

OSC2005

# SPECjAppServer2004によるJBossの性能・信頼性評価

-商用APサーバとの比較-  
-TIPS-

2005年3月25日

(株)NTTデータ  
オープンソース開発センタ  
金子 崇之





## 検証目的

- SPECjAppServer2004の**測定手順の確立**
  - JBossクラスタ環境
  - **WebLogicクラスタ環境**
- OSSアプリケーションサーバの**適用領域を明確化**
  - 商用と比較することで、OSSだけでは分からないことを

## 検証内容

- APサーバ台数を**1台～4台**に変化させ性能比較
  - スループット/応答時間
  - CPUリソース等のサーバリソース
- スケーラビリティを確認
- **HttpSessionレプリケーション**を使用した測定を実施
  - ネットワークを介したサーバメモリ間のHttpSessionコピー
  - システム全体の性能と信頼性を高める方式



## 測定パターンと比較

測定	APサーバ	HttpSession レプリケーション	比較A	比較B	比較C	比較D
1	JBoss	有り	○			○
2	JBoss	無し	○		○	
3	WebLogic	有り		○		○
4	WebLogic	無し		○	○	

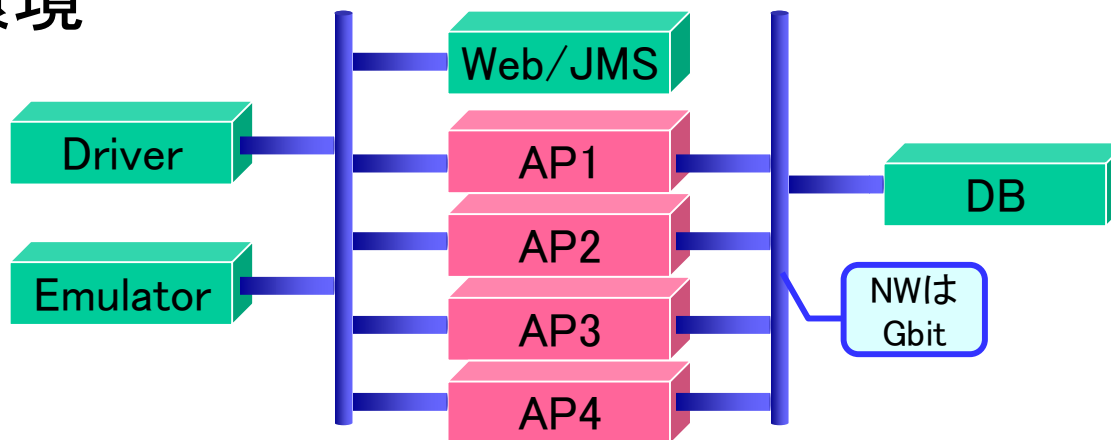
- 比較A: JBossのHttpSessionレプリケーション有り/無し比較
- 比較B: WebLogicのHttpSessionレプリケーション有り/無し比較
- 比較C: JBossとWebLogicの単体比較
- 比較D: JBossとWebLogicのクラスタ比較



これらの比較を元に、適用領域を考察



## 検証環境



サーバ	Driver	Emulator	Web/JMS	DBサーバ	APサーバ
名称	DL360G4				DL360G3
CPU	2x3.4GHz Xeon-(e)				1x2.8GHz Xeon
Mem	2GB			4GB	2GB
Disk	2x36GB HDD(SCSI), RAID0, write-cache				2x36GB HDD(SCSI), RAID1

APサーバのスペックを抑え、DBサーバネックを回避

種別	ソフトウェア
OS	Red Hat Enterprise Linux 3.0 AS(U3)
JDK	J2SE 1.4.2_04 / JRockit1.4.2_04
DB Server	PostgreSQL 7.4.6
AP Server	JBoss 4.0.0 / WebLogic8.1SP3
Web Server	Apache httpd 2.0.46
Connector	mod_jk2 2.0.4 / wl_proxy
Application	SPECjAppServer2004 V1.03



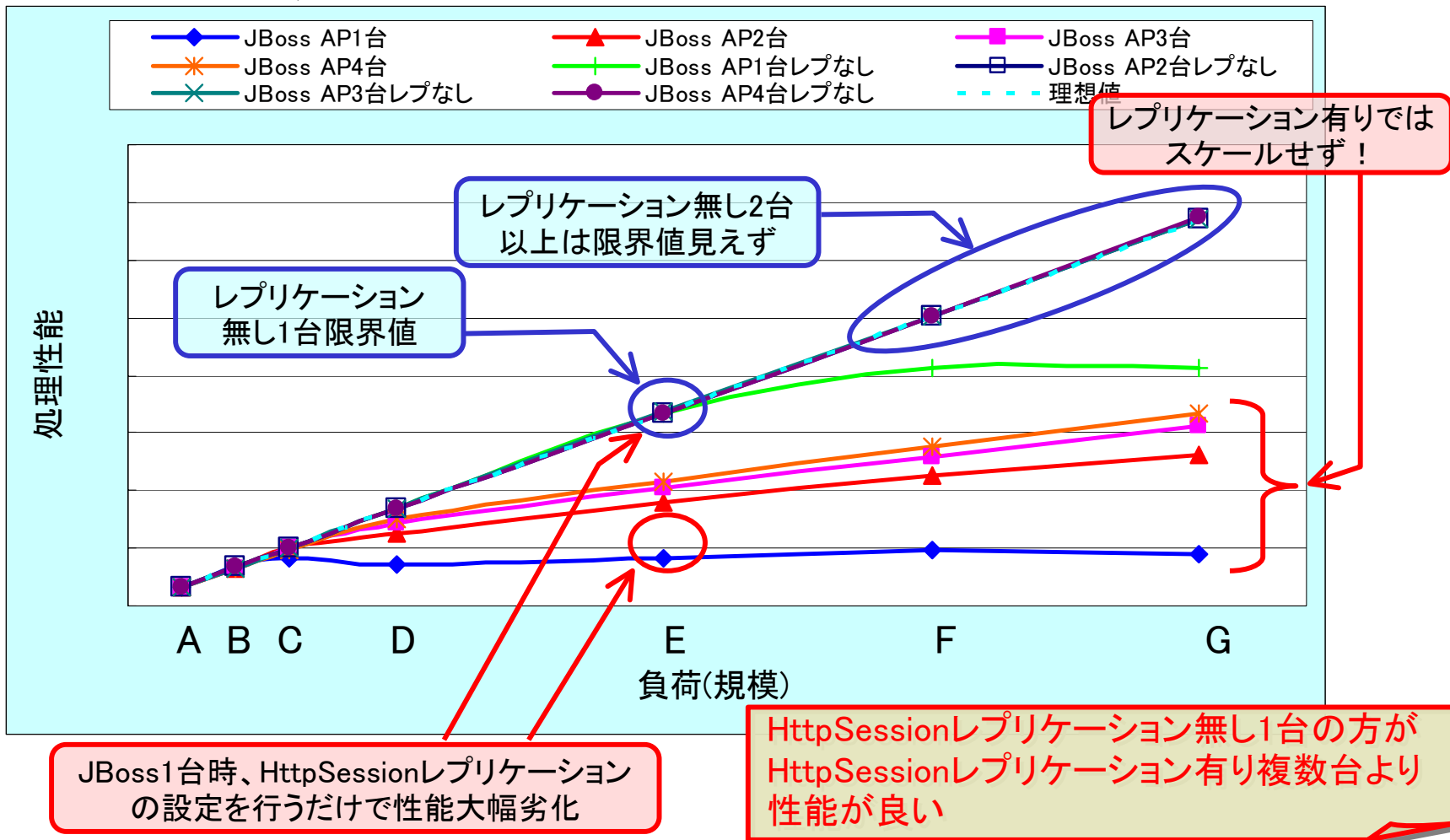
## 比較A: JBossのHttpSessionレプリケーション有り/無し比較

測定	APサーバ	HttpSession レプリケーション	比較A	比較B	比較C	比較D
1	JBoss	有り	○			○
2	JBoss	無し	○		○	
3	WebLogic	有り		○		○
4	WebLogic	無し		○	○	



## JBossクラスタ(HttpSessionレプリケーション有り/無し)

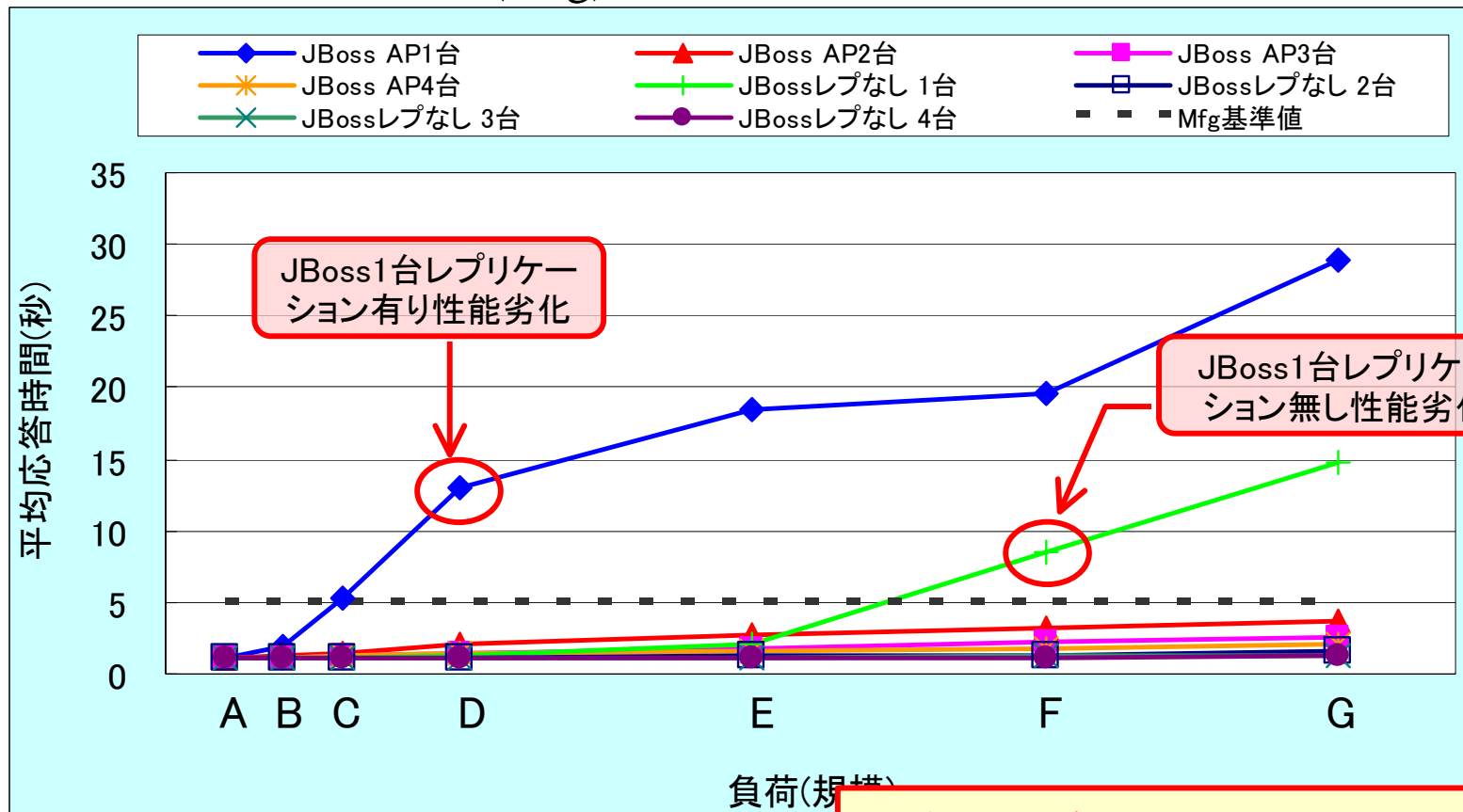
### - スループット





## JBossクラスタ(HttpSessionレプリケーション有り/無し)

- レスポンスタイム(Mfg)

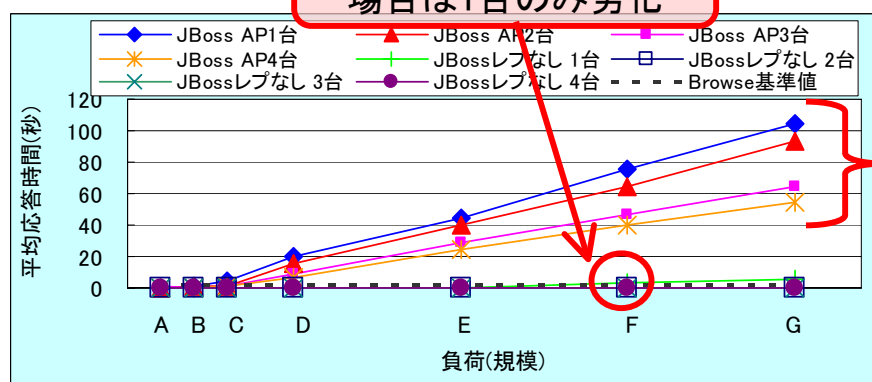
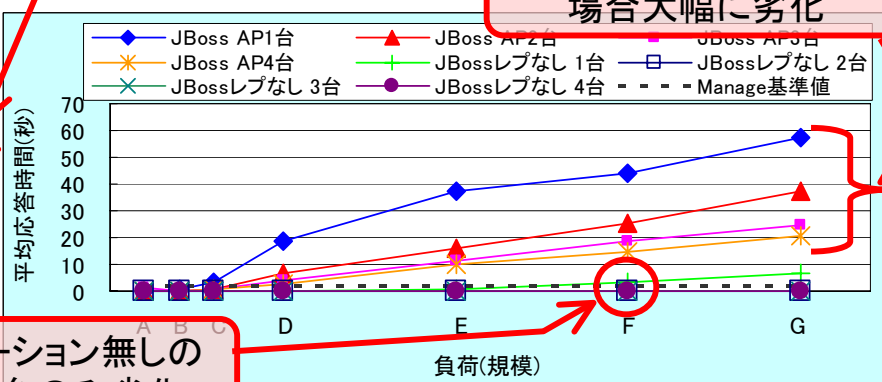
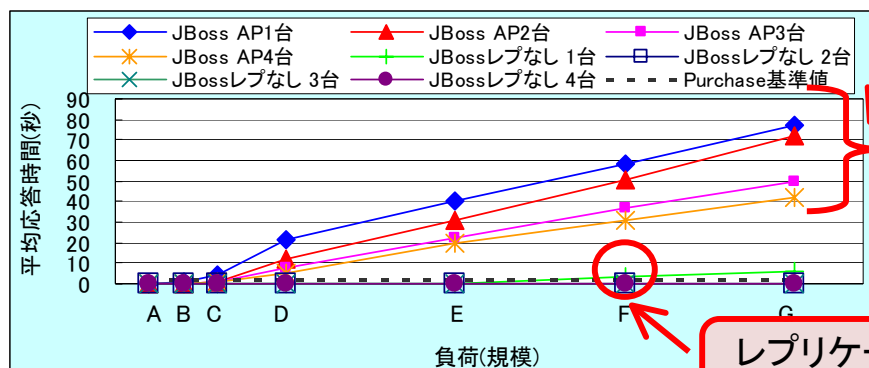


応答時間が劣化しているのは、JBoss AP1台時のみ



## JBossクラスタ(HttpSessionレプリケーション有り/無し)

- レスポンスタイム(Purchase, Manage, Browse)



レプリケーション有りの場合大幅に劣化

レプリケーション無しの場合は1台のみ劣化

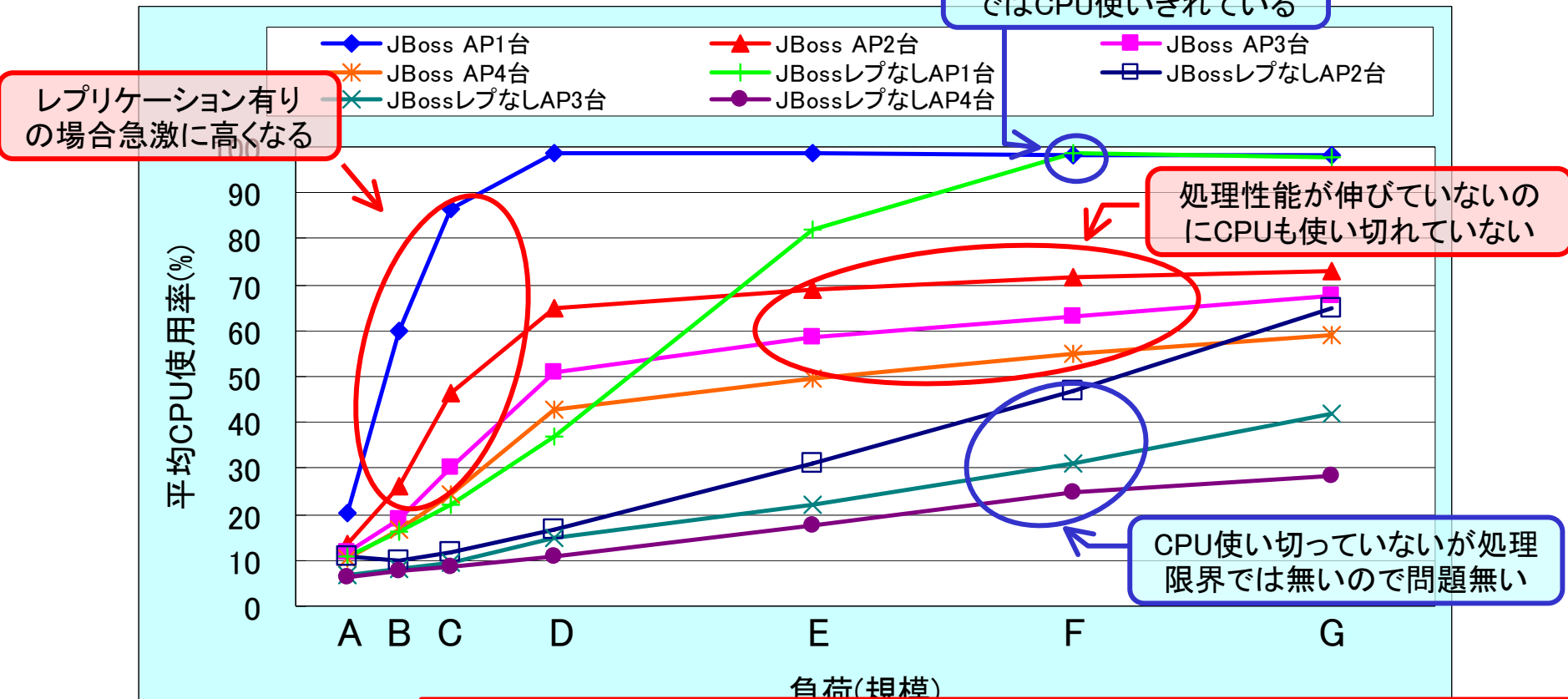
HttpSessionレプリケーションにより大幅に応答時間が劣化





## JBossクラスタ(HttpSessionレプリケーション有り/無し)

### - 平均CPU使用率



レプリケーションが実際に行われているとCPUが使い切れていない  
Mfg(EJB直接アクセス)応答時間はCPU使用率と関連？



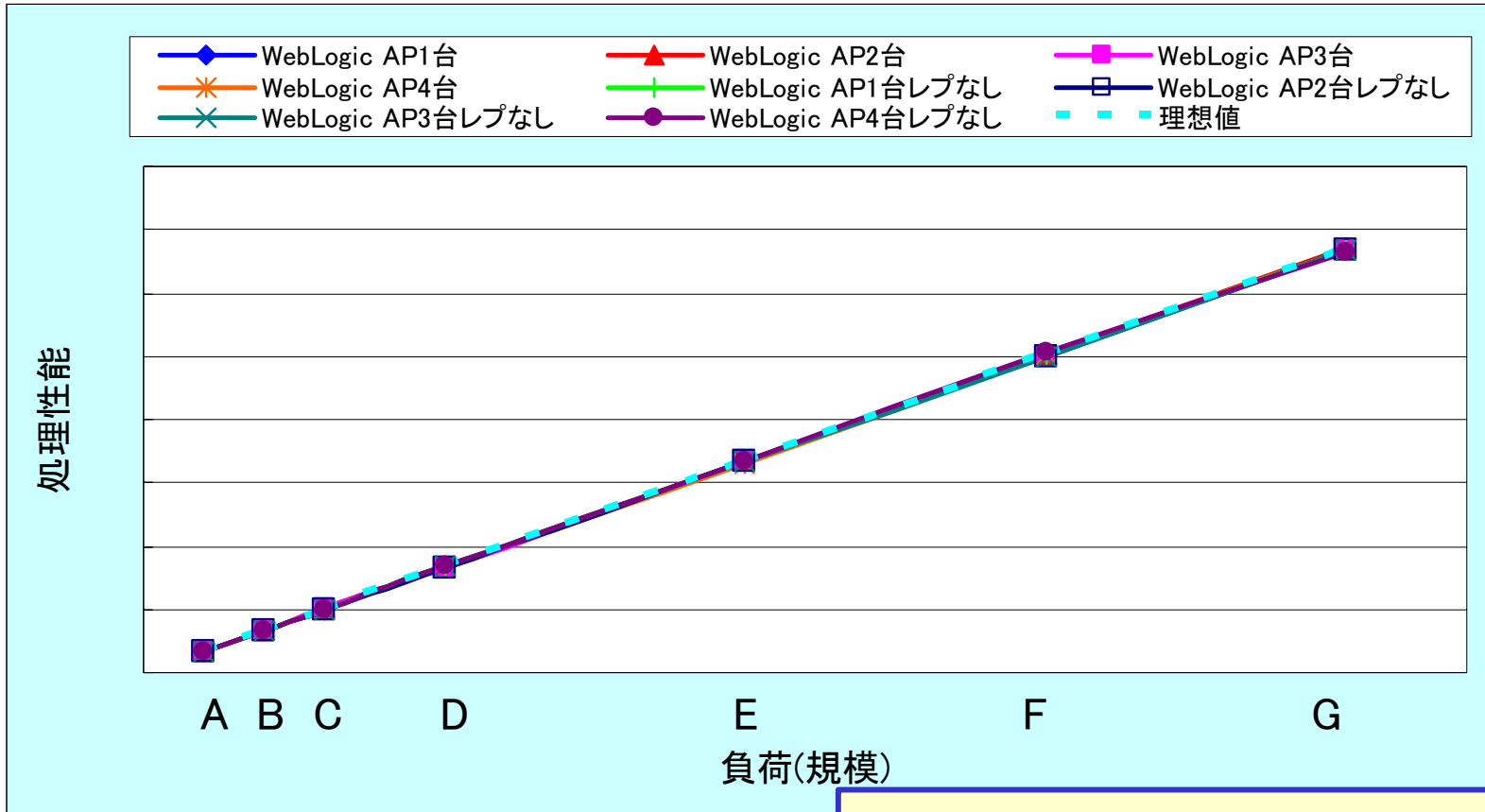
## 比較B: WebLogicのHttpSessionレプリケーション有り/無し比較

測定	APサーバ	HttpSession レプリケーション	比較A	比較B	比較C	比較D
1	JBoss	有り	○			○
2	JBoss	無し	○		○	
3	WebLogic	有り		○		○
4	WebLogic	無し		○	○	



## WebLogicクラスタ(HttpSessionレプリケーション有り/無し)

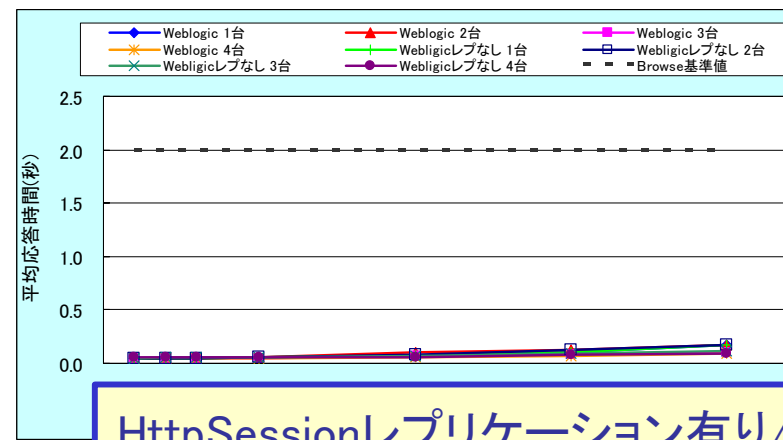
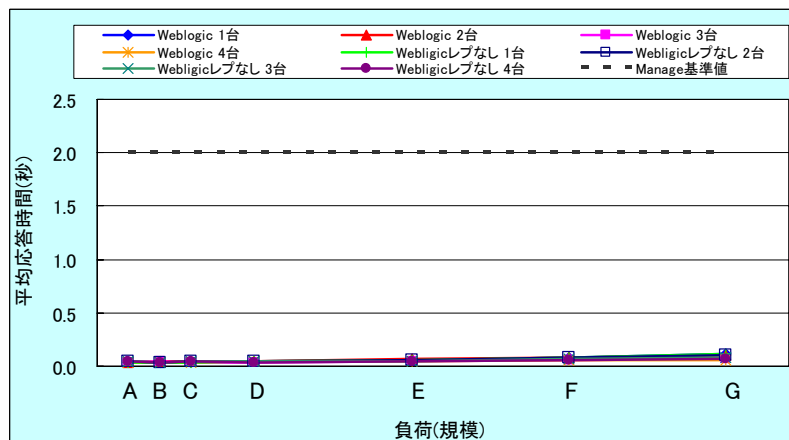
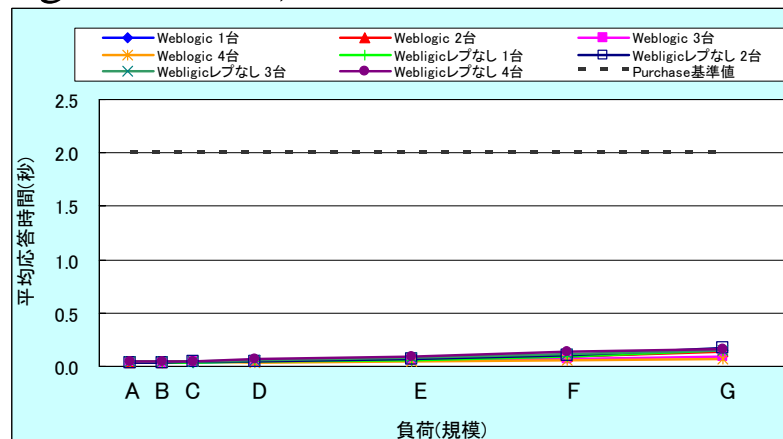
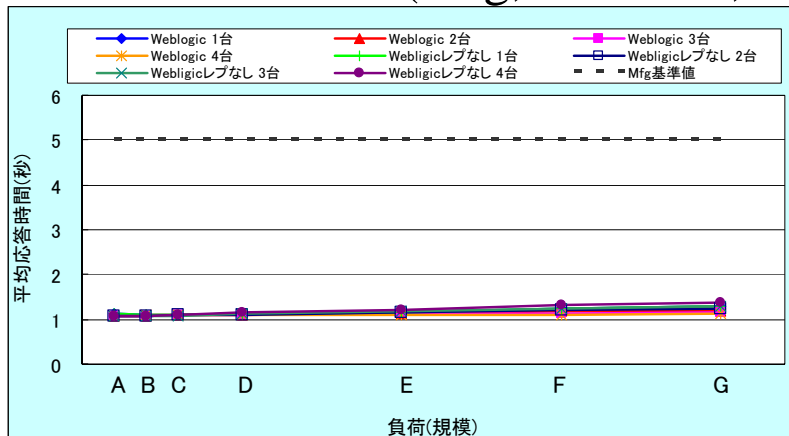
### - スループット



HttpSessionレプリケーション有り/無し  
に関わらず処理限界を迎えていない



## WebLogicクラスタ(HttpSessionレプリケーション有り/無し) - 応答時間(Mfg, Purchase, Manage, Browse)

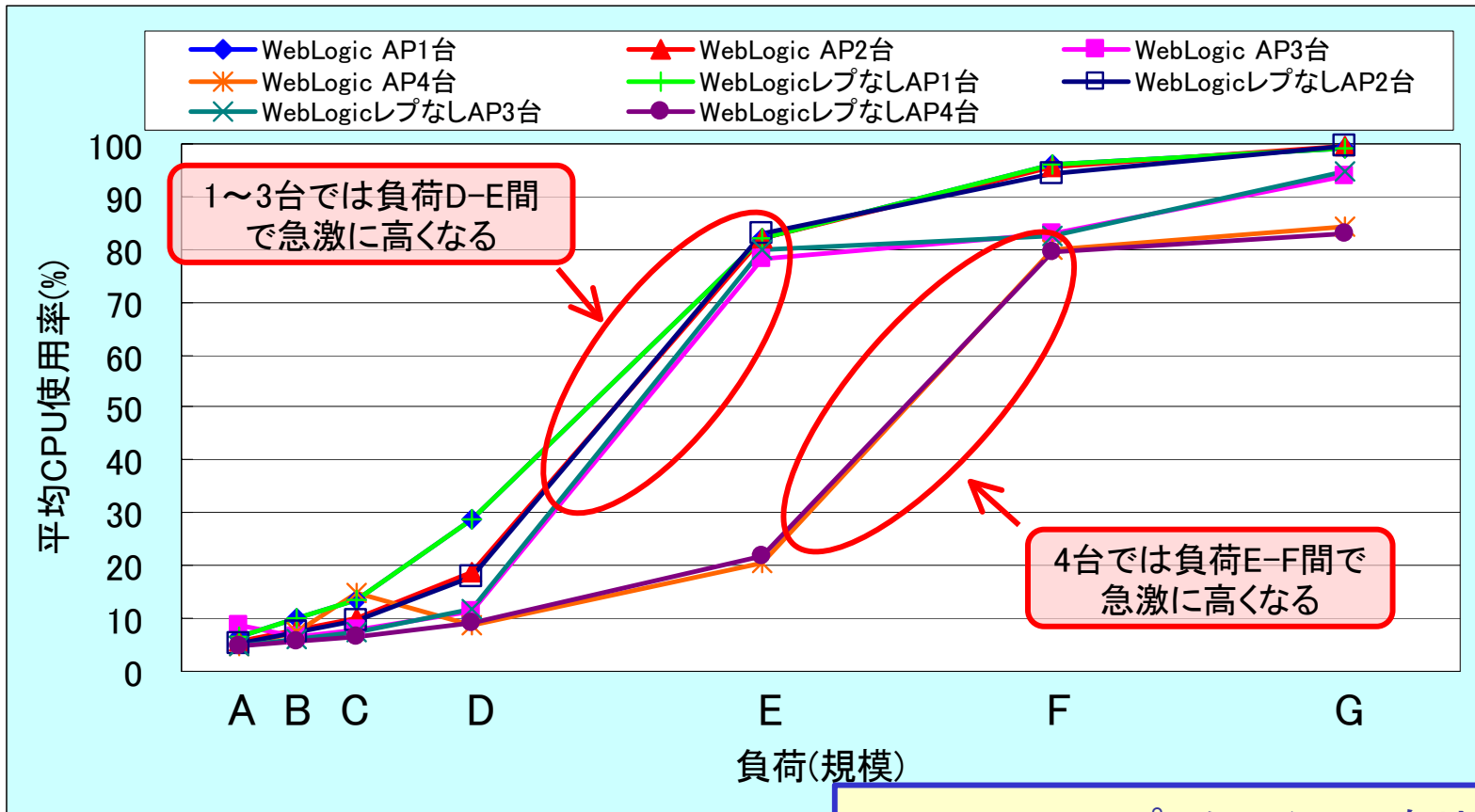


HttpSessionレプリケーション有り/無し  
に関わらず応答時間は劣化していない



## WebLogicクラスタ(HttpSessionレプリケーション有り/無し)

### - 平均CPU使用率

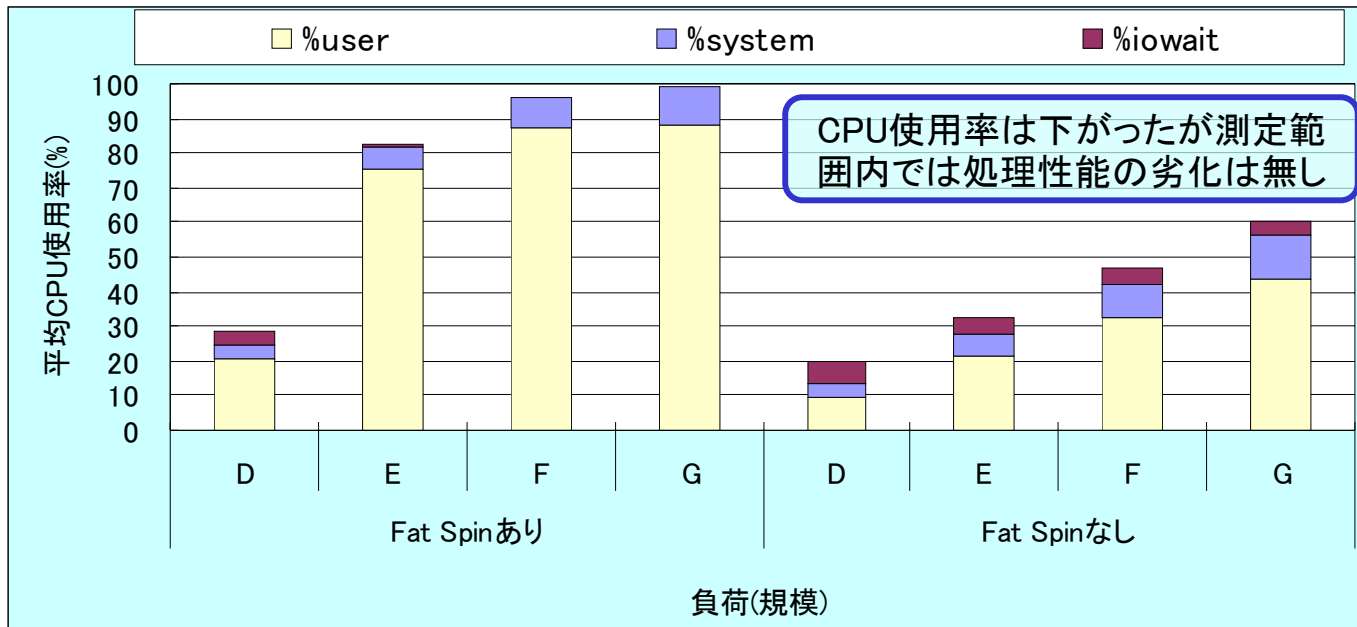


HttpSessionレプリケーション有り/無し  
に関わらず傾向は同じ



## Fat Spinについて

- JRocket1.4.2\_04(WebLogic8.1SP3付属)ではFat Spinを使用
- ロック獲得時、まずspinロックを行い待ち時間が長くなったらスリープさせる方式
- -XX+disablefatspinオプションにより無効化



Fat Spinを無効化することにより、CPU使用率が大幅に下がる



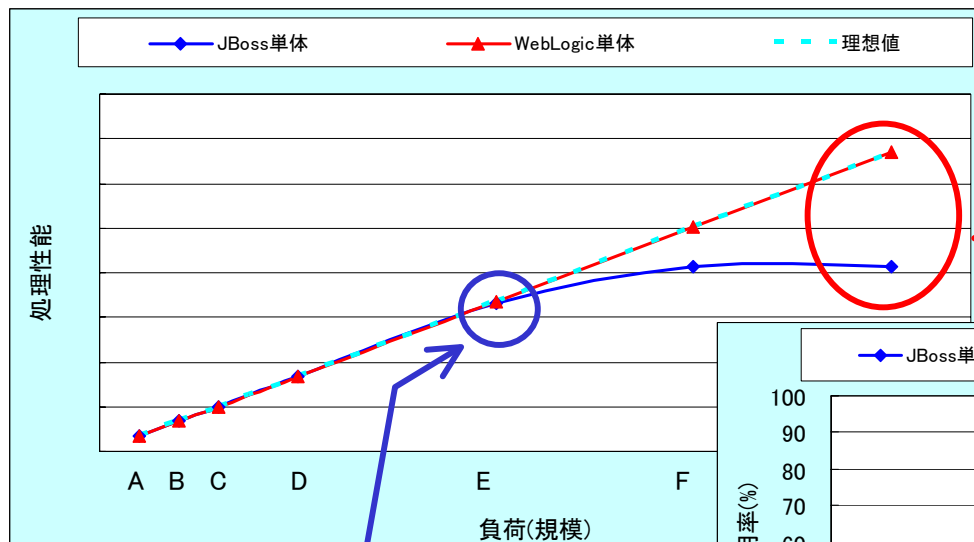
## 比較C: JBossとWebLogicの単体比較

測定	APサーバ	HttpSession レプリケーション	比較A	比較B	比較C	比較D
1	JBoss	有り	○			○
2	JBoss	無し	○		○	
3	WebLogic	有り		○		○
4	WebLogic	無し		○	○	



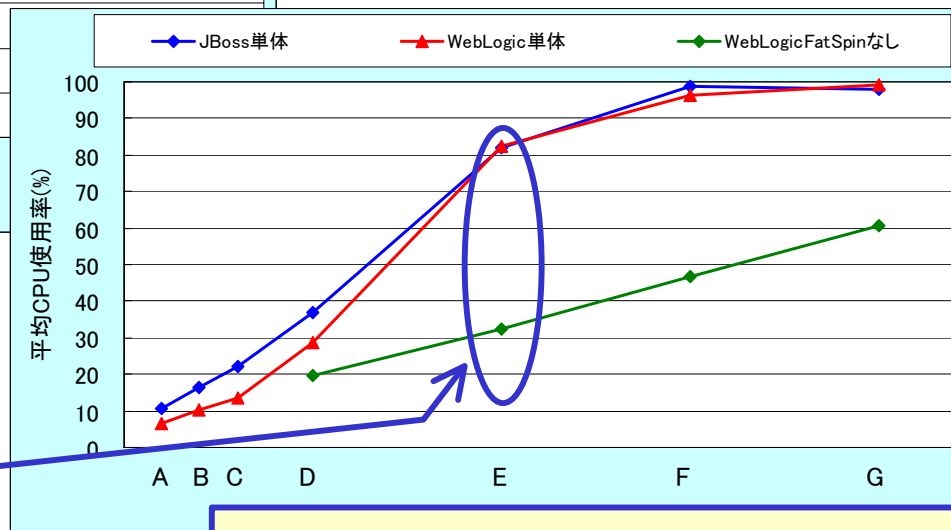
## JBoss/WebLogic単体比較

### - スループットとCPU使用率



WebLogicが処理限界を迎えていないので、処理能力の差が測れない

JBoss処理限界の負荷Eで、WebLogicのCPU使用率が33%



単体での性能差を確認



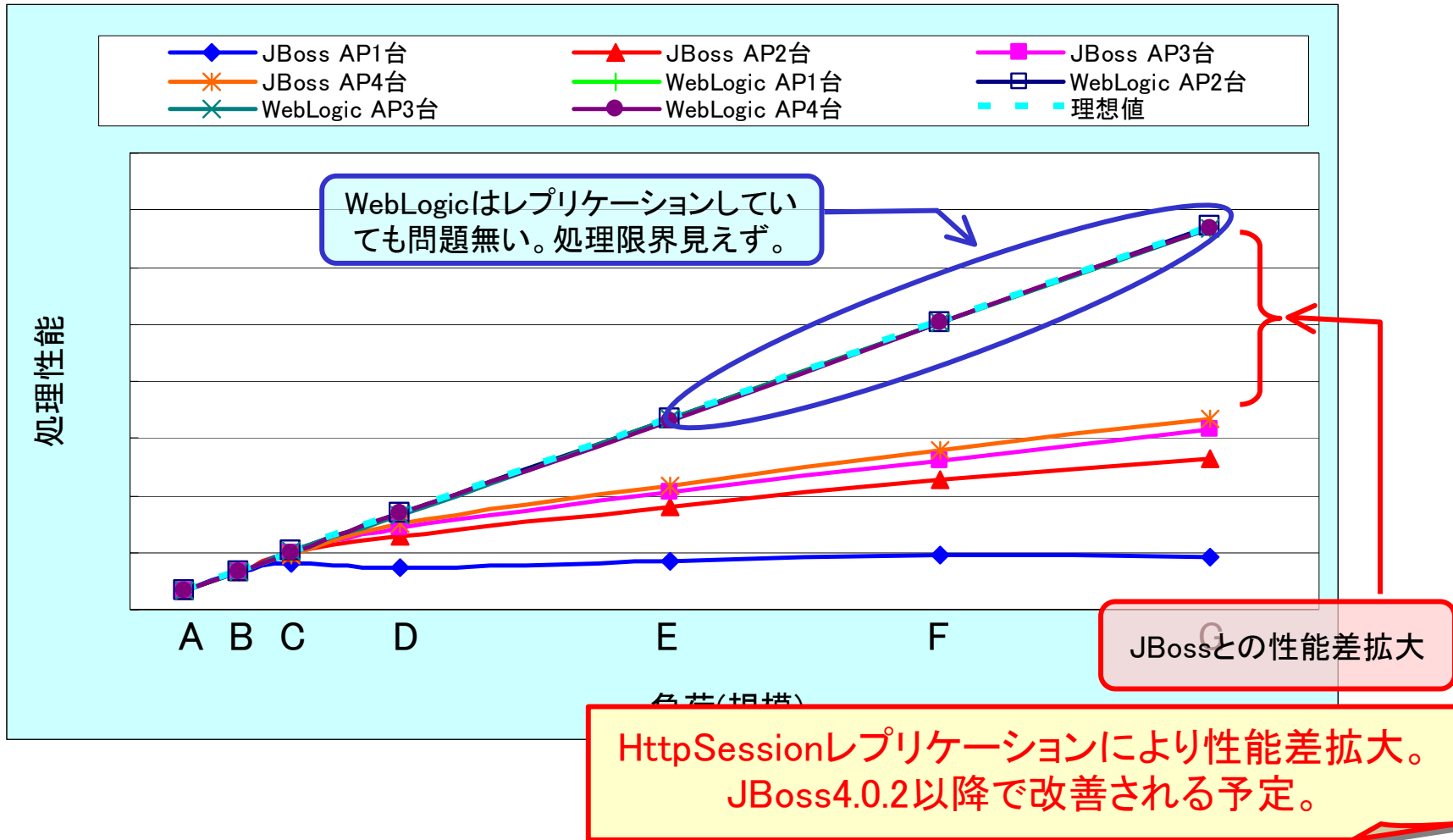


## 比較D: JBossとWebLogicのクラスタ比較

測定	APサーバ	HttpSession レプリケーション	比較A	比較B	比較C	比較D
1	JBoss	有り	○			○
2	JBoss	無し	○		○	
3	WebLogic	有り		○		○
4	WebLogic	無し		○	○	



## JBoss/WebLogicクラスタ比較





# 考察



## 単体比較結果について

- **SPECjAppServer2004**ではJBossとWebLogicの性能差を確認

## クラスタ比較結果について

- JBossクラスタはスケールアウトしないため、**比較にならず**

## JBossクラスタ(HttpSessionレプリケーション)が遅い理由

- 設定するだけで遅い→仕組み上、遅い
  - レプリケーションの粒度が大きい
    - JBoss4.0.0では**HttpSession全体**でしかレプリケーションできない
    - **変更のあった属性のみレプリケーションする設定(ATTRIBUTE)が動作しない(バグ)**
  - CPUを使い切れていない→レプリケーション時に同期ポイントが存在？
- ➡ HttpSessionレプリケーションについては、現時点では使わない方が無難

## JBossの適用領域は？

- 高い性能が求められないシステム
- 信頼性については、APの作りこみでカバーできるシステム



# TIPS紹介



- 検証で得られたノウハウをドキュメントに付録として提供
- 本セッションでは3つ紹介

## SPECjAppServer2004ノウハウ

### 設定上の注意点

- Driver側の設定(ヒープメモリサイズ設定、EJBのロードバランシング)
- アプリケーション又はアプリケーションサーバの設定(SequenceEntのrow-locking、XA処理の必要性、例外対応)
- 動作させる上での注意点(REPEATABLE\_READの解釈)

### 測定上の注意点

- レポートのFAILEDについて
- 「Error vehicle already sold」について
- Response Timeの90% Tileが10秒になる
- Dealer.detailの情報が欠落する

## WebLogicノウハウ

### 設定上の注意点

- 実行スレッドキューの分割

## PostgreSQLノウハウ

### チューニングノウハウ

- データ投入直後のreindex、analyze、vacuum full
- **index付与によるチューニング** ③

## JBossノウハウ

### 設定上の注意点

- JBoss4.0.0バグ情報(PreparedStatementCache、JvmRouteFilter)
- 設定(レプリケーションのタイムアウト、UDPエラー対応)

### チューニングノウハウ

- ear-deployer.xml(IsolatedとCallByValueの設定)
- JBossの不要サービス停止
- Apache mod\_jk2(**Apache MPMをworkerに**) ①

### HttpSessionレプリケーションについて

- ノウハウ(仕組み、バージョン、範囲) ②
- バグ情報(**ATTRIBUTE**)

### Entity Beanについて

- JBoss4.0.0の機能(分散ロック、キャッシュ)
- SPECjAppServer2004向けの設定(CMP設定)

### 測定上の注意点

- 高負荷時の例外(ClassCircularityError、InstanceAlreadyExistException)
- HttpSessionレプリケーション時のConcurrentModificationException
- 60秒おきのFullGC

## その他

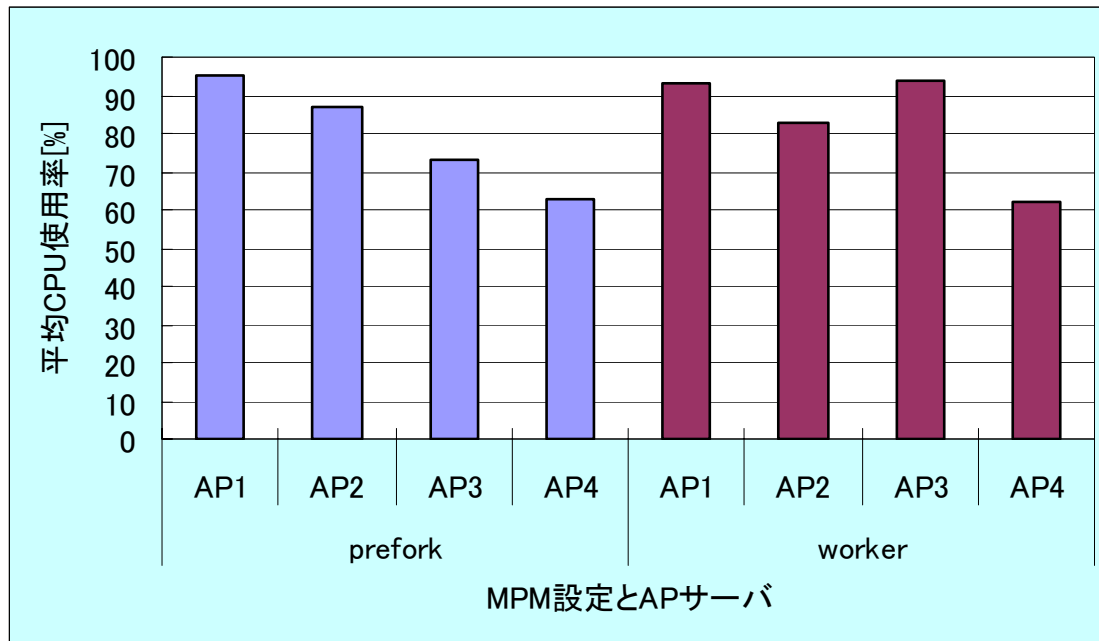
### 測定上の注意点

- sarのバージョンによる情報の欠落



## Apache MPMをworkerに

- Apache MPMがpreforkの場合、mod\_jk2による振り分けが偏る
  - 常にAPサーバ1→2→3→4の順でCPU使用率が高い状態
- workerに変更することにより、バラつきはあるものの偏りは改善



- mod\_jk2ではどのAPサーバに割り振るかをプロセス単位で管理
- プロセスが多数起動するpreforkでは、管理がバラバラになり偏りが生じる



## HttpSessionレプリケーションのreplication-granularity

- JBoss4.0.0では、バグによりATTRIBUTEの設定が動作しない
  - 高負荷時にJBossCache(TreeCache)で二重コミットが発生する問題
  - 不正なメソッド呼び出しの問題
  - トランザクションがPREPAREフェーズで既にアクティブな問題

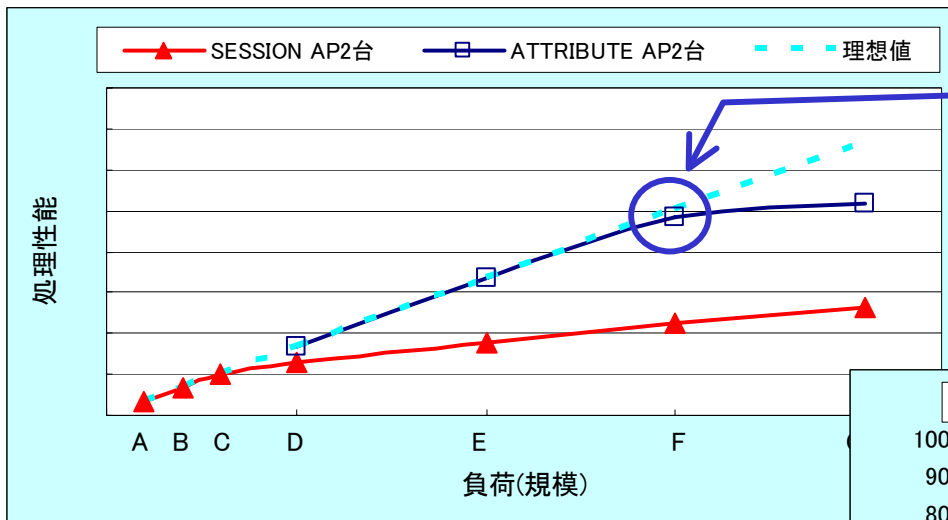
- 上記バグについてJBoss Forumやjiraにおいて開発者にバグを報告
  - 4.0.2では修正される予定

- CVSの最新版を元に性能がどの程度変わるかを確認
  - 動作させてみたところ、同期部分で例外が発生
  - 若干修正を入れ、AP2台で測定(結果は次ページ)



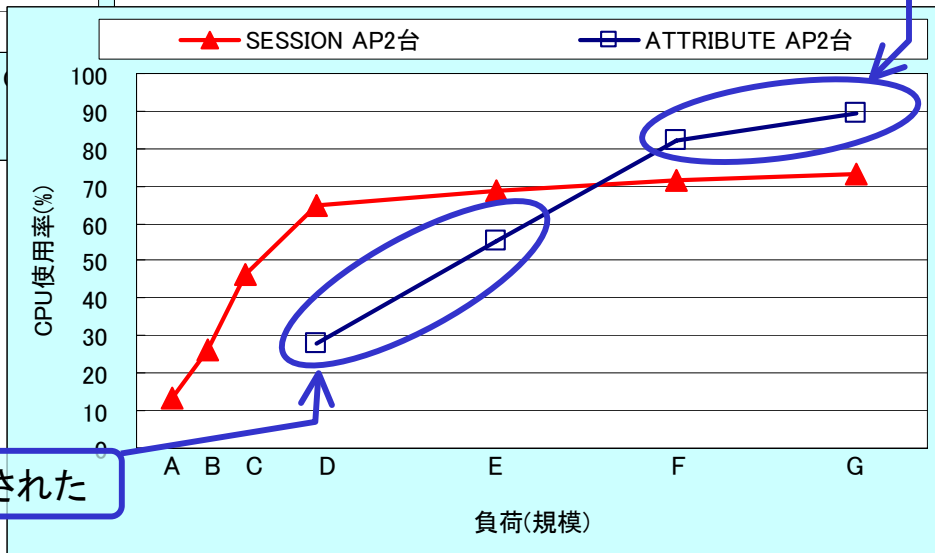


# SESSIONとATTRIBUTEの比較



処理能力向上!  
 ・負荷Fまで性能が出るようになった  
 ・JBoss1台よりも性能が出る

高負荷時、使いきれなくなった!



低負荷時、CPU使用率が改善された

かなりまとまな状態になっている。期待!



## index付与によるチューニング

- SPECjAppServer2004では、DBスキーマに対し十分なチューニングがされておらず、DBサーバがボトルネックに
- indexを付与する事により(付け忘れ?)、DBサーバの性能が大幅に改善され、DBサーバボトルネックが解消された

```
ossf=# explain analyze SELECT ci_id FROM c_customerinventory WHERE (ci_customerid=47862);
```

QUERY PLAN

```
Seq Scan on c_customerinventory (cost=0.00..25646.80 rows=7 width=4) (actual time=198.758..613.254 rows=5 loops=1)
  Filter: (ci_customerid = 47862)
  Total runtime: 613.313 ms
(3 rows)
```

遅い!

シーケンシャルスキャンしている

```
CREATE INDEX C_ci_customerid_idx ON
C_customerinventory (ci_customerid);
```

```
ossf=# \d c_customerinventory
Table "public.c_customerinventory"
Column      | Type          | Modifiers
-----
ci_id       | integer      | not null
ci_itemid   | character(20)
ci_quantity | integer
ci_customerid | integer
ci_value    | numeric(12,2)
Indexes:
  "c_ci_idx" unique, btree (ci_id)
```

インデックスが無い!

```
ossf=# explain analyze SELECT ci_id FROM c_customerinventory WHERE (ci_customerid=47862);
```

QUERY PLAN

```
Index Scan using c_ci_customerid_idx on c_customerinventory (cost=0.00..3.08 rows=7 width=4) (actual time=0.021..0.029 rows=5 loops=1)
  Index Cond: (ci_customerid = 47862)
  Total runtime: 0.065 ms
(3 rows)
```

早い!

インデックスを使用している



# まとめ



## 今年度のまとめ

- SPECjAppServer2004による評価手順を確立した
  - 評価手順並びに設定ファイル群を一般公開
    - <http://www.ipa.go.jp/software/open/forum/>  
(日本OSS推進フォーラム)
- 商用APサーバと比較し、適用領域を明確化した

## 今後の課題

- JBossの継続的な評価の実施
  - JBoss 4.0.1, 4.0.2, ... で性能・信頼性の向上を期待
- より容易なベンチマーク測定を可能にしたい
  - SPECjAppServer2004は環境構築が大変(合計78ページ)
  - SPECjAppServer2004は結果の公開に制限



ご清聴ありがとうございました。