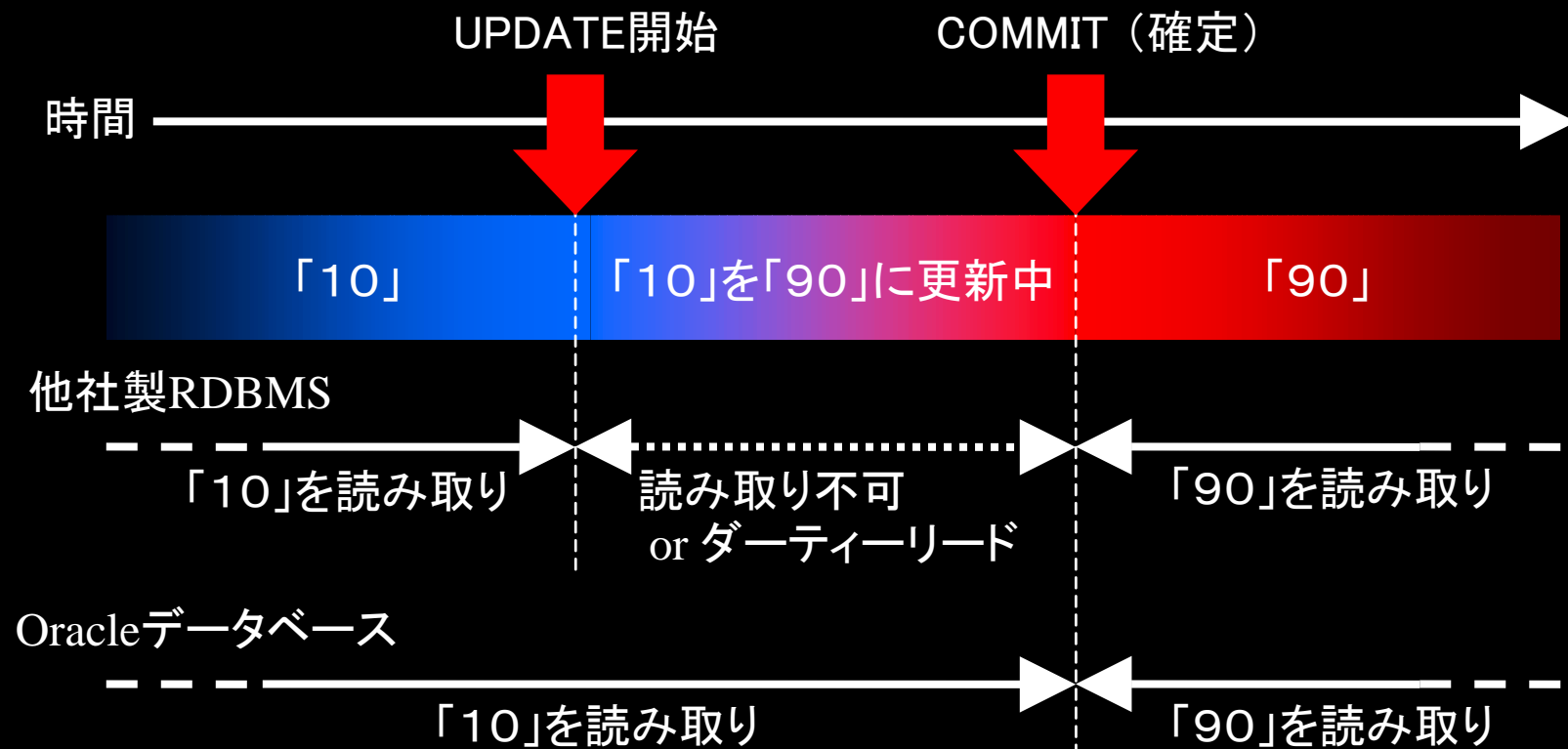


UNDO表領域

同時実行性とデータの一貫性の両立

どのデータを読むのが正しいのか？

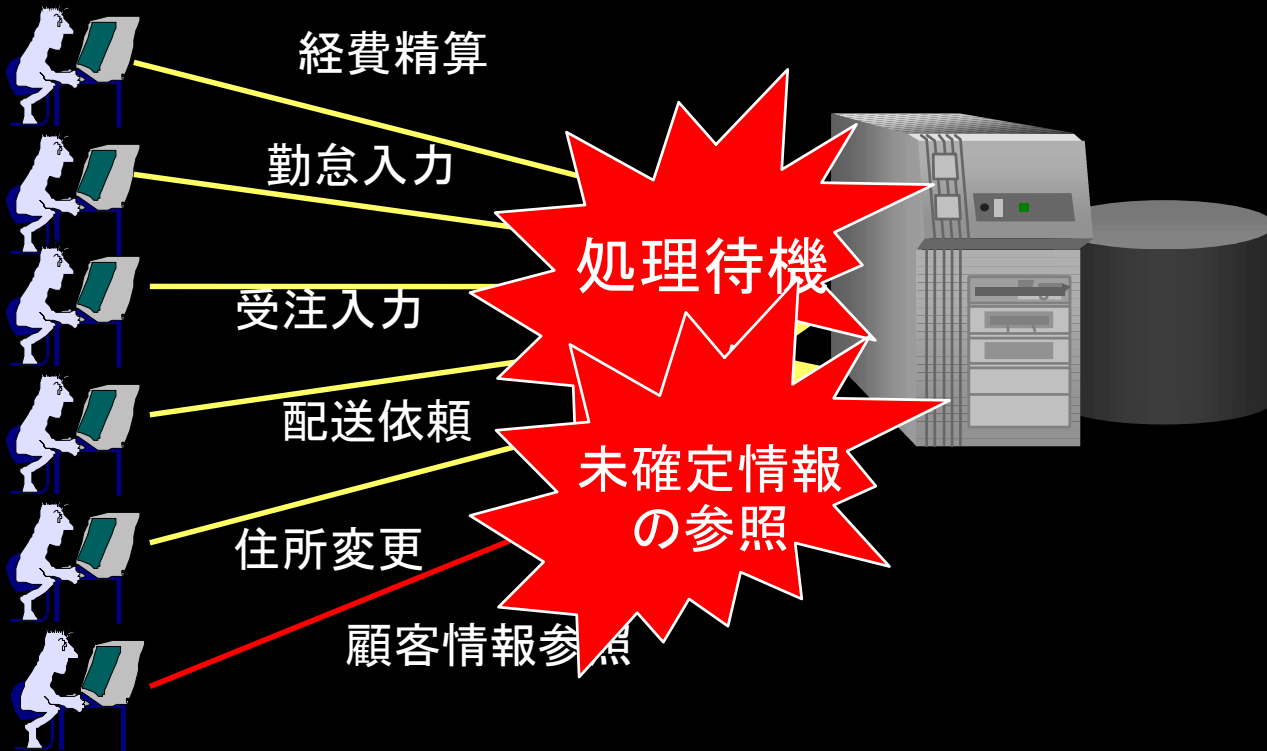
「10」を「90」に更新する場合。



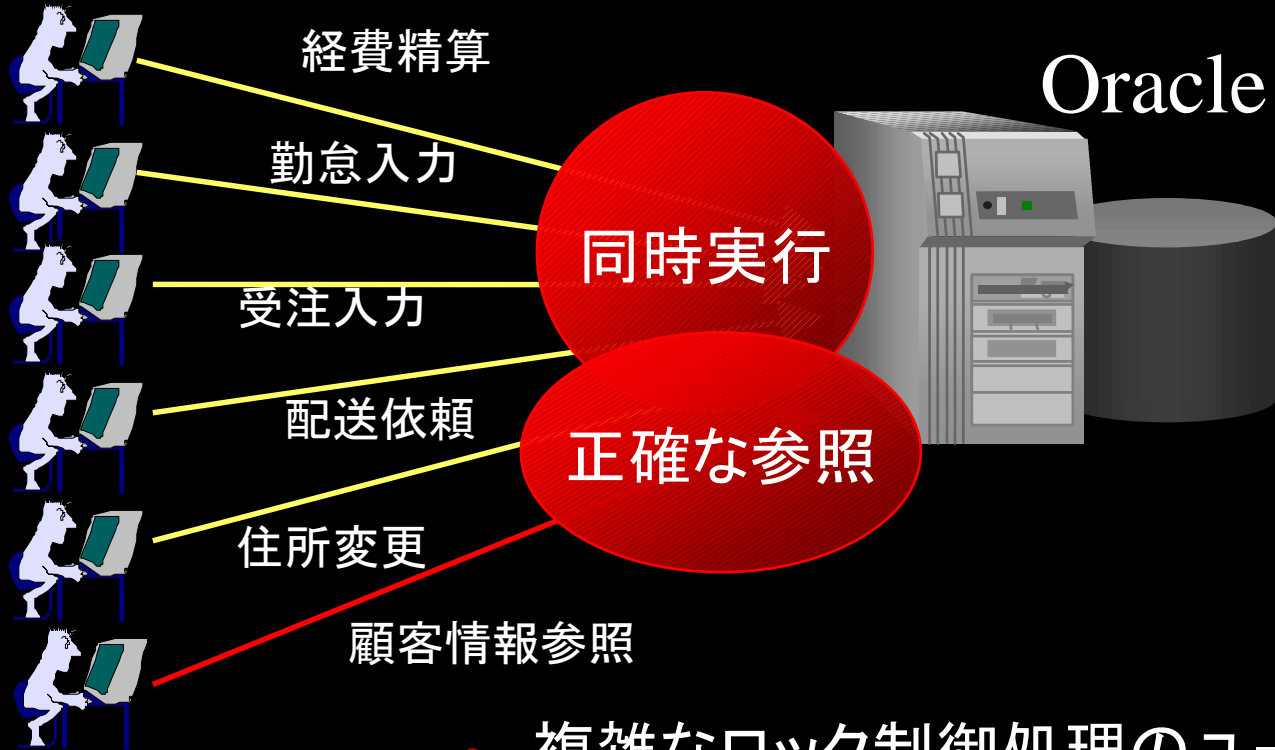
整合性の確保

なぜロックを使用しない読み取り一貫性が重要か？

他のRDBMSで、複数ユーザからのアクセスに対して、Oracleと同等のレコード制御や読み取り一貫性を実現しようとすると...

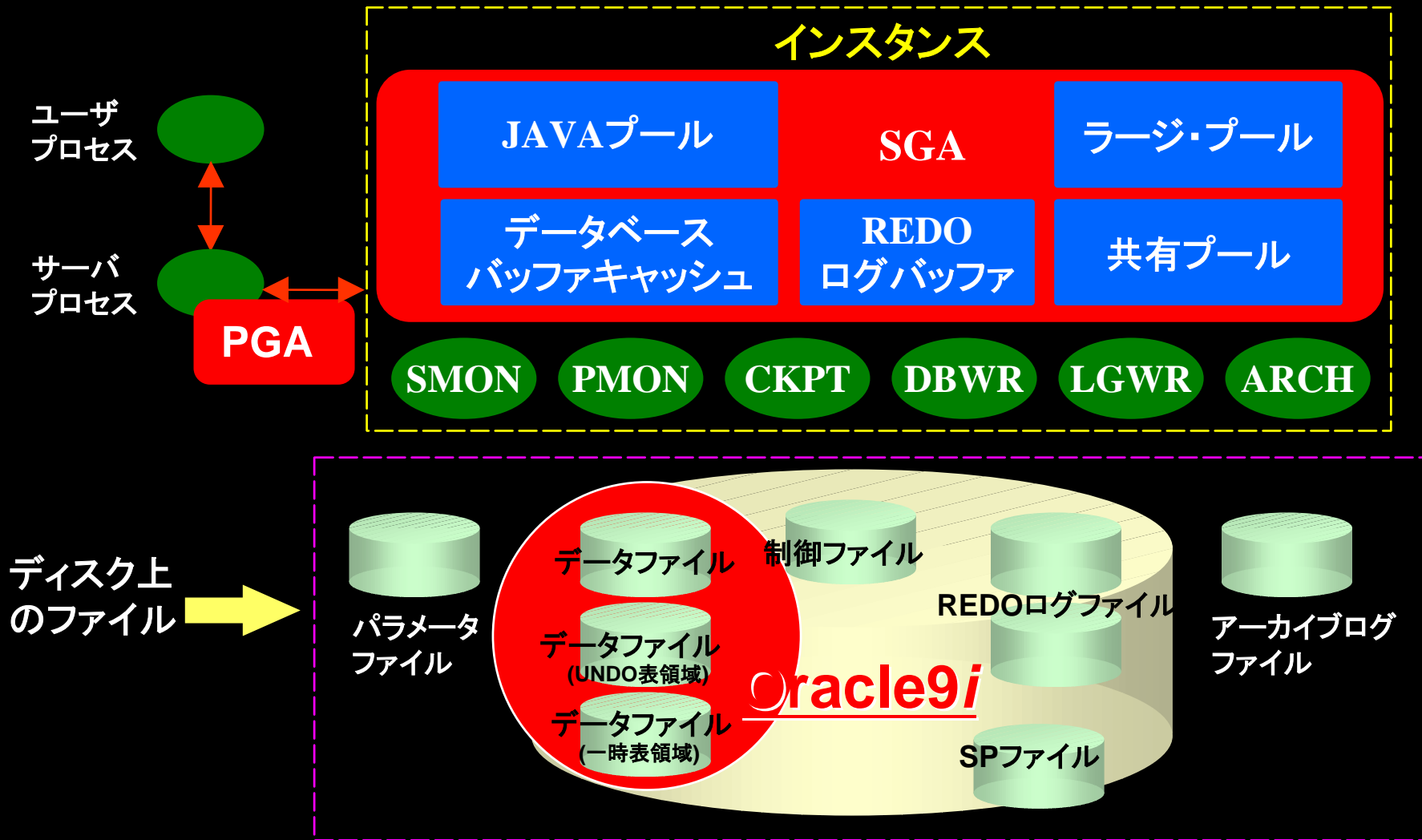


整合性の確保 – Oracleでは...

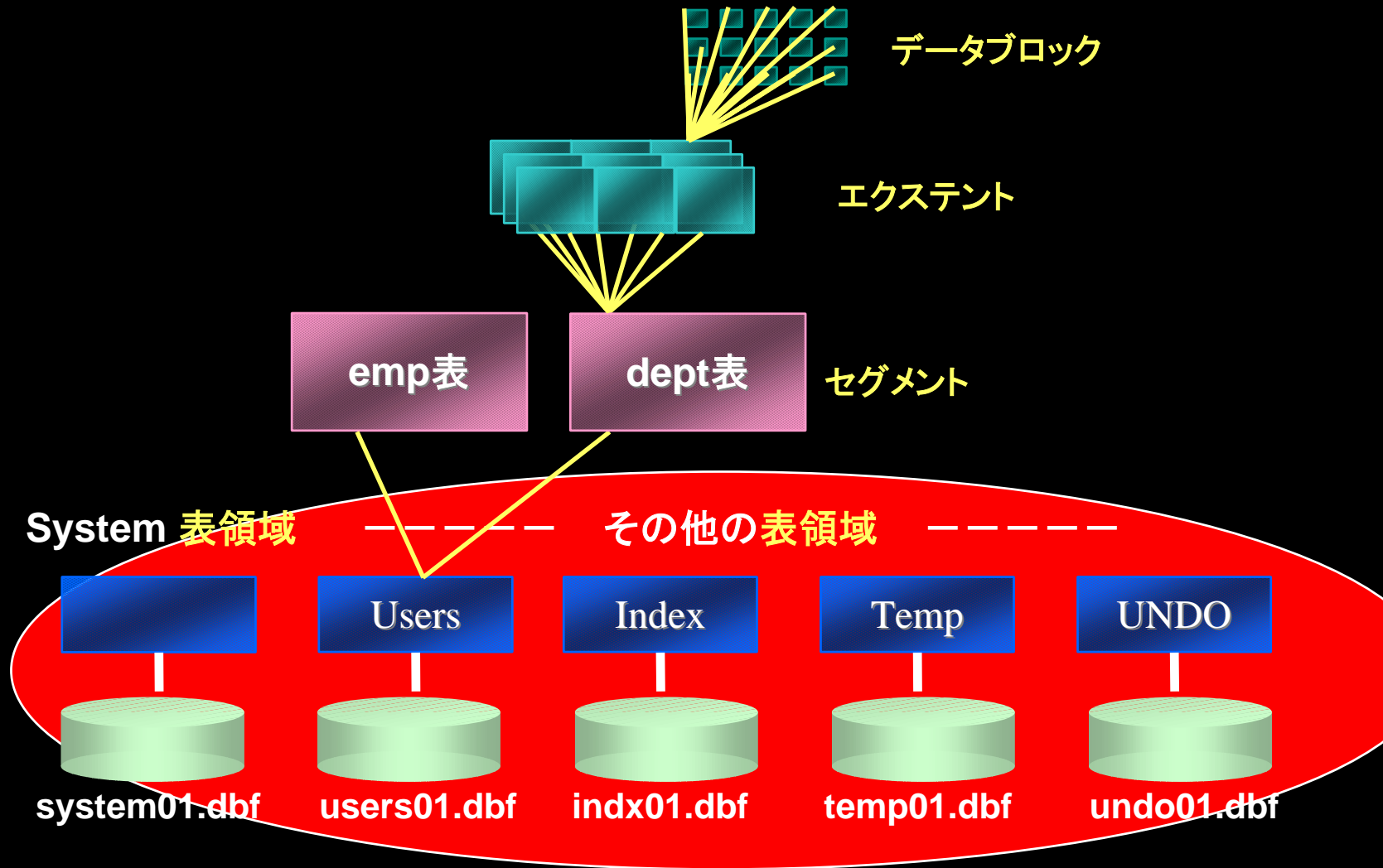


- 複雑なロック制御処理のユーザ開発
 - 読み取り一貫性保証処理のユーザ開発
- などは不要

アーキテクチャ概要



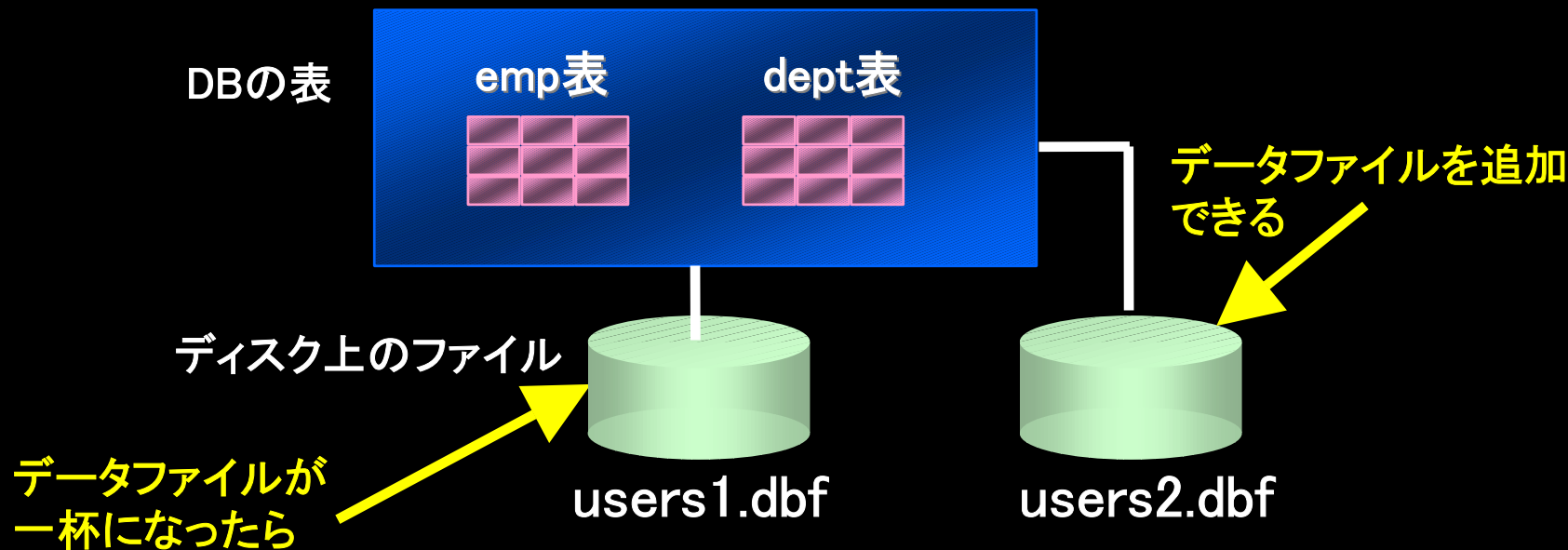
記憶領域管理



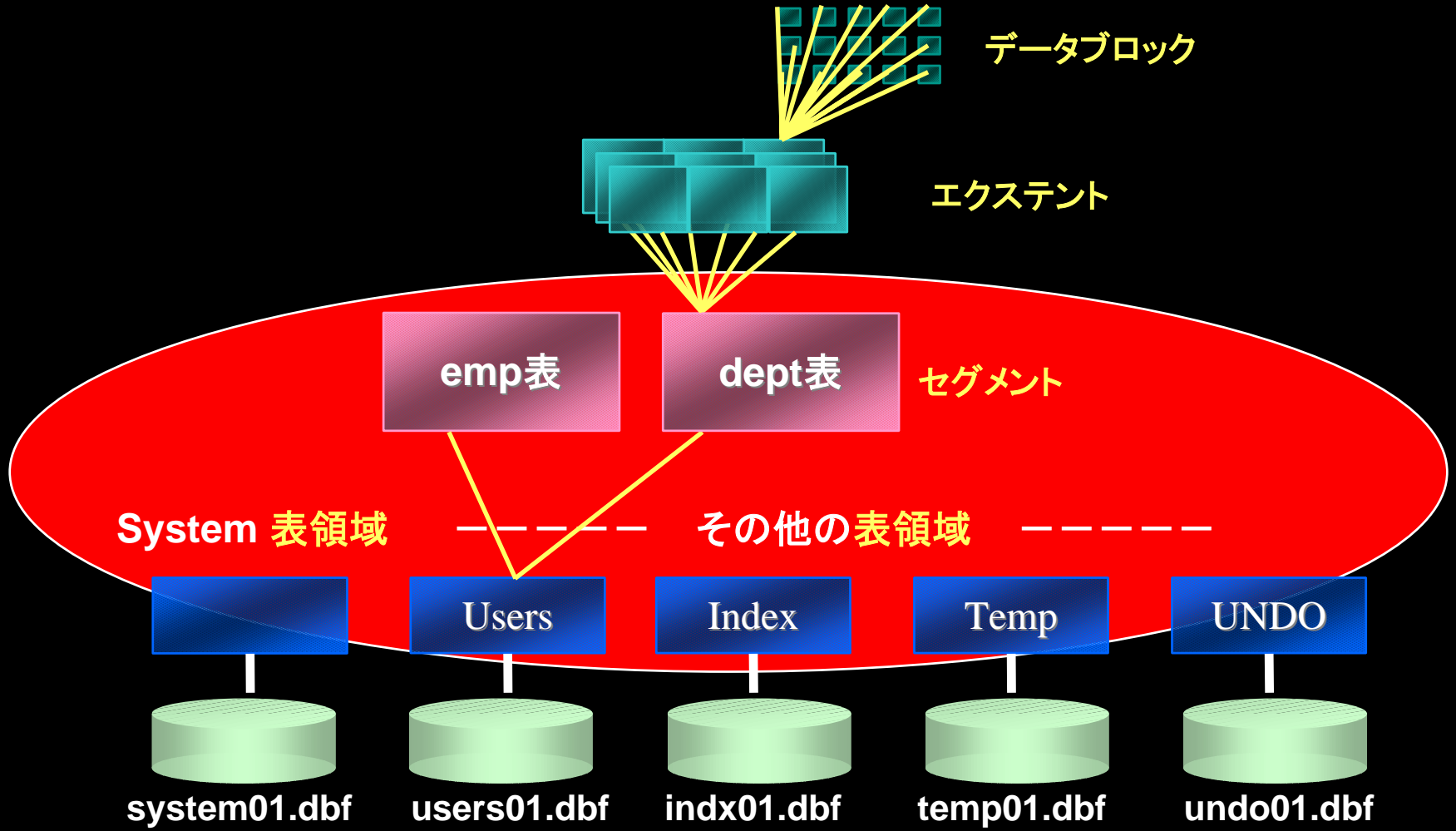
表領域とディスク上のファイルの関係

- 表を入れる枠を作成し、表領域と呼ばれる領域とディスク上のファイルを対応させる。

表領域 USERS

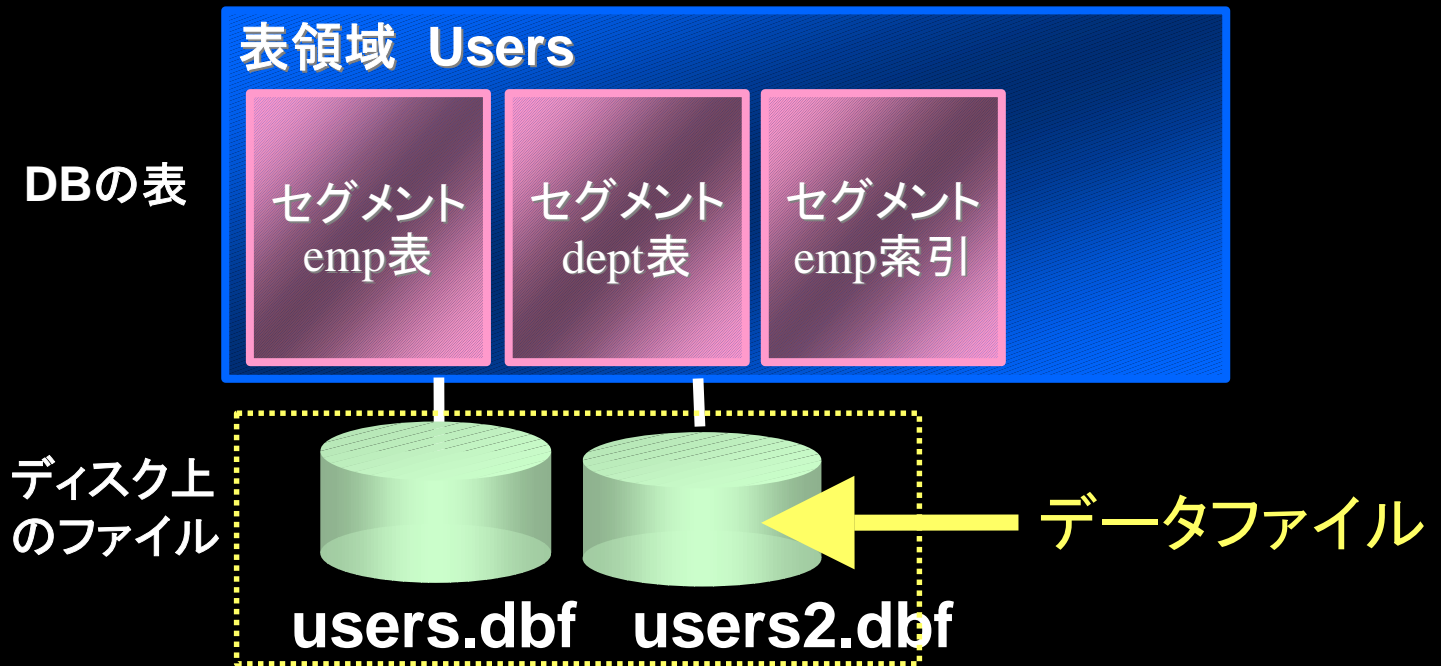


記憶領域管理

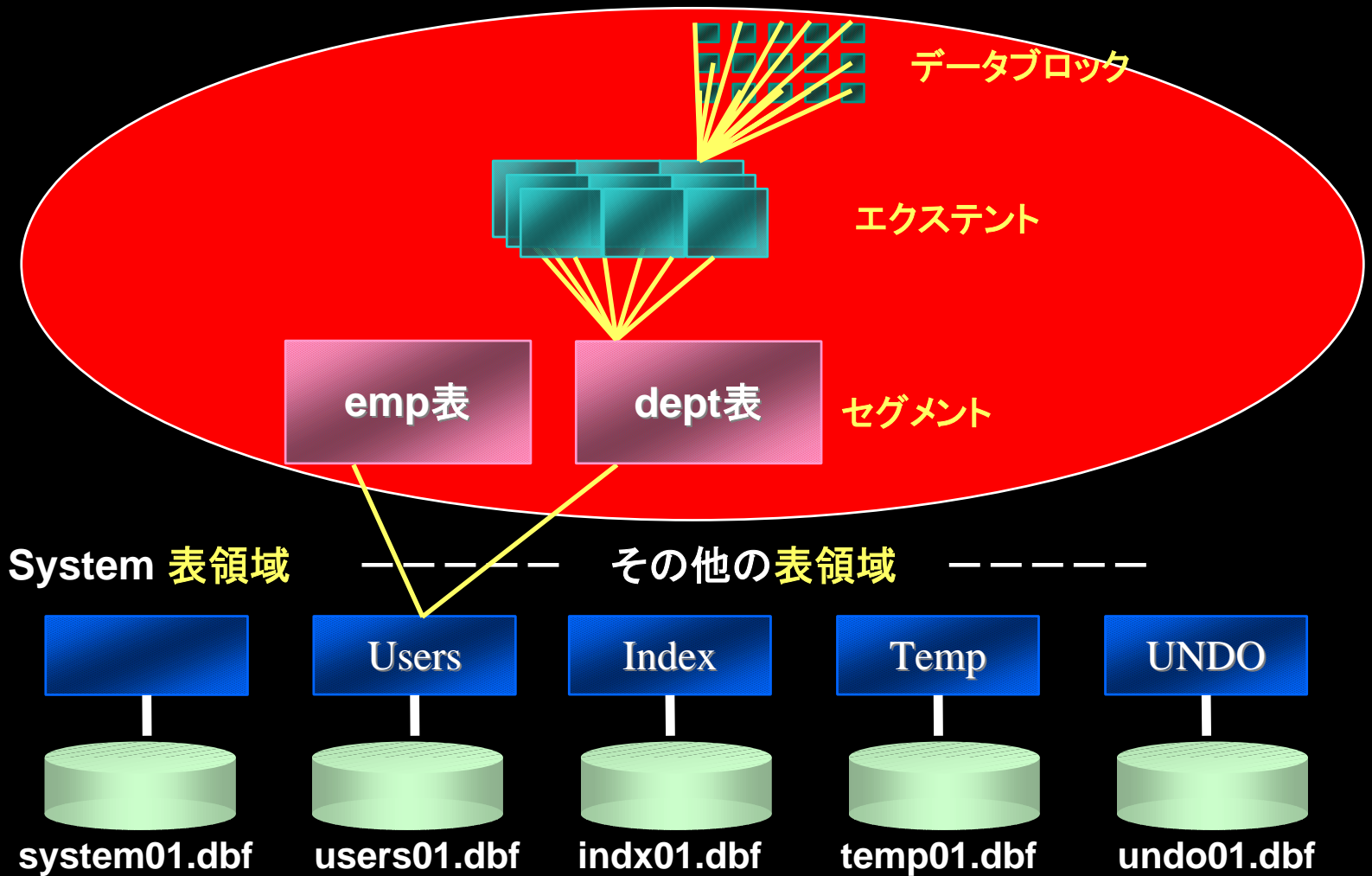


表領域とセグメントの関係

- 表領域は、複数のセグメントから構成される。



記憶領域管理



セグメントとその構成要素

セグメント

一つの表や索引などのオブジェクトが使用する領域

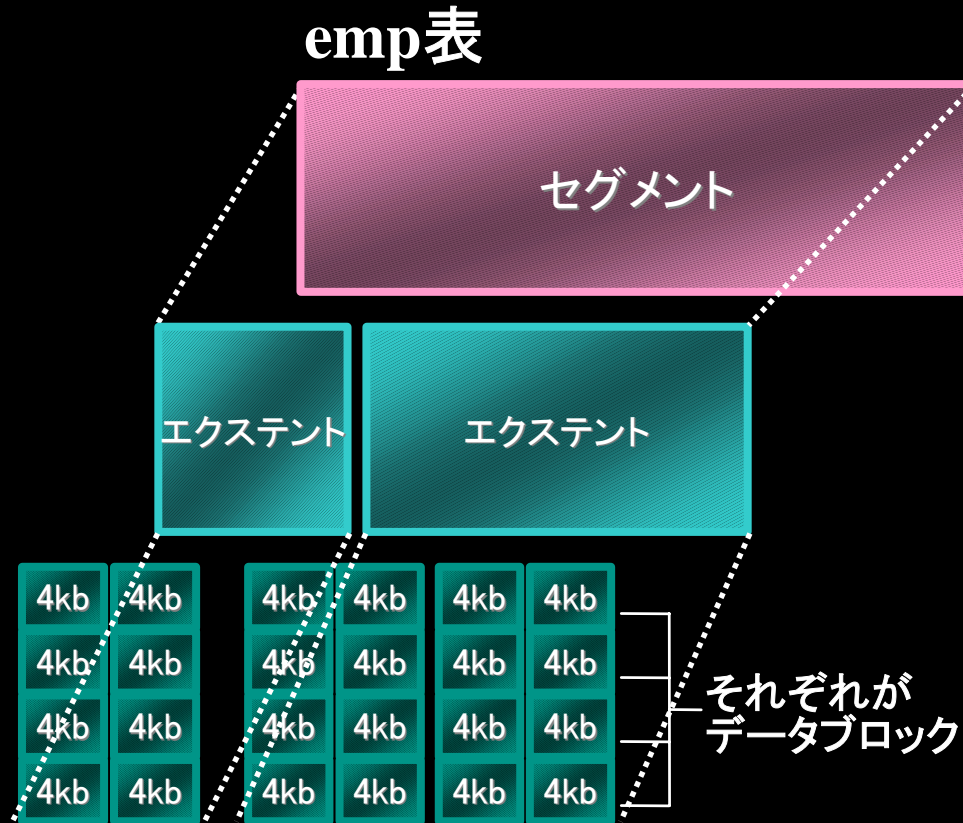
エクステント

セグメントの中の連続したデータブロックで構成される領域

データブロック

Oracleが一度にデータをI/O(読み書き)する最小の単位

OLTP系、DWH系でサイズを変える



System表領域

- 必須の表領域
 - データディクショナリ(データベース全体の情報)
 - ビュー、シノニム、順序などの情報
 - パッケージ、ストアドプロシージャ、ストアドファンクシオン
 - データベーストリガー
 - などが格納されている
- ユーザーのテーブルやインデックスを格納することも可能ですが、好ましくありません

Undo表領域

- ロールバックデータの格納用の表領域
 - Oracle8iまでは「ロールバックセグメント」
 - ロールバックデータの管理
 - ロックを使用しない読み取り一貫性
 - トランザクションのロールバック
 - データベースのリカバリ
 - Oracle9iから自動管理
 - データの保持期間を設定
 - フラッシュバッククエリに使用

一時表領域

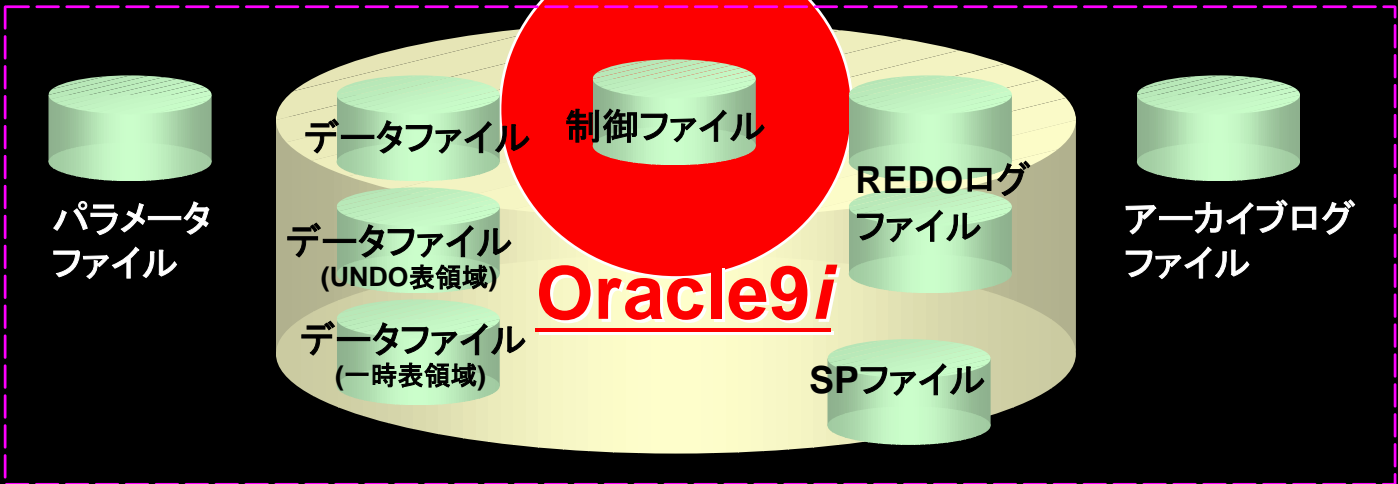
- ソート処理の時に使用する領域
 - DatabaseのDefault一時表領域の利用(Oracle9i)
 - Oracleのソート処理
 - 最初はメモリ(sort_area)を使用してソートする
 - メモリ領域でソートしきれない場合、一時表領域を使用する

制御ファイル



データ・Redoログファイルに関する名称や状態、チェックポイント番号などデータベース全体の整合性を維持するために必要な情報が格納されます。

ディスク上のファイル



REDOログファイル

ユーザ
プロセス

サー
プロ

インスタンス

JAVAプール

SGA

ラージ・プール

共有プール

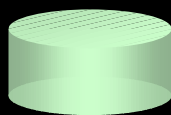
LGWR

LGWR

ARCH

データベースに対する更新履歴をすべて記録します。障害発生時にバックアップファイルと更新履歴により直前のcommitまでデータを回復することが可能となります。

ディスク上
のファイル



パラメータ
ファイル



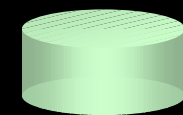
データファイル



制御ファイル



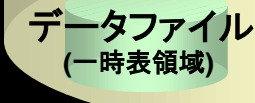
REDOログ
ファイル



アーカイブログ
ファイル



データファイル
(UNDO表領域)

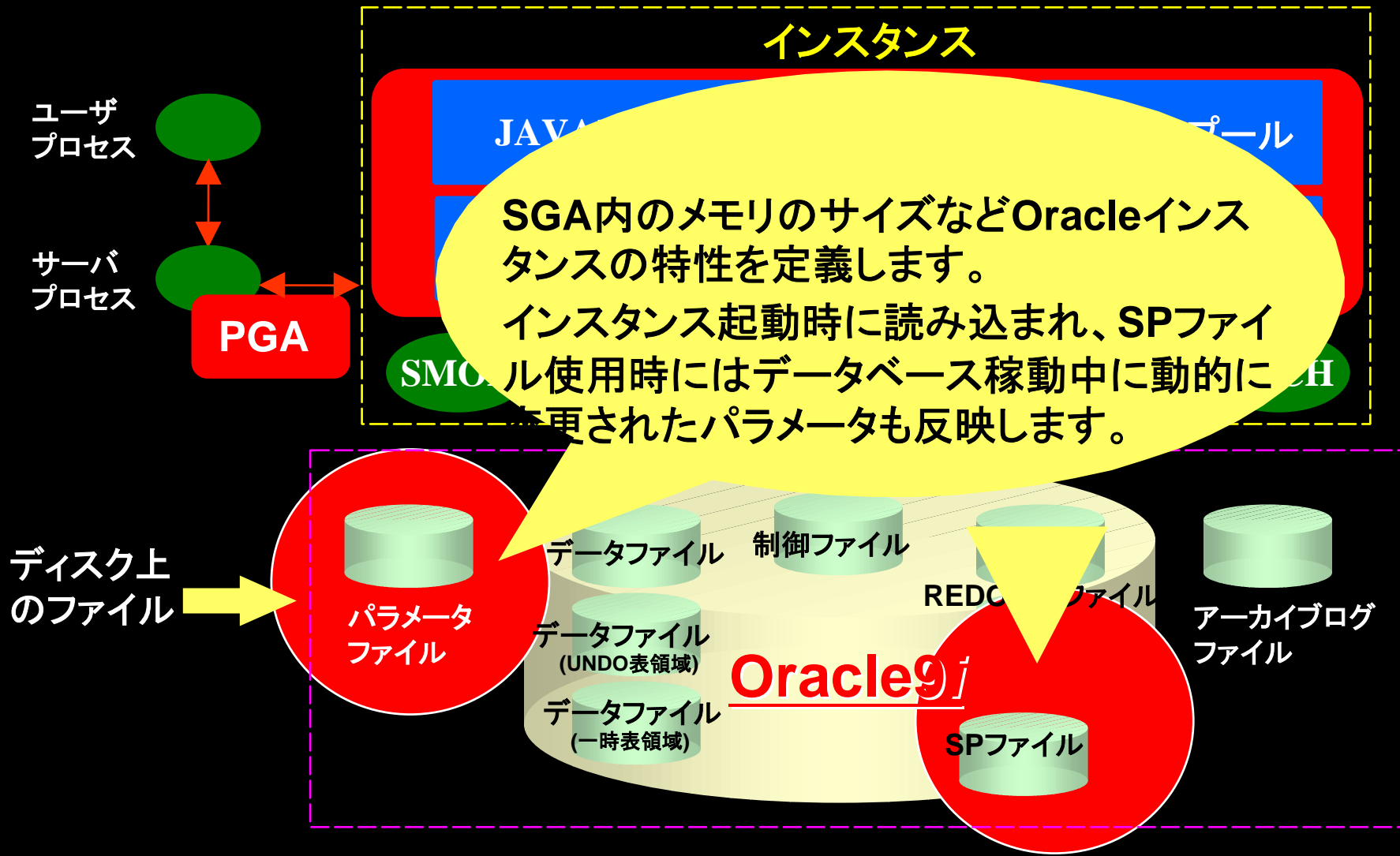


データファイル
(一時表領域)

Oracle9i

SPファイル

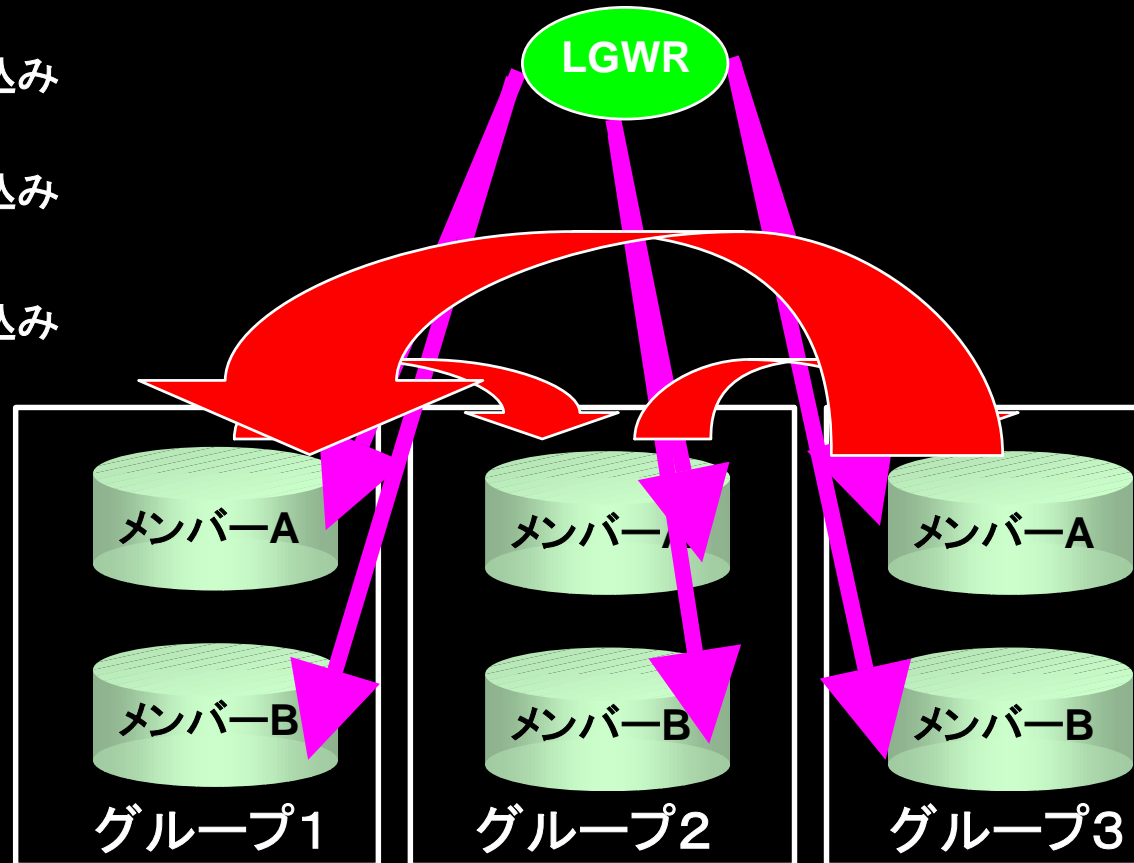
パラメータファイル



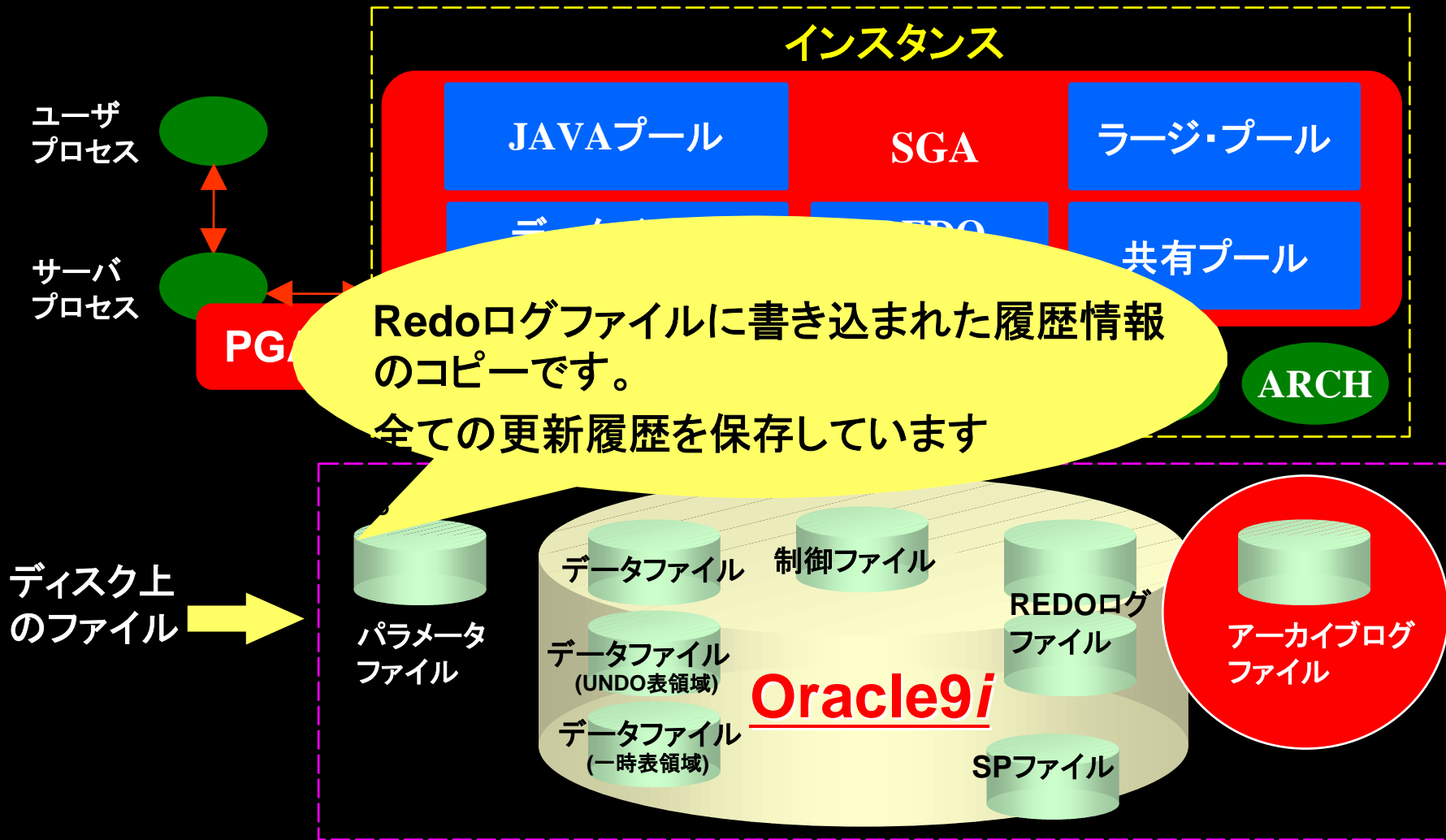
REDOログファイル

ログ書込みの動作

1. グループ1のメンバーに書込み
→ ログスイッチが発生
2. グループ2のメンバーに書込み
→ ログスイッチが発生
3. グループ3のメンバーに書込み
→ ログスイッチが発生
4. グループ1のメンバーに書込み

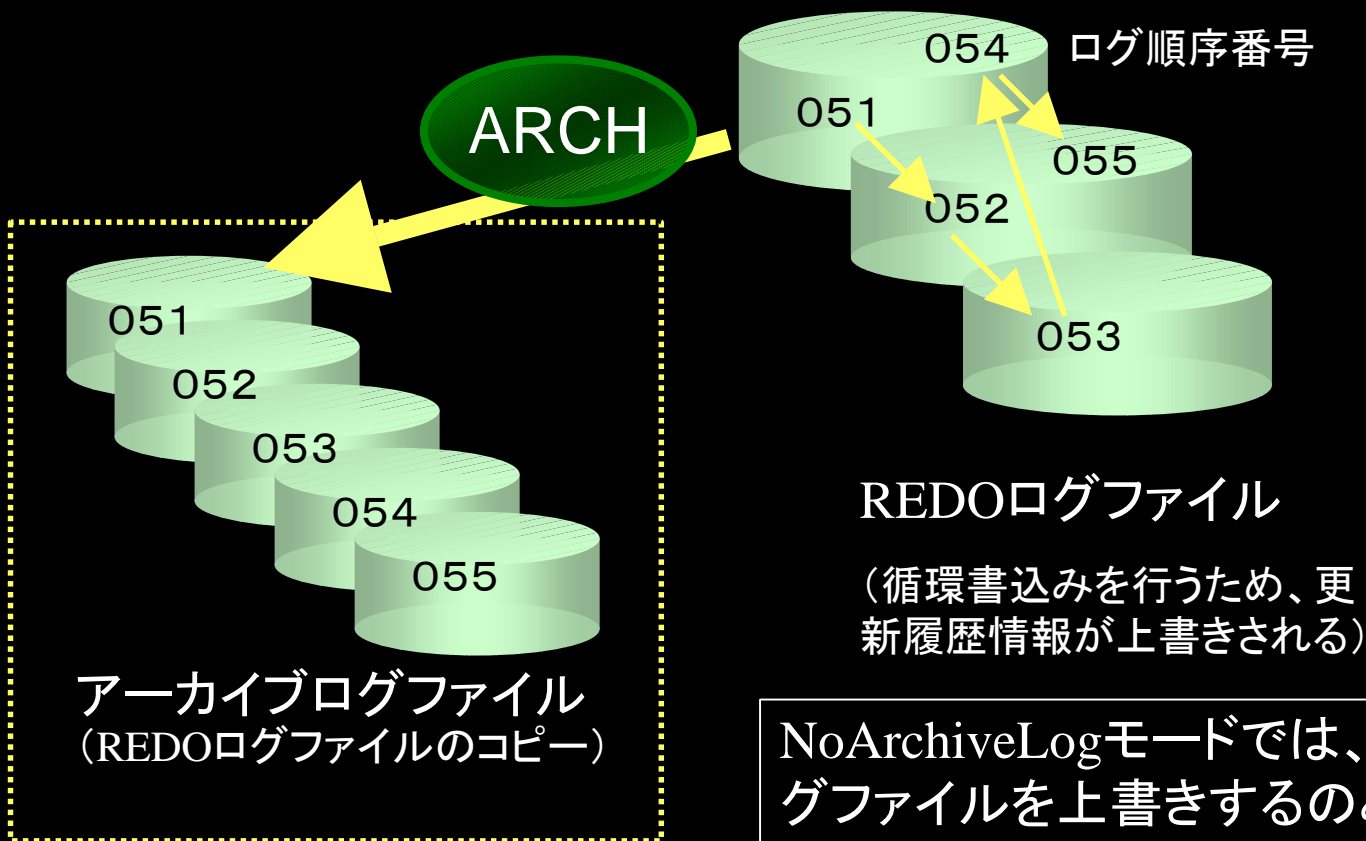


アーカイブログファイル

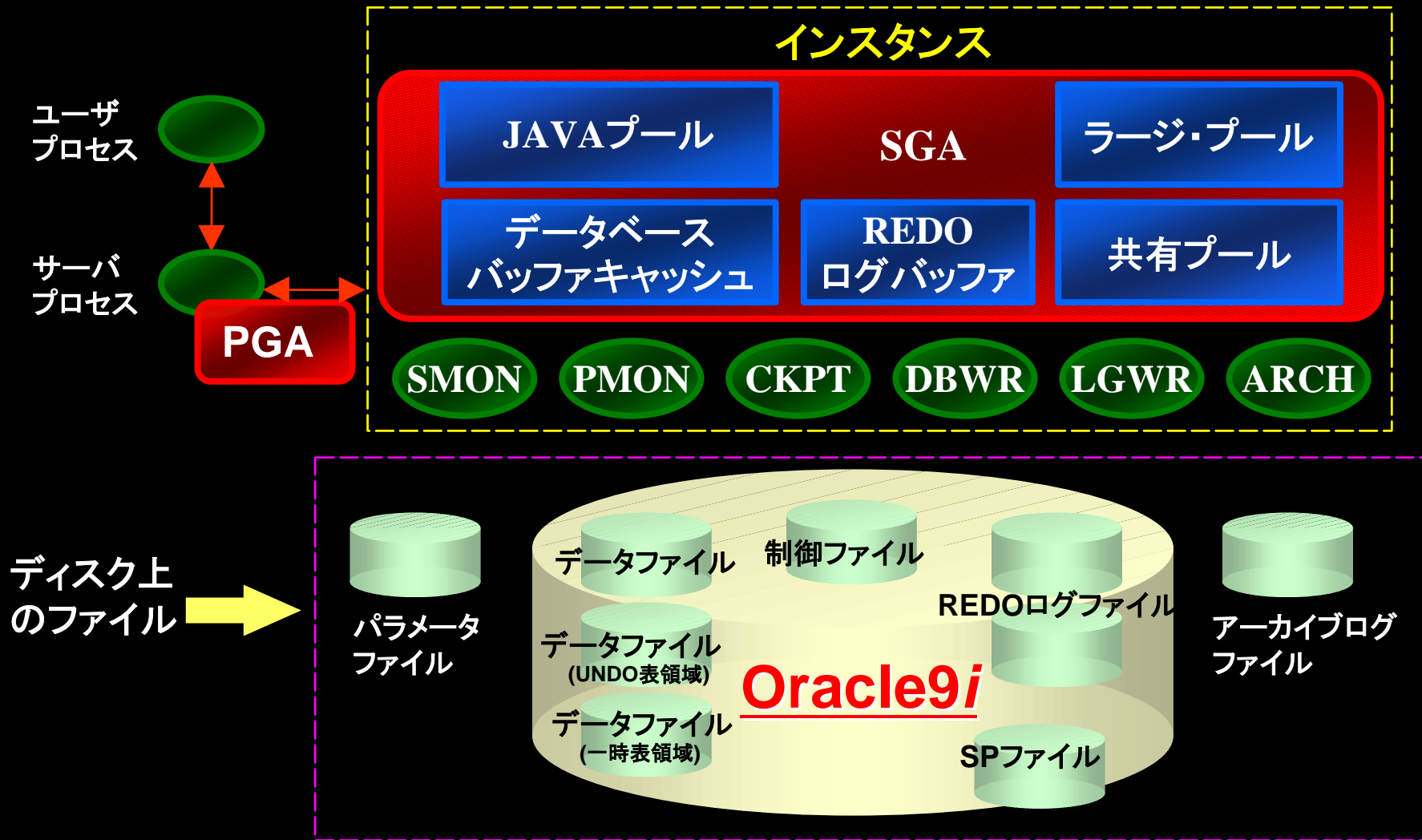


アーカイブログファイル

ArchiveLogモードでの運用



アーキテクチャ概要



SGA(System Global Area)



Javaプール/ラージプール/共有プール

ディクショナリ情報/解析済みSQL文などを保持するメモリ

データベースバッファキャッシュ

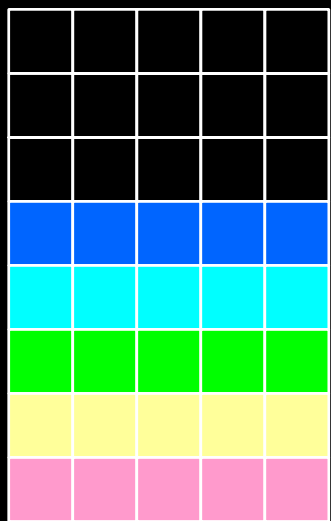
データファイルから読み込んだデータを保持するメモリ

Redoログバッファ

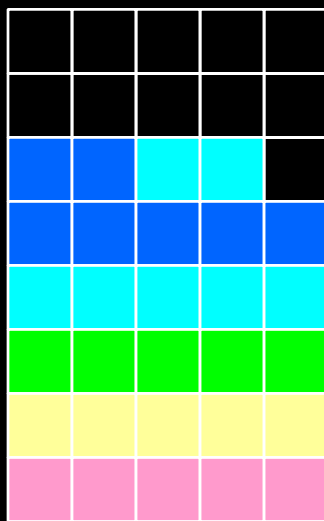
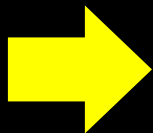
データベースに対する更新履歴の一時格納メモリ

SGA(System Global Area)

SGA合計
(SGA_MAX_SIZE)



OEMで動的にサイズ変更



SGA_MAX_SIZEの範囲内での動的な変更(縮小・拡張)可能



インスタンス稼動中の動的な変更は不可



□ = グラニュール



グラニュール単位で領域が確保される

バックグラウンド・プロセス



SMON

システムを監視し、起動時にインスタンス回復などを行う

PMON

プロセス群を監視し、異常終了した接続をクリーンアップ

CKPT

チェックポイント発生時にDBWRへシグナルを送る

バックグラウンド・プロセス



DBWR

データベースバッファキャッシュ上の更新済みデータをデータファイルに書込む

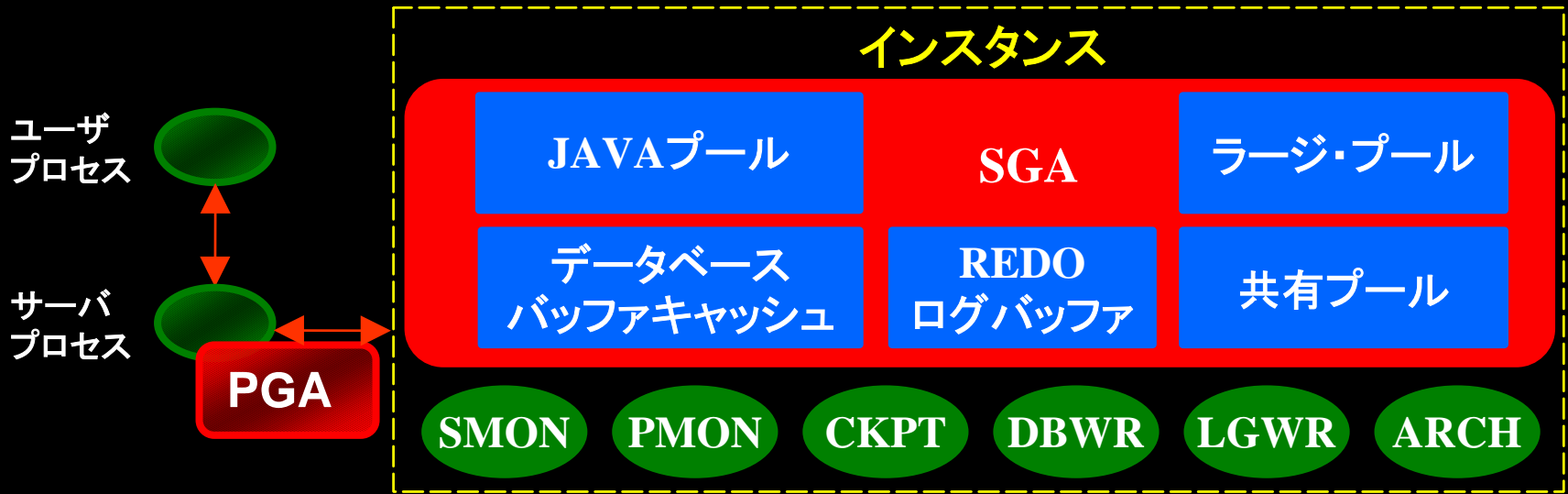
LGWR

Redoログバッファ上の更新記録をRedoログファイルに書込む

ARCH

アーカイブログファイルにRedoログファイルのコピーを書込む

ユーザプロセスとサーバプロセス

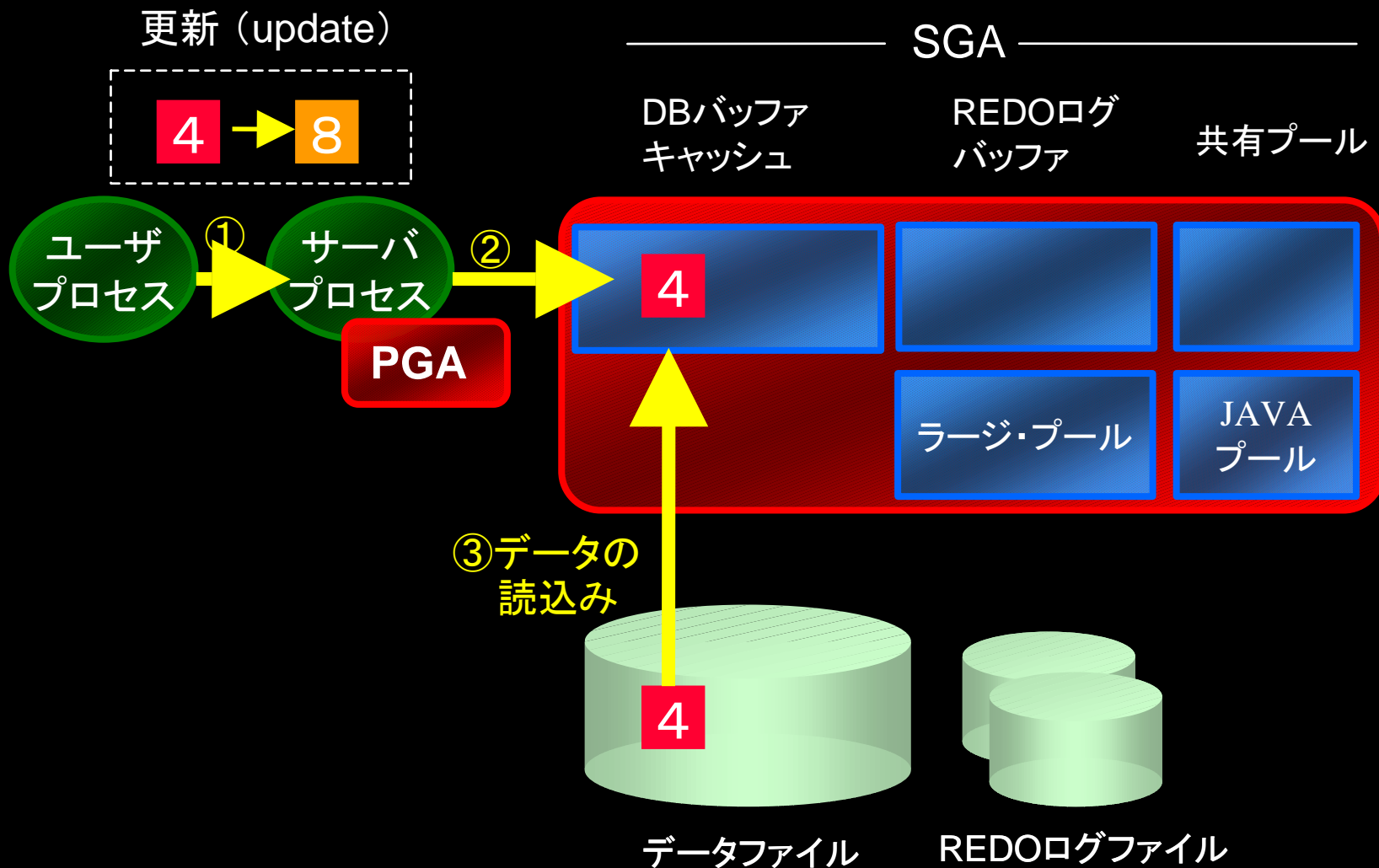


サーバプロセスはインスタンスに含まれず、インスタンスとユーザプロセスの間の命令のやり取りを行い、基本的に一つの接続ごとに一つ必要となる

PGA(Program Global Area)

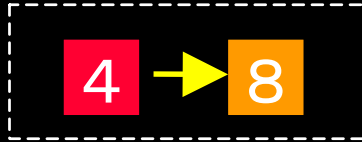
サーバプロセス毎に存在し、処理結果のソート用の領域や接続情報などを保持
データベース全体で使用するPGAのサイズの指定が可能。

データ更新時の動作（読込み）



データ更新時の動作（更新履歴の記録）

更新 (update)



SGA

DBバッファ
キャッシュ

REDOログ
バッファ

共有プール

ユーザ
プロセス

サーバ
プロセス

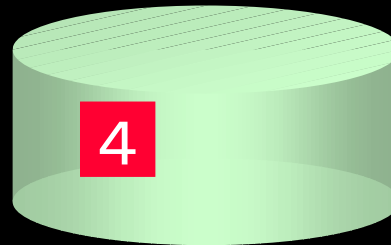
PGA

⑤更新前イメージをRBSに
格納し行データの更新

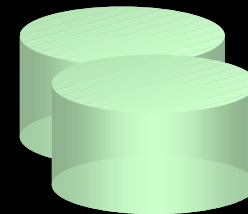
ラージ・プール

JAVA
プール

④更新履歴を記録



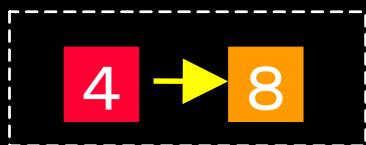
データファイル



REDOログファイル

データ更新時の動作 (commit)

⑥ commit

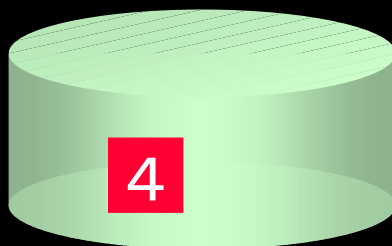


SGA

DBバッファ キャッシュ REDOログ バッファ 共有プール



データファイルに更新データは反映されていない。

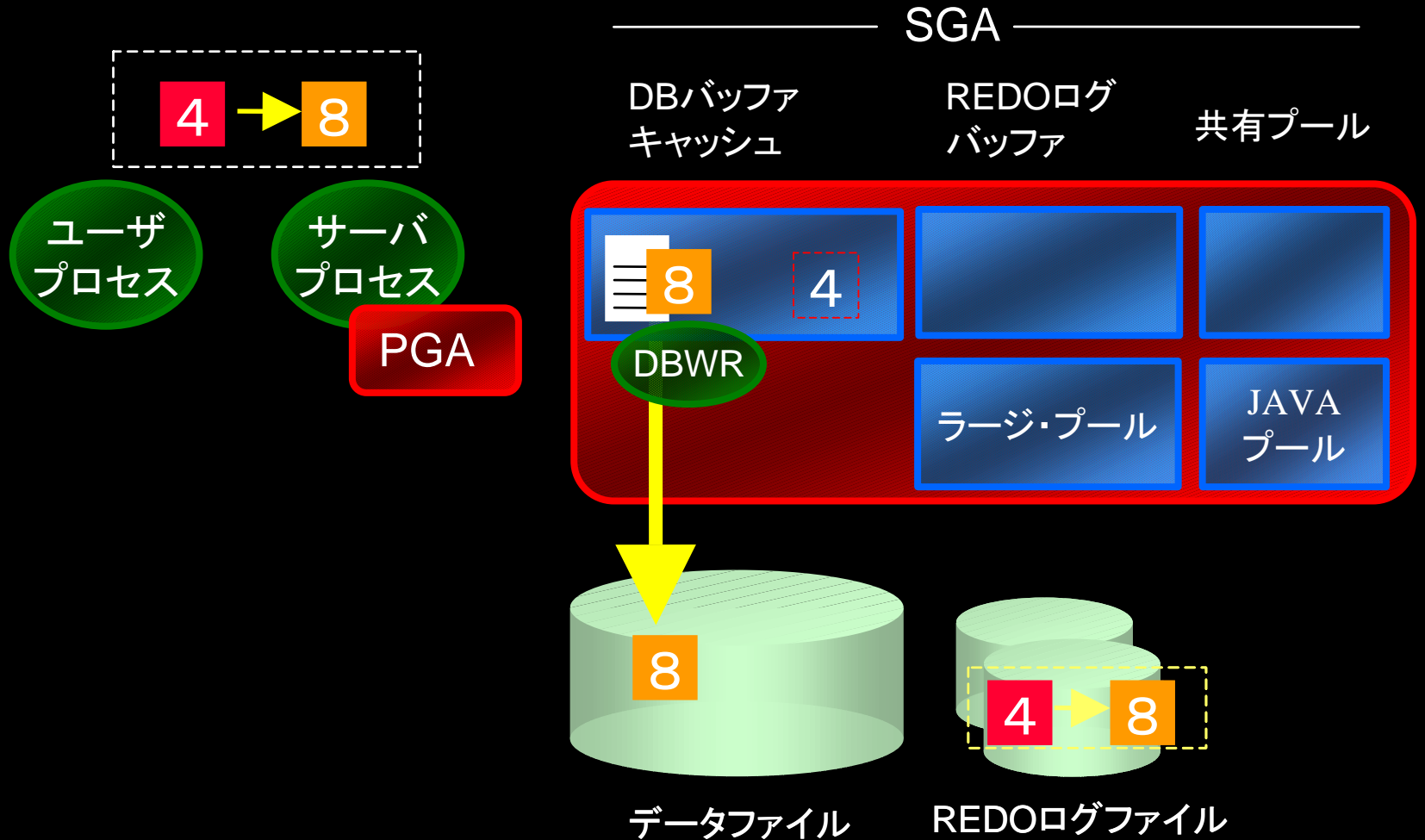


データファイル



REDOログファイル

データ更新時の動作（遅延書き込み）



データ更新時の動作 (更新中のインスタンス障害)

commit完了

4 → 8

SGA

DBバッファ
キャッシュ

REDOログ
バッファ

共有プール

ユーザ
プロセス

サーバ
プロセス

PGA

8

インスタンス障害

JAVA
プール

メモリ内の更新データが無くなる

DISK上のデータファイル
には更新データが反映
されていない

4

データファイル

4

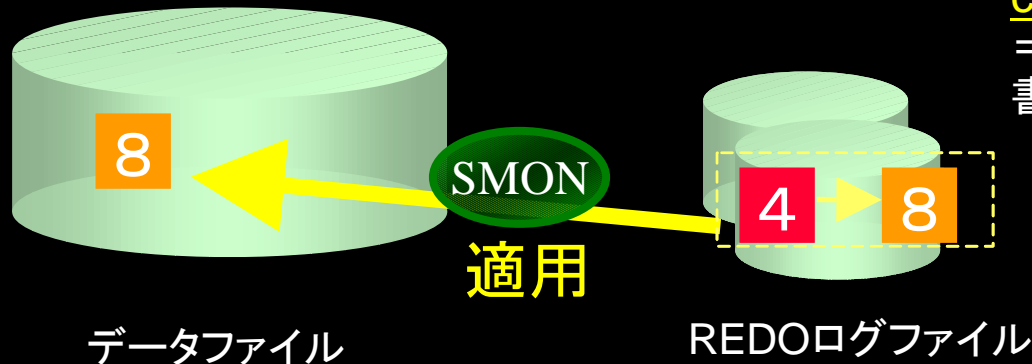
8

REDOログファイル

データ更新時の動作 (更新中のインスタンス障害)

インスタンス再起動時

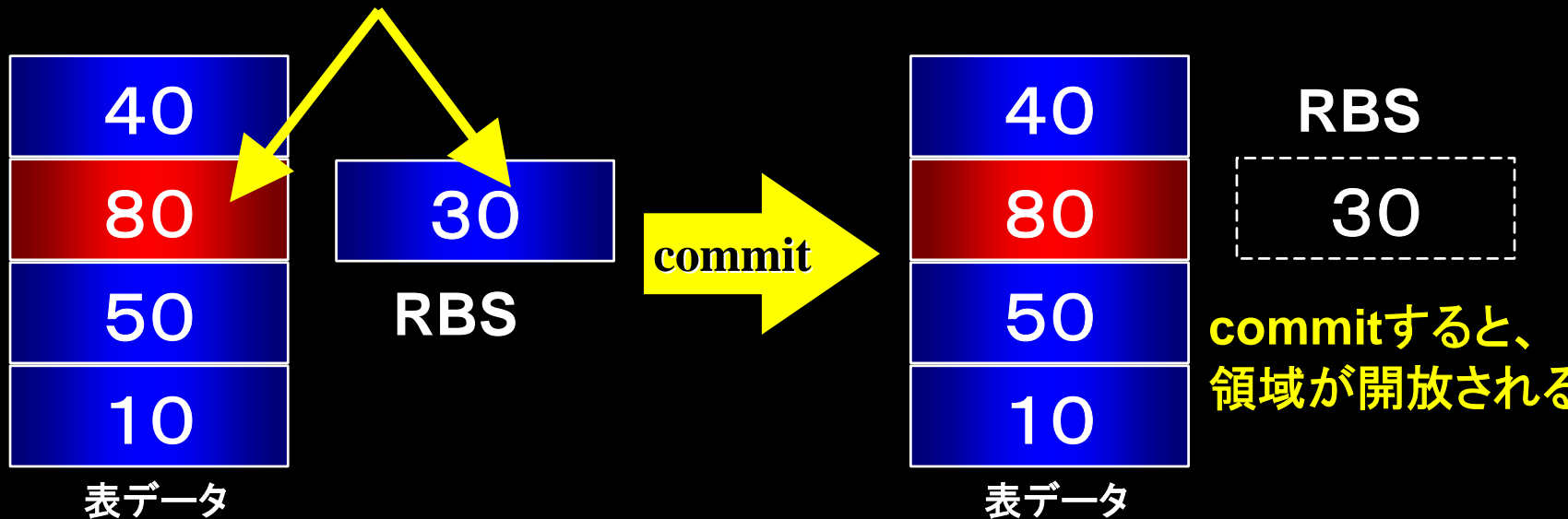
SGA



commit完了
=REDOログへの
書込みは完了

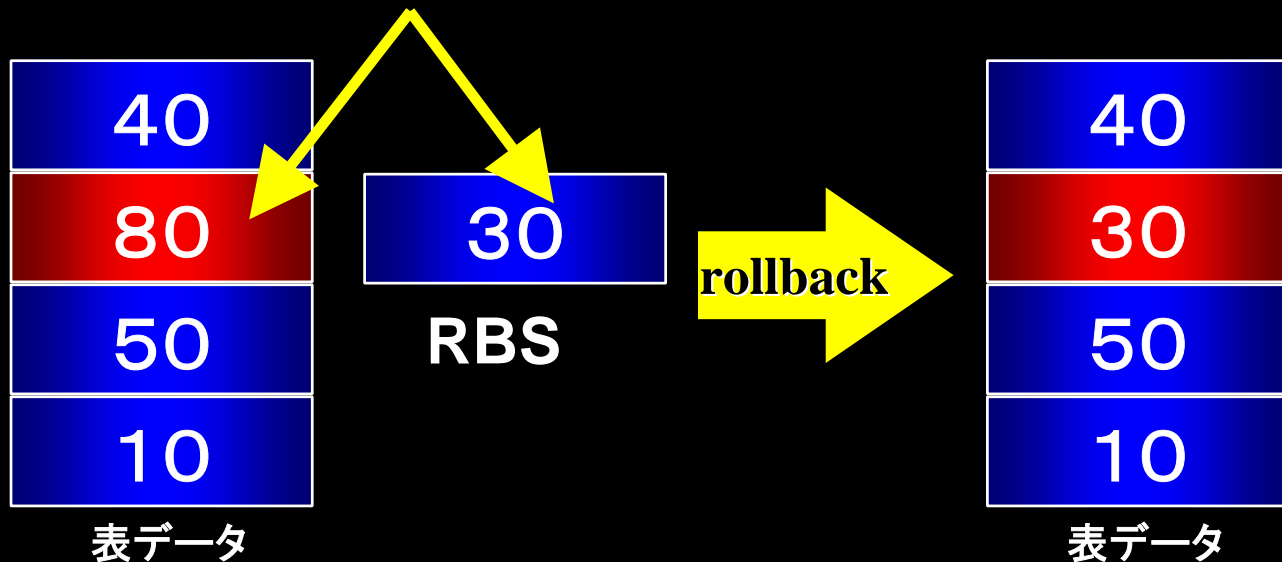
ロールバックセグメント (commit)

更新 30 → 80
(commit はまだ実行していない)



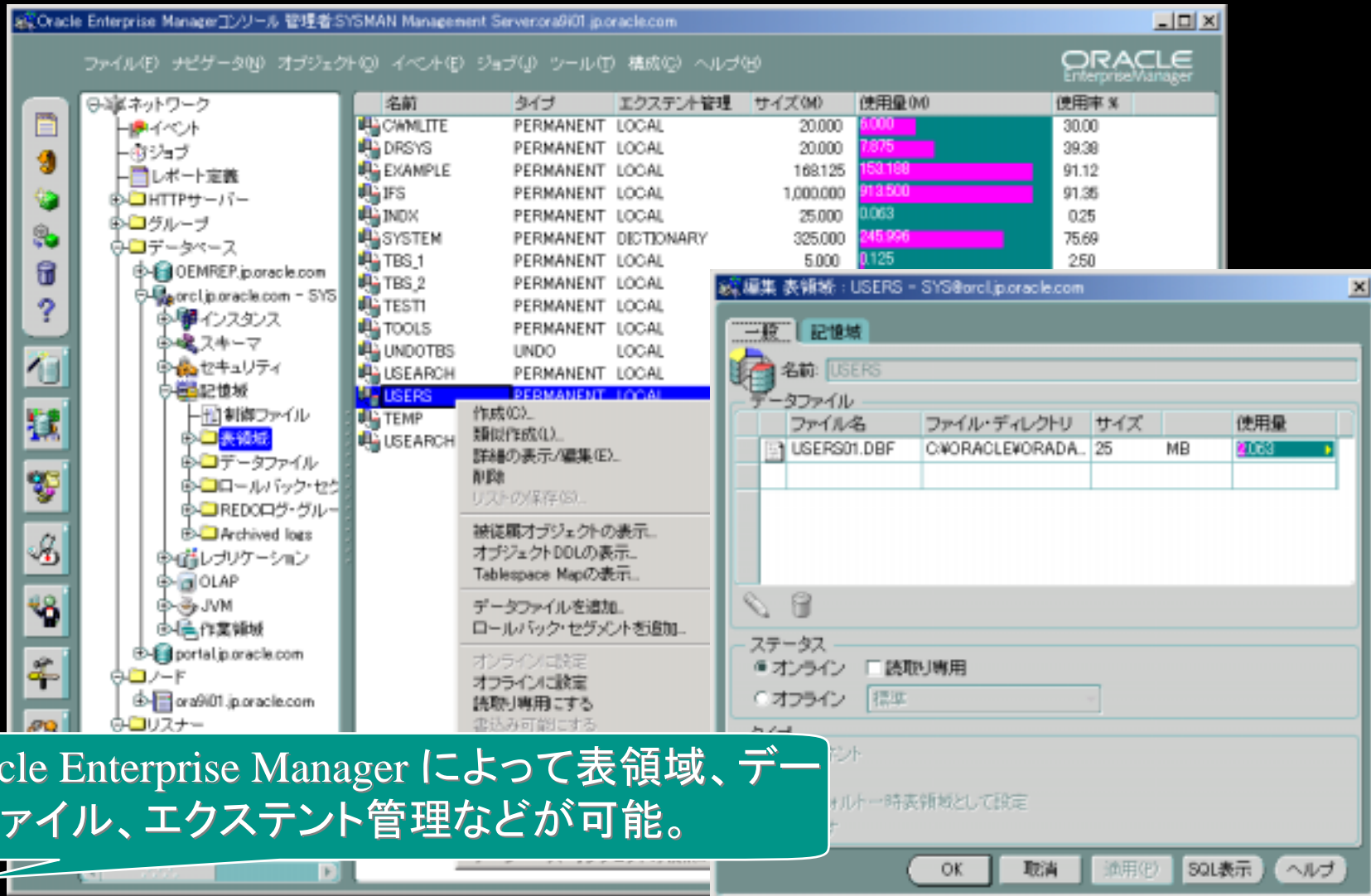
ロールバックセグメント (rollback)

更新 30 → 80
(commit はまだ実行していない)



RBSのデータを
利用し、元に戻す

参考: GUIによる記憶領域管理



Oracle Enterprise Manager GUI showing database memory management. The main window displays a tree view on the left and a table of memory segments in the center. A 'USERS' tablespace is selected, and a secondary window shows its datafiles.

名前	タイプ	エクステンツ管理	サイズ(M)	使用量(M)	使用率 %
CWMLITE	PERMANENT	LOCAL	20,000	6,000	30.00
DRSYS	PERMANENT	LOCAL	20,000	7,875	39.38
EXAMPLE	PERMANENT	LOCAL	168,125	153,100	91.12
IFS	PERMANENT	LOCAL	1,000,000	913,500	91.35
INDX	PERMANENT	LOCAL	25,000	0,003	0.25
SYSTEM	PERMANENT	DICTIONARY	325,000	245,920	75.69
TBS_1	PERMANENT	LOCAL	5,000	0,125	2.50
TBS_2	PERMANENT	LOCAL			
TEST1	PERMANENT	LOCAL			
TOOLS	PERMANENT	LOCAL			
UNDOTBS	UNDO	LOCAL			
USEARCH	PERMANENT	LOCAL			
USERS	PERMANENT	LOCAL			
TEMP					
USEARCH					

名前	記憶域
USERS	表領域

名前	データファイル
USERS01.DBF	C:\ORACLE\ORADA...

Oracle Enterprise Manager によって表領域、データファイル、エクステンツ管理などが可能。

参考: GUIによるパラメータ変更

Oracle Enterprise Manager によってパラメータを変更可能。
それぞれのパラメータの説明も表示できる。

パラメータ名	値	デフォルト	ダイナミック	カテゴリ
tape_asynch_io	TRUE	✓		バックアップおよびリストア
thread	0			
timed_os_statistics	0			
timed_statistics	TRUE			
trace_enabled	TRUE			
tracefile_identifier				
transaction_auditing	TRUE			
transactions	308	✓		トランザクション
transactions_per_rollback_segme...	5	✓		システム管理UNDOおよびロールバック
undo_management	AUTO			システム管理UNDOおよびロールバック
undo_retention	900	✓	✓	システム管理UNDOおよびロールバック
undo_suppress_errors	FALSE	✓	✓	システム管理UNDOおよびロールバック
undo_tablespace	UNDOTBS		✓	システム管理UNDOおよびロールバック
use_indirect_data_buffers	FALSE	✓		キャッシュおよび I/O
user_dump_dest	C:\oracle\admin\orcl\udump		✓	診断および統計
utl_file_dir		✓		PL/SQL

説明
説明:
UNDO_RETENTION パラメータは、データベースに保持するコミット済UNDO情報量の指定に使用されます。このパラメータ値は $UndoSpace = RD * UPS$ (ここで、UndoSpaceはUNDOブロック数を、RDはUNDO_RETENTION(秒)を、UPSは秒当りのUNDOブロック数を示す。) 値の範囲: 許可される最大値は $2 ** 32$ 秒です。

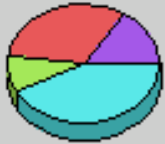
参考: GUIによるメモリ管理

データベース: 構成の編集 - SYS@ora9201.khattori02.jp.oracle.com

一般(G) **メモリ(M)** リカバリ(C) リソース・モニター(S) UNDO(U)


SGA

- 共有プール: 12 MB
- バッファ・キャッシュ: 24 MB
- ラージ・プール: 8 MB
- Javaプール: 32 MB
- SGA合計: 76.932 MB
- SGA最大サイズ: 105.069 MB



PGA

- 集計PGAターゲット: 10 MB
- 現行のPGA割当て: 14613 KB
- 最大PGA割当て(起動時):
- キャッシュ・ヒット率:

 PGAとSGAの合計なメモリを引く

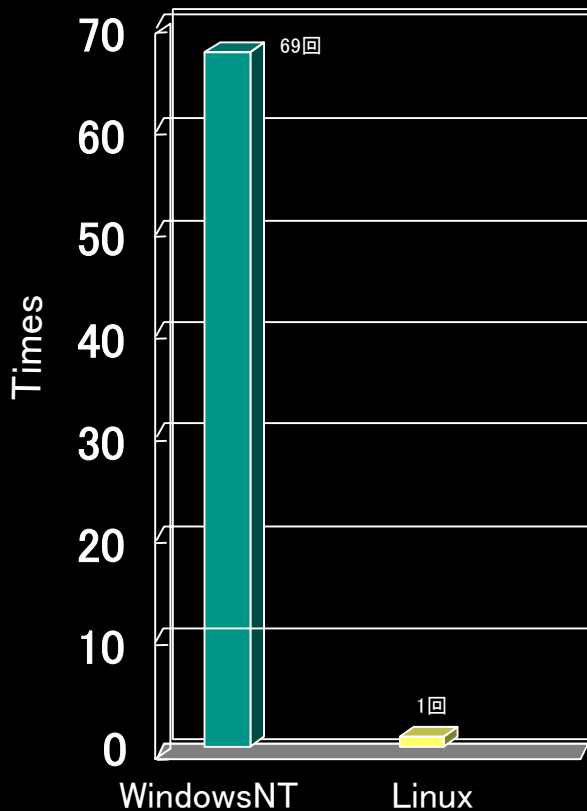
**メモリの動的な管理が可能。
また運用に合わせたバッファ・キャッシュ・サイズ・アドバイス機能がOracle9iから追加された。
PGAのサイズ変更によるキャッシュヒット率の変化もアドバイス**

Linuxのメリット

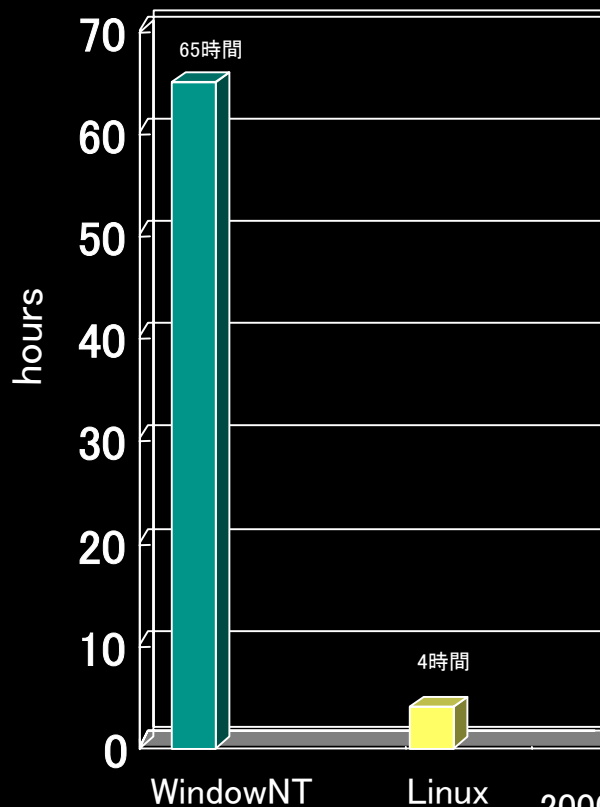
- 高い信頼性
 - 連続運転に強い(メモリ・リーク等による再起動は不要)
 - 障害原因の究明可能
- 高い性能
 - Linuxカーネルの急速な進化
- 圧倒的に安い導入コスト

高い信頼性・安全性

サーバー障害回数/1年



サーバーダウンタイム/1年



2000/01 Bloor Resaerch

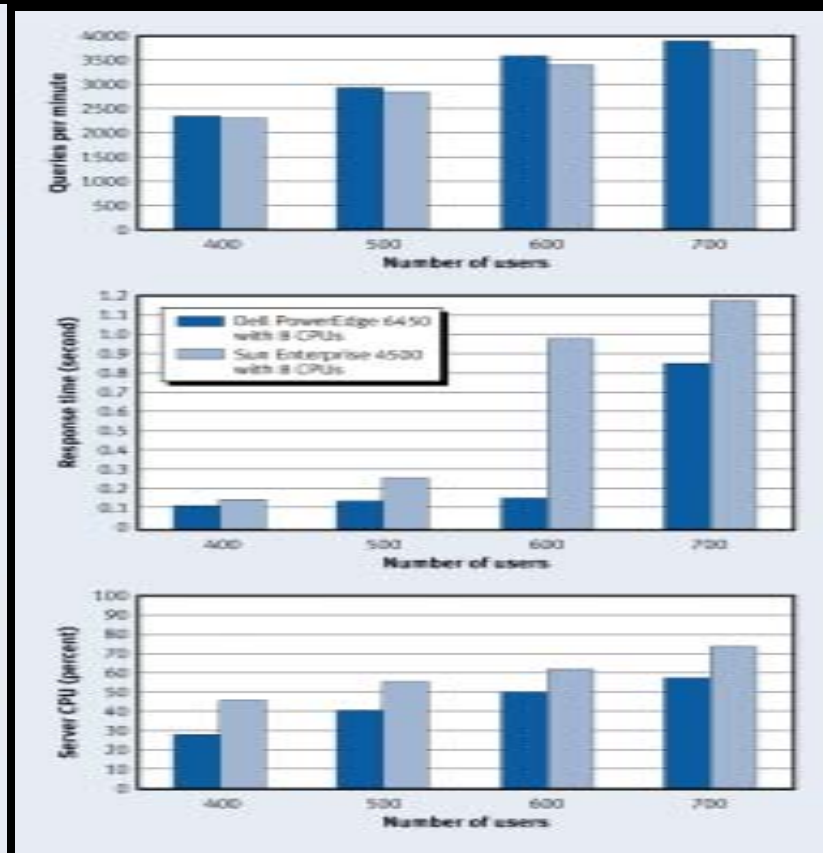
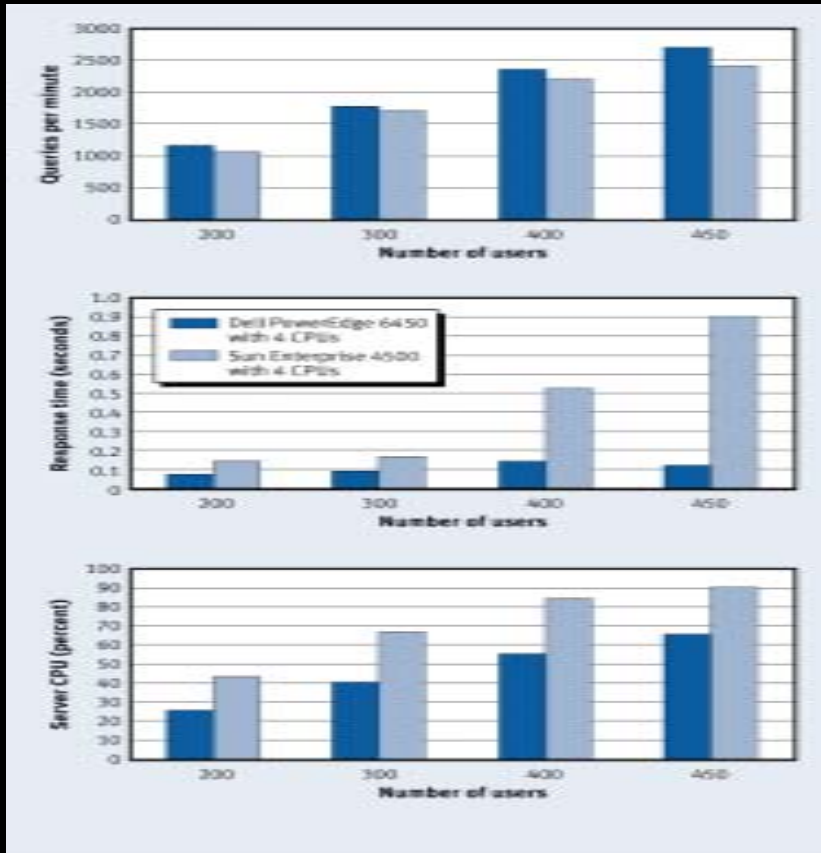
<http://gnet.dhs.org/stories/bloor.php3>

高い信頼性・安全性 ～Linuxの特徴～

- UNIXクローン
- マルチタスク
 - 様々な処理を平行して行うことが可能
- マルチユーザ
 - 複数ユーザの同時使用が可能
- 進化の早さ
 - セキュリティーフィックス
 - exec-shield等、新しい試み

データベースプラットフォームとして高い性能 ~INTEL vs SPARC on Oracle~

データベースやファイルサーバの性能はCPUクロックにほぼ比例



データベースプラットフォームとして高い性能 ～Linux vs Windows～

最新データベースベンチマーク結果
TPC-Cの結果(2002/09/16現在)

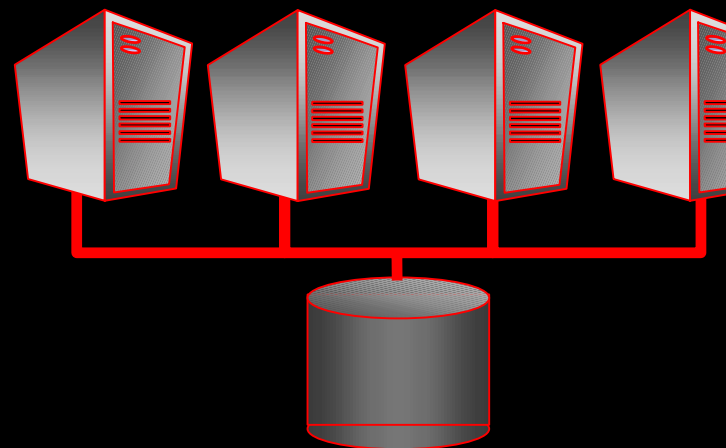
Oracle9iDB Release2
on Linuxでの結果

TPC-C スループット	138,362
価格 / tpmC	17.21 US \$

Oracle9iDB Release2
on Windowsでの結果

TPC-C スループット	137,260
価格 / tpmC	18.46 US \$

<http://www.tpc.org>

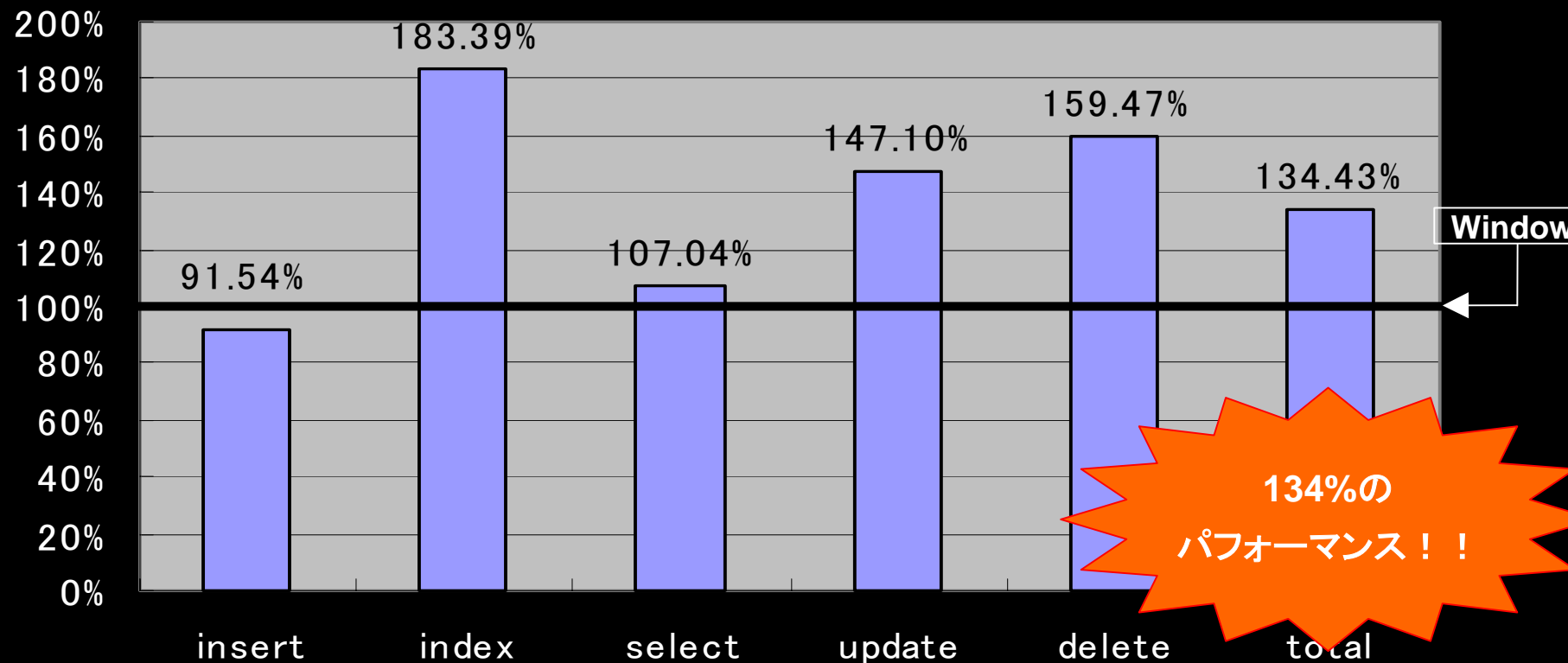


ハードウェア：
HP DL580 4CPU x 8ノード

データベースプラットフォームとしても高い性能

~Linux vs Windows~
Oracle9i DB (Linux vs Windows)

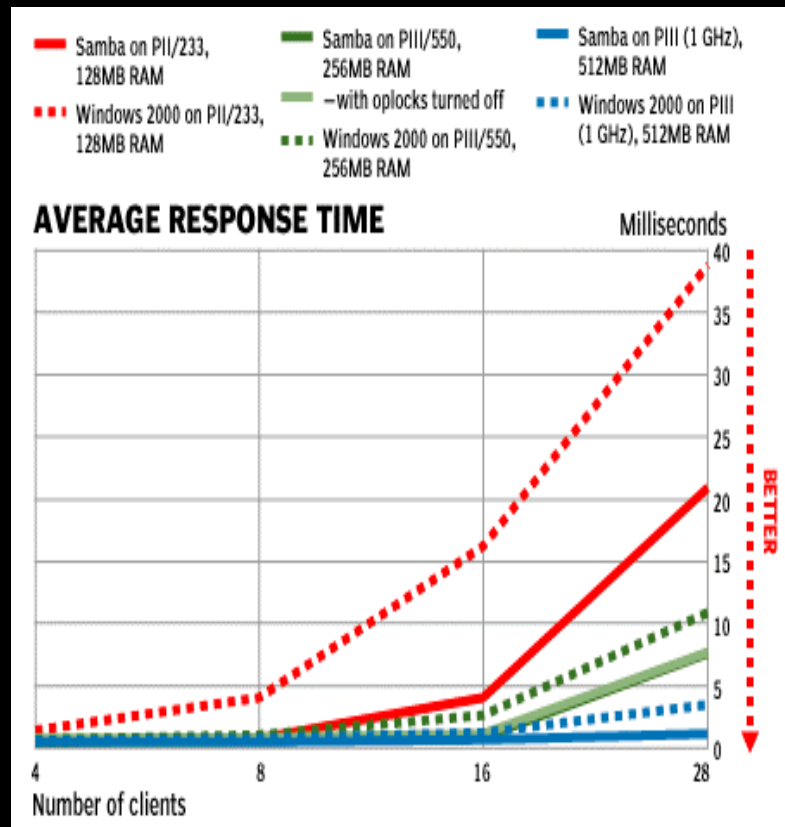
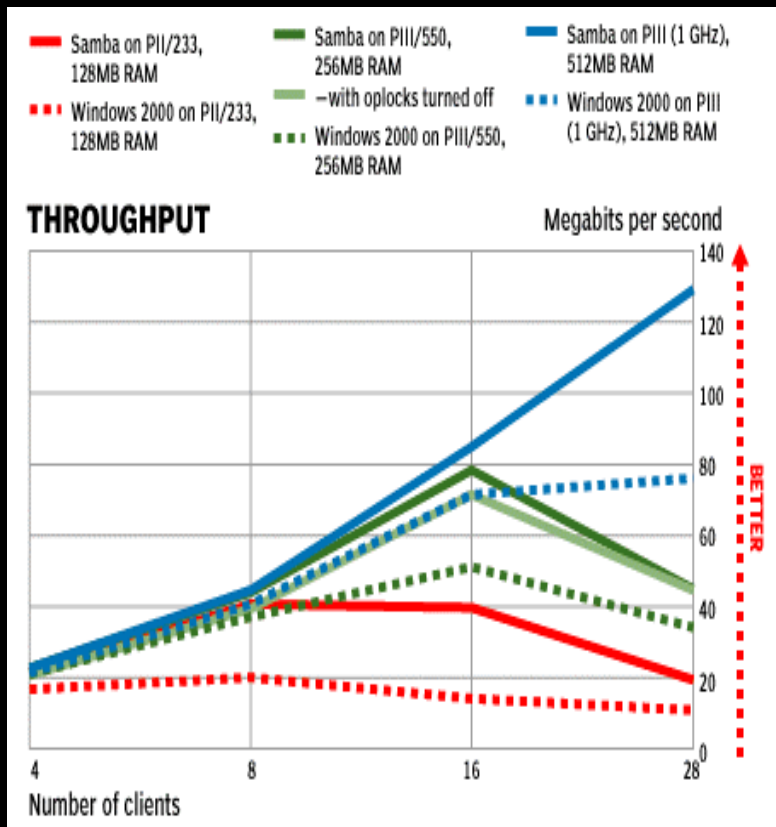
Windowsの処理速度に対する
MIRACLE LINUXの処理速度の割合



ファイルサーバ性能向上

Windows 2000よりLinux+Sambaは高性能

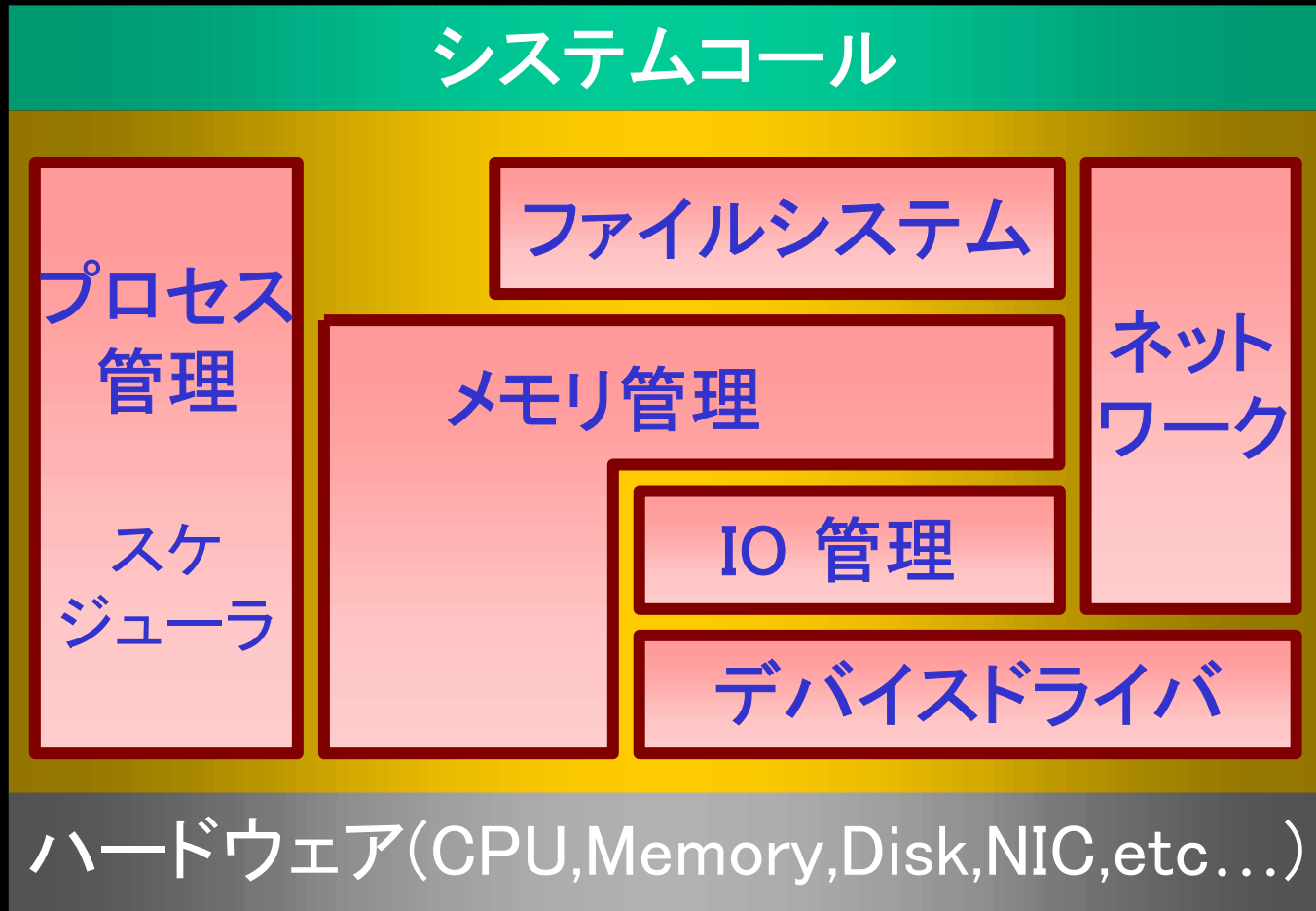
<http://www.pcmag.com/article/0,2997,s%253D1474%2526a%253D16554,00.asp>



Linuxのコア:カーネル

- カーネルとは
 - ソフトウェアの中で最もハードウェアと密接に関わっている部分
 - ハードウェアを直接コントロールしているソフトウェア

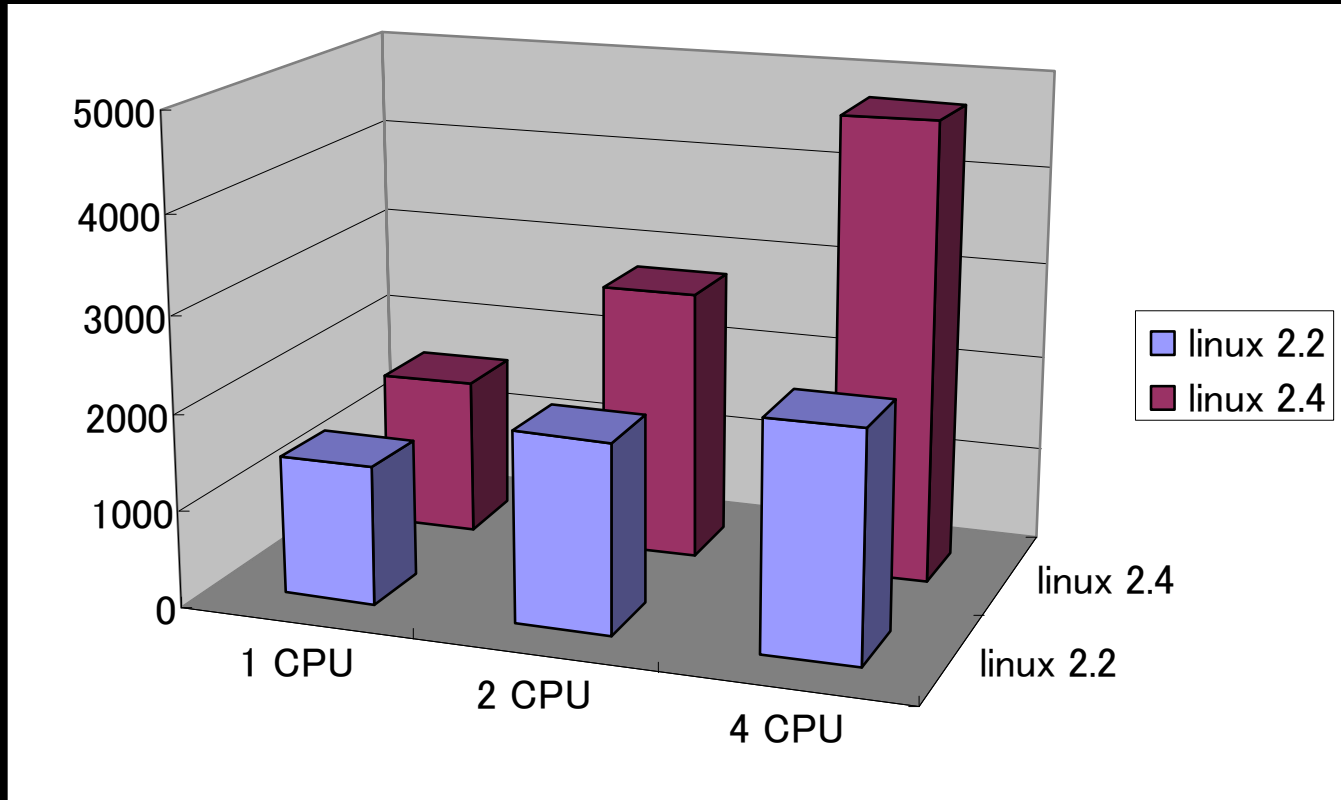
Linuxカーネルの構造



進化するLinuxカーネル

～Linux 2.2 と2.4 のスケーラビリティ～

- WebBench で測定



進化するLinuxカーネル

～更に機能強化されたカーネル～

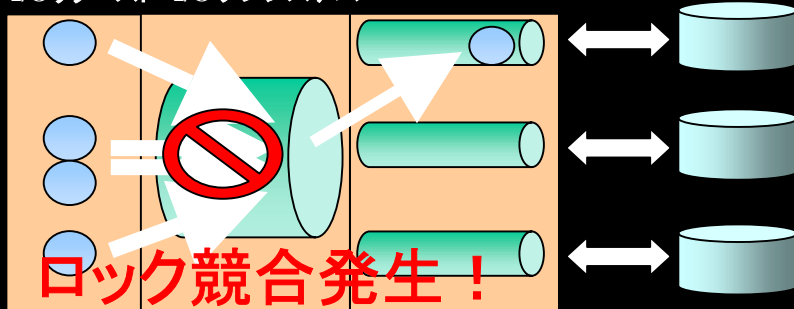
- エンタープライズ領域で要求されるスケーラビリティを実現
- Oracle 9iR2 と組み合わせることによって、より大規模な DBシステムを構築可能
 - 非同期I/Oサポート
 - VLM (Very Large Memory) 機能
 - O(1) プロセス スケジューラの導入
 - I/Oリクエストロックの細分化

OSスケールABILITY向上

I/Oリクエストロックの細分化

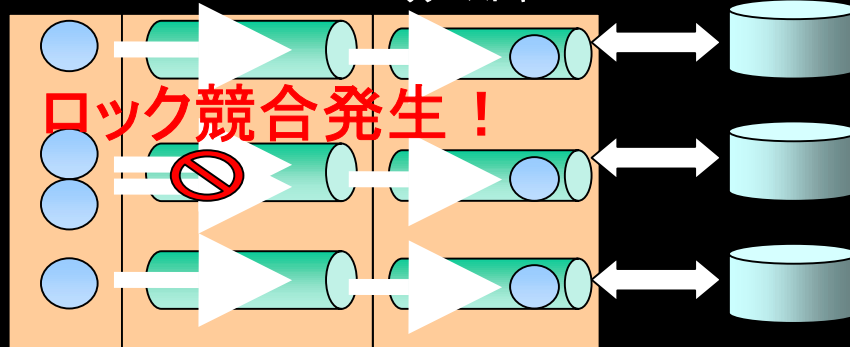
カーネルがI/Oリクエストを制御するためのロックを細分化することにより、多数のデバイスを使用するような大規模システムにおいて、スケールABILITYが向上します

I/Oリクエスト I/Oサブシステム リクエストキュー



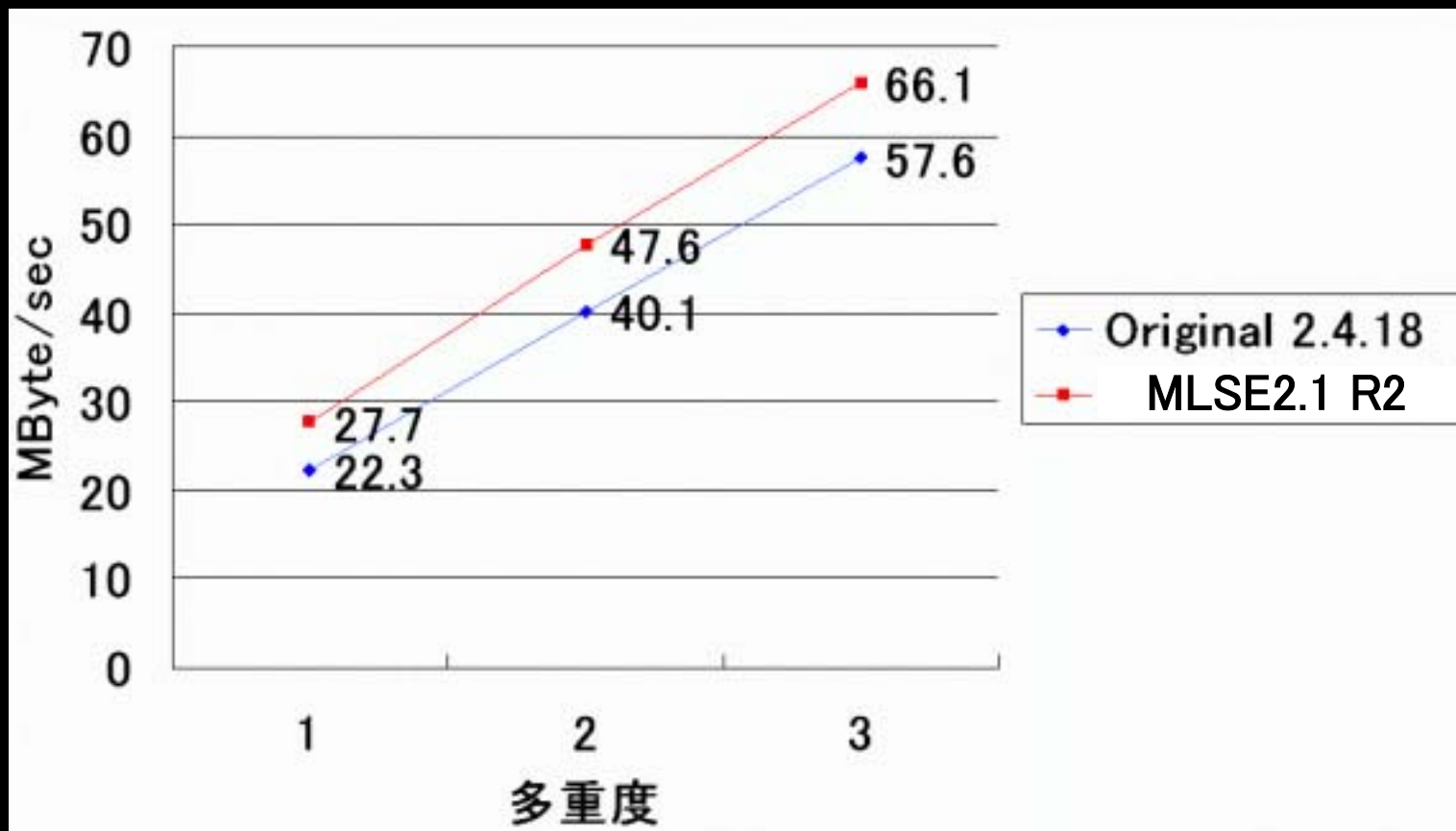
ロック競合頻度の減少によってI/O性能が向上

I/Oリクエスト I/Oサブシステム リクエストキュー



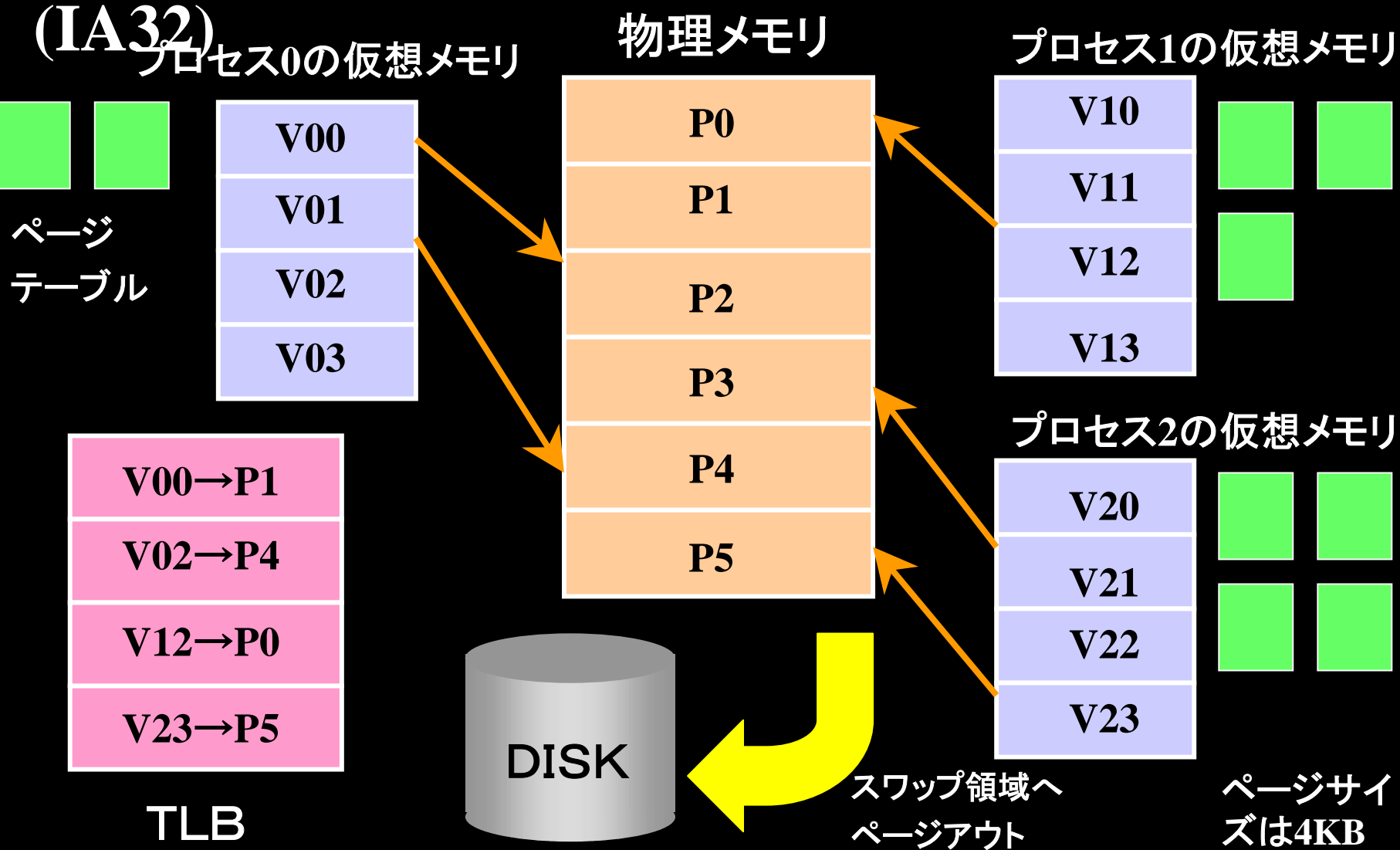
I/Oリクエストロックの細分化

- 効果測定結果



仮想アドレスから物理アドレスへのマッピング

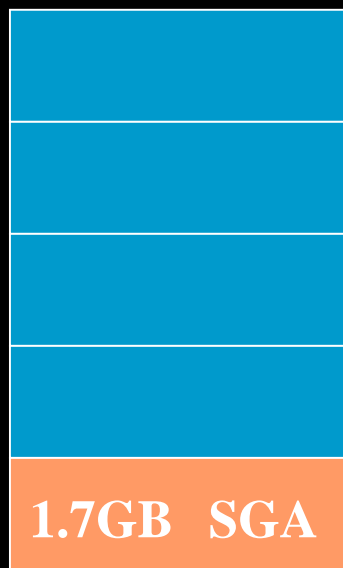
(IA32)



Oracle VLM 機能

- 大規模メモリを Oracle のデータベースバッファキャッシュとしてフルに利用可能
 - 従来ではSGAのサイズは最大約1.7GBという制限
 - MLSE2.1R2+Oracle9iR2 ではメモリファイルシステム使って物理メモリをフルに利用可能

64GBメモリの
システムの場合



従来の Linux + Oracle



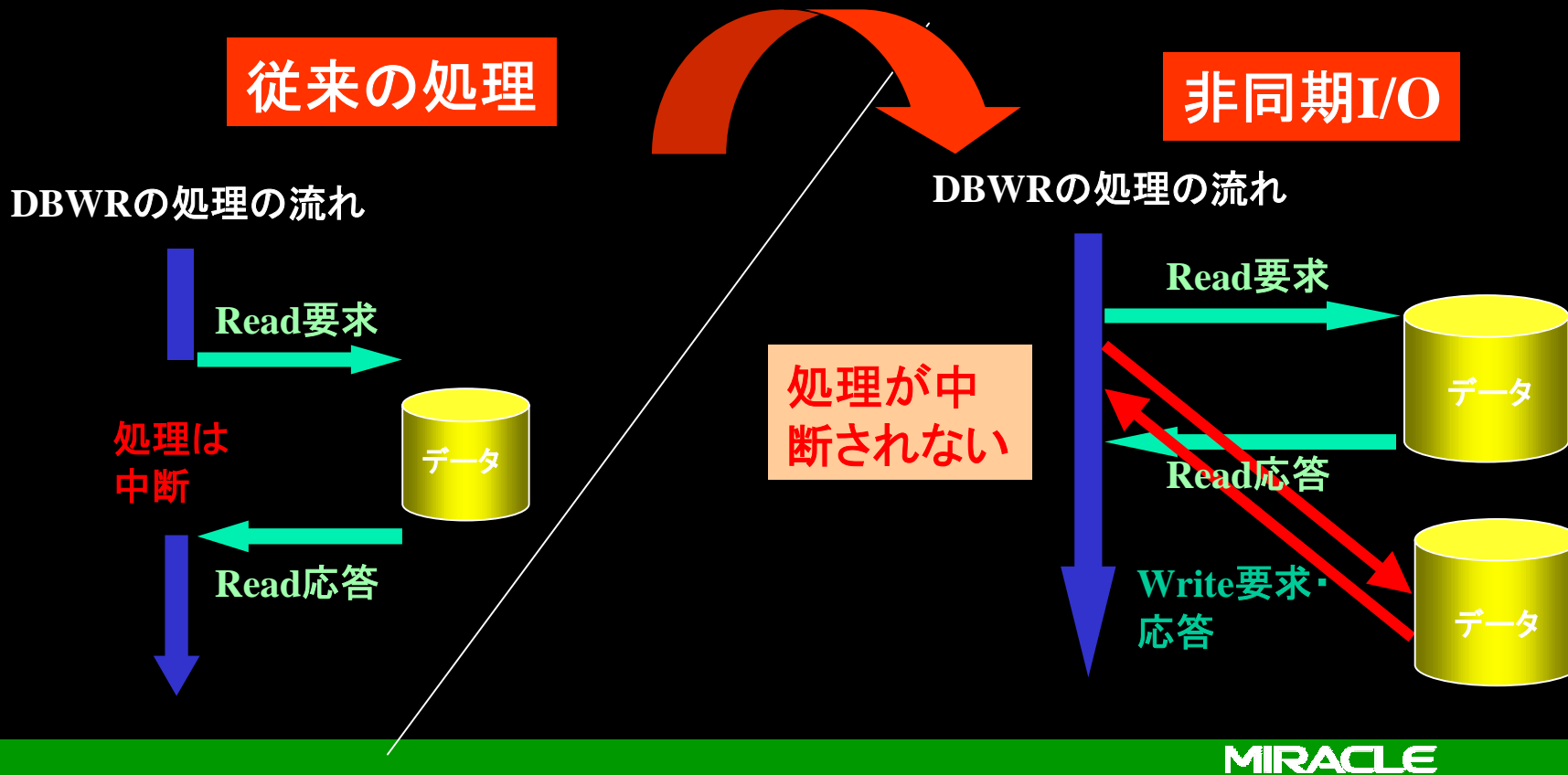
ディスクI/O
の大幅削減



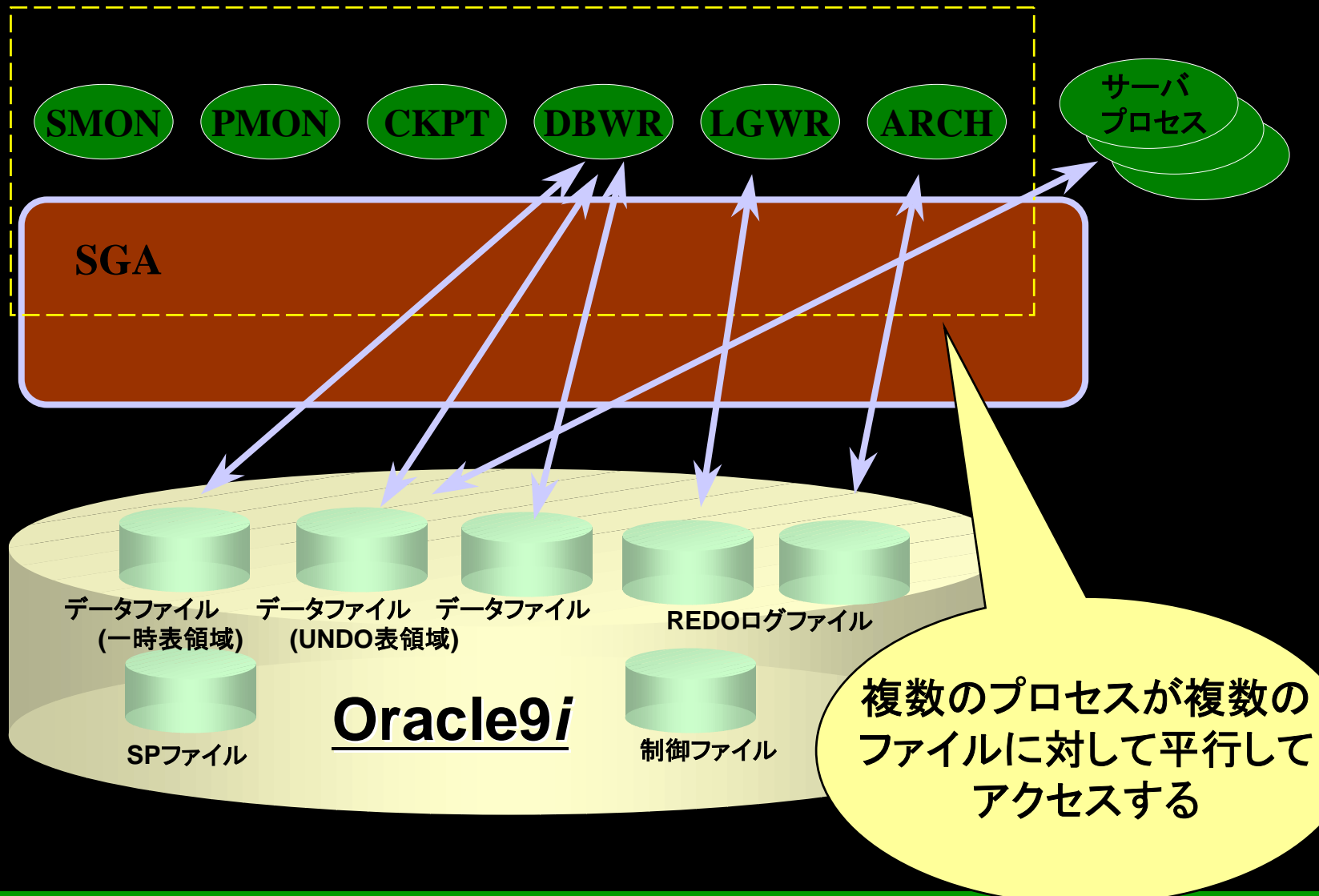
MLSE2.1R2 + Oracle9iR2

大規模データベースパフォーマンス向上

- 非同期 I/O (Oracle DBWRのスループット向上)
 - オラクルデータベースライタプロセスのI/O待ちが軽減され、データベースのスループットが向上。大規模・高負荷なデータベースシステムが構築可能になる。



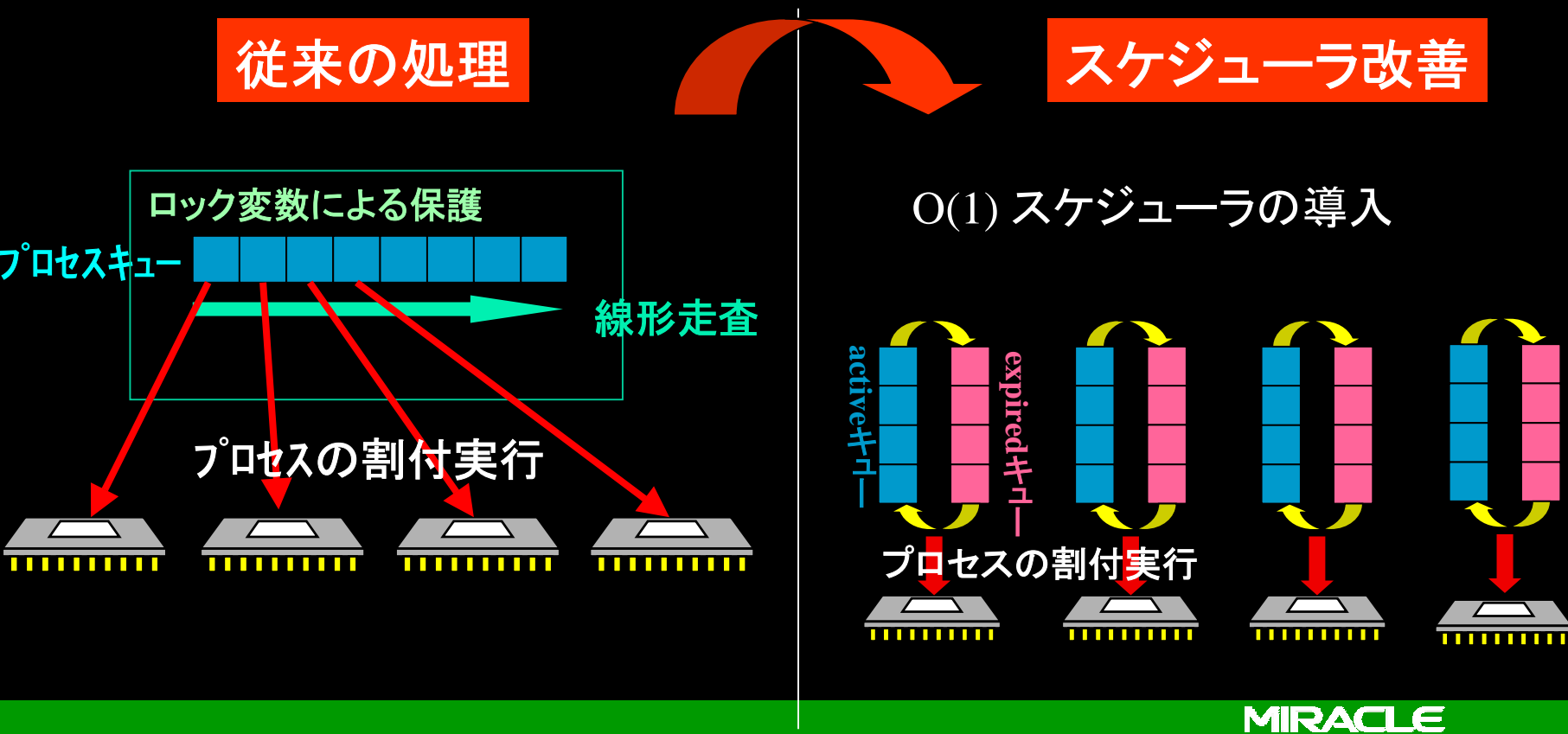
Oracleデータベースのアーキテクチャ



OSパフォーマンス向上(1)

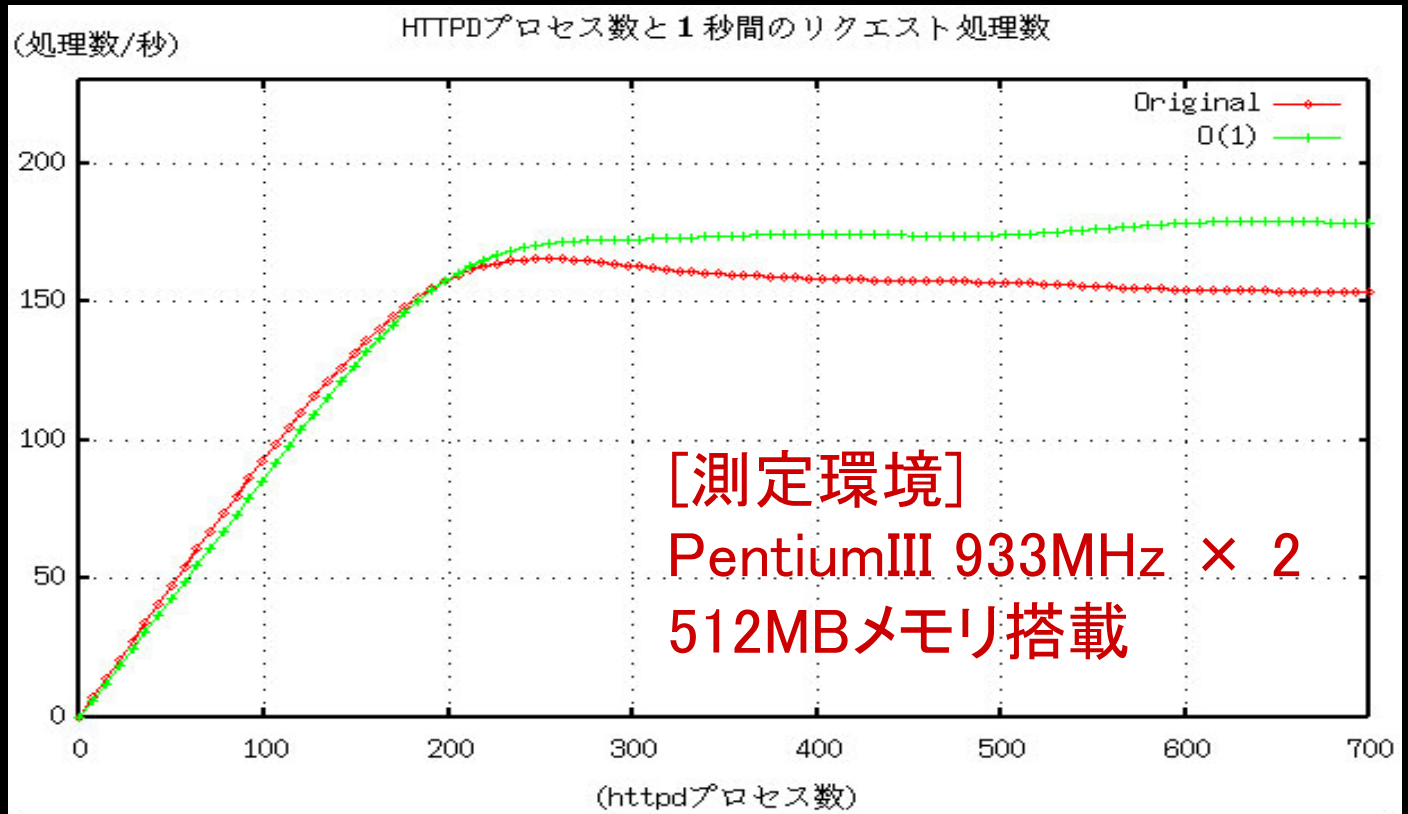
- プロセススケジューラの改善

- 同時に多数の要求が来た際のプロセス処理が改善され、OSとしての処理速度が著しく向上。
- 複数のCPUを搭載した大規模データベース/アプリケーションサーバシステムの性能が向上。



OSパフォーマンス向上(2)

O(1)スケジューラ測定結果



プロセス数の増加によるボトルネックが抑えられ、
パフォーマンスが向上

MIRACLE

【お問い合わせ先】

info@miraclelinux.com

<http://www.miraclelinux.com>

ミラクル・リナックス株式会社 【無断転載を禁ず】

この文書はあくまでも参考資料であり、掲載されている情報は予告なしに変更されることがあります。ミラクル・リナックス(株)は本書の内容に関していかなる保証もいたしません。また、本書の内容に関連したいかなる損害についても責任を負いかねます。又、本資料の著作権は特に指定されている箇所を除いて、ミラクル・リナックスが有します。ミラクル・リナックスが著作権を有するコンテンツにつきましては、ミラクル・リナックスに対して無断で複製、改変、頒布などを行うことはできません。

MIRACLE LINUX の製品名、ロゴ、サービス名などは、ミラクル・リナックスが所有するか、使用権許諾を受けている商標もしくは登録商標です。その他、本 Web サイトに掲載されている他社の製品名、ロゴなどは、それぞれ該当する各社が所有する商標もしくは登録商標です。