

日経Linux2005セミナー資料

## OSSの基幹業務への適応を前提にした 性能・信頼性評価手法

2005年7月25日

日本OSS推進フォーラム  
開発基盤WG  
主査 鈴木友峰  
((株)日立製作所 OSSテクノロジセンタ)



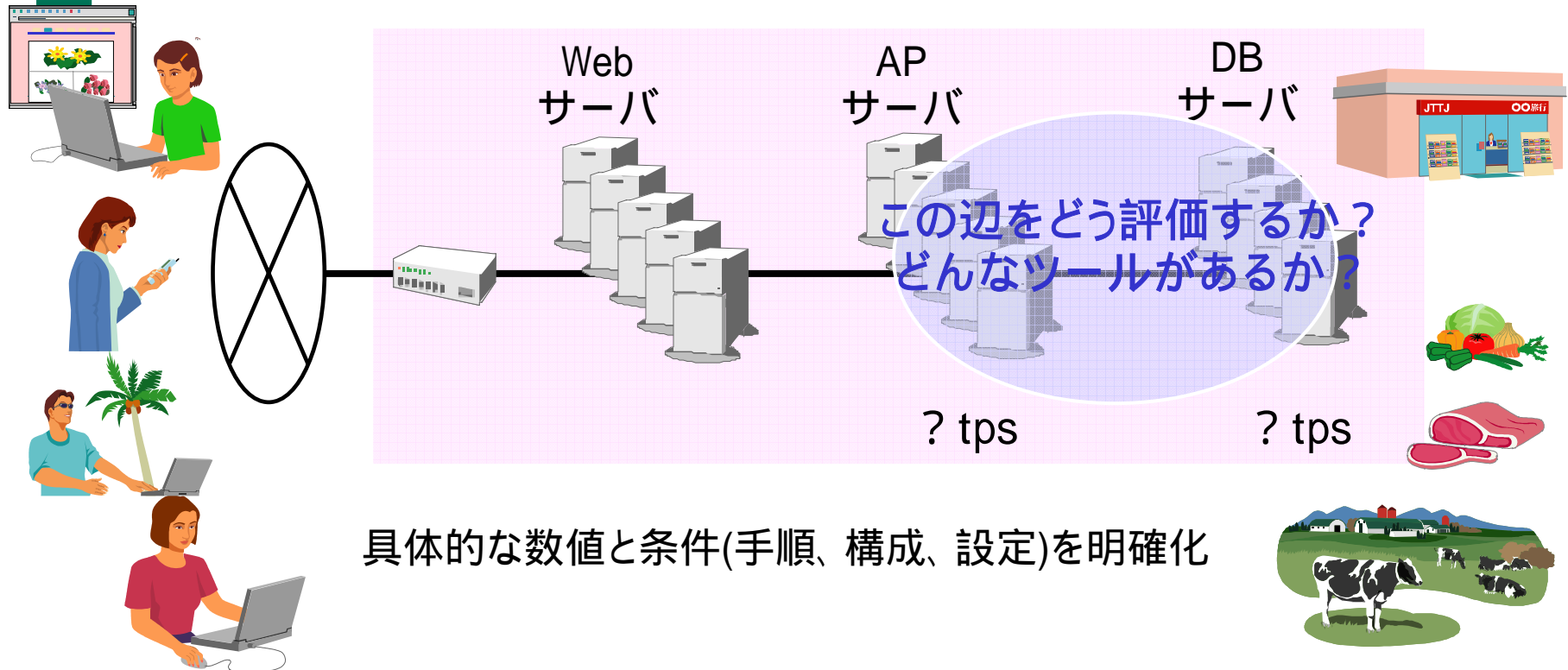
本プロジェクトの成果の一部は、独立行政法人 情報処理推進機構 (IPA)  
オープンソースソフトウェア活用基盤整備事業に係る委託業務の一環として開発しました。

# 今日のお話

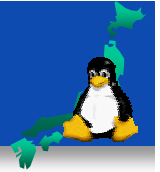


## OSSを基幹業務で使うための性能・信頼性評価手法

- 単なるベンチマークではない(最大性能のPR目的でない)
- 業務を想定したベンチマーク(そもそも基幹業務って?)  
= できるだけ実システムに近い
- 対象はOSS(でも商用との比較がしたい)
- 中身と結果に対する詳細な解析ができること(解析ツールとセット)



# 1. 日本OSS推進フォーラムの概要 (1) 設立背景と目的



## 組織構成

- 2004年2月設立
- 幹事団7名、顧問団14名(企業・団体のトップ、学識経験者で構成)
- オブザーバとして経済産業省、総務省。事務局 IPA

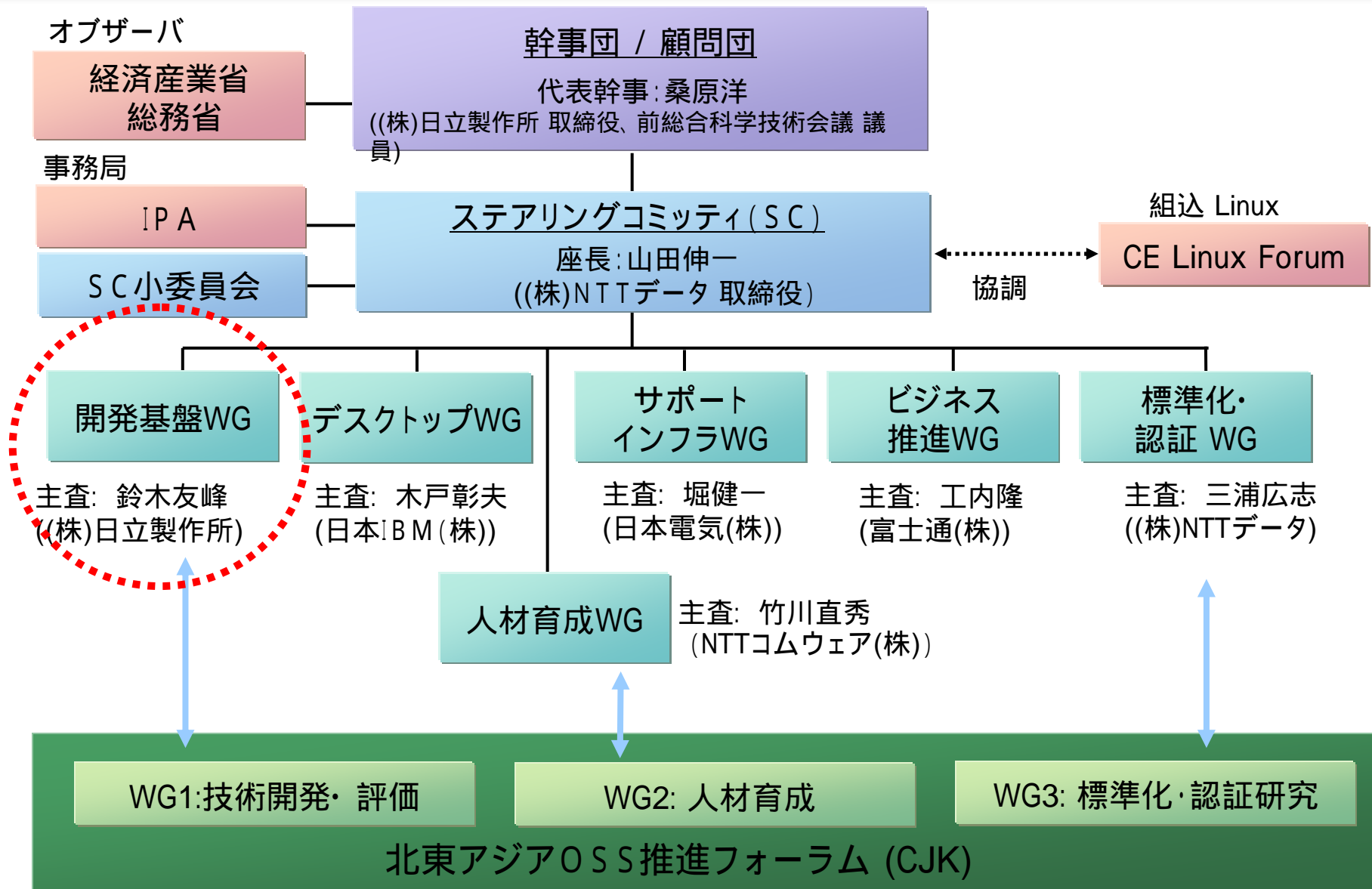
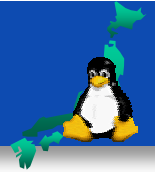
## 設立背景

- 2003年9月の日中韓経済貿易大臣合意及び日中韓IT担当大臣合意、11月の日中韓オープンソースビジネス懇談会の成果  
日本を代表してOSSについて国際協力していく団体が必要  
OSS普及について民間の意見を集約する場が必要
- OSSのシステムへの適用の進展  
ユーザが安心してOSSを利用するための技術的、制度的課題  
解決の必要性

## 設立目的

- 政府、民間で協力することによる日本国内でのOSS普及拡大
- ユーザが安心して使えるための技術的、制度的課題の解決と  
新たな選択肢の提供
- 日中韓、世界のコミュニティとの協調によるOSS発展への貢献

# 1. 日本OSS推進フォーラムの概要 (2) 組織



## 2. サーバ向けOSSの現状における課題(1)



### ベンダ、Slerから見た課題

ニーズがLinuxだけでなく、ミドルにまで拡大し、OSS適用システムが複雑化

それにもかかわらず...



- ・信頼性・性能等のシステム設計・構築に必要なデータが不足  
(結果として、各社が同じような評価を実施)
- ・障害解析ツールが不足しており、原因究明に時間がかかる



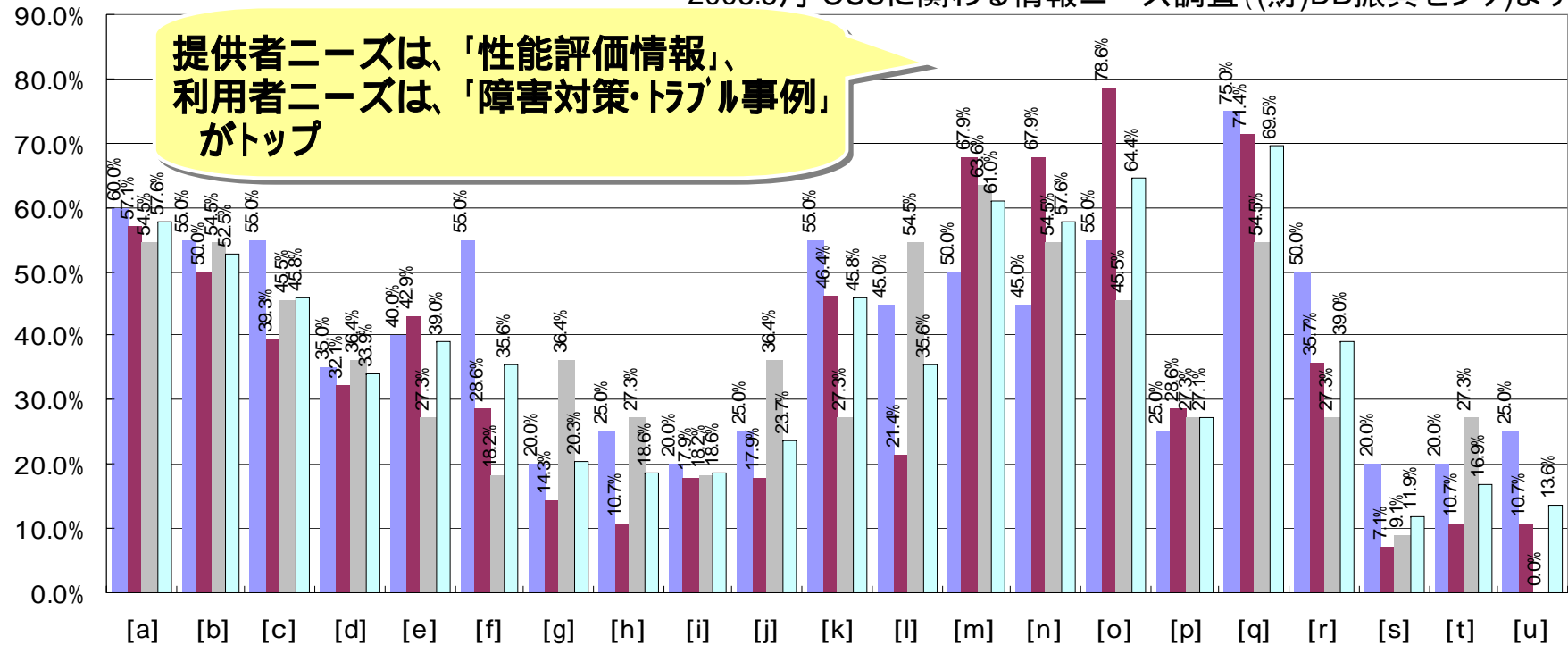
## 2. サーバ向けOSSの現状における課題(2)



### Q1 × Q3 取組の立場と情報ニーズの選択率

2005.5月 OSSに関わる情報ニーズ調査((財)DB振興センタ)より

提供者ニーズは、「性能評価情報」、  
利用者ニーズは、「障害対策・トラブル事例」  
がトップ

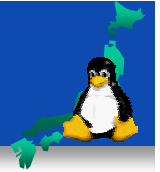


■ [a]利用者 ■ [b]提供者 ■ [c]その他 ■ 合計

利用者20人、提供者28人、その他11人  
(複数回答 全465件)

[a]一覧表、[b]導入や設定、[c]利用可能環境、[d]バージョン等の特徴・差異、[e]バグ等やその対処法  
[f]適用ライセンス、[g]有償サポートの入手先、[h]利用者コミュニティ、[i]開発コミュニティ、[j]依頼先ベンダ情報  
[k]アプリケーションの開発支援、[l]技術研修・教育支援、[m]業種・分野・用途別の事例  
[n]実績のある構成情報、[o]性能評価、[p]マイグレーション、[q]障害対策・トラブル事例、[r]知的財産権  
[s]イベント情報、[t]関連書籍・雑誌、[u]人材リクルート情報

## 3. プロジェクトの目的



### プロジェクトの目的

サーバLinux、OSSの更なる普及・拡大のためのベンダサイドの課題解決  
各企業内にあるOSSノウハウのDB化とオープン化

#### 1. ベンダ共同のOSSの性能・信頼性評価によるシステム設計・構築ノウハウの共有

- 結果だけでなく、手順やデータも共有し、標準化を図る
- 広くコミュニティに公開することで、OSSの普及に貢献
- ベンダにおけるOSS評価コストの低減(特にカーネル2.6、AP層、DB層などの新分野)
- 多様なノウハウをベースとしたシステム構築によるシステム信頼性向上

#### 2. 障害情報解析ツールの開発(例えばダンプ解析、トレーサ等)とノウハウの共有

- ツールの利用ノウハウのベンダ間での共有とブラッシュアップ
- 必要な機能はプロジェクトで開発し、公開することでコミュニティに貢献
- 障害解析時間の短縮
- ミッションクリティカルシステムへの適用ニーズに対応

## 4. プロジェクトのロードマップ



国内におけるサーバLinux、OSS普及のための「企業コミュニティ」の  
形成・育成・発展を目指し、以下のロードマップで推進

### 2004年度

#### フェーズ1 コミュニティの形成

1. 国内ベンダ、Sierの結集 & 民間技術者の連携意識の醸成  
具体的実施事項:
  - ・共同ベンチマーク実施とノウハウの共有
  - ・高信頼化ツールの開発とノウハウの共有
2. 中韓連携の土壌開拓  
具体的実施事項:
  - ・人脈の確立
  - ・共通意識の確立

### 2005年度

#### フェーズ2 コミュニティの育成

1. 高信頼化のための共同評価範囲拡大  
追加実施事項:
  - ・評価指針(手順)の標準化
  - ・評価範囲の拡大(クラス等)
  - ・ベンチマークツール設計・開発
  - ・チューニングノウハウの共有
  - ・障害予防ツールの整備
2. 新しい人材の投入
  - ・参加企業拡大
  - ・学との連携
3. 中韓との共同開発検討

### 2006年度

#### フェーズ3 コミュニティの発展

1. 国内ベンダ共同評価成果の公開範囲拡大、利用ユーザの拡大
2. 適用ユーザ事例の公開
3. 中韓との共同開発

## 5. メンバと状況



「OSS技術開発コンソーシアム」として2004/10～2005/2まで、  
IPAの「OSS活用基盤整備事業」として委託を受け具体的作業を実施。  
2005/6-10は11社で実施中

### ■ WG メンバ企業

#### -主査

-(株)日立製作所

#### -メンバ企業(コンソーシアムメンバ)

-(株)SRA.

-(株)NTTデータ

-新日鉄ソリューションズ(株)

-住商情報システム(株)

-(株)野村総合研究所(04年度)

-ミラクル・リナックス(株)

-ユニアデックス(株)

#### -メンバ企業(非コンソーシアムメンバ)

-NTTコムウェア(株)

-Red Hat KK

-日本ユニシス(株)

#### -オブザーバ

-Novell, Inc., OSDL,

#### -新メンバ(2005年1月から加入)

-日本電気(株)

-ターボリナックス(株)

-(株)日立システムアンドサービス

-(株)テンアートニ

-富士通(株)

-日本HP(株)

## 6. 実施事項の概要(1)



### 具体的な実施事項

#### 1. ベンダ共同のOSS性能・信頼性評価による システム設計・構築ノウハウの共有

Javaアプリケーション層の評価  
DB層の評価  
OS層の評価



・結果だけでなく、ツール・手順も共有  
・広くコミュニティに公開

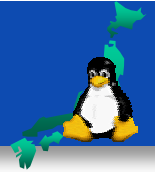
#### 2. 障害解析ツールの開発と利用ノウハウの共有

ダンプデータ解析ツール(Alicia)の開発  
カーネル性能評価ツール(LKST)の開発  
ディスク割当評価ツール(DAV)の開発



・障害解析時時間の短縮  
・高信頼システム適用ニーズに対応

## 6. 実施事項の概要(2)



### 想定されるアウトプット(ベンチマーク評価)

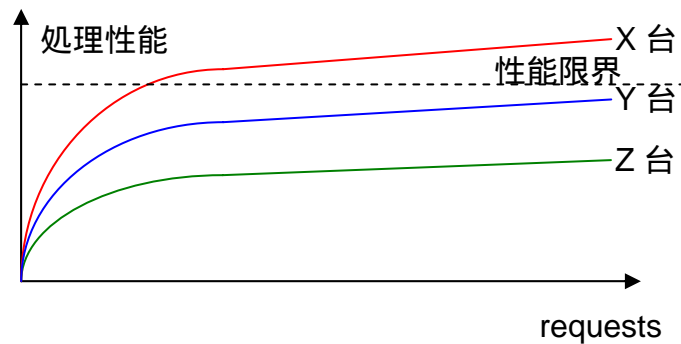
#### 評価環境定義書

- ・評価HW、SW(OSS)構成
- ・評価ツール

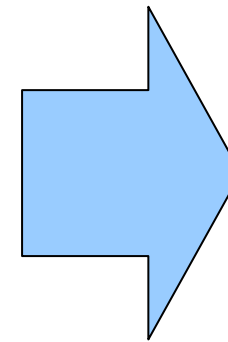
#### 評価手順書

- ・SW(OSS)インストール、設定
- ・評価項目
- ・評価手順

#### 評価結果



評価項目毎



公開、共有

OSS  
Community

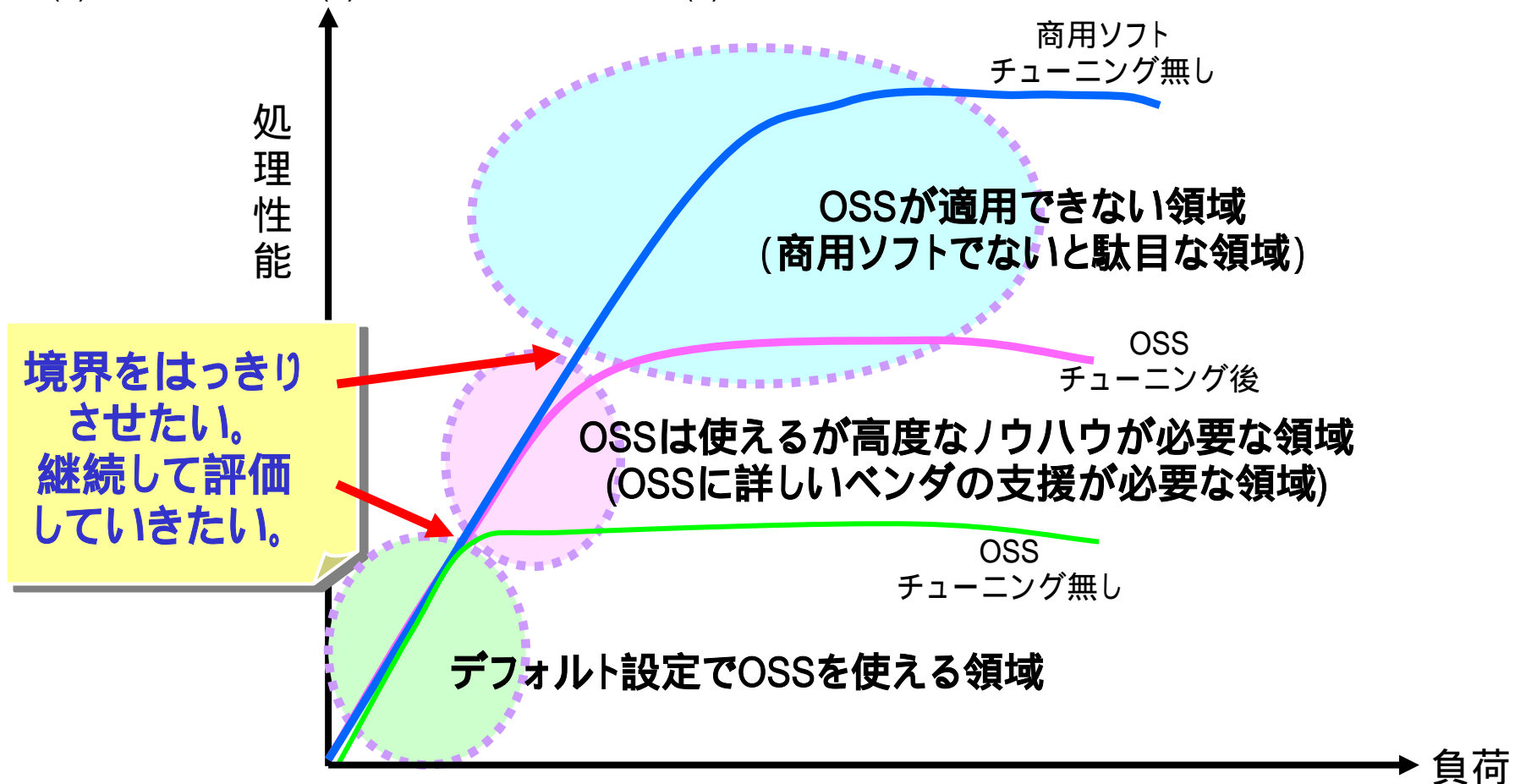


## 6. 実施事項の概要(3)



### 評価目的

OSSモデルが、現状でどこまで使えるかを、評価手順を明らかにしたうえで、明確化する  
具体的にはOSSの処理性能はチューニングにより大きく変化するので以下3パターンを明らかにする  
(1)デフォルトOSS、(2)チューニング後OSS、(3)商用



## 7. 成果の概要(1)



### 主な成果一覧(手順書関連)

#### 1. Java AP層

- ・ [SPECjAppServer2004](#)によるJBoss、WebLogic共通の性能評価手順を確立  
(RedHatAS2.1,3、MIRACLE LINUX V3.0、SuSE Linux ES9 + PostgreSQLでの手順を検証)

#### 2. DB層

- ・ [OSDL DBT-1](#)によるPostgreSQL、MySQL(MaxDB)、Oracle共通の性能評価手順を確立  
(RedHatAS3、MIRACLE LINUX V3.0、SuSE Linux ES9での手順を検証)
- ・ [OSDL DBT-3](#)によるPostgreSQL性能評価手順を確立
- ・ 大量データのロードなど実SI場面を想定したPostgreSQL対応性能評価手順の確立

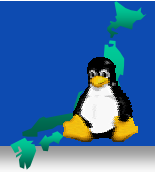
#### 3. OS層

- ・ CPU/IO負荷状態を想定したボトルネック解析手法の確立(iozone、oprofile、LKST)
- ・ DBMSと相関を持つベンチマークの開発(diskio)

#### 4. ツール

- ・ ダンプ、トレーサ、ファイルシステムのフラグメンテーション可視化ツールのそれぞれについて、効果を測定し、障害解析手順をまとめた

## 7. 成果の概要(2)

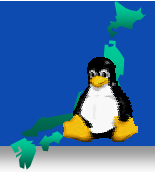


### 評価手順をまとめたベンチマーク一覧

対象	ベンチマーク ツール	OS	対象ミドル	備考
AP	SPECj AppServer 2004	RedHatAS2.1、3 MIRACLE V3.0 SuSE9	JBoss4.0.0 (RMS) WebLogic (MS)	<ul style="list-style-type: none"> <li>・SPEC.orgが開発</li> <li>・有償(2K\$)</li> <li>・EJB対応</li> </ul>
DB	OSDL DBT-1	RedHatAS3 MIRACLE V3.0 SuSE9	PostgreSQL7.4 (RMS) MaxDB (RMS)	<ul style="list-style-type: none"> <li>・OSDLが開発</li> <li>・TPC-W相当</li> </ul>
	OSDL DBT-3	MIRACLE V3.0	Oracle10g (M) PostgreSQL7.4、8.0	<ul style="list-style-type: none"> <li>・OSDLが開発</li> <li>・TPC-H相当</li> </ul>
	pgbench	MIRACLE V3.0	PostgreSQL7.4、8.0	<ul style="list-style-type: none"> <li>・PostgreSQL付属</li> <li>・TPC-B相当</li> </ul>
OS	Iozone	MIRACLE V3.0	-	・Iozone.orgから公開
	diskio	MIRACLE V3.0	-	・新規開発(DBサイジング用)

括弧内: R:RedHat, M:Miracle, S:SuSE

# 環境定義書・評価手順書の例



## 1.1 環境定義

### 1.1.1 システム構成

今回の評価では、PostgreSQL7.4.6用と8.0.0beta5用の、2つのシステムを使用して検証を行った。7.4.6用のシステムを表 1.1-1 に、8.0.0beta5用のシステムを表 1.1-2 に示す。

表 1.1-1 PostgreSQL7.4.6用のシステム

製品名	HP ProLiant DL380 Generation 3		
プロセッサ	インテル Xeon プロセッサ 2.80GHz、2CPU		
メモリ	2.5GByte		
ハードディスク	Ultra SCSI 320 HDD 15000rpm 内蔵ドライブベイに、3台		
ディスク構成	36GB HDD	/boot	1GB
		swap	4GB
		/	4GB
		/usr	4GB
		/var	1GB
		/tmp	1GB
	/home	20GB	
	36GB HDD	/db_xlog	すべて
	72GB HDD	/dbt3_0	すべて

すべてのパーティションは ext3 ファイルシステムでフォーマットして、ext3 のジャーナリングのモードも、デフォルトの orderd モードを使用する。  
以下のディレクトリについては、その所有者を pgsq1 ユーザにしておく。  
/db\_xlog、/dbt3\_0

表 1.1-2 PostgreSQL8.0.0beta5用のシステム

製品名	HP ProLiant DL380 Generation 3
プロセッサ	インテル Xeon プロセッサ 2.80GHz、2CPU
メモリ	2.5GByte

#### 1.1.1.1 PostgreSQL のインストール

今回の評価では、バージョン 7.4.6 と 8.0.0beta5 を使用した。以下の手順でインストールできるが、二つのバージョンを同時にインストールする場合は、使用するディレクトリが競合しないように、適切に変更する必要がある。

root ユーザで、PostgreSQL の所有者となる Linux のユーザとして、pgsq1 ユーザを作成する。

```
# useradd pgsq1
```

PostgreSQL のソースコードアーカイブを展開するディレクトリと、PostgreSQL をインストールするディレクトリを作成して、ディレクトリの所有者を pgsq1 ユーザにする。

(a) PostgreSQL7.4.6 の場合

```
# mkdir /usr/local/src/postgresql-7.4.6
# chown pgsq1 /usr/local/src/postgresql-7.4.6
# mkdir /usr/local/pgsq1
# chown pgsq1 /usr/local/pgsq1
```

(b) PostgreSQL8.0.0beta5 の場合

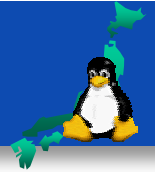
```
# mkdir /usr/local/src/postgresql-8.0.0beta5
# chown pgsq1 /usr/local/src/postgresql-8.0.0beta5
# mkdir /usr/local/pgsq1
# chown pgsq1 /usr/local/pgsq1
```

pgsq1 ユーザで、PostgreSQL のソースコードアーカイブを展開し、展開したディレクトリに移動する。PostgreSQL のソースコードアーカイブは、/tmp ディレクトリにあるものとする。

(a) PostgreSQL7.4.6 の場合

```
# su pgsq1
```

誰でも再現できる  
レベルで記述



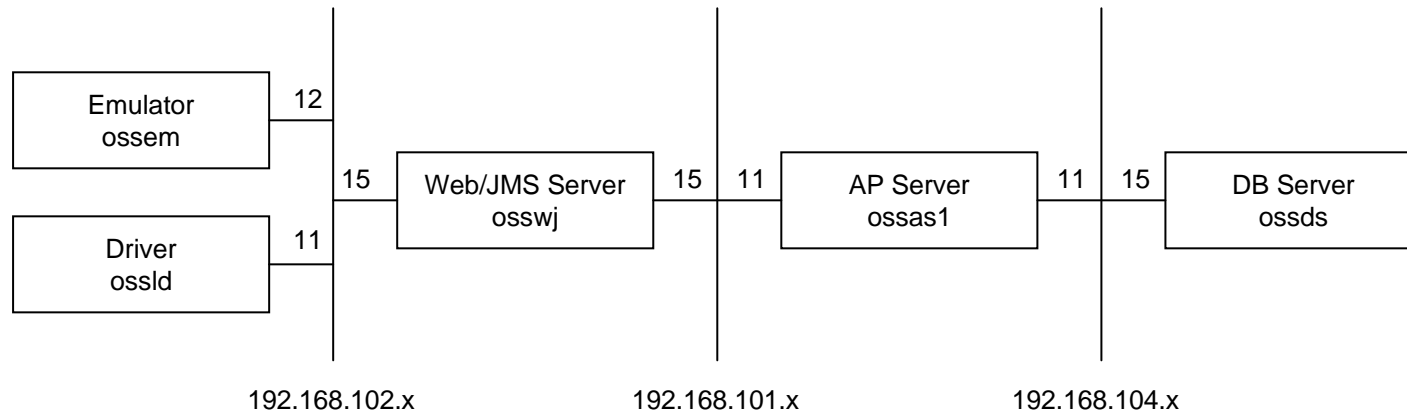
### SPECjAppServer2004の概要

- SPEC.orgから購入可能(\$2,000)
  - ソースコードは購入者以外非公開、ドキュメント・測定結果等は公開
    - <http://www.spec.org/jAppServer2004/>
- J2EE 1.3以降に準拠したAPサーバ上で可動
  - JaveServer Pages, Servlet API, JDBC等を使用
  - 個々のAPサーバ上で動かす為にはカスタマイズが必要
    - 配備記述子、DataSource設定、Message Queue設定、等
- EJB 2.0の以下の機能群を使用
  - Stateless/Stateful Session Bean
  - Message Driven Bean
  - CMP Entity Bean
- J2EE/EJBコンテナの総合的なベンチマークの代表格
  - 現状OSSでは同様な機能を提供するものは存在しない模様

# 7.1 ベンチマークの概要 SPECjAppServer2004とは?(2)



## ハードウェア・ソフトウェア環境(例)



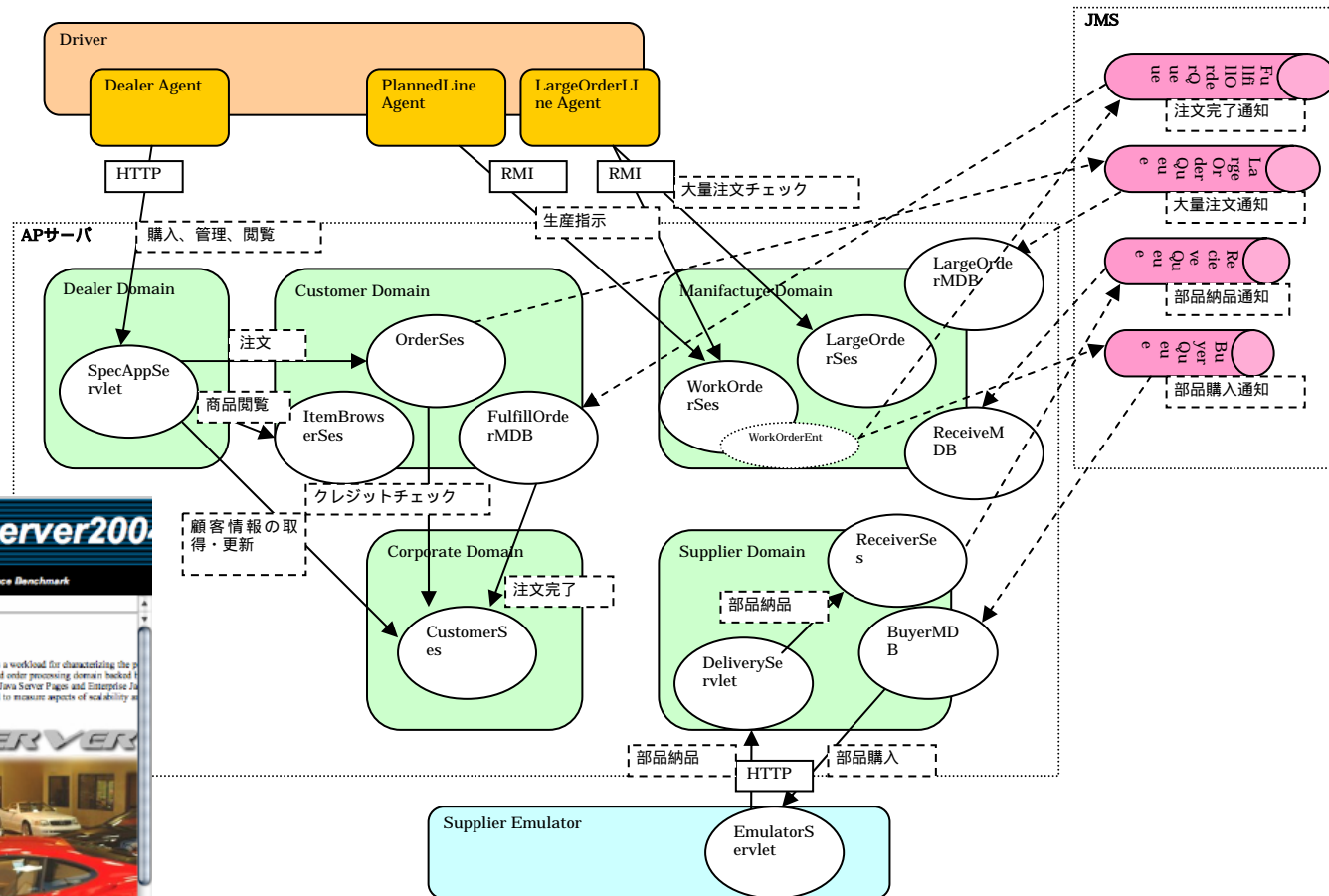
Driver	1x3GHz P4, 1GB Memory, 1x120GB HDD, 1xGigabit
Emulator	1x3GHz P4, 1GB Memory, 1x120GB HDD, 1xGigabit
Web/JMS	2x2.8GHz Xeon, 2.5GB Memory, 2x36GB HDD, 2xGigabit
AP Server	2x2.8GHz Xeon, 2.5GB Memory, 2x36GB HDD, 2xGigabit
DB Server	2x2.8GHz Xeon, 3GB Memory, 2x73GB HDD, 2xGigabit

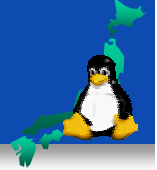
OS	Red Hat Enterprise Linux AS 2.1(2.4.9) SUSE LINUX Enterprise Server 9(2.6.5)
JDK	Java 2 SDK, Standard Edition 1.4.2_04
DB Server	PostgreSQL 7.4.6
AP Server	JBoss 4.0.0
Web Server	Apache httpd 2.0.46
Web Connector	Jakarta mod_jk2 2.0.4
Application	SPECjAppServer2004 V1.03

# 7.1 ベンチマークの概要 SPECjAppServer2004とは?(3)



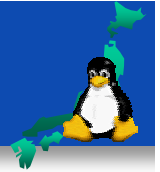
## SPECjAppServer2004のアーキテクチャ





### SPECjAppServer2004の内部構造

- 5つのejb-jarにより構成(33 EJBs/20 tables)
  - corp(4 EJBs/5 tables)
  - mfg(12 EJBs/5 tables)
  - orders(7 EJBs/3 tables)
  - supplier(8 EJBs/6 tables)
  - util(2 EJBs/1 tables)
- 2種のトランザクションに大別(35 patterns)
  - Dealer's Web interaction(HTTP, 14 patterns)
    - doLogin, doPurchase, doClearCart, doInventory, etc.
    - 3つのビジネストランザクションに大別
      - » Purchase(25%), Manage(25%), Browse(50%)
  - EC transaction(RMI/IIOP, 21 patterns)
    - ScheduleWorkOrder, UpdateWorkOrder, etc.
    - 2種のLineに大別(PlannedLine, LargeOrderLine)



### ベンチマーク測定後に得られる結果

- 処理性能(総合、トランザクション種別毎)
  - 単位時間あたりのトランザクション処理性能(JOPS)
    - $JOPS = \text{jAppServer Operation Per Second}$
- 応答時間(トランザクション種別毎)
  - 平均応答時間、最大応答時間、90%応答時間等
- 種々のテスト結果(約30のテスト項目)
  - トランザクション整合性、応答時間が要求範囲内か、等
- エラーログ、トランザクション詳細データ、等

### 測定結果の公開に対する制限

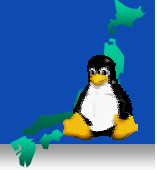
- IR値およびJOPS値の公開には、SPEC.orgのレビューが必要
  - IR = Injection Rate(サーバへの負荷並びにDBの規模を示す)
  - <http://www.spec.org/jAppServer2004/results/> に手順・結果が公開
    - ただし現状では商用APサーバのみ



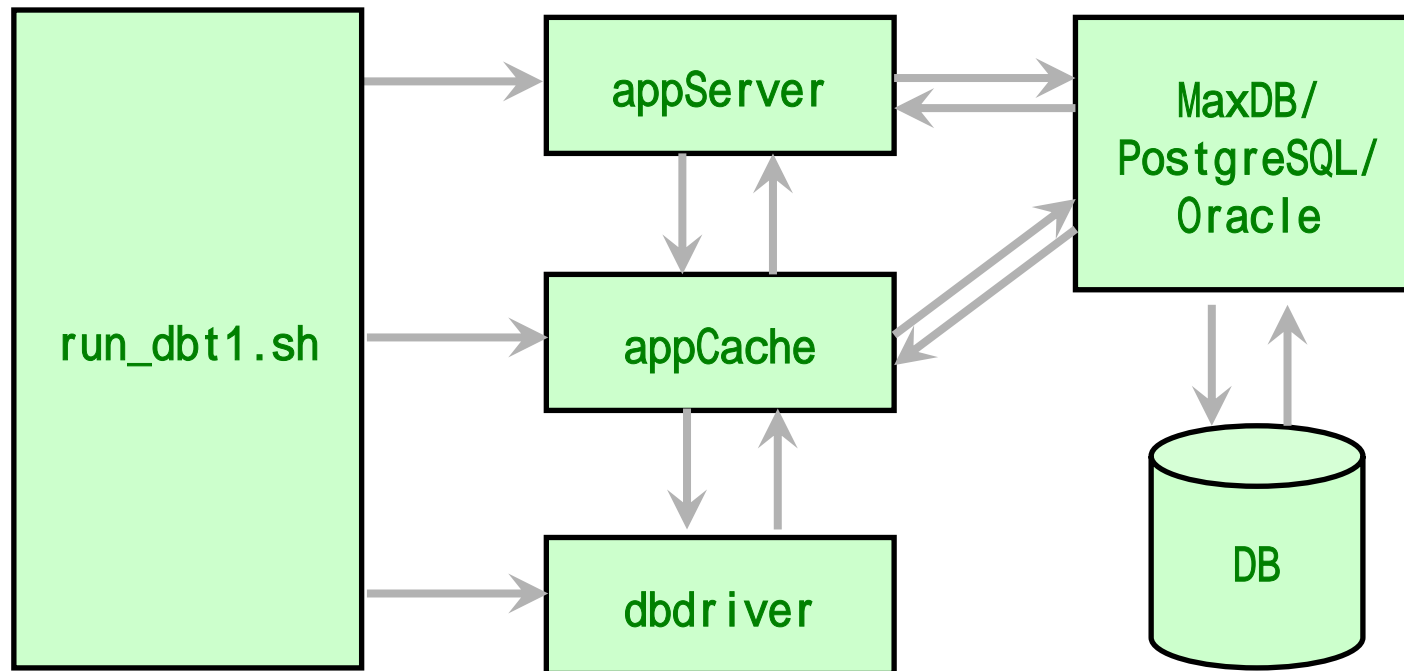
### OSDLが公開しているフリーのベンチマークツール

DBT-1	TPC-W の簡易版実装 Webベースのトランザクション・パフォーマンステスト。オンライン書店におけるユーザのアクティビティをシミュレートする
DBT-2	TPC-C の簡易版実装 OLTPトランザクション・パフォーマンステスト。複数の作業者が1つのデータベースへアクセスし、顧客情報を更新し、部品の在庫を確認する卸売業者をシミュレートする
DBT-3	TPC-H の簡易版実装 意思決定支援のためのワークロードを実行しパフォーマンスを測定する。業務用の特別なクエリや並行動作するデータ更新処理のスイートで構成される

各DBTで得られた値を、TPCの結果として利用することは禁止されている



### OSDL DBT-1の構成



- 全てを1台にインストールして実行可能
- 1つのシェルスクリプトで、一連の処理を行う
- 全インタラクションの処理時間が記録されており、測定終了時トランザクション/秒が計算される



### OSDL DBT-1のインタラクション構成

- Admin Confirm
  - Admin Request
  - Best Sellers
  - Buy Confirm
  - Buy Request
  - Customer Registration\*
  - Home
  - New Products
  - Order Display
  - Order Inquiry
  - Product Detail
  - Search Request\*
  - Search Results
  - Shopping Cart
- \*は、未実装

・各インタラクションは、ストアプロシージャで実装されている

MaxDB 7.5.00.16

PostgreSQL 7.4.6

他

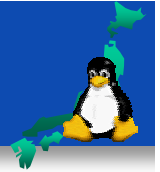
・各インタラクションの実行時間を取得できる



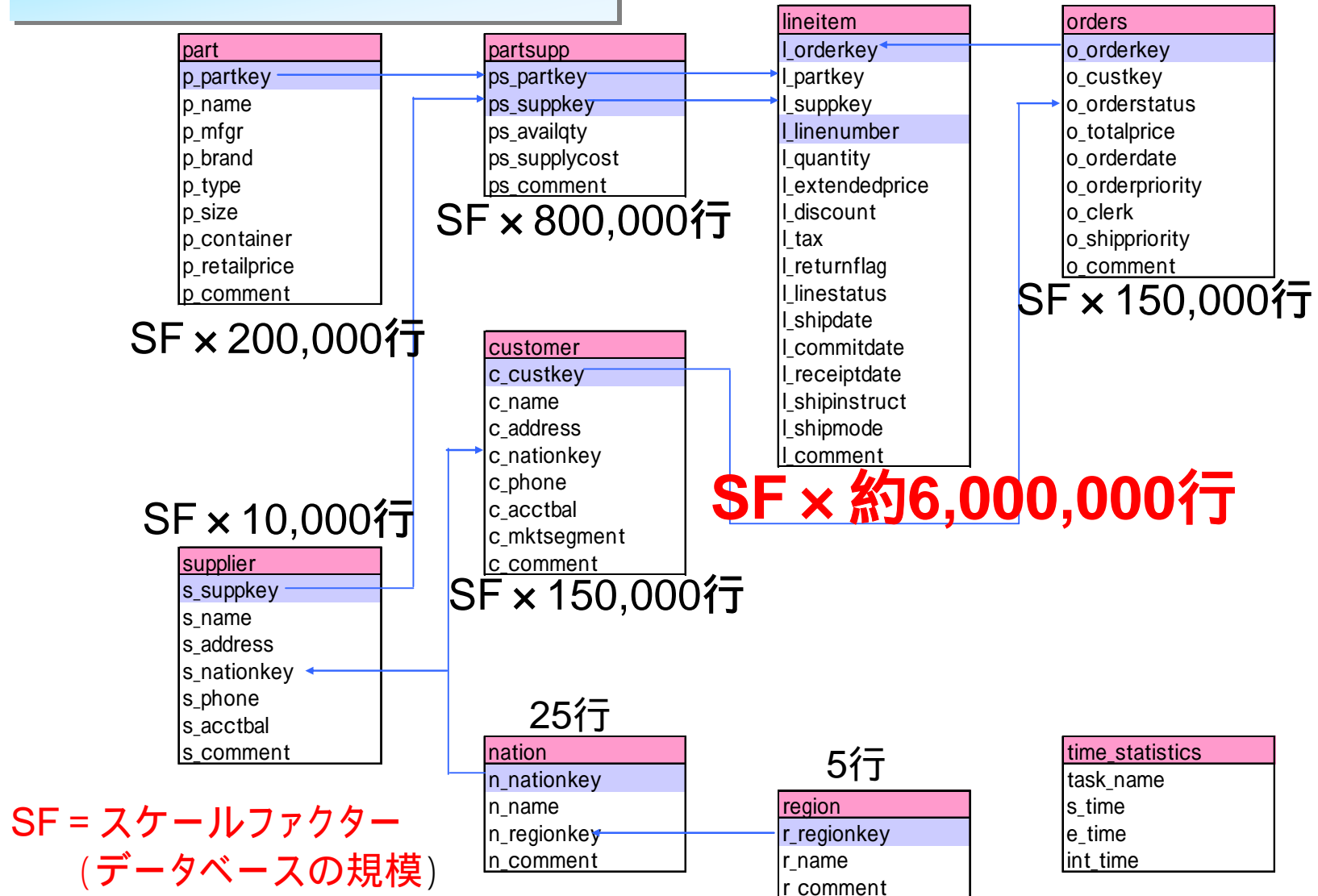
### OSDL DBT-3とは

- 意志決定支援環境のベンチマーク
  - 大規模なデータベース  
(キャッシュが効果的に使えない)
  - 複雑なクエリー  
(インデックスが効果的に使えない)
- TPC-Hを簡略化
  - DBT-3の実行結果は、TPC-Hとは無関係

# 7.3 ベンチマークの概要 OSDL DBT-3とは(2)



## OSDL DBT-3のDB構造





### クエリーの例

- 22個の複雑な検索クエリー  
(意志決定支援情報の検索)

Q7	<pre>select supp_nation, cust_nation, l_year, sum(volume) as revenue from (   select n1.n_name as supp_nation, n2.n_name as cust_nation, extract(year from l_shipdate) as l_year,   l_extendedprice * (1 - l_discount) as volume   from supplier, lineitem, orders, customer, nation n1, nation n2   where s_suppkey = l_suppkey and o_orderkey = l_orderkey and c_custkey = o_custkey and s_nationkey =   n1.n_nationkey and c_nationkey = n2.n_nationkey   and ((n1.n_name = 'JAPAN' and n2.n_name = 'VIETNAM') or (n1.n_name = 'VIETNAM' and n2.n_name   = 'JAPAN'))   and l_shipdate between date '1995-01-01' and date '1996-12-31') as shipping group by supp_nation, cust_nation, l_year order by supp_nation, cust_nation, l_year;</pre>
----	--

- 2個のリフレッシュクエリー  
(システム運用中のデータの増減をエミュレート)

RF1	insert into lineitem ′ ′ ′; Inesrt into orders ′ ′ ′;
RF2	delete from lineitem ′ ′ ′; delete from orders ′ ′ ′;



### DBT-3で測定する内容

- 以下の3つテストを連続実行する。
  - **ロードテスト**
    - データベースにデータをファイルからロードした時間を測定する。
    - データファイルのサイズは、「スケールファクター × 1ギガバイト」。
  - **パワーテスト**
    - 同時に実行するトランザクション数 = 1
    - 22 + 2個のクエリーを実行した時間を測定する。
  - **スループットテスト**
    - 同時に実行するトランザクション数 = 任意指定(今回は4)
    - 22 + 2個のクエリーを実行した時間を測定する。

# 7.3 ベンチマークの概要 OSDL DBT-3とは(4)

- 測定結果は、HTMLレポート

Compsite

下記2つの平均値。

Query Processing Power

同時トランザクション数 = 1で、  
「1時間あたりに実行できる  
クエリー数 × SF」

Throughput Numerical Quantiry

同時トランザクション数 = nで、  
「1時間あたりに実行できる  
クエリー数 × SF」

スクロールすると、  
詳細な情報がある

## DBT-3 Test Result

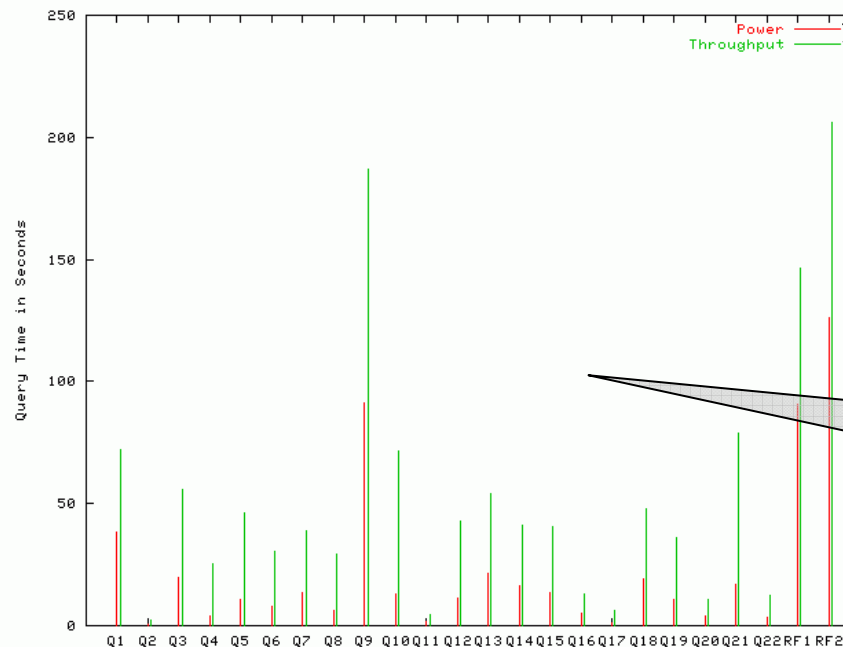
### Configurations:

Software Version	Hardware Configuration	Run Parameters
Linux Kernel: 2.4.21-9.38AXsmp	4 CPUS @ 2787.511 MHz	Database Scale Factor: 1
PostgreSQL: 8.0.0beta5	CPU model Intel(R) Xeon(TM) CPU 2.80GHz	Number of streams for throughput run: 4
sysstat: 4.0.7	2573188 kB Memory	shmmax: 2147483648
distribution: MIRACLE LINUX V3.0 (koumei)	Node: srpc2408.co.jp	database parameter:
procps: 2.0.13		Put pgsql_tmp on different drive: 0
Test Kit Version 1.5		Put WAL on different drive: 1

### DBT-3 Metrics:

Composite	Query Processing Power	Throughput Numerical Quantity
275.16	336.99	224.68

### Query Times



スケール  
ファクター  
(SF)

ストリーム数  
(同時接続数)

Throughput  
ストリーム数 = n  
のベンチマーク

Power  
ストリーム数 = 1  
のベンチマーク

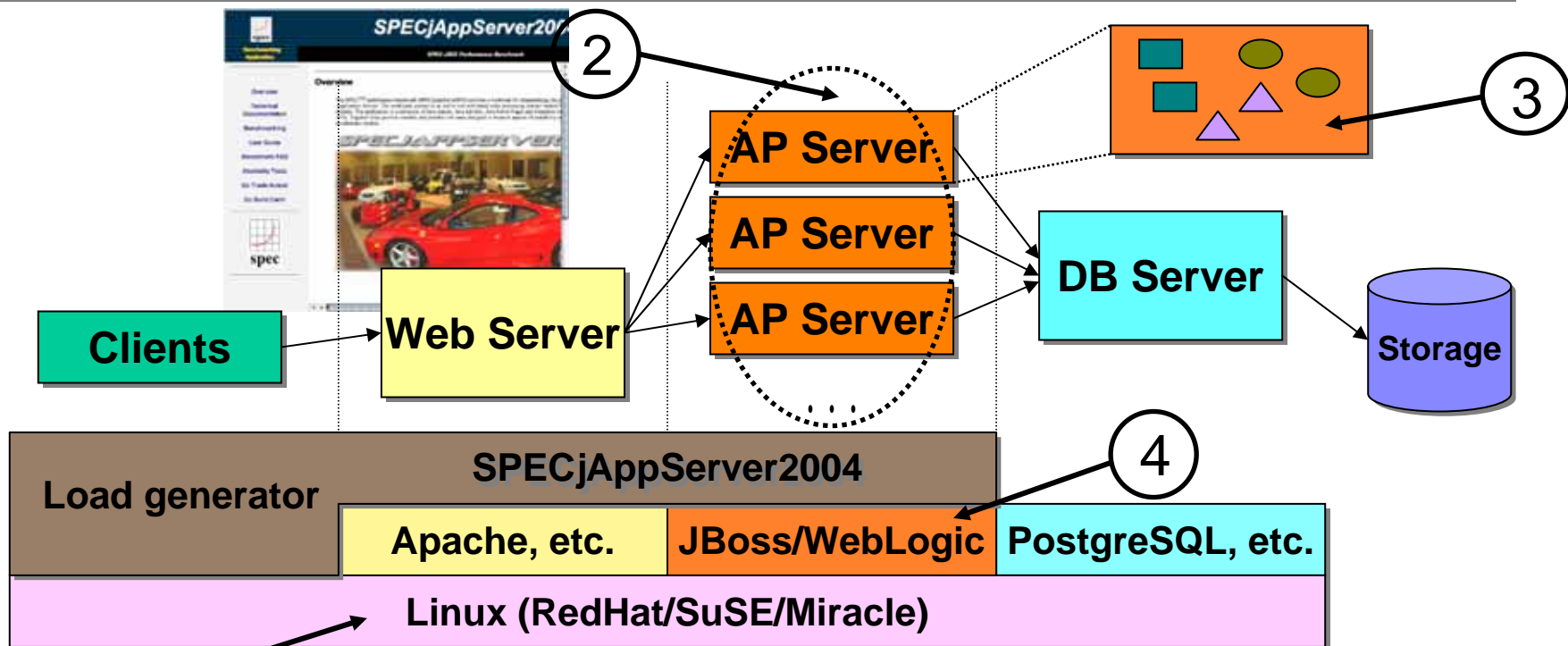
クエリーごとの  
所要時間の  
グラフ(秒)

## 8. Java AP層評価結果 (1)評価項目



### 2004年度ベンチマーク評価項目

1. カーネル2.4と2.6の比較(2.6の性能や信頼性はどうか、等)
2. 分散処理性能の評価(大規模システムにおける性能・信頼性等)
3. トランザクション特性の解析(マイクロなレベルでの解析を行いたい)
4. 商用APサーバとの比較(JBossとWebLogicとの比較)



1

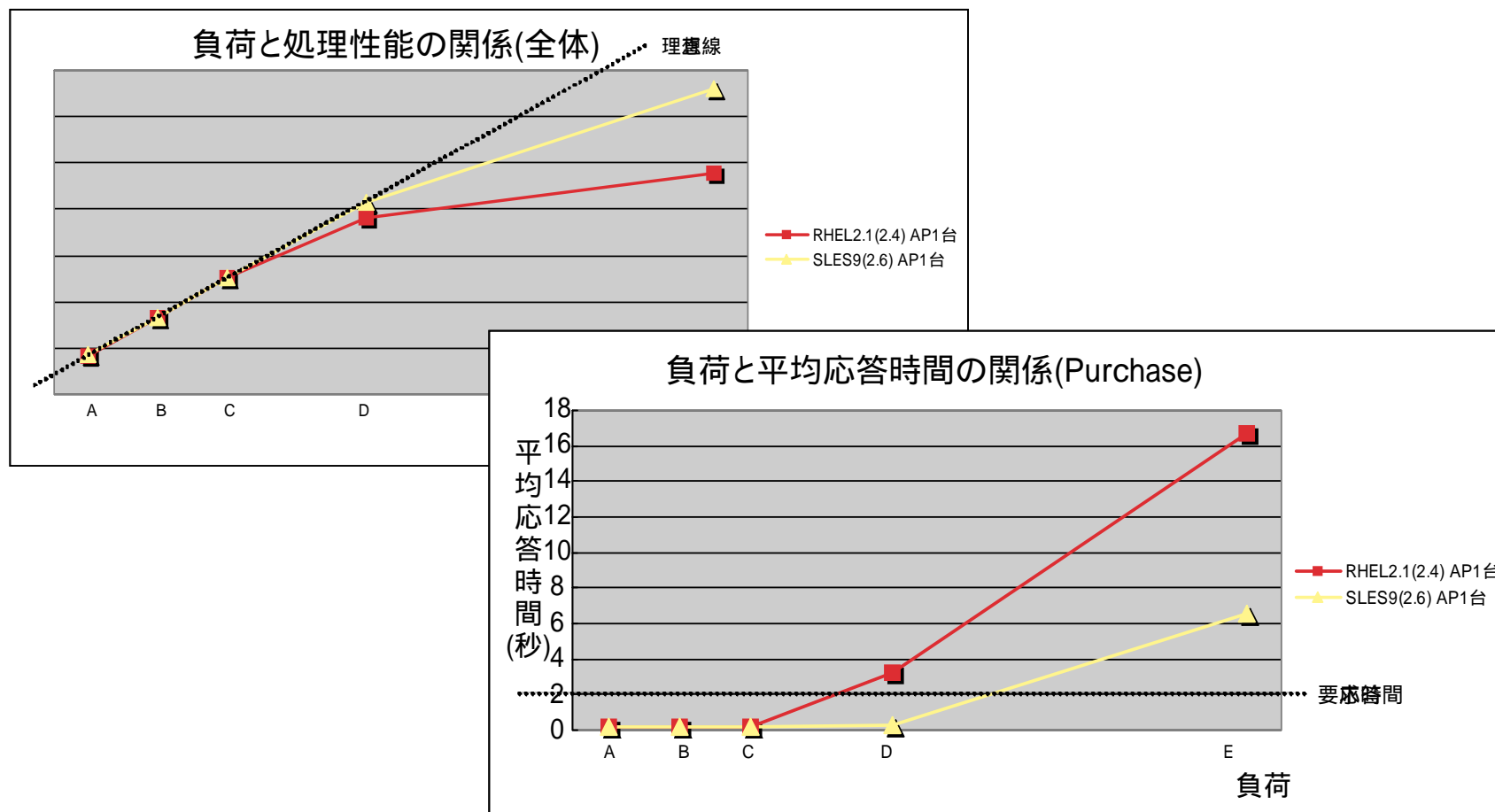
Javaアプリケーション層ベンチマークのSW/HW構成

Copyright (C) 2005, Development Infrastructure WG, Japan OSS Promotion Forum

## 8. Java AP層評価結果 (2)評価結果の詳細

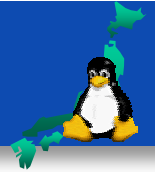


### カーネル2.4と2.6の比較



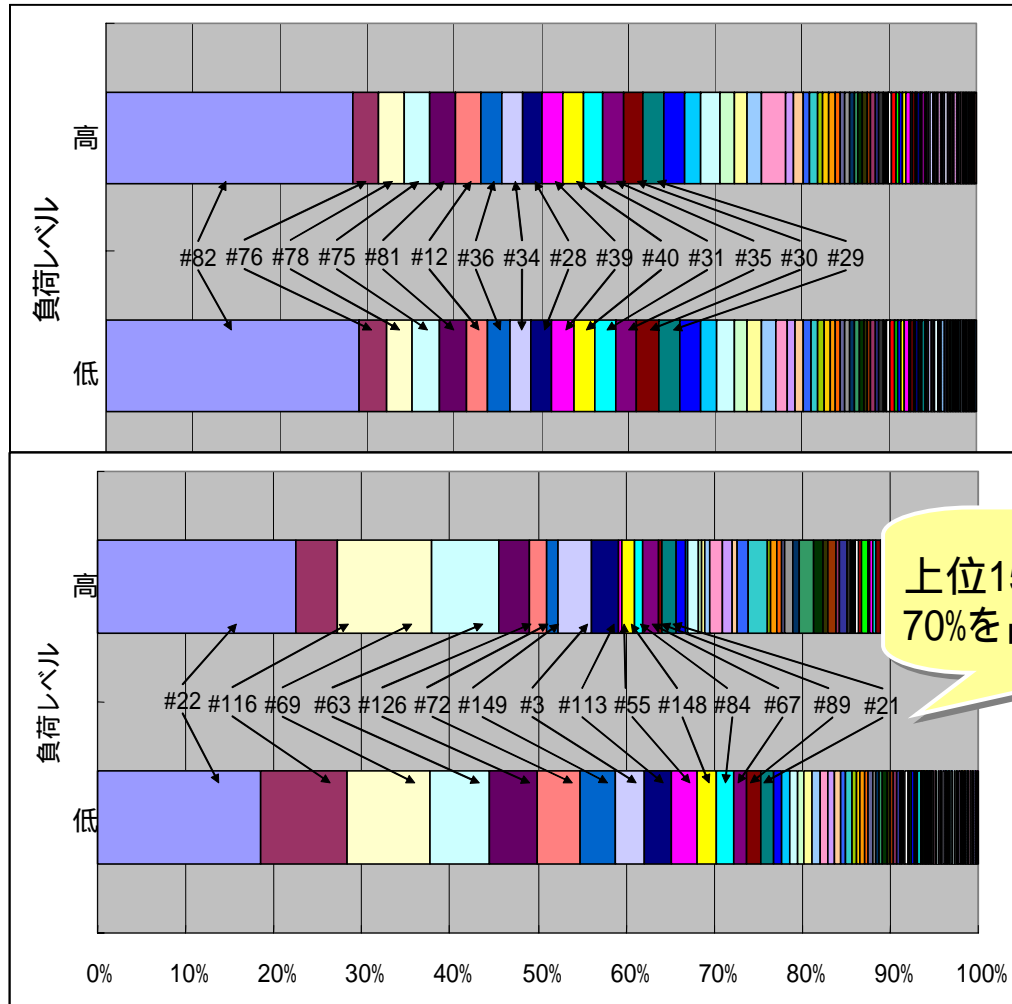
処理性能・平均応答時間共に、高負荷時には、カーネル2.6の方が、やや高性能  
:カーネル2.4:RedHatAS2.1、カーネル2.6:SuSE9

# 8. Java AP層評価結果 (2)評価結果の詳細



## トランザクション特性の解析

メソッドレベルの  
実行回数の割合

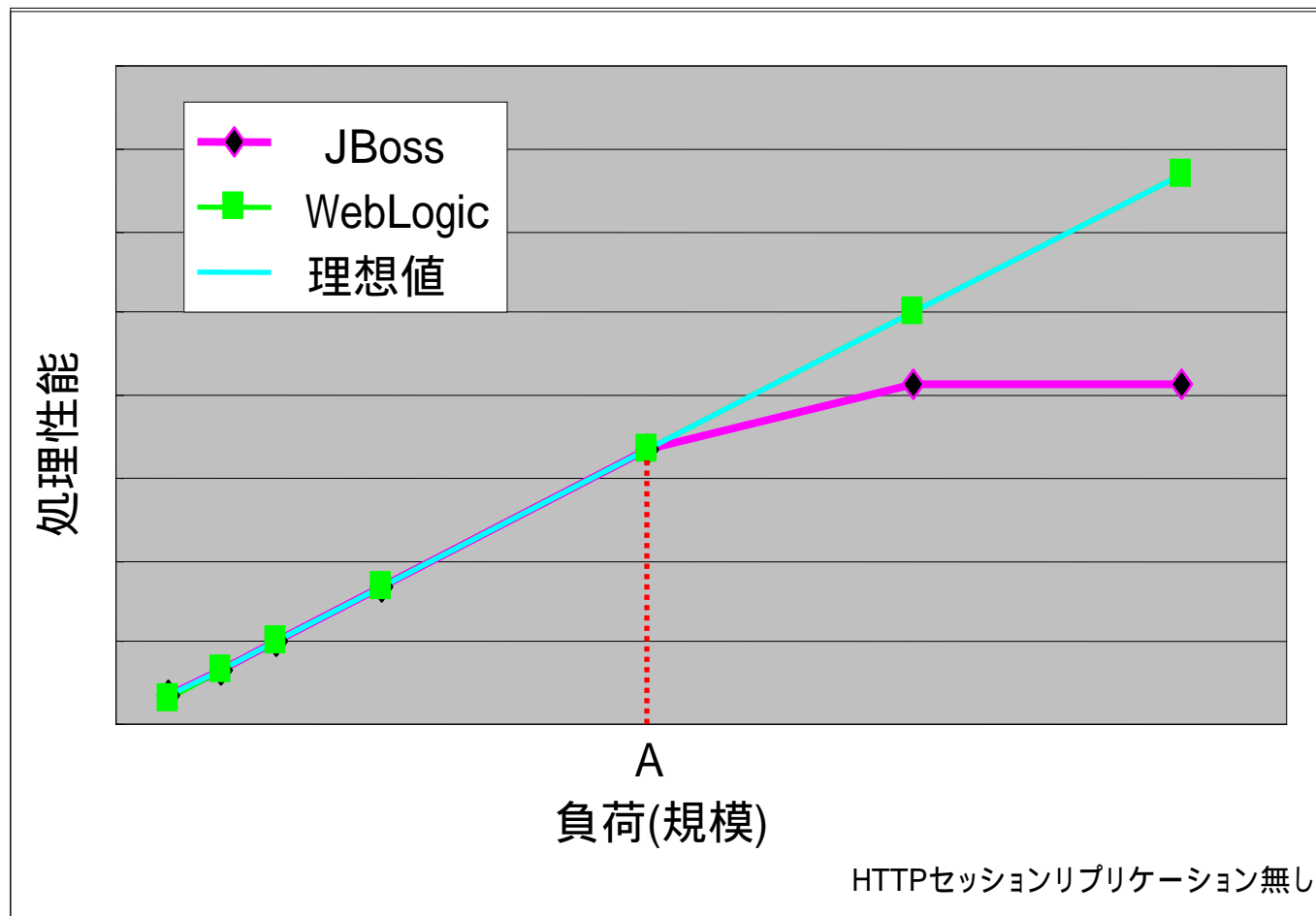


メソッドの実行回数・時間の可視化により、ボトルネック解析と効率的なチューニングに有効

## 8. Java AP層評価結果 (2)評価結果の詳細



### 商用ソフトとの比較

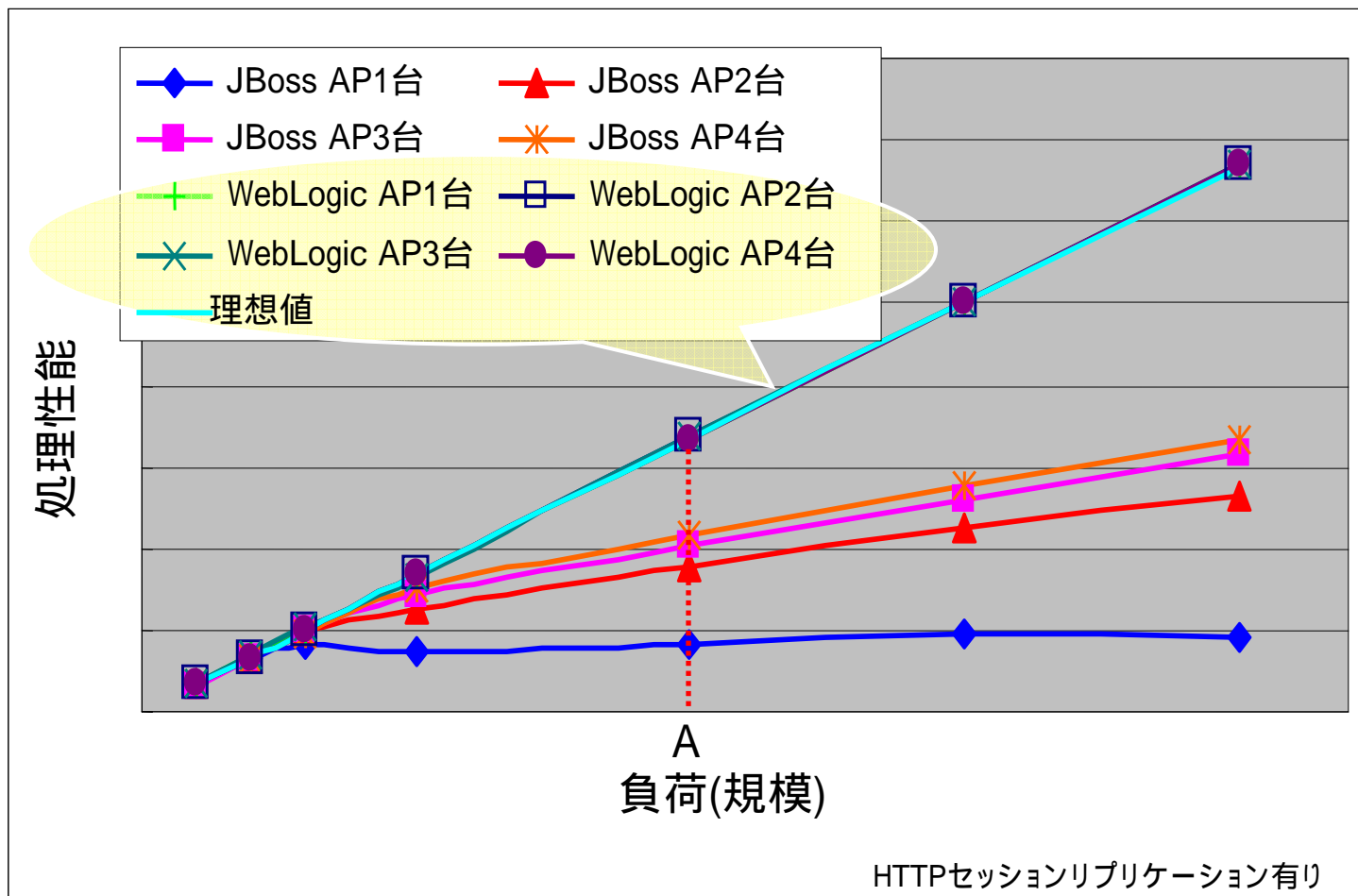


測定した負荷の範囲内では、WebLogicの性能劣化は見られなかった(性能限界が高い)。JBossでは、負荷A程度までが限界で、それ以上の負荷ではレスポンスタイムが低下する。

## 8. Java AP層評価結果 (2)評価結果の詳細

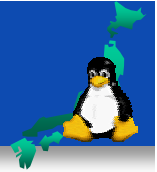


### 商用ソフトとの比較



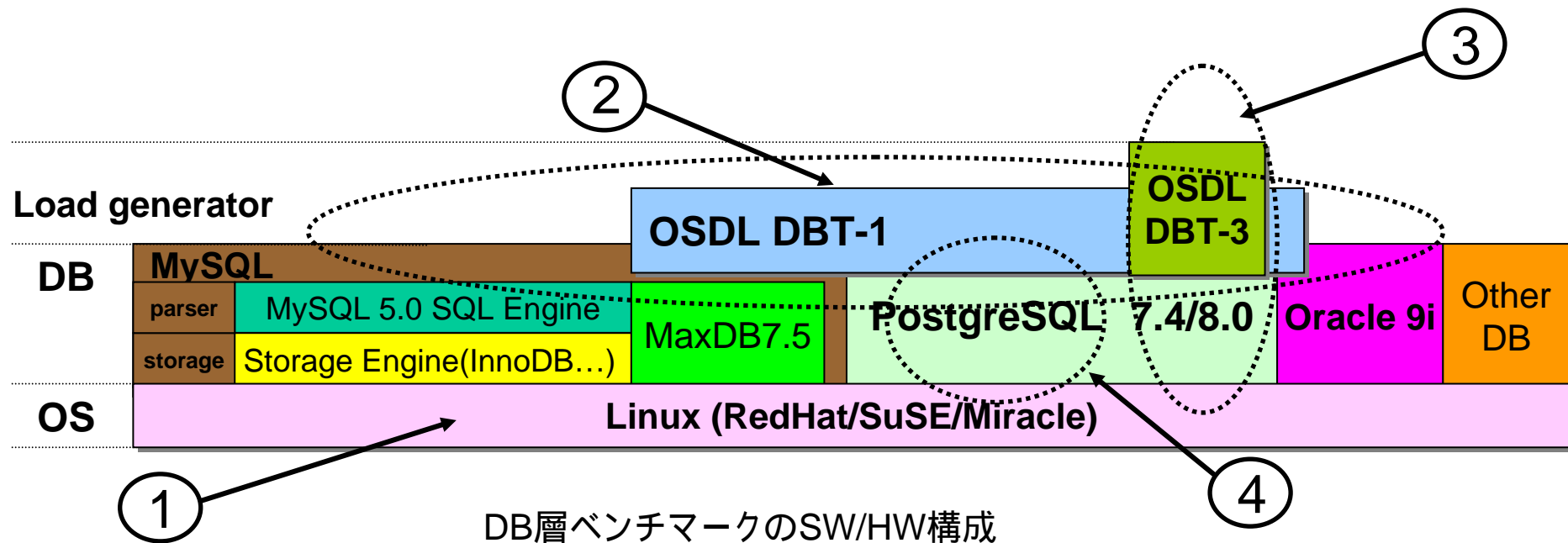
HTTPセッションリプリケーションありでは、JBossとWebLogicの性能差は拡大する。  
JBoss4.0.2では改善される予定(バグ)。

## 9. DB層評価結果 (1)評価項目



### 2004年度ベンチマーク評価項目

1. カーネル2.4と2.6の比較(2.6の性能や信頼性はどうか、等)
2. Web系処理性能の評価(Webシステムにおける性能・信頼性、等)
3. DSS系処理性能の評価(DSSシステムにおける性能・信頼性、等)
4. 大規模DB性能(運用性)の評価(大規模データのバックアップ、ロード、インデックス再構成、等)

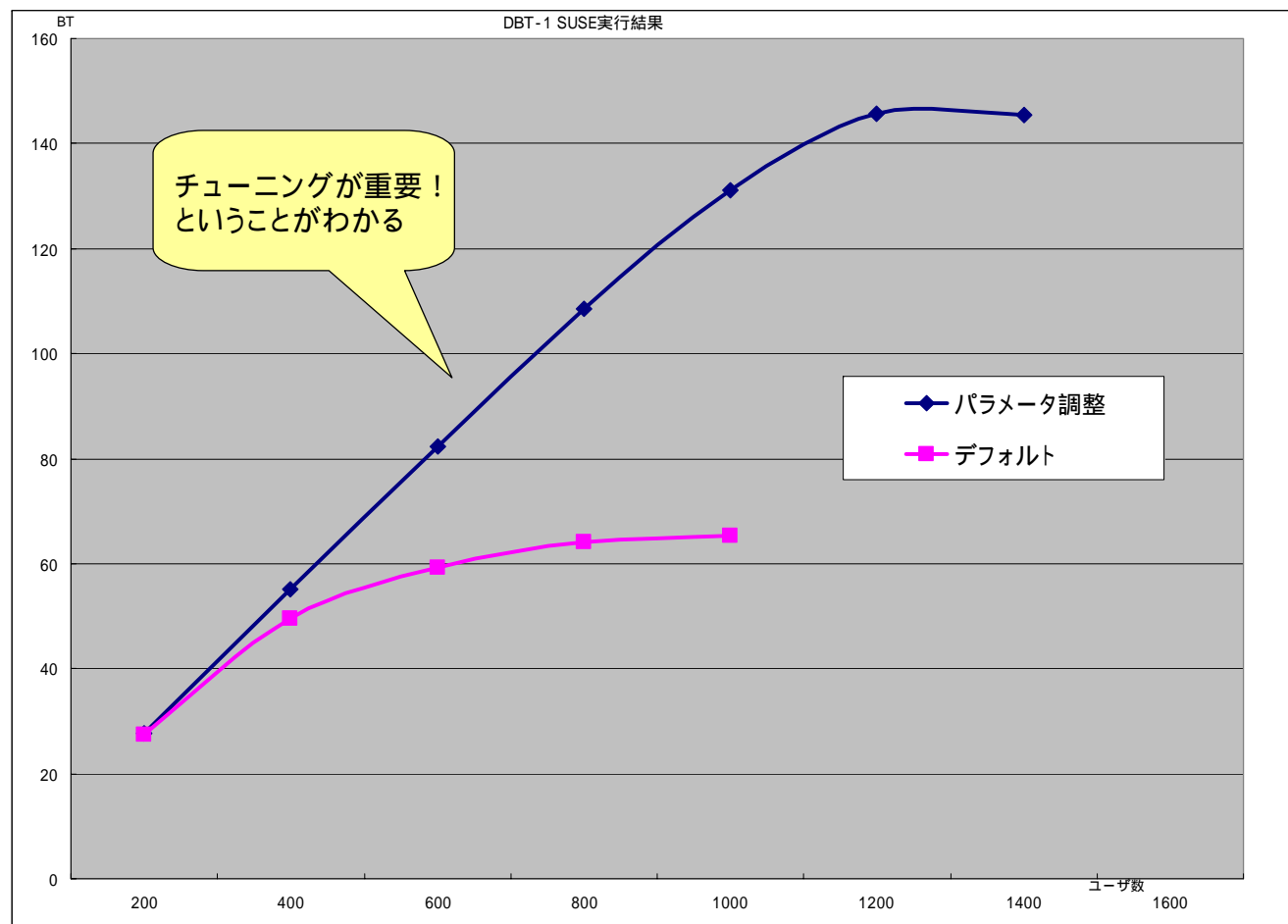


## 9. DB層評価結果 (2)評価結果の詳細



### OSDL DBT-1の結果

-MaxDB7.5によるDBT-1のBT値(擬似トランザクション処理数)遷移



#### BT値

-simultaneous connection=1

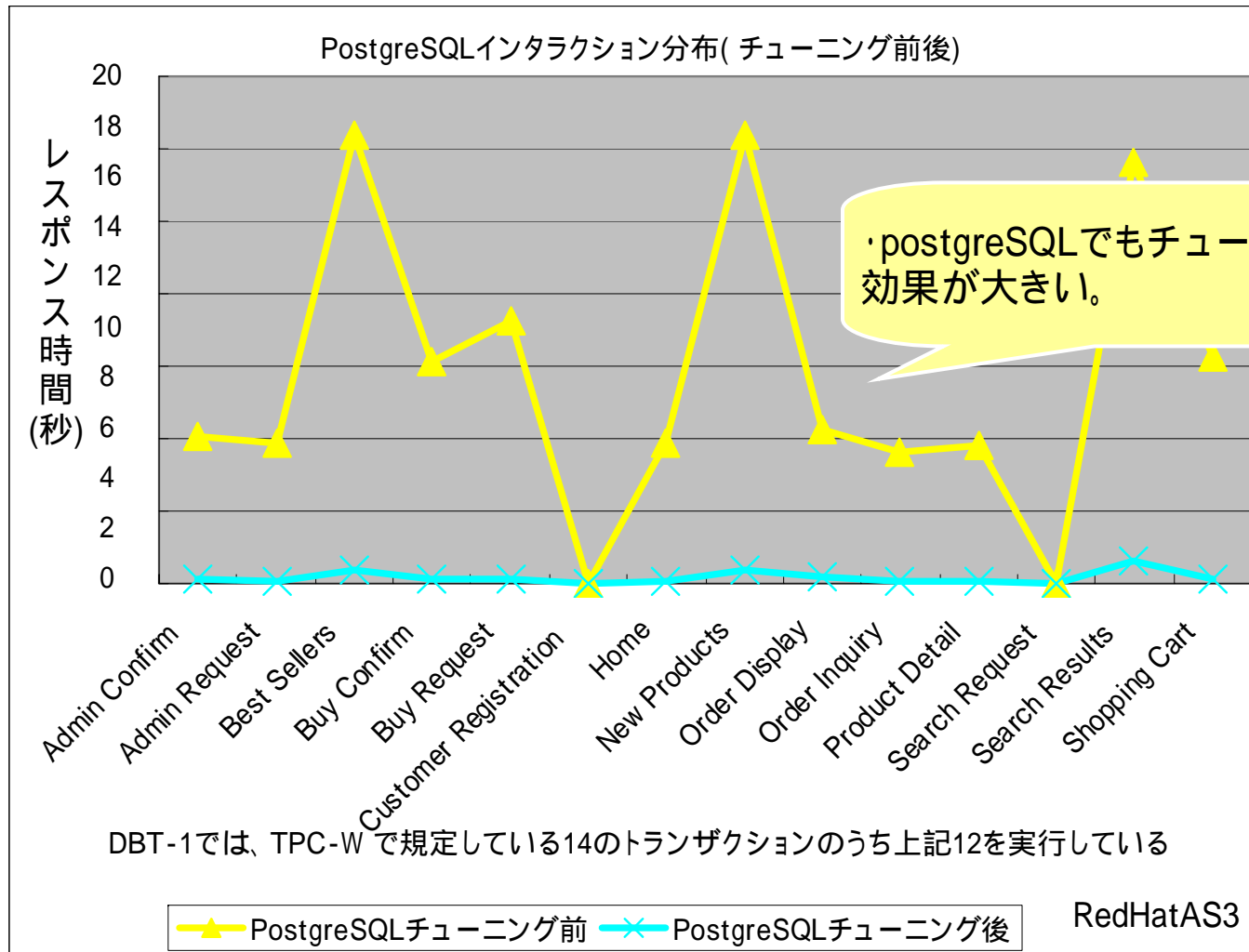
SUSE LINUX Enterprise Server 9  
MaxDB7.5  
CPU Intel Xeon 3.6GHz (HT) Dual  
Memory 4GB

## 9. DB層評価結果 (2)評価結果の詳細



### OSDL DBT-1の結果

#### -PostgreSQLのチューニング前後の比較

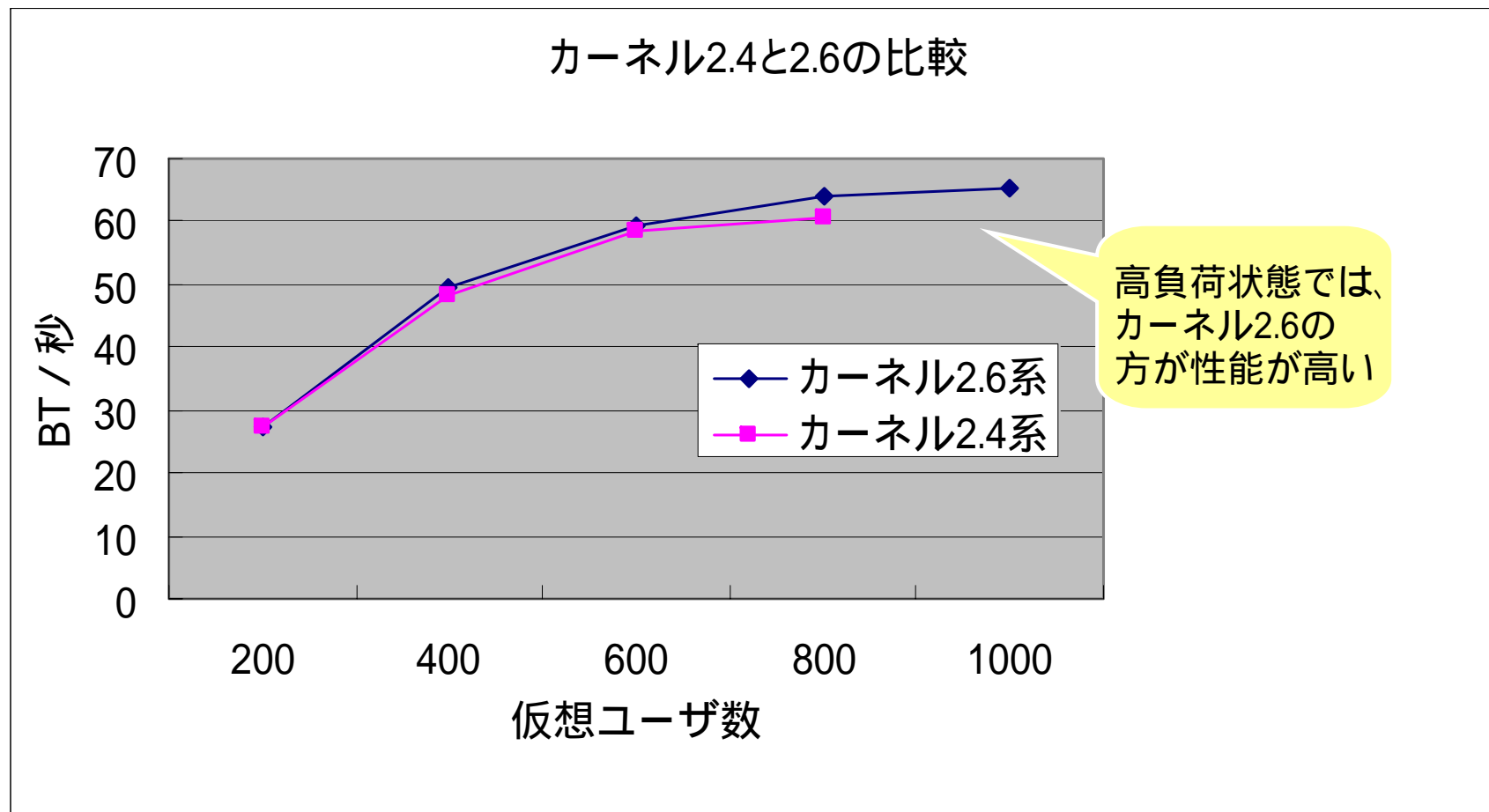


## 9. DB層評価結果 (2)評価結果の詳細



### OSDL DBT-1の結果

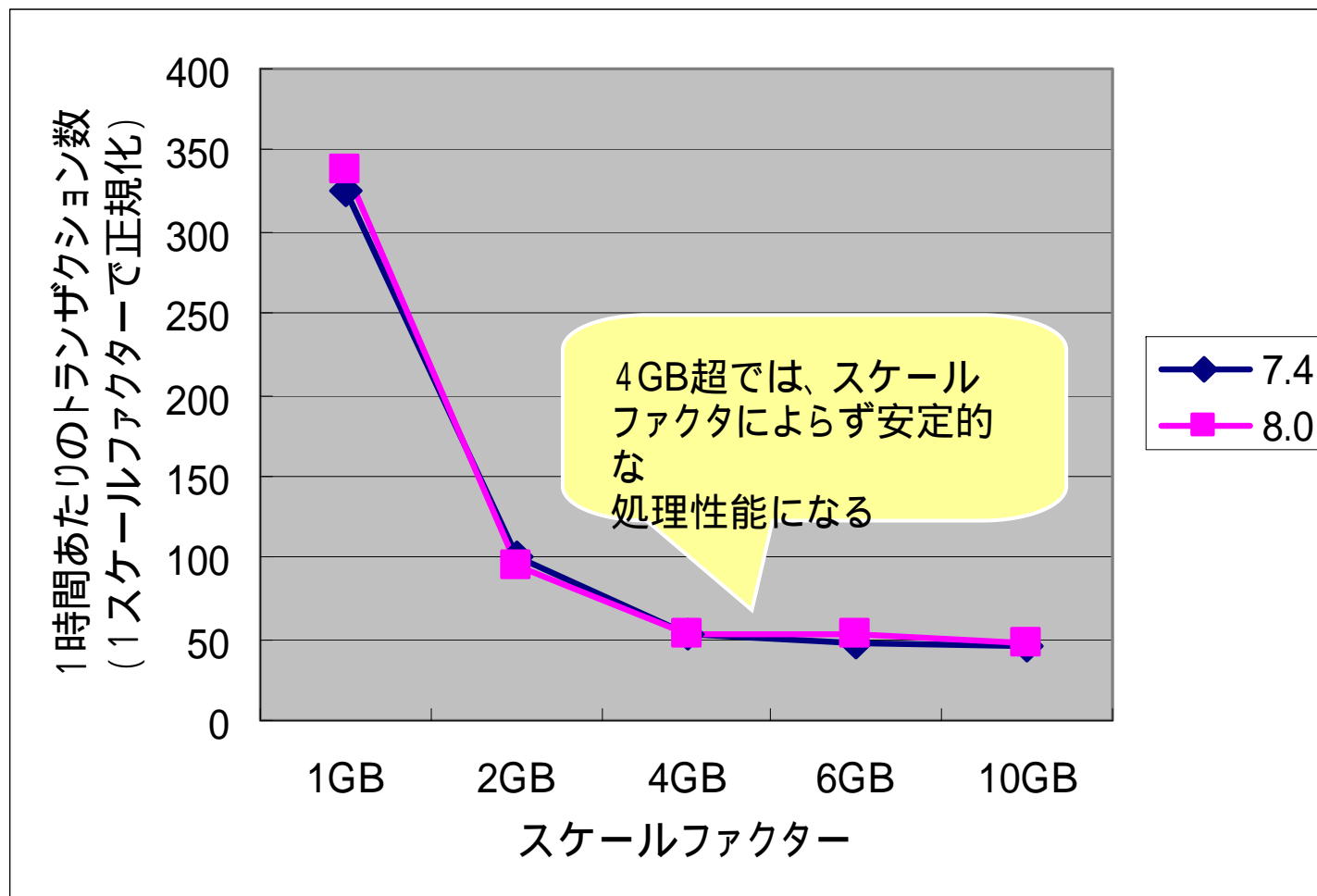
- カーネル2.4と2.6の比較



## 9. DB層評価結果 (2)評価結果の詳細



### OSDL DBT-3の結果

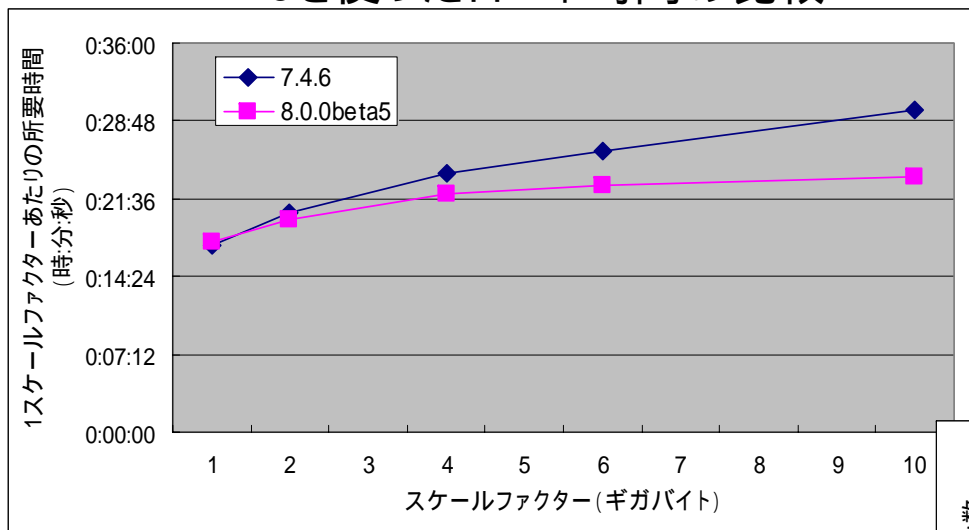


## 9. DB層評価結果 (2)評価結果の詳細



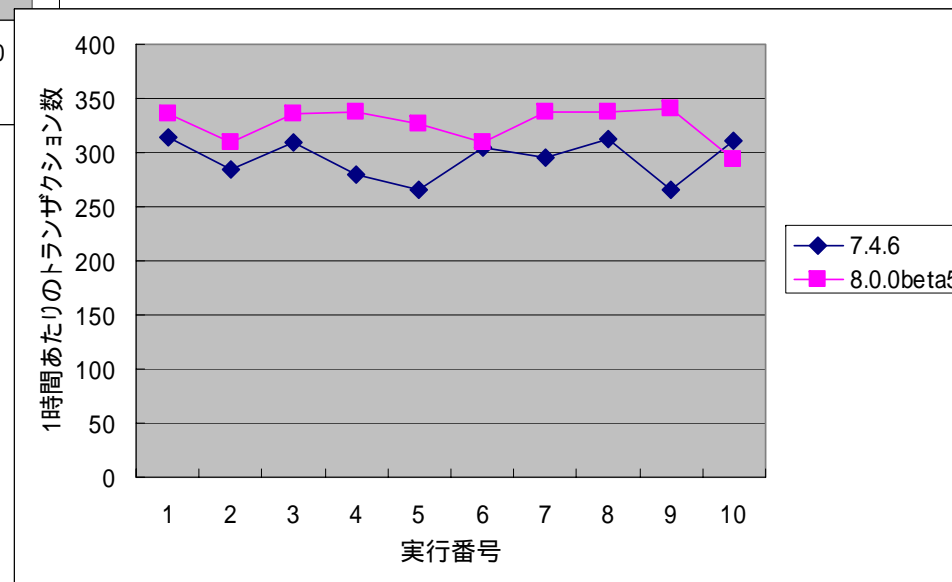
### DBT-3によるPostgreSQL7.4と8.0の比較

- DBT3を使ったロード時間の比較



PostgreSQL8.0では  
着実な進歩が見られる

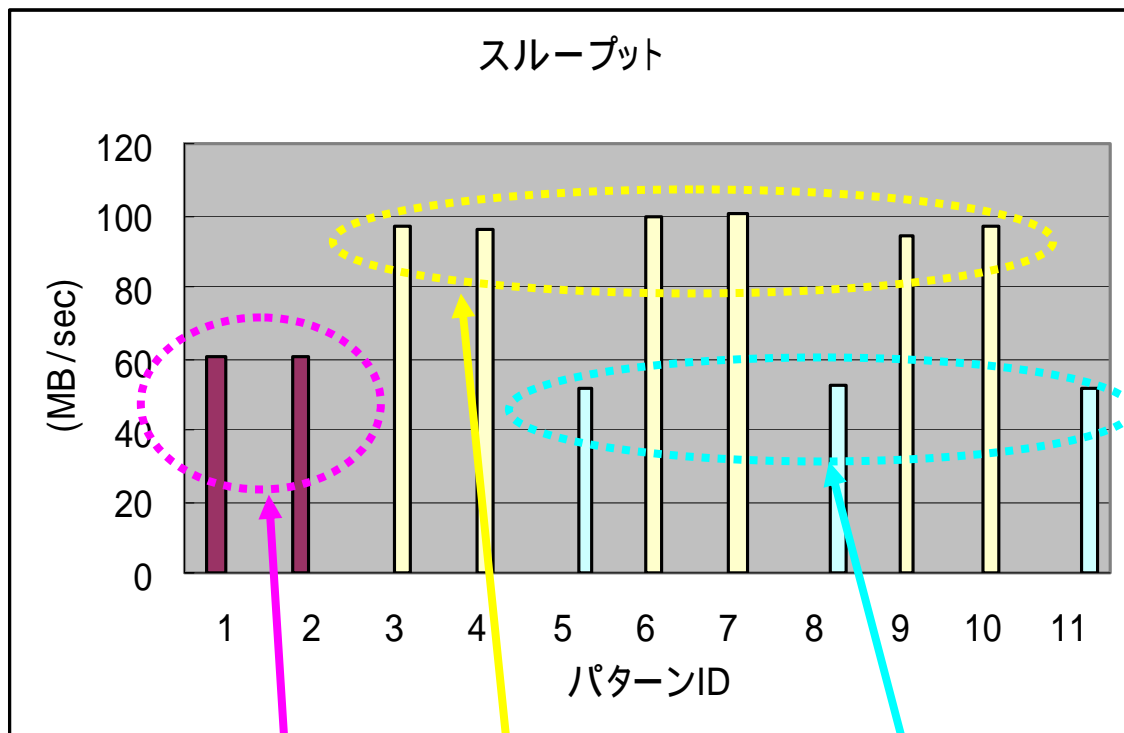
- DBT3パワーテストによる比較



# 1.0. OS層評価結果 (1)高負荷状態の解析



iozoneを様々なパラメータ(ID1 ~ 11)で実行し、高負荷状態におけるカーネル内部での挙動の違いをOprofile,LKSTを使って検証した



CPUアイドル型

CPUビジー型

ロック競合型

CPUアイドル型: 8GBのファイルのI/O待ちで発生(I/Oネック)

CPUビジー型: iozoneプロセスからページキャッシュへのデータコピー処理がネック

ロック競合型: グローバルなカーネルロックで発生(複数のiozoneコマンド起動時)

パターンID	ファイルサイズ	プロセスorスレッド数	スレッドの使用	iozoneコマンド実行数
1	8G	1	P	1
2	8G	1	T	1
3	4G	2	P	1
4	4G	2	T	1
5	4G	1	P	2
6	2.7G	3	P	1
7	2.7G	3	T	1
8	2.7G	1	P	3
9	2G	4	P	1
10	2G	4	T	1
11	2G	1	P	4

# 1.0. OS層評価結果 (1)高負荷状態の解析



## CPUアイドル型 (ID=1)とCPUビジー型 (ID=9)をOprofileで分析

### 1. CPUビジー型

Oprofileによるサンプリング結果(ID=9)

順位	シンボル	サンプル数	占有率(%)
1	do_generic_file_write	8,462	16.16
2	get_hash_table	4,584	8.754
3	unlock_buffer	3,364	6.424
4	__br_write_lock	2,763	5.276
5	__write_lock_failed	1,989	3.798

- ・上の結果から、do\_generic\_file\_write() の実行頻度が一番高いことがわかる。
- ・細分化したところ0xc0160de8の命令が実行頻度が一番高いことが判明。
- ・ソースとの突合せの結果、\_copy\_from\_user()が最も実行頻度が高いことが判明。
- ・\_copy\_from\_user()はユーザ空間のデータをカーネル空間のページキャッシュに書き込む(コピーする)処理  
I/Oの中心処理なので、この処理の実行頻度が高いことは効率よい書き込み処理がされたことを示している

### 2. CPUアイドル型

Oprofileによるサンプリング結果(ID=1)

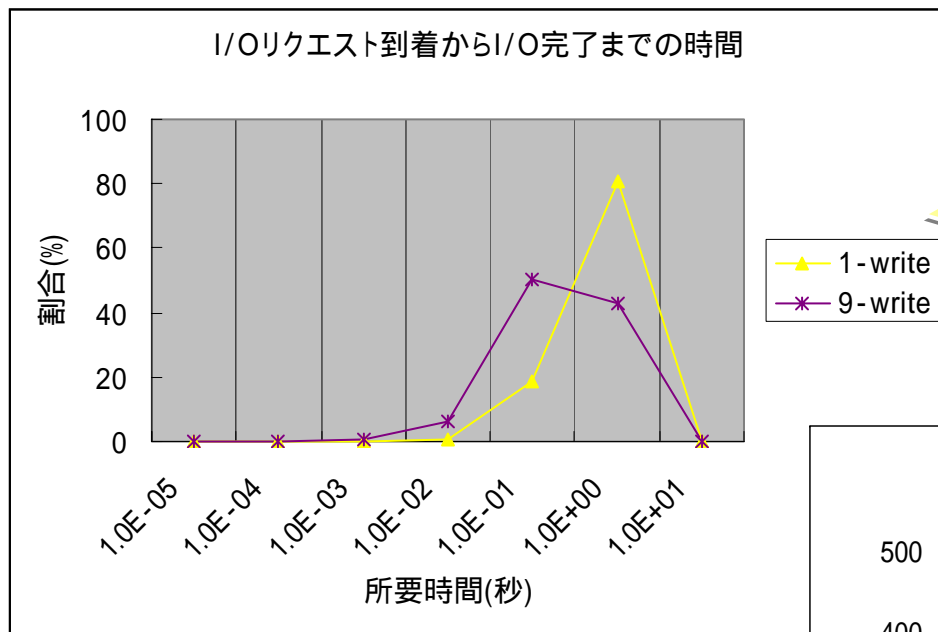
順位	シンボル	サンプル数	占有率(%)
1	default_idle	1,993	24.37
2	try_to_free_buffers	532	6.504
3	launder_page	532	6.504
4	LKST_ETYPE_MEM_SWAPOUT_HEADER_hook	371	4.536
5	unlock_page	365	4.462

- ・上の結果から、default\_idle() の実行頻度が一番高いことがわかる。
- ・細分化したところhltの命令が実行頻度が一番高いことが判明。  
I/O待ちでCPUがアイドル状態であったことを示している

# 1.0. DB層評価結果 (1)高負荷状態の解析

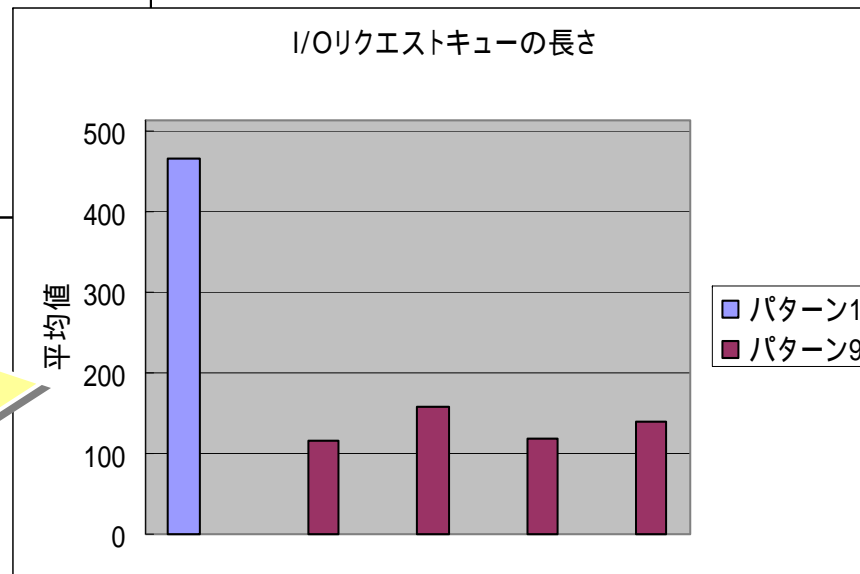


## CPUアイドル型 (ID=1)とCPUビジー型 (ID=9)をLKSTで分析



CPUアイドル型(ID=1)では、CPUビジー型(ID=9)に比べ、I/O処理にかかる時間のピークが10倍違う(1秒vs0.1秒)

CPUアイドル型(ID=1)では、キュー長が最大の512に近いことから、ディスクの処理限界性能付近で動作していたことがわかる

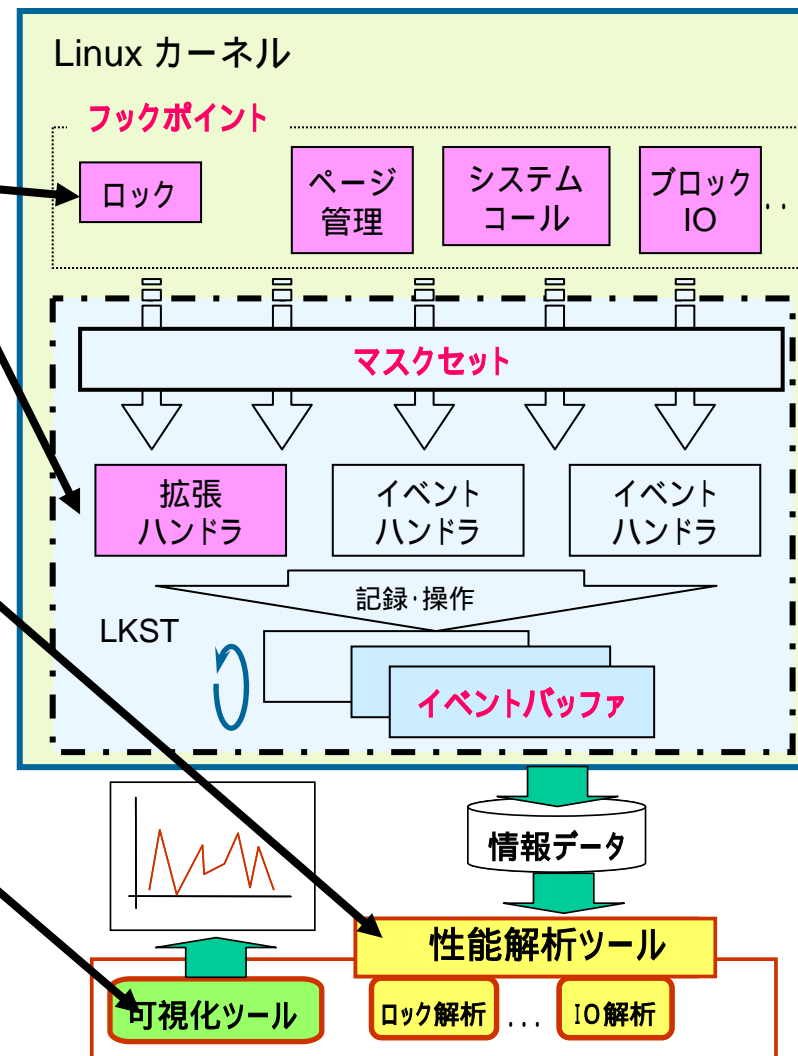


# 1.1. 解析ツール (1)トレーサ(LKST)



## カーネル性能評価向けLKST(Linux Kernel State Tracer)

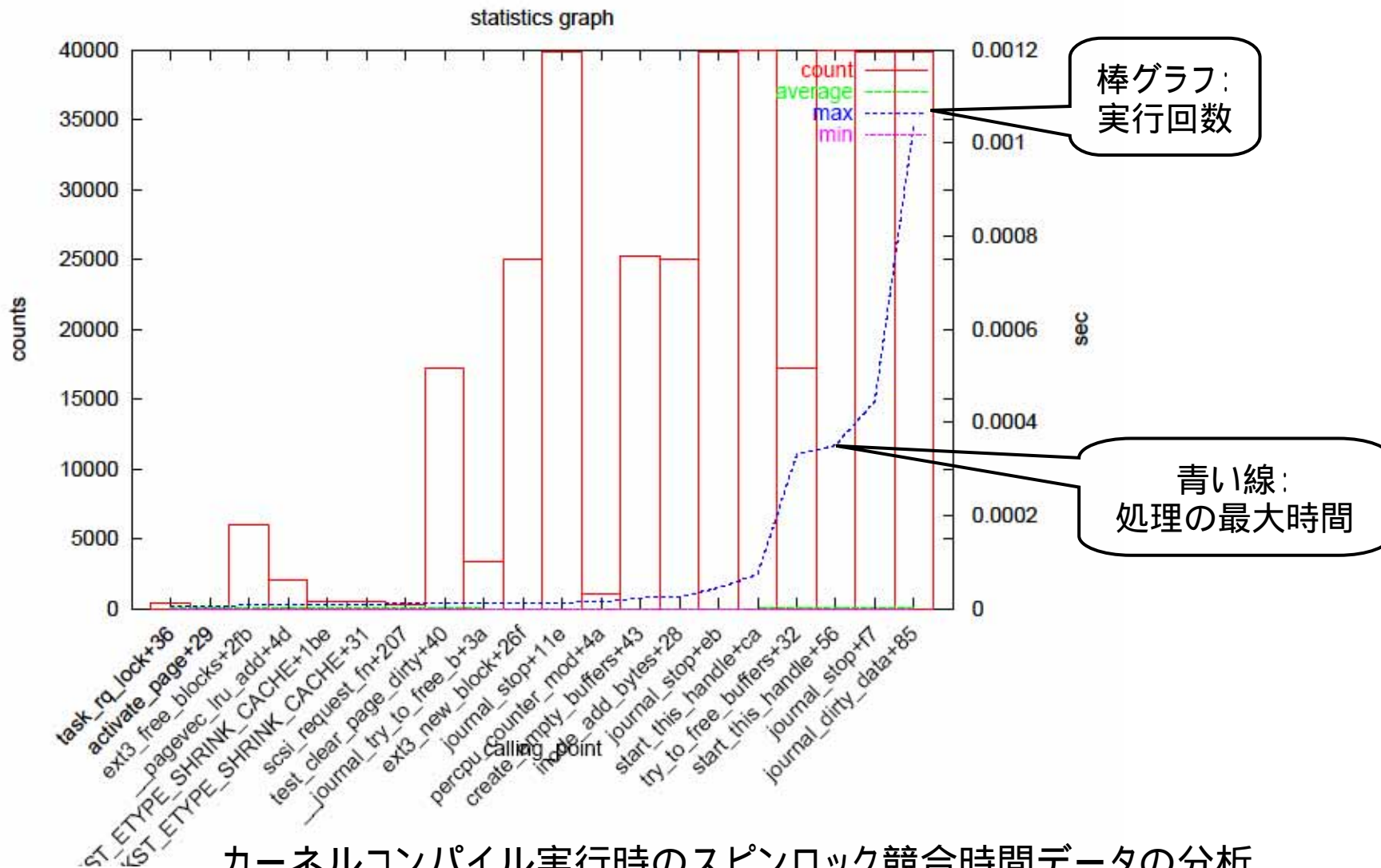
- 性能評価ポイントで情報を取得する「LKST機能拡張」
- 取得した情報を解析する「性能解析ツール」
- 解析した情報を可視化する「可視化ツール」



# 1.1. 解析ツール (1)トレーサ(LKST)



## LKSTの出力結果



# 1.1. 解析ツール (2)ディスク割当評価ツール(DAVL)



## 2004年度開発結果

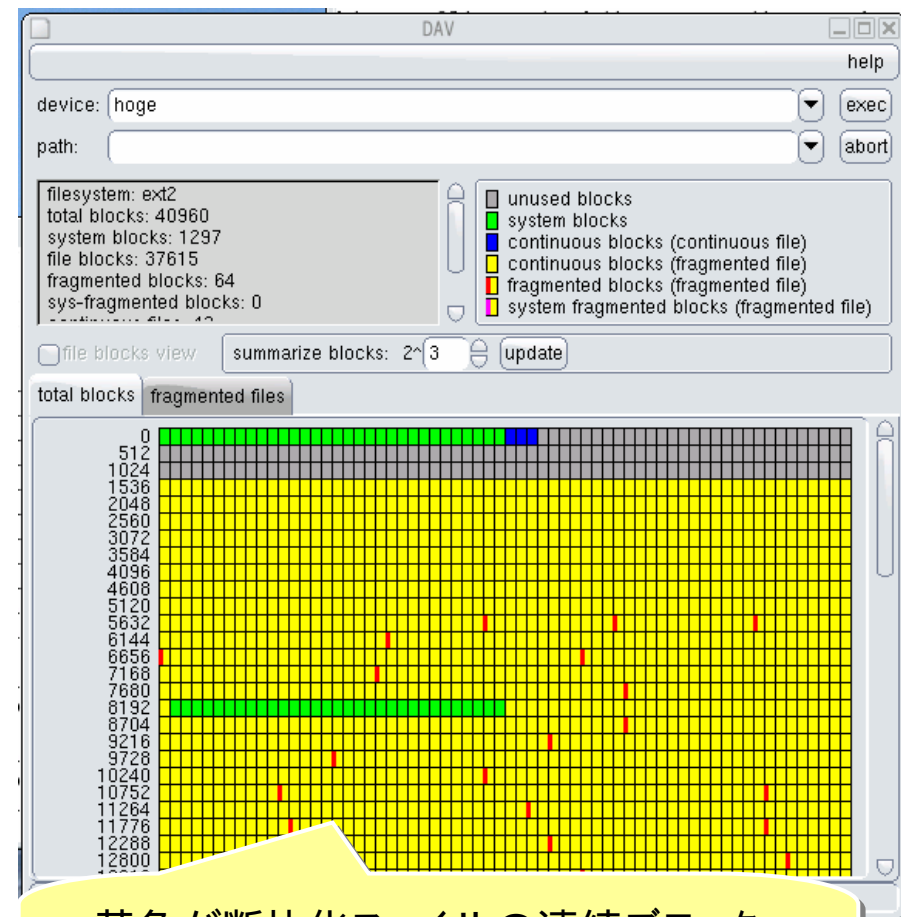
性能劣化の原因やデフラグツールの効果を、視覚的に裏付け情報を取得する手段の提供 (右図)

- ・対象パーティション、ディレクトリ、ファイルのフラグメンテーション状態がどうなっているかの確認。
- ・デフラグツール適用前後における、Disk内の状況を把握可能。
- ・DAVLはフラグメンテーション情報を収集するツール(dac)と情報を可視化するツール(dav)で構成

Diskのフラグメント状態の遷移を取得する手段および視覚化する手段の提供

- ・dacで定期的に情報を保存
- ・保存したデータファイルを指定して、davを実行  
フラグメンテーションがどう進んでいくのかを確認可能。  
事前評価として行なうことで、サイジングに活用できる。  
但し、本年度は手動で保存。

日経Linux7月号特選フリーソフトで紹介



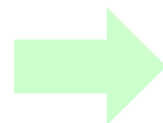
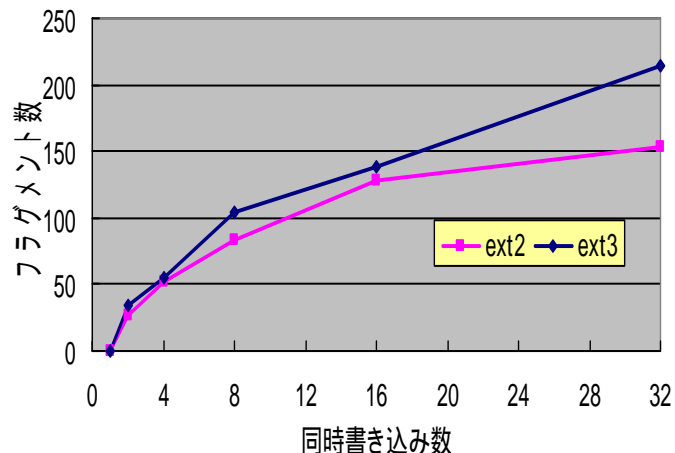
黄色が断片化ファイルの連続ブロック、オレンジが断片化部分状態を表す。(青は断片化していないファイル)

# 1.1. 解析ツール (2)ディスク割当評価ツール(DAVL)

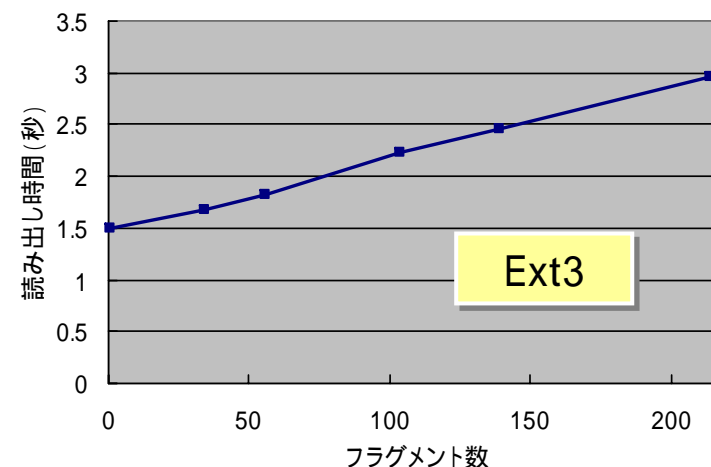
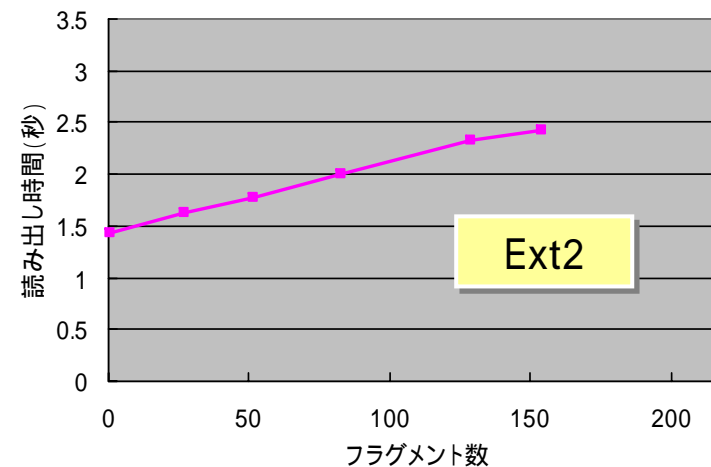


## フラグメンテーションとファイルread性能の関係

### ツールでフラグメンテーションを発生させる



### ファイルのRead性能を測定



### フラグメンテーション発生ツール:

- ・子プロセス毎に32MBのファイルを他パーティションにcopyする。子プロセスは合計32個。
- 同時書き込み数とはその子プロセスの同時実行数。
- ・上記の性能はcopy後のファイルをreadした性能。

### 実行結果:

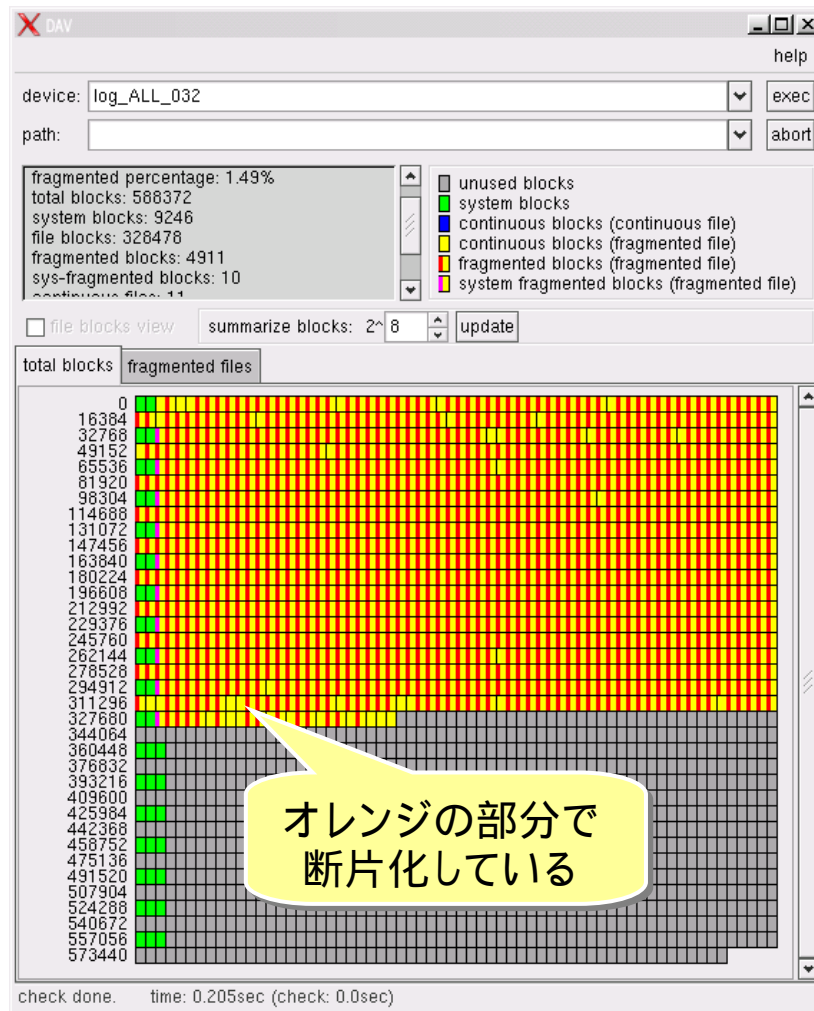
- ・並列実行度が上がるにつれてフラグメンテーション数が増加する
- ・フラグメンテーション数が増加するとread性能が低下する。

### 測定条件:

ハードウェア:  
・CPU:P4 2GHz  
・メモリ:512MB  
・HDD:30GB\*2パーティション

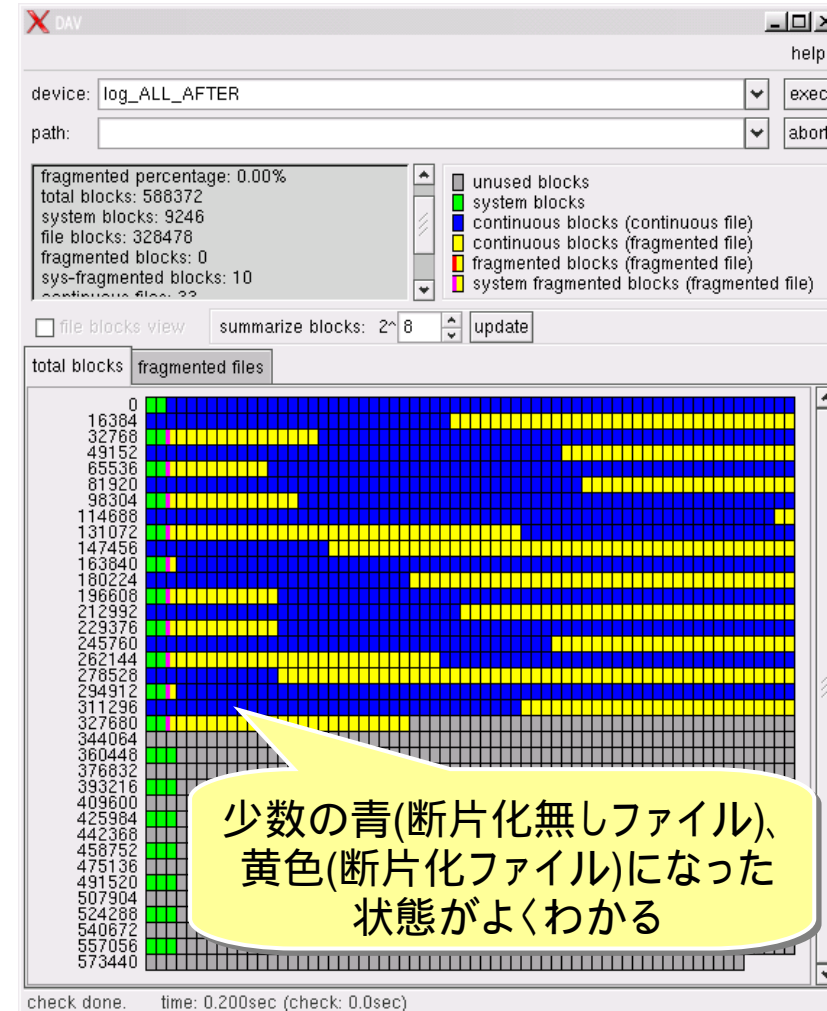
ソフトウェア:  
・OS - Miracle Linux 3.0  
・Filesystem - Ext2/Ext3  
(ブロックサイズ:4KB)

# 1.1. 解析ツール (2)ディスク割当評価ツール(DAVL)



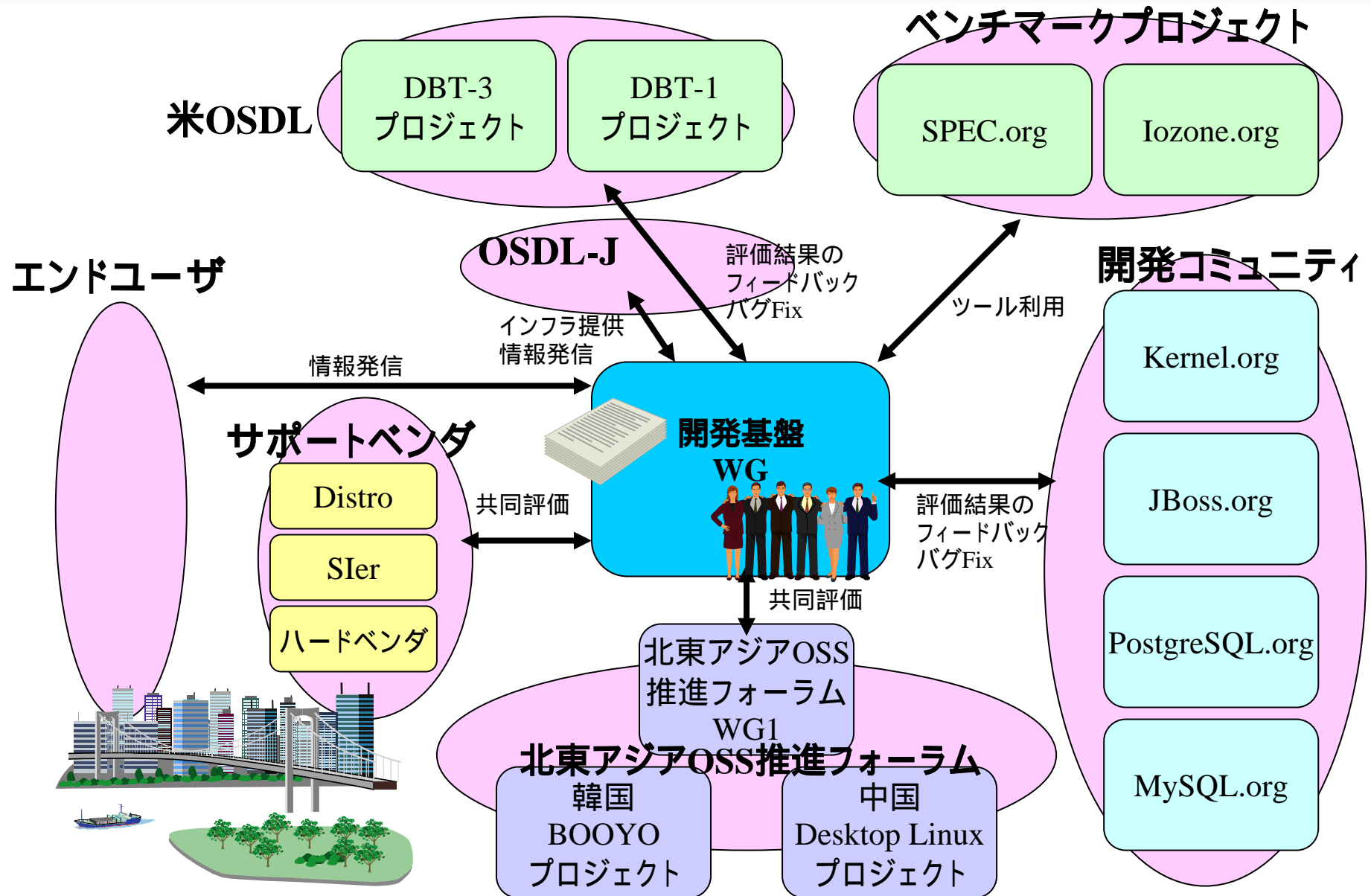
デフラグ実行前

デフラグツール: defrag 0.73 pjml



デフラグ実行後

# 12. 開発基盤WGとコミュニティとの関係

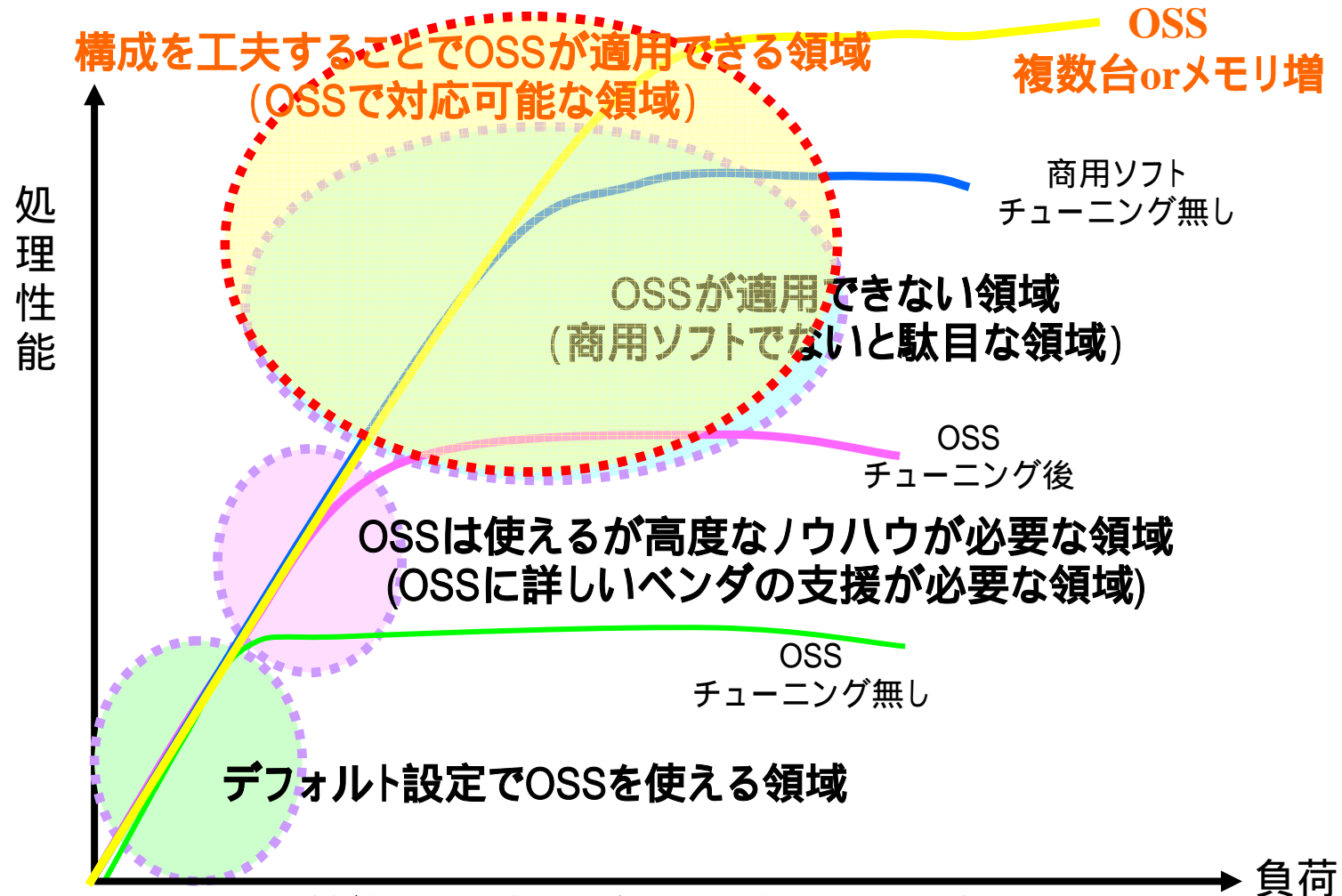


# 1.3. 2005年度の活動(1)



## 2005年度の活動目標

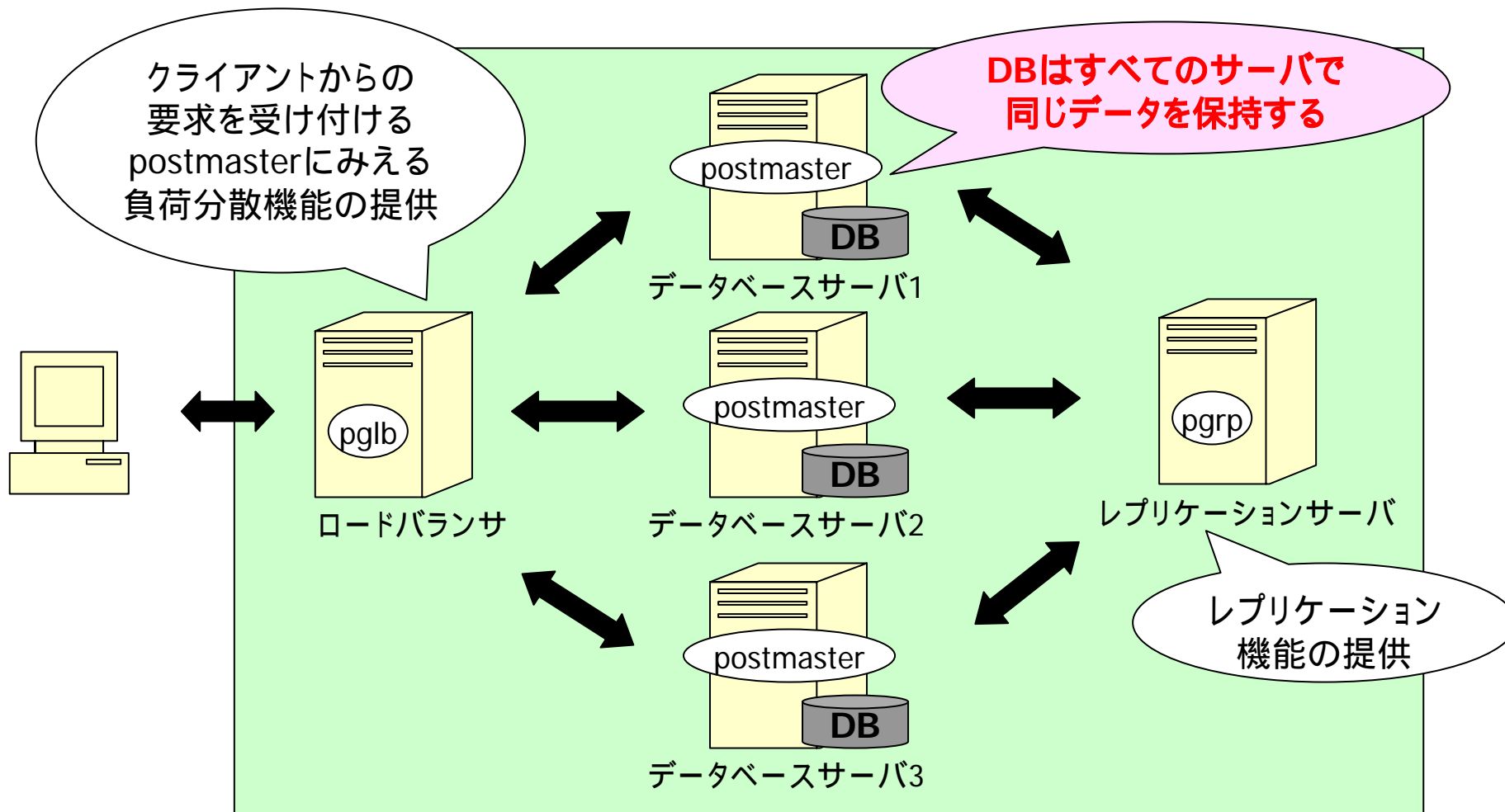
04年度は単体構成(同一条件)での比較だったが、05年度はクラスタ化、メモリ増強等のシステム構成の工夫によりOSSが適用可能な領域をノウハウを含めて明らかにする。



# 1.3. 2005年度の活動(2)



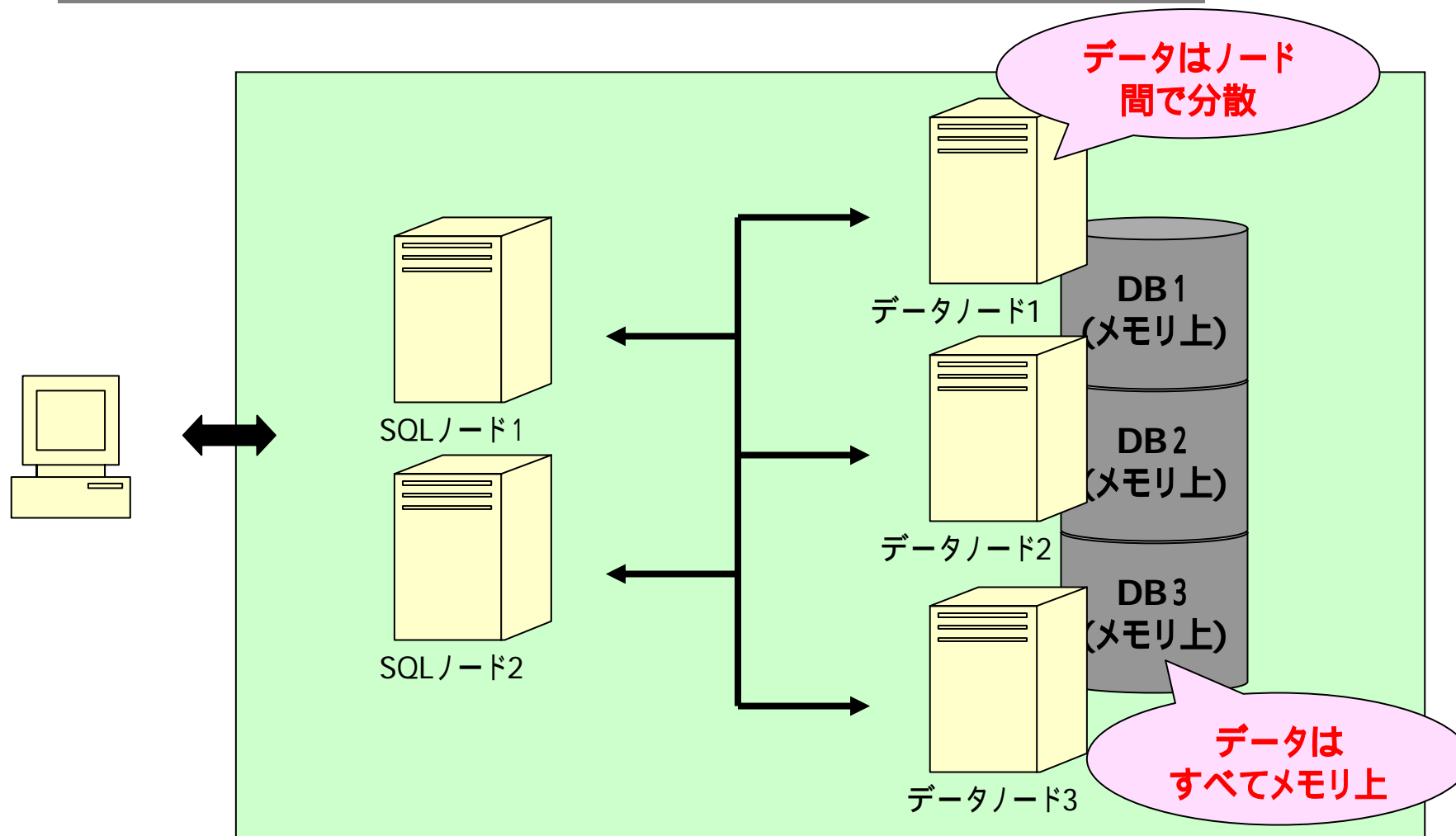
## DBのスケラビリティ評価(1) ~ PGCluster評価の構成



# 1.3. 2005年度の活動(3)

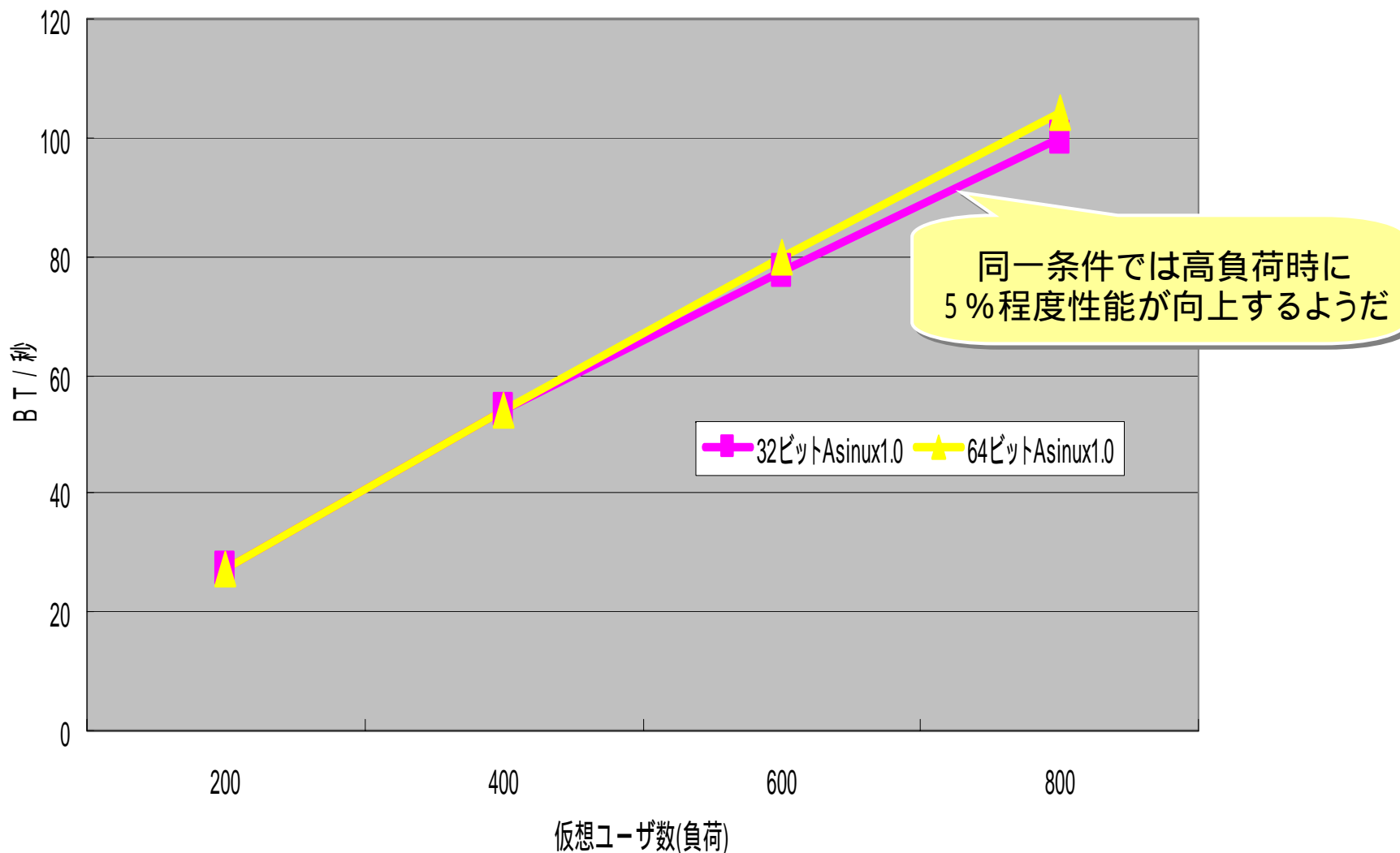


## DBのスケラビリティ評価(1) ~ MySQL Cluster評価の構成





## 評価途中データ EM64Tで性能がどうなるか？



## 今日のお話(まとめ)



### OSSを基幹業務で使うための性能・信頼性評価手法

- 単なるベンチマークではない(最大性能のPR目的でない)
- 業務を想定したベンチマーク(そもそも基幹業務って?)  
= できるだけ実システムに近い
- 対象はOSS(でも商用との比較がしたい)
- 中身と結果に対する詳細な解析ができること(解析ツールとセット)

- できるだけ実システムに近いベンチマークツール  
SPECjAppServer2004, OSDL DBT-1,3

- 商用との比較  
WebLogic、Oracle(手順のみ)

- 中身と結果に対する詳細な解析  
EJBProfiler EJB  
Oprofile、LKST Linuxカーネル

## まとめ



- Linux、OSSミドルは着実に進歩している。また、OSSの開発方式により、今後も発展が期待される
- これまでは明確な性能評価基準がなかったが、ベンダ共同で評価手順を策定し、また、これを共有し、育成していく土壌ができたことは大きな価値がある
- 開発コミュニティにとっても、1つの明確な評価基準が明らかになったことで、これを1つの評価尺度として利用しながら開発を進めることができるはず
- ユーザにとっては、自社のシステムにおいて、OSSがどこまで適用できるのかといった疑問に対し、1つの判断基準を提供できたと考える。また、業務システムを想定したベンチマークと、その結果の解析ツールは、OSSの業務システムへの適用評価に有効と考える。
- 今後は
  - ・評価対象のバリエーションを増やしていくこと
  - ・WorldWideに広く普及させるための活動を継続していくことを実施していきたい

詳細資料: <http://www.ipa.go.jp/software/open/forum/DevInfraWG.html>

ダイジェスト版: <http://www.thinkit.co.jp/free/compare/9/1/1.html>

