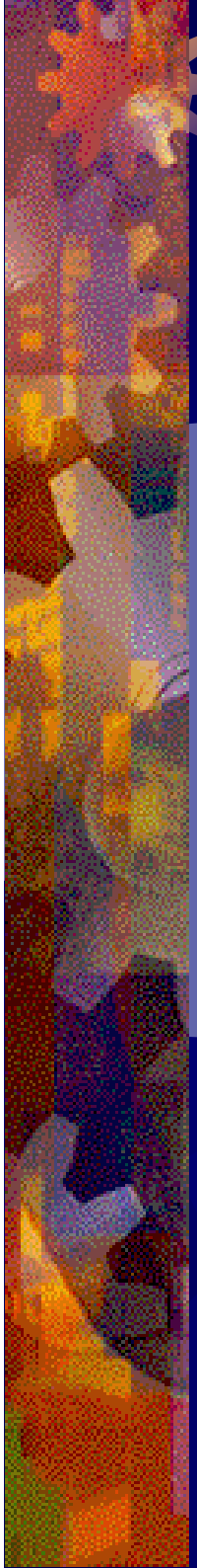


PostgreSQL 9.0
ストリーミングレプリケーションの実力

2011年2月25日

株式会社日立製作所 ソフトウェア事業部
福岡博
NECソフト株式会社 PFシステム事業部
岩浅晃郎

自己紹介

NEC

ミドルウェア領域のオープンソースソフトウェアを対象にしたサポート・サービスを体系化して提供
「OSSSミドルウェアサポートサービス」
http://www.nec.co.jp/oss/middle_support/

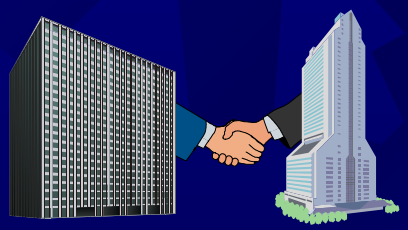
日立

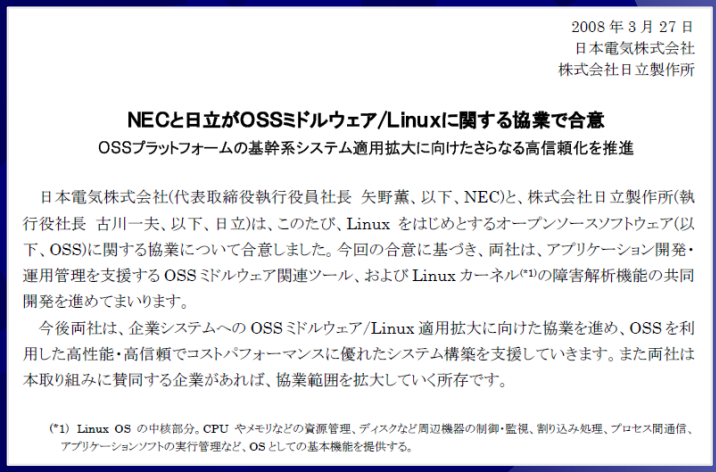
PostgreSQLなどOSSミドルウェアを対象としたサービスを提供
「日立サポート360」
「かんたんOSSサポートサービス」
http://www.hitachi.co.jp/Prod/comp/linux/service/solution/oss/

2008年3月NECと日立でOSSミドルウェア/Linux協業合意

OSS　DBMS分野の活動

- 「JP1」「DB Monitor」連携
- メンテナンスツール検証





2008 年 3 月 27 日
日本電気株式会社
株式会社日立製作所

NECと日立がOSSミドルウェア/Linuxに関する協業で合意
OSSプラットフォームの基幹系システム適用拡大に向けたさらなる高信頼化を推進

日本電気株式会社(代表取締役執行役員社長　矢野薫、以下、NEC)と、株式会社日立製作所(執行役社長　古川一夫、以下、日立)は、このたび、Linux をはじめとするオープンソースソフトウェア(以下、OSS)に関する協業について合意しました。今回の合意に基づき、両社は、アプリケーション開発・運用管理を支援する OSS ミドルウェア関連ツール、および Linux カーネル(*1)の障害解析機能の共同開発を進めてまいります。

今後両社は、企業システムへの OSS ミドルウェア/Linux 適用拡大に向けた協業を進め、OSS を利用した高性能・高信頼でコストパフォーマンスに優れたシステム構築を支援していきます。また両社は本取り組みに賛同する企業があれば、協業範囲を拡大していく所存です。

(*1) Linux OS の中核部分。CPU やメモリなどの資源管理、ディスクなど周辺機器の制御・監視、割り込み処理、プロセス間通信、アプリケーションソフトの実行管理など、OS としての基本機能を提供する。

http://www.hitachi.co.jp/New/cnews/month/2008/03/0327.pdf
http://www.nec.co.jp/press/ja/0803/2701.html

2010年からOSSクラウド基盤の可能性に向けたOSS情報交換会およびOSS DBMSの検証を実施中

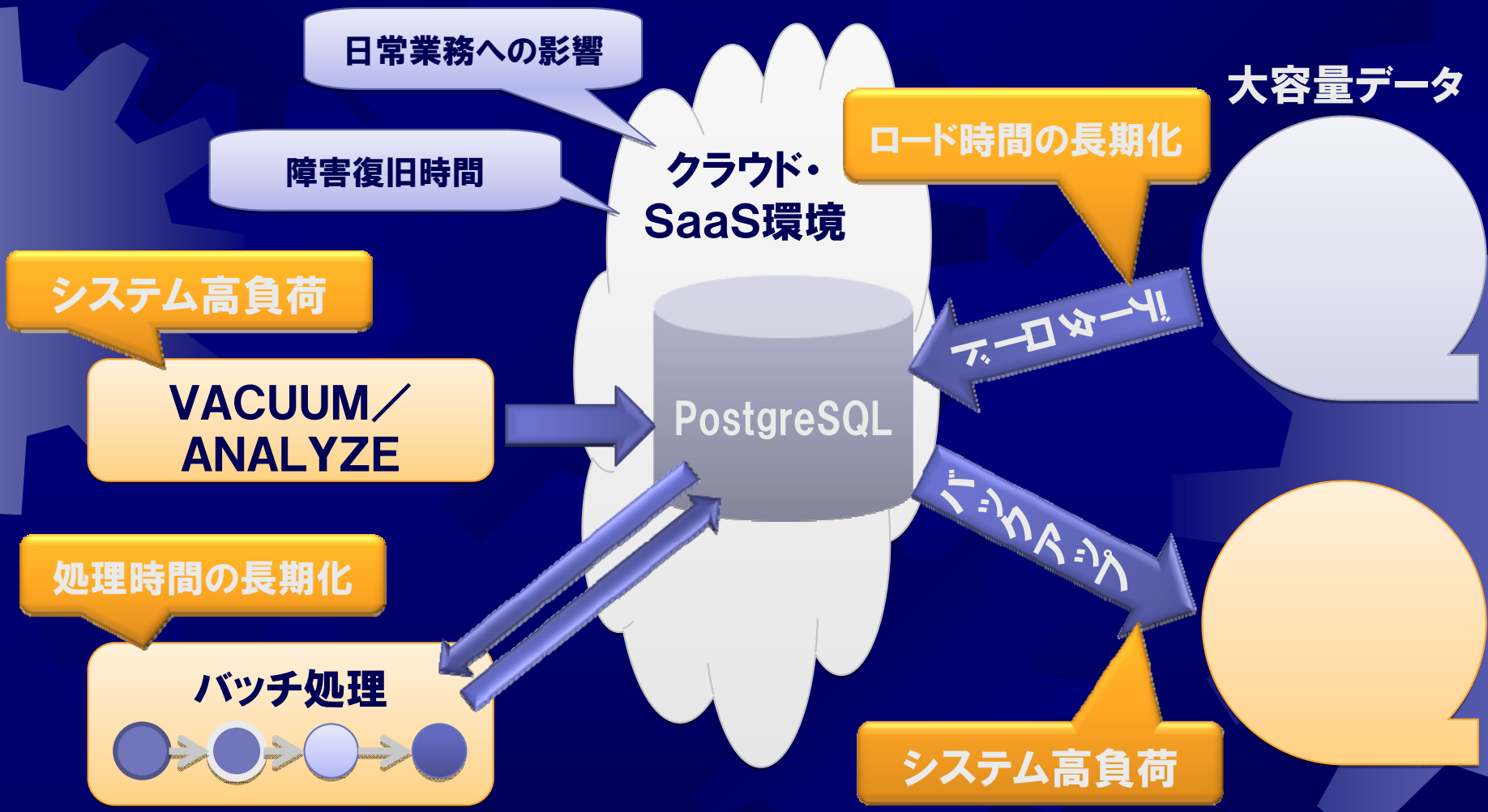
クラウド・SaaS領域へのOSS DBMS適用における課題

- データ量、リクエスト数の予測が困難
- 仮想サーバ追加による負荷分散への対応
- スケールアップによる性能改善の限界

スケールアウト構成による課題の解決

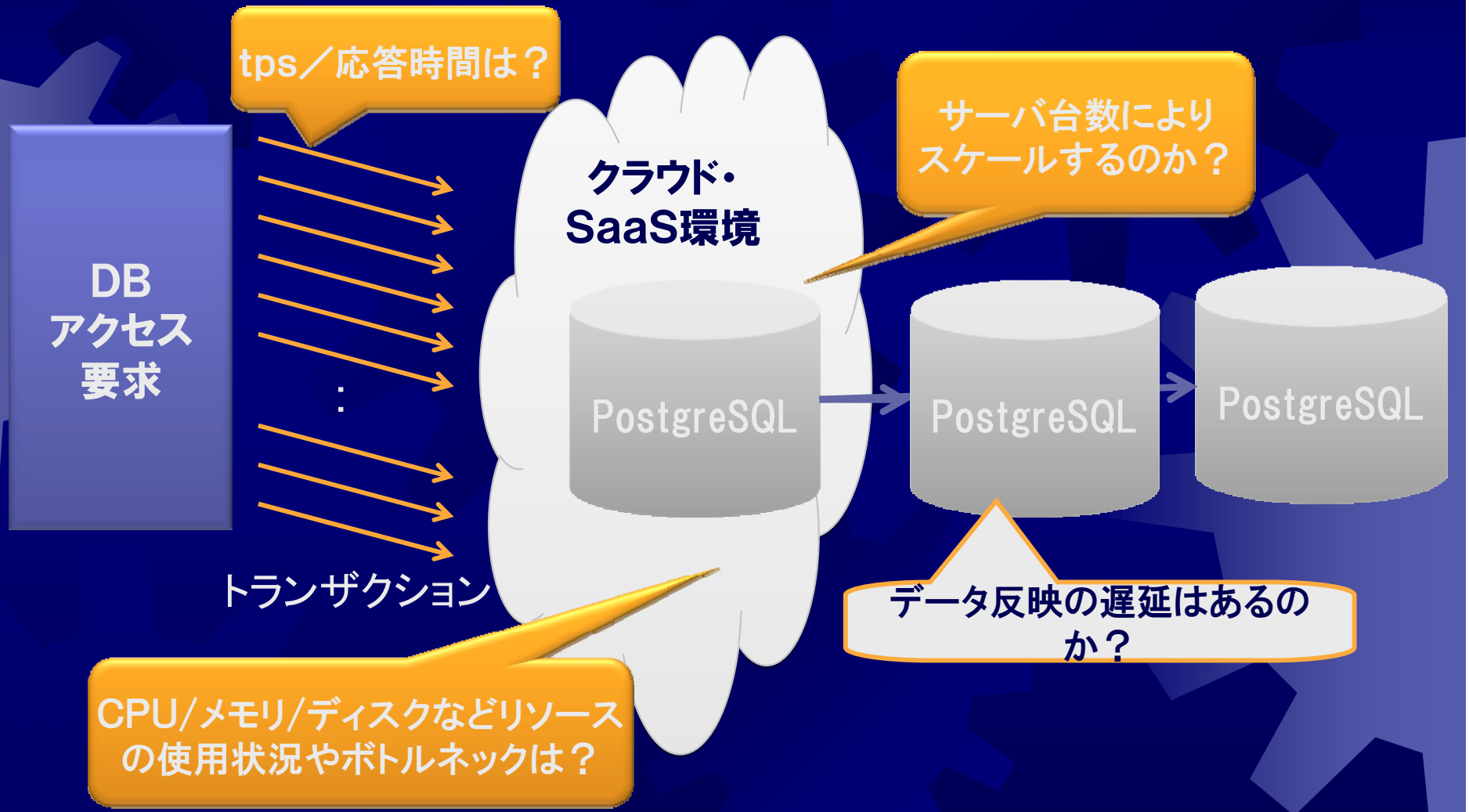
大容量データ処理

- 大量データのロード／バックアップ／バキューム
- バッチ処理



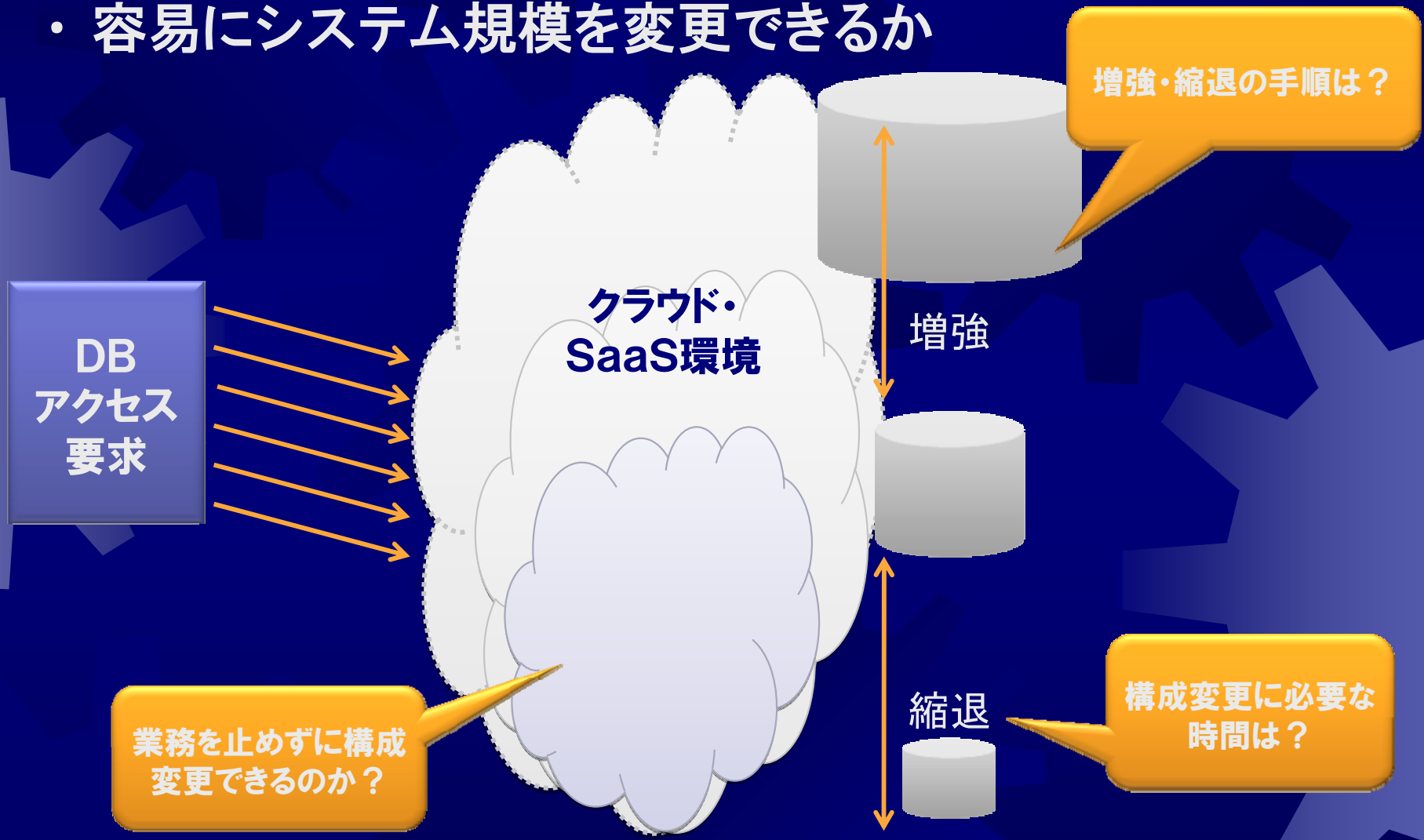
トランザクション数増加

- トランザクション数増加に対する許容範囲
- 負荷の予測サイジングが困難



柔軟な構成変更

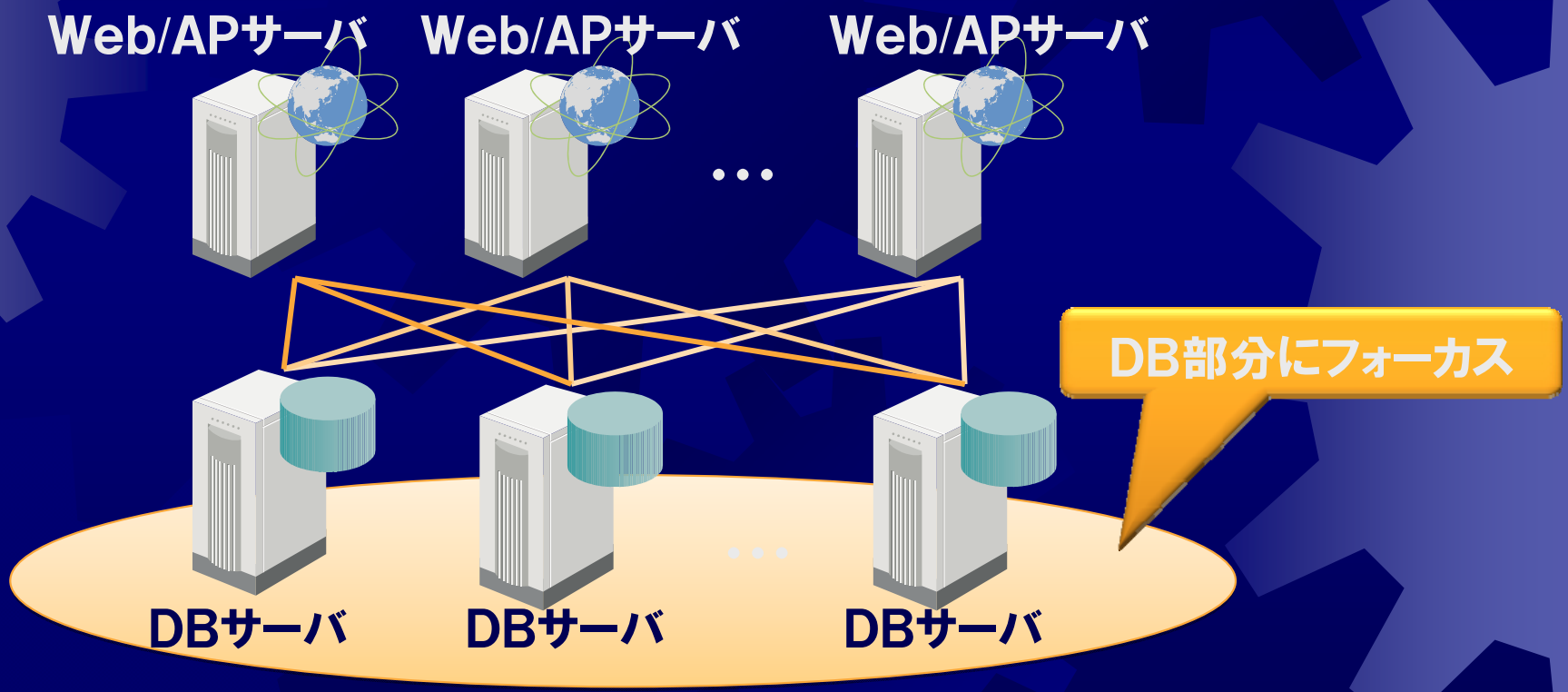
- 繁忙期には増強、閑散期には縮小
- 容易にシステム規模を変更できるか

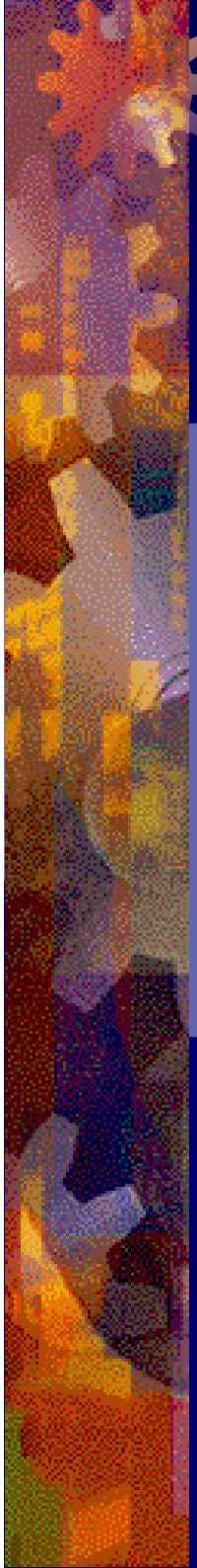


クラウド領域でのOSS DBMS適用分野

- プライベートクラウド：ITリソースの有効活用、統合運用管理
- パブリッククラウド：インターネット経由のBtoB/C、不特定多数

PostgreSQLは現実解となり得るのか？



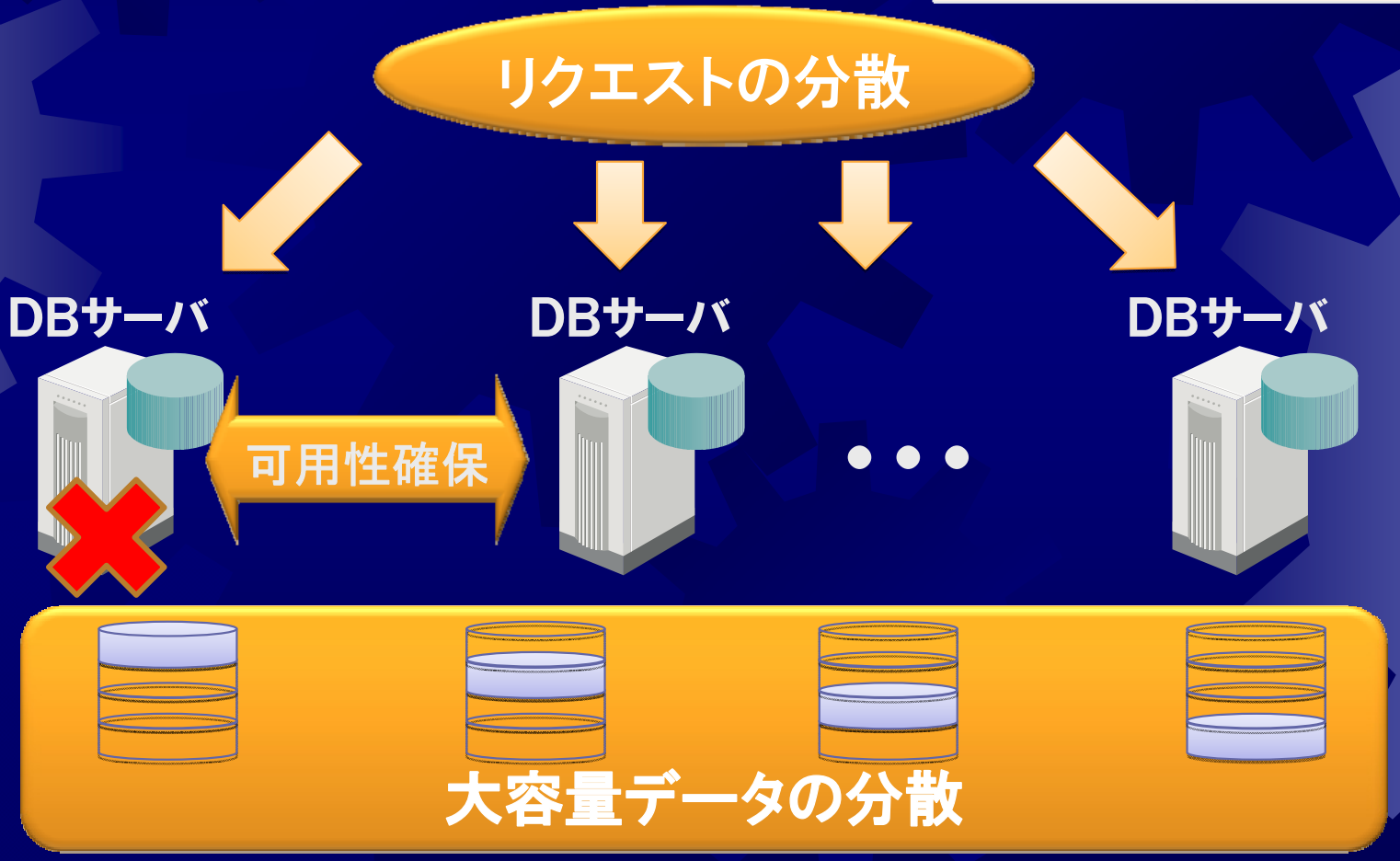


PostgreSQLのスケールアウト（クラスタ）

DBMSクラスタリングによる性能・可用性確保

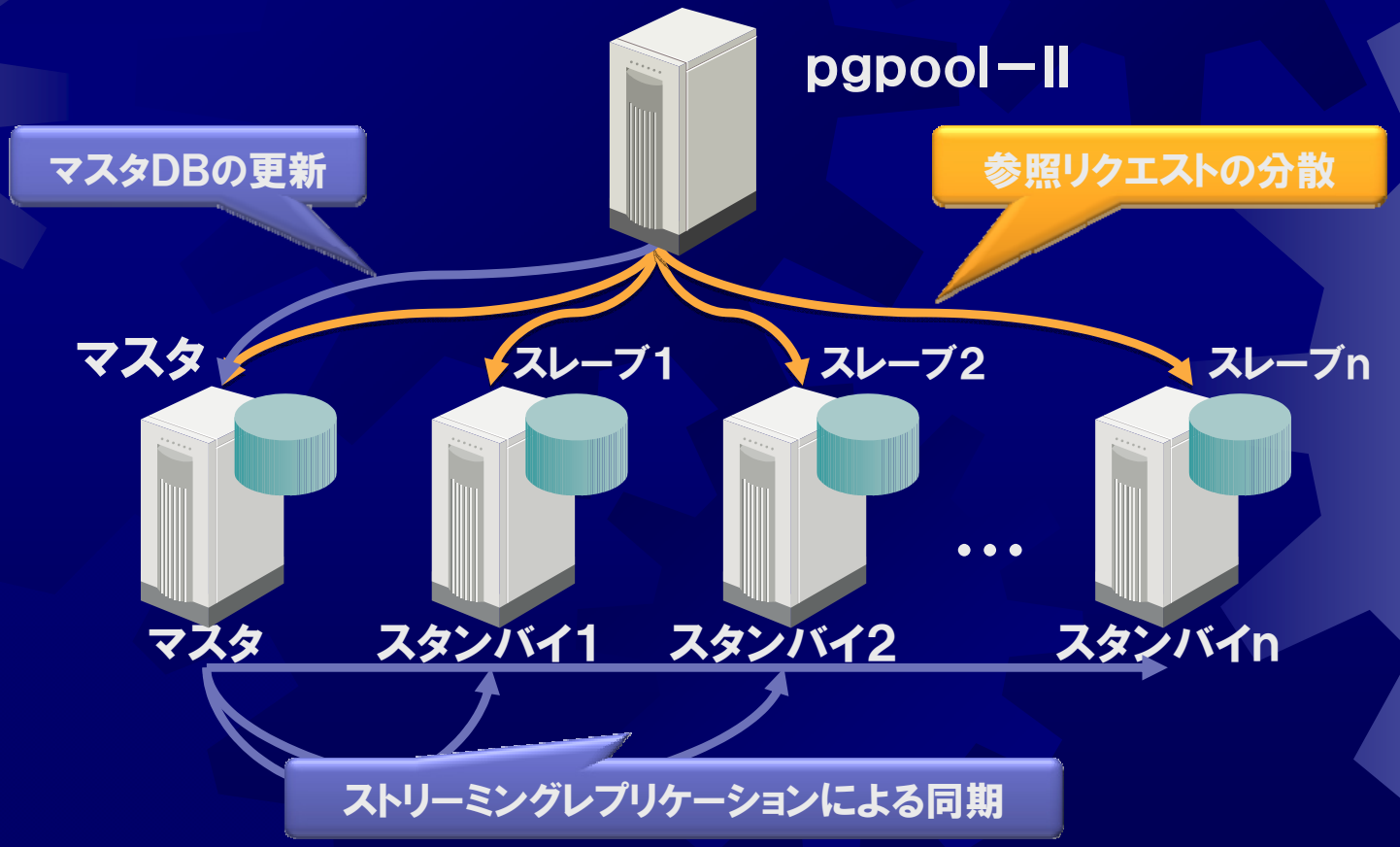
- クラスタリングの目的
  - 可用性確保による業務の継続性
  - リクエストの分散
  - 大容量データの分散

クラスタ構成例	
ソフトウェア	pgpool－II
ハードウェア	ロードバランサ
Webサーバ	リバースプロキシ
:	:



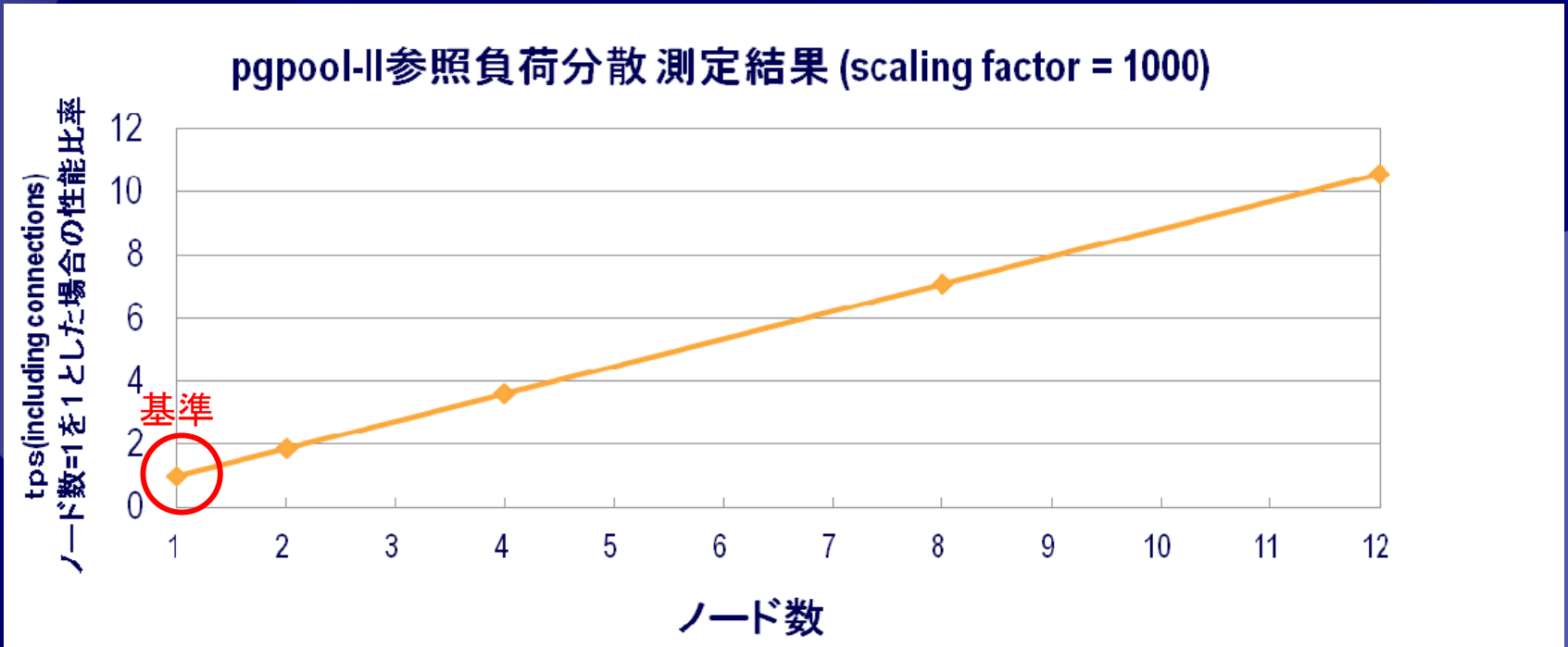
スケールアウト基盤としての ストリーミングレプリケーション

- PostgreSQL9.0で登場したストリーミングレプリケーションとpgpool-Ⅱ3.0の組合せ
  - ストリーミングレプリケーションによるマスタ／スレーブモードでのデータ同期
  - トランザクションレベルでは非同期 → 参照負荷分散

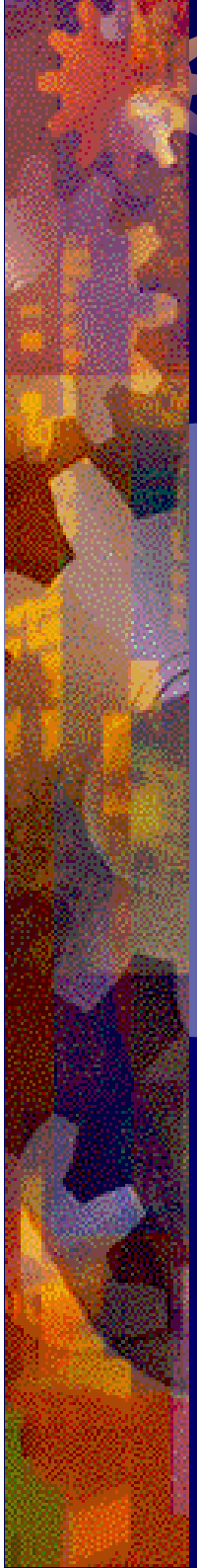


参照負荷分散でのスケールアウト例

- pgpool－Ⅱによる参照負荷分散
  - pgpool－Ⅱはマスタ／スレーブモード
  - DBノード間はストリーミングレプリケーションで同期



※DBノードで高負荷となるよう一部設定を変更し、pgbenchの”SELECT ONLY”で計測

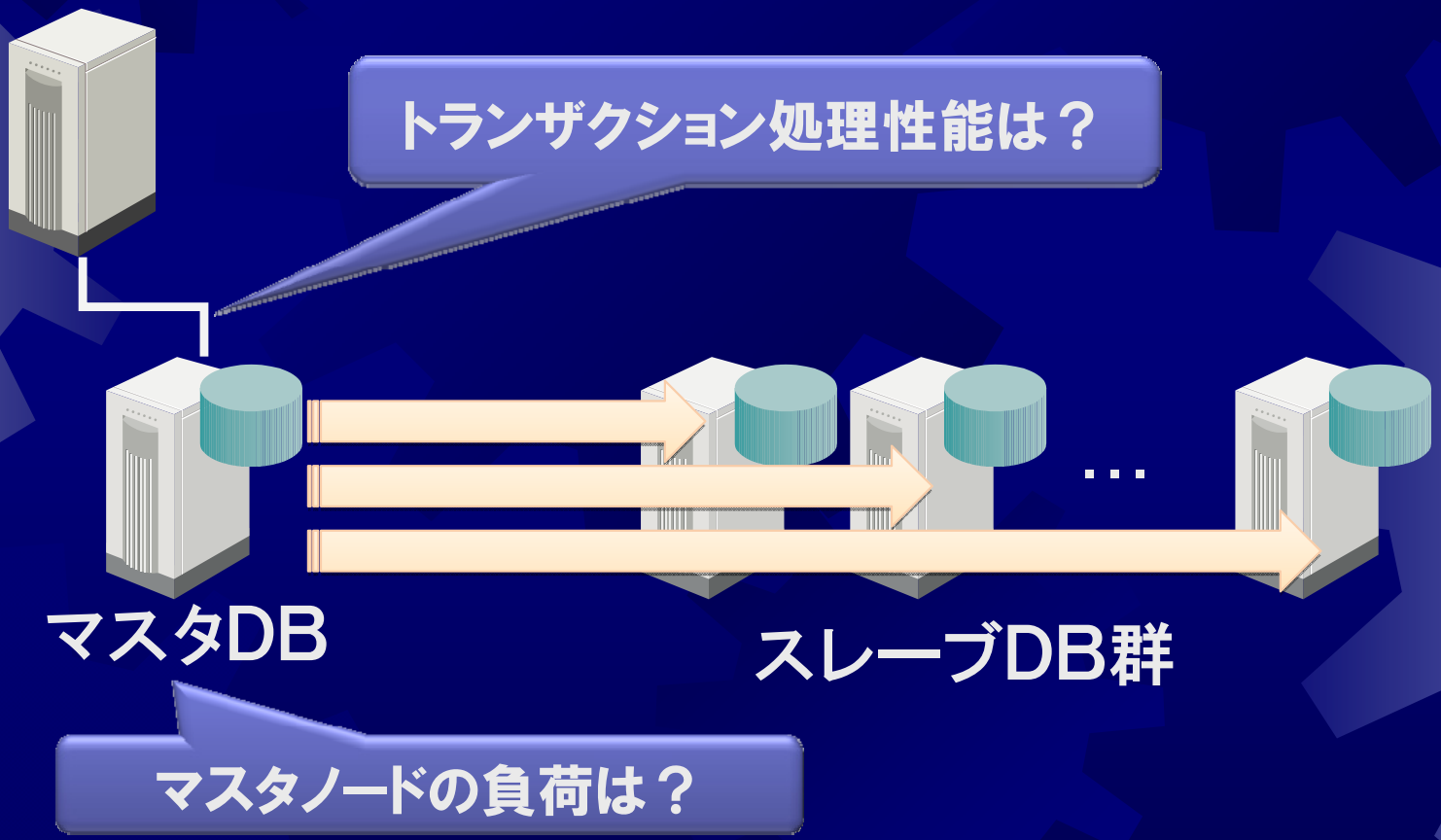


実機検証

実機検証①

スレーブDBノードは何台まで追加できるのか

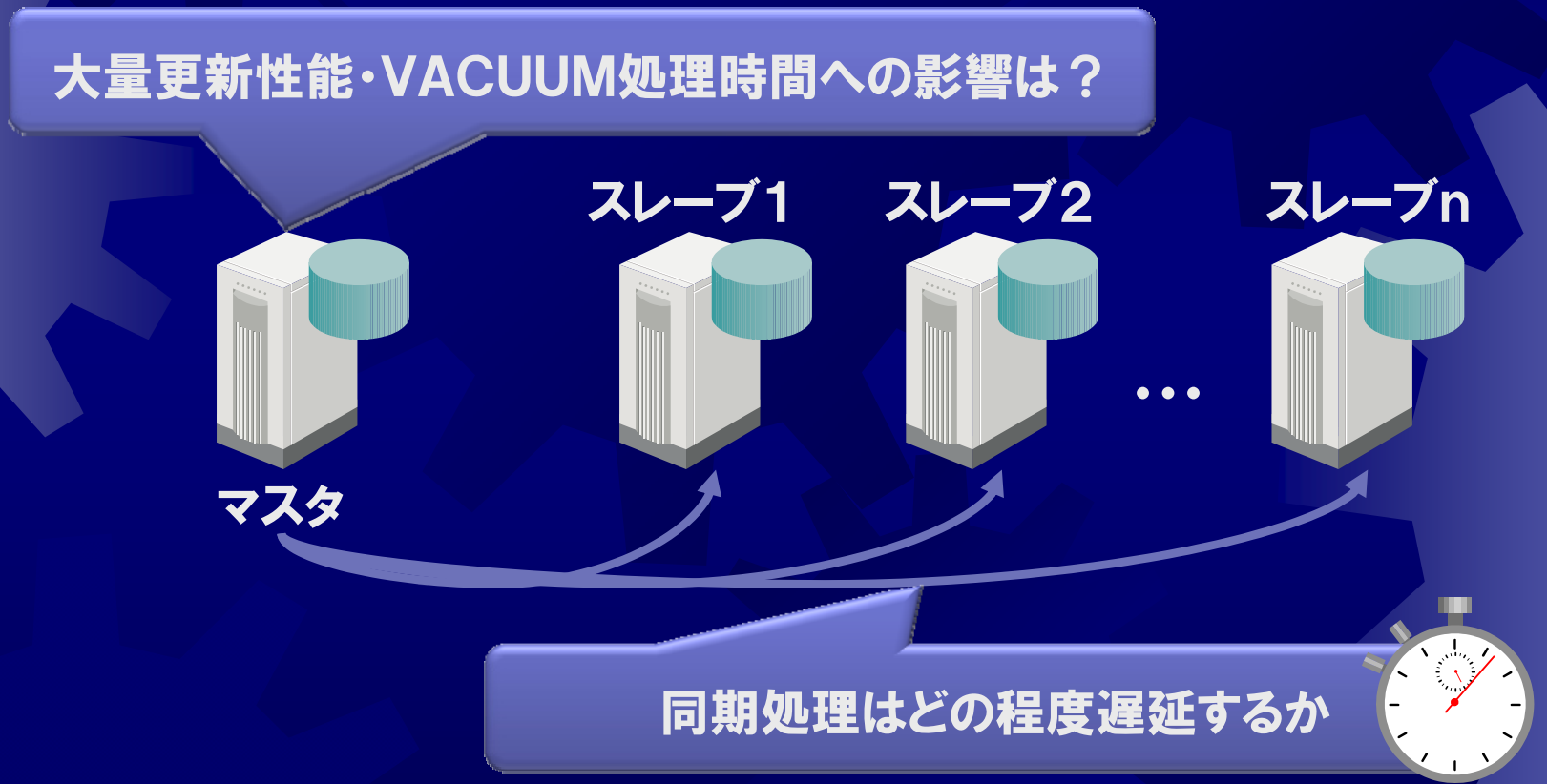
- マスタDBノードにおけるトランザクション処理性能への影響
- マスタDBノードの負荷



実機検証 ②

大量更新処理性能・運用への影響

- 夜間バッチ処理、VACUUMなど、大量更新処理性能への影響
- スレーブノードの同期遅延



検証環境

①サーバ：小型高集積モデルBS320 R5

CPU Intel Xeon X5670 2.93GHz (6コア) × 2
メモリ 48GB

#	用途
1	PostgreSQL9 マスタサーバ
2	PostgreSQL9 スタンバイサーバ
3	〃
4	〃
5	〃
6	〃
7	〃
8	〃
9	〃
10	〃
11	〃
12	〃

②サーバ：HA8000/TS20

CPU Intel Xeon X5670 2.93GHz (6コア) × 2
メモリ 16GB

#	用途
13	pgpool-Ⅱ

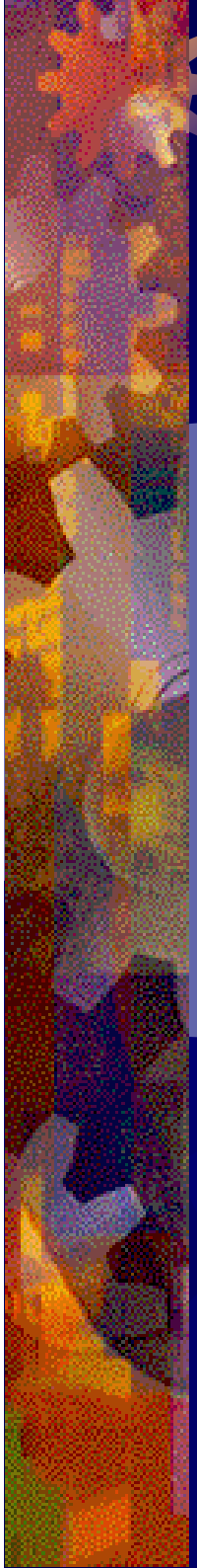
③ストレージ：Hitachi Adaptable Modular Storage 2010

SAS 15,000min-1, FC接続, 各ブレード毎

領域	容量
システム領域	20GB(RAID5)
データ領域	80GB(RAID5)
WAL領域	10GB(RAID5)

④ソフトウェア

#	用途
DBMS	PostgreSQL 9. 0. 2
クラスタ	pgpool-Ⅱ 3. 0. 1
OS	Red Hat Enterprise Linux 5.4 (AMD/Intel 64)

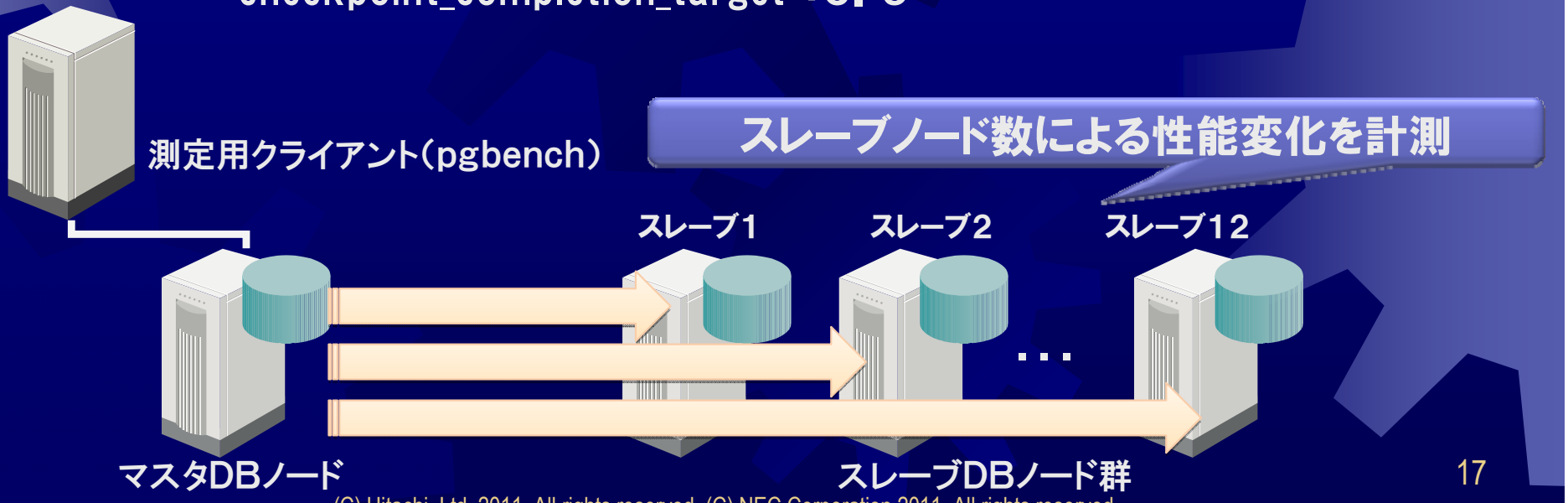


ベンチマークツールによる性能測定結果

検証結果

スレーブノード数別ベンチマーク

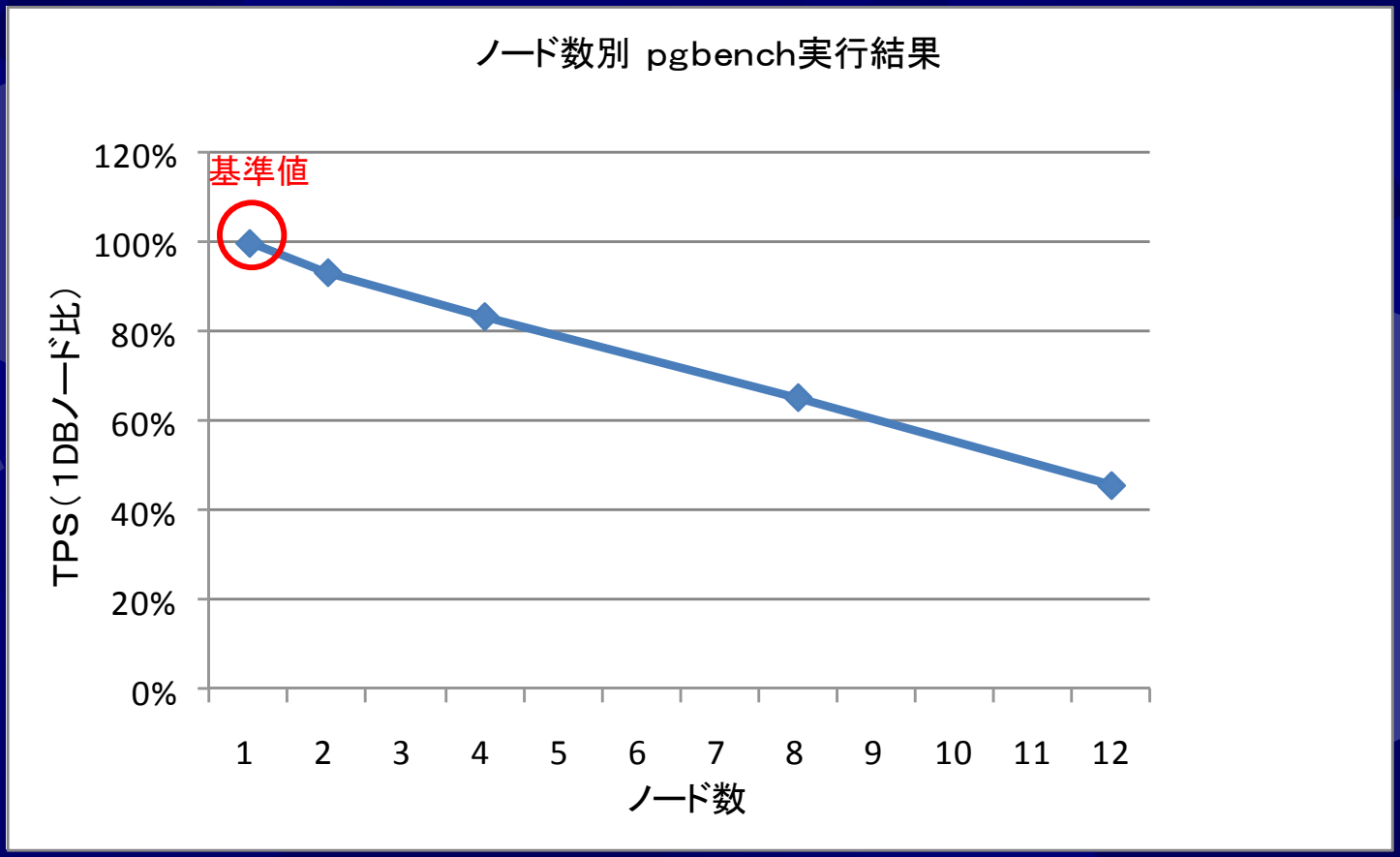
- pgbenchを使用した性能測定
  - DBノード数：1（単体）、2、4、8、12ノード
  - データ件数： 1,000,000件
  - 接続数： 100
  - PostgreSQLパラメータ設定
    - shared_buffers：480MB
    - wal_buffers：256kB
    - checkpoint_segments：50
    - checkpoint_completion_target ：0. 9



スレーブノード数別ベンチマーク

ノード数の増加に応じて少しずつ処理性能が低下

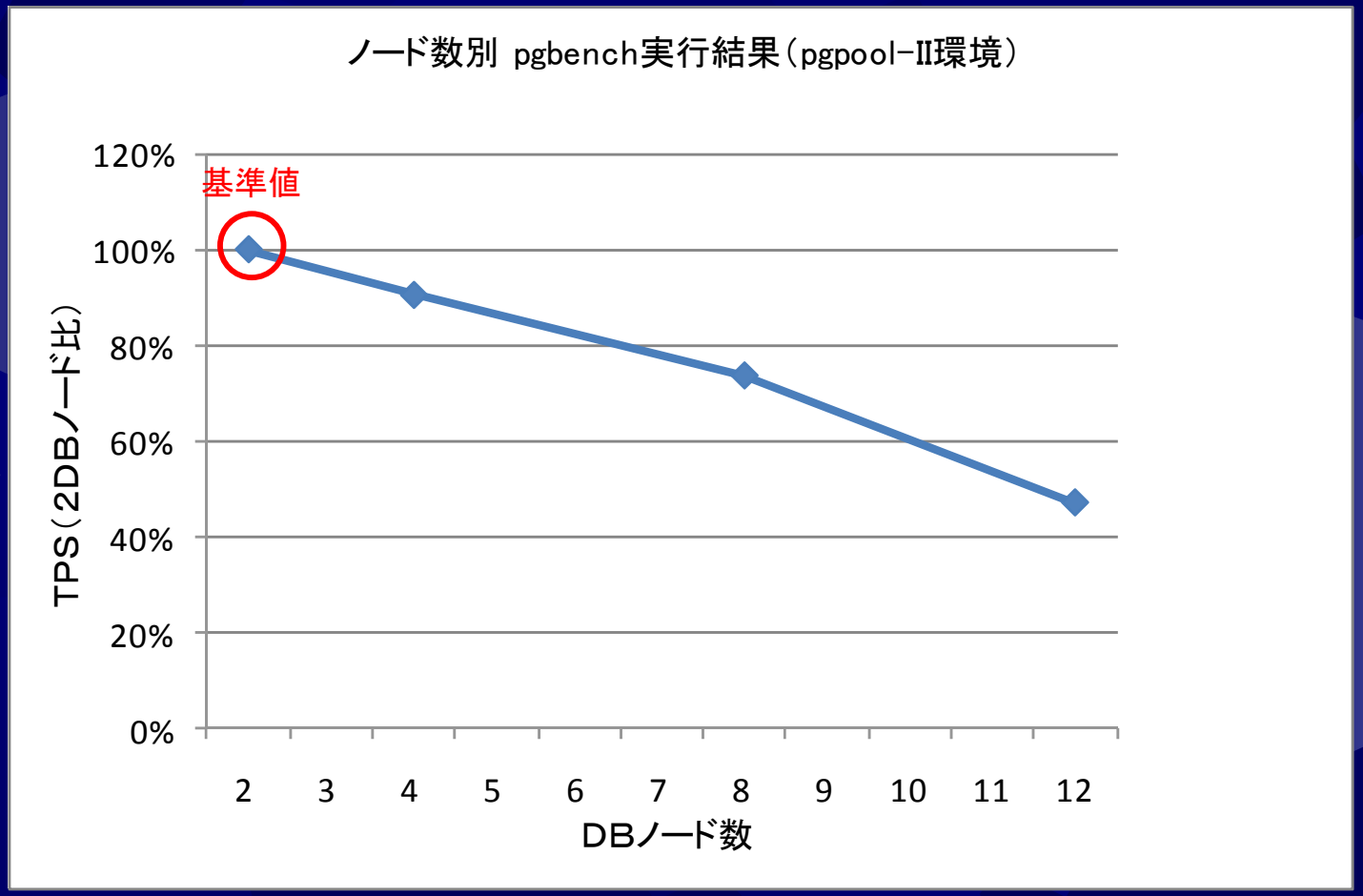
基準値（1DBノード、100クライアント）の処理性能（TPS）を基準とした比率



スレーブノード数別ベンチマーク

pgpool-II環境(Master/Slave)の場合も同様の傾向

基準値(2DBノード、100クライアント)の処理性能(TPS)を基準とした比率

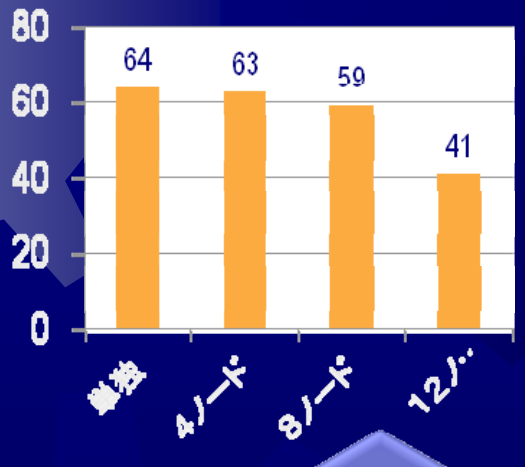


スレーブノード数別ベンチマーク

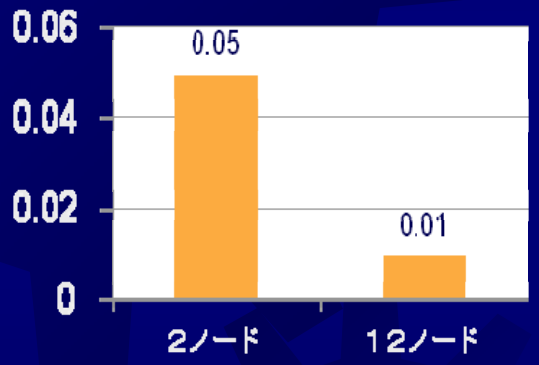
マスタ・スレーブDBやクライアントの高負荷による性能劣化ではない

ベンチマーク実行中のCPU使用率

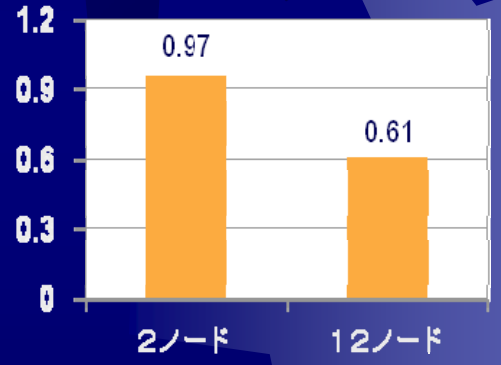
マスタDBノード



スレーブDBノード



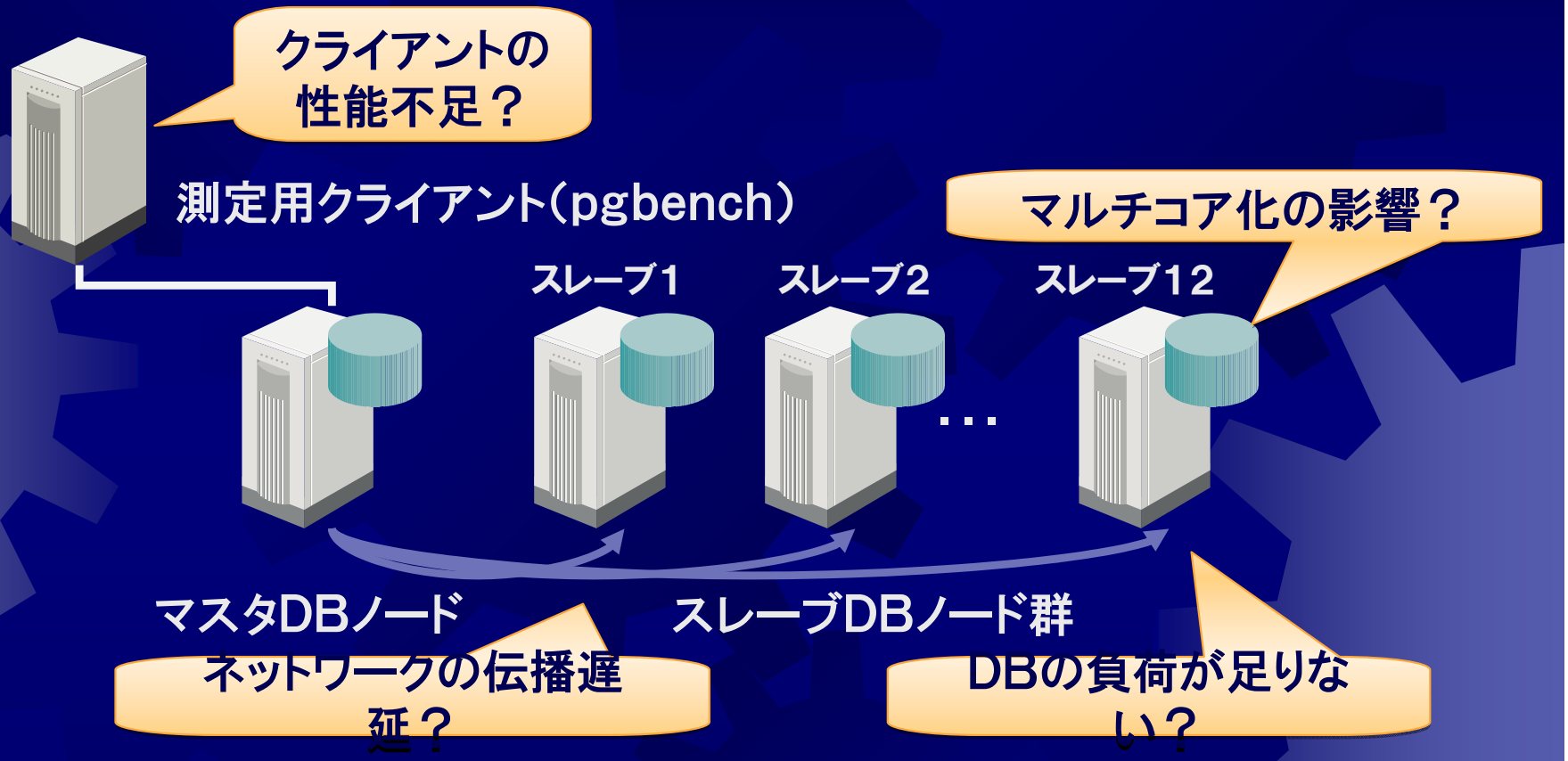
クライアント



マスタDBノードのCPU使用率はむしろ低下

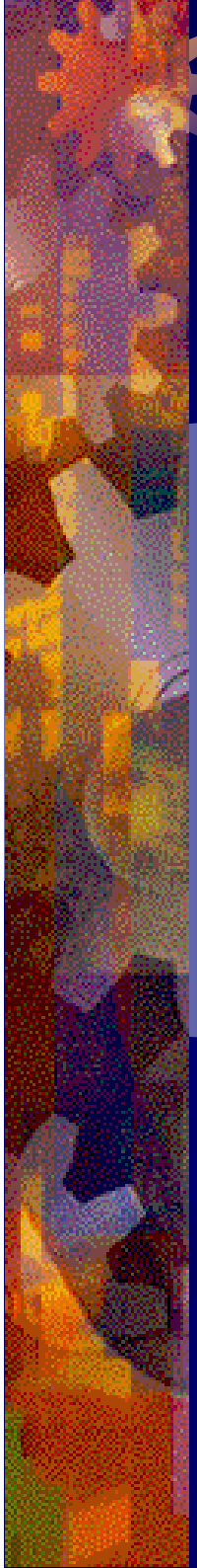
スレーブノード数別ベンチマーク

ボトルネックはどこか



最も大きなボトルネックはいずれの可能性も低い

OProfileサンプリング結果では「LWLockAcquire」などがやや上位に。
ロック待ちが影響している?



データロード・大量更新処理性能
VACUUM処理性能

検証結果

データロード・大量更新処理性能

バッチ処理を想定し、マスタDBノードにおけるデータロード・大量更新処理性能を計測

①データロード処理時間測定条件

- DBノード数：2、12ノード
- データロード件数： 100,000,000件

②UPDATE処理時間測定条件

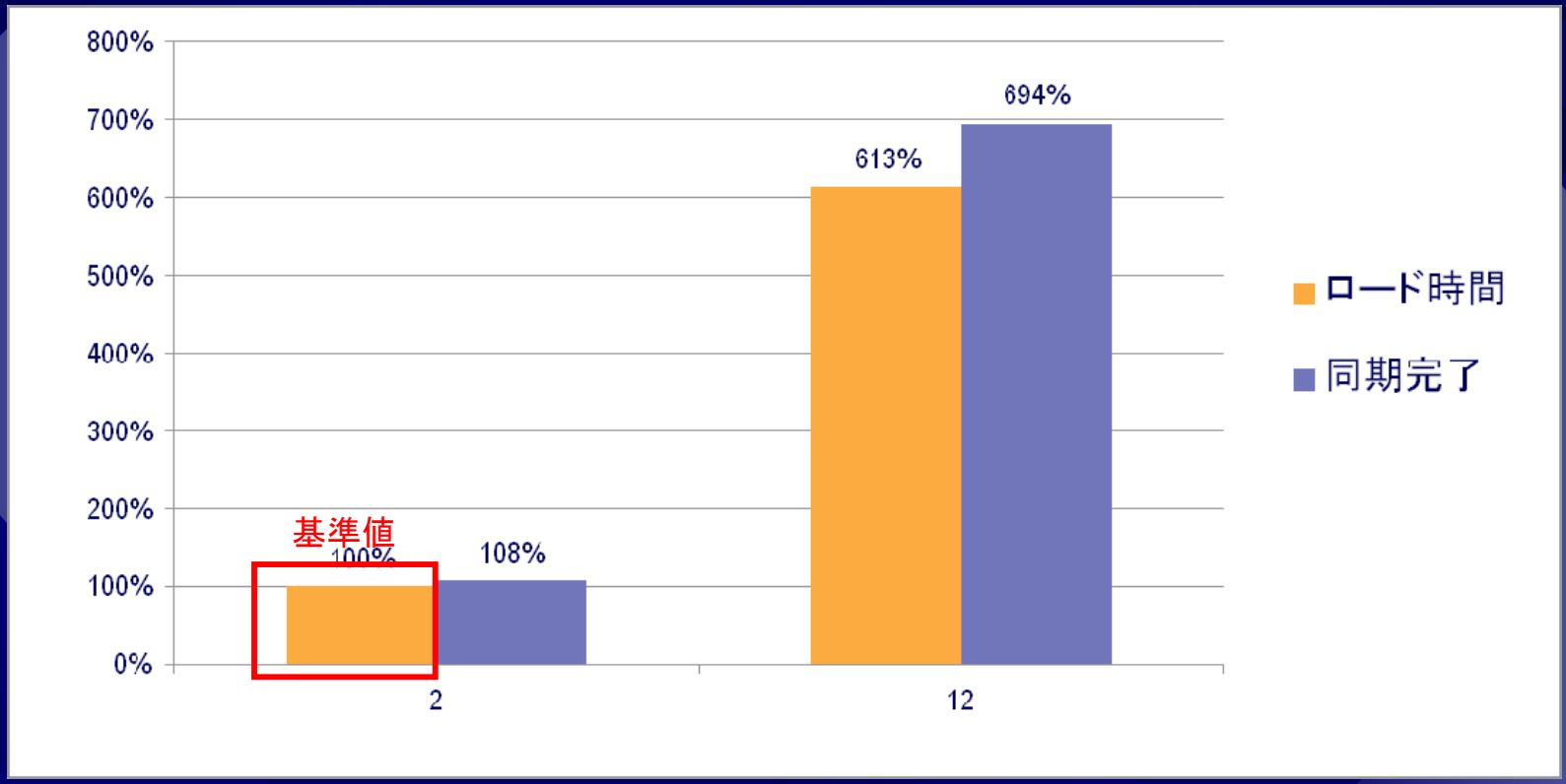
- UPDATEレコード件数：10,000、100,000、500,000、1,000,000

③VACUUM処理時間測定条件

- pgbench実行直後のVACUUM処理時間
  - pgbenchを500クライアントで10分間実行
  - 母集団のデータ件数は1億件
- autovacuumは無効

データロード処理性能

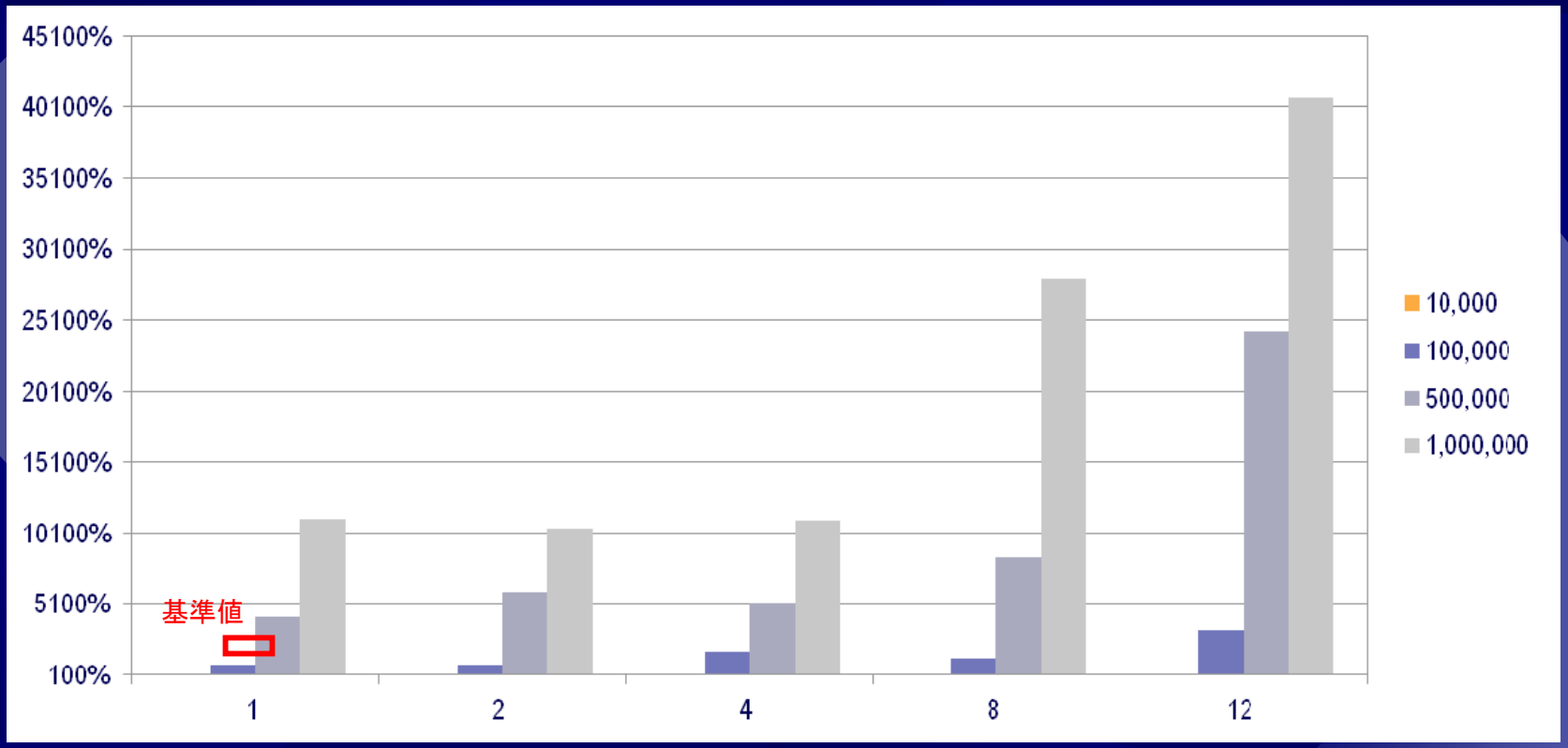
- スレーブノード数がデータロード時間に大きく影響
- データロード完了からノード数の同期までに10分以上



- 2ノードのデータロード時間を基準として、同期完了時間と12ノードにおける処理時間の比率をグラフ化しています。
- 同期の完了は、マスタ／スレーブノードのWAL格納位置が同期するまでの時間です。

大量更新処理性能

- スレーブノード数増加に応じて処理時間が拡大
- 更新件数増加により差が顕著に表れる

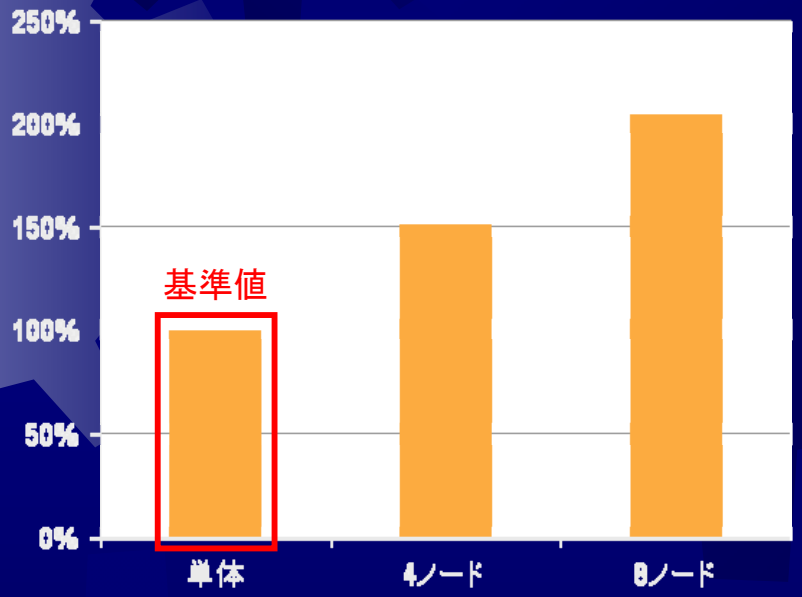


- 1DBノード、100000件の更新処理時間を基準として、更新件数・ノード別の処理時間の比率をグラフ化しています。

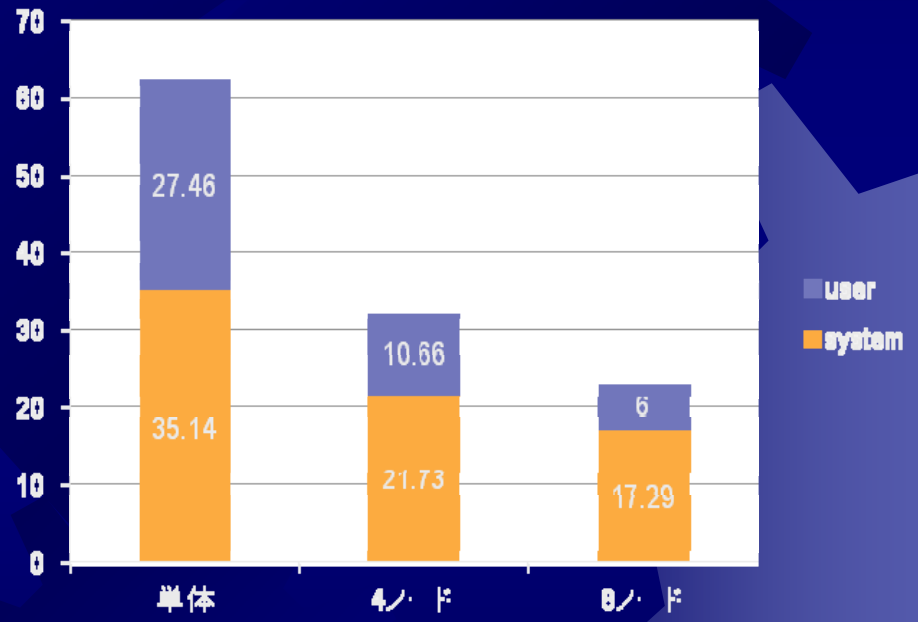
VACUUM処理

•スレーブノード数に比例してVACUUM処理時間が拡大
•CPU使用時間はノード数増加に応じて減少

VACUUM処理時間



CPU使用時間



•VACUUM処理時間は、DB単体の処理時間を基準とした比率をグラフ化しています。
•VACUUM処理時間、CPU使用時間は、VACUUMのメッセージから情報を採取しています

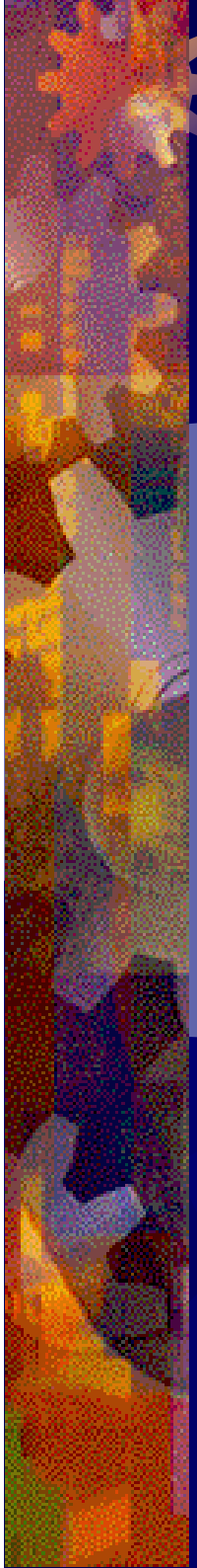
データロード、大量更新、VACUUM処理性能

マスタDBノードへの影響

WALを大量に出力する可能性があるデータロード、更新、VACUUM、は、スレーブDBノード数に比例して処理時間が延びる

スレーブDBノードへの影響

- 大量にWALを出力する処理では、スレーブへの反映完了が大幅に遅れる可能性がある
- 未反映WALファイルが大量に残留し、マスタDBノードのwal_keep_segmentsを超える可能性がある



ストリーミングレプリケーション運用上の注意

運用上の注意点など：規模の拡大・縮小

ノードの追加

マスタサーバ

SELECT pg_start_backup('label');
・・・*rsyncなどで転送*
SELECT pg_stop_backup();

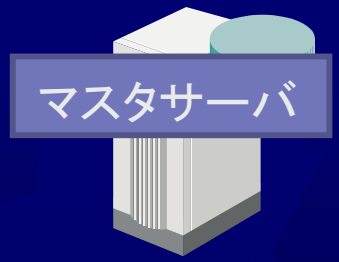
追加ノード

スレーブサーバ

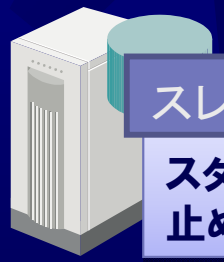
recovery.conf編集
postgresql.conf編集
pg_ctl start

- **難しい操作ではない**
- **スタンバイサーバ側で一括操作できると運用は楽になる**
- **スタンバイサーバ追加を見込み、max_wal_senders = xxx の値を設定しておく必要がある**

ノードの削除



マスタサーバ



削除ノード

スレーブサーバ

スタンバイサーバのPostgreSQLを止めるだけ

とても簡単♪

更新性能への影響をお忘れなく

ノードの追加・削除により、特に更新性能に影響を与えるので十分考慮する

運用上の注意点など:WAL関連

- スタンバイサーバのWAL格納領域はディスク容量に余裕を

  特にスタンバイサーバのWALファイル格納領域は、WAL送受信の遅延によりディスクサイズが増える可能性がある。最悪の場合、ディスクフルとなってPostgreSQLが停止してしまう。

```
[postgres@standby]$ ls $PGDATA/pg_xlog
000000010000000100000E2
000000010000000100000E3                    ←WALファイルが大量に存在
:
```

- wal_keep_segmentsは十分に大きな値を設定する

  レプリケーションが追いつかないとWALセグメントがマスタから削除済みになり、レプリケーションできなくなる。ただしスタンバイサーバは参照用DBとして生き続けるため、失敗の検知は必須。

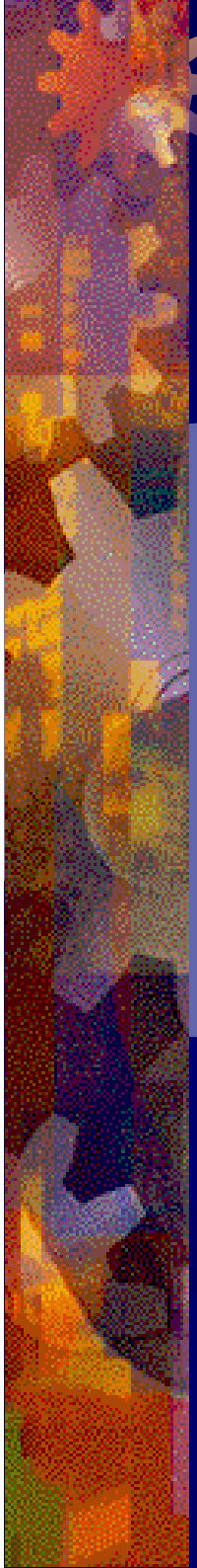
```
マスタ側 メッセージ
FATAL:  requested WAL segment
000000010000000000000040 has already been removed
```

```
スタンバイ側 メッセージ
FATAL:  could not receive data from WAL stream: FATAL:  requested WAL
segment 000000010000000000000040 has already been removed
```

- その他

  今回の測定環境では、以下のパラメータによる性能向上は確認できなかった。

  - マスタ:full_page_writes
  - スタンバイ:fsync、checkpoint_segments



検証結果 まとめ

スケールアウト基盤としての ストリーミングレプリケーション

① スケールアウト基盤としては十分に利用可能

- 今回の性能検証では、サーバ台数の増加に対して少しずつリニアな性能劣化が確認できる程度だった。
- マスタサーバへの負荷が低い
- 大量更新以外の測定では、ノード数が増加しても目立った反映の遅延は発生しなかった。

② 導入時に考慮すべき点

- マスタ／スレーブDBノードのWAL領域には十分な注意が必要
- バッチによる大量更新、VACUUMは、スレーブDBノードへの反映の遅延を見越した設計が必要
- ログ、システム管理関数を使用したデータの反映状況およびレプリケーション失敗の監視は必須

③ 今後の検証項目

- ロングランによる傾向分析

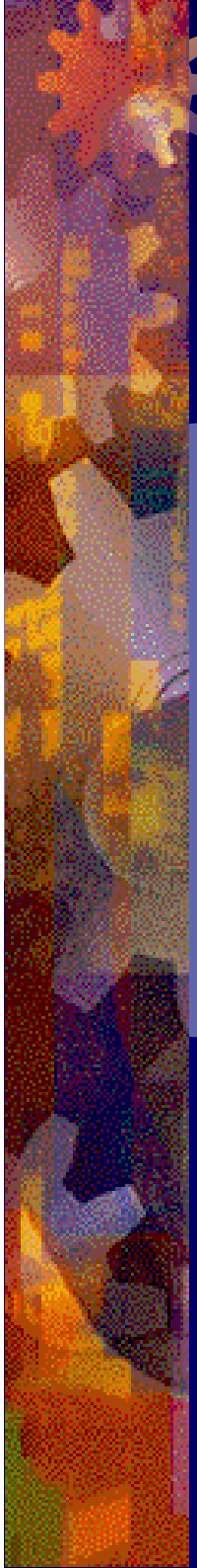
今後への期待

- スレーブDBノード数増加によるマスタDBノードの負荷軽減
  - 部分データベースレプリケーションの実現
  - レプリケーション処理のカスケード構成対応
  - レプリケーションのマスタDBノード分散
- スレーブDBのリカバリ性能改善
  - 大量更新後の反映の遅延が、WAL領域の圧迫や同期失敗の引き金となっている
- 環境構築時の支援強化
  - 稼動中のアプリケーションからはレプリケーションで発生した問題がわかりにくいため、設計段階で問題を回避できるよう情報提供が必要。
  - 設定方法等のガイドの充実
    - WAL領域サイズ等、最適なリソース設定
    - スレーブノードへの反映タイミング
    - 更新処理やVACUUMの処理時間への影響
  - 環境設定の簡易化またはツール

参考文献および商標

- 参考文献
  - PostgreSQL9.0.2日本語マニュアル
  （日本PostgreSQLユーザ会 文書・書籍関連分科会 訳）
  http://www.postgresql.jp/document/
  - pgpool-II + HS/SRクラスタ構成
  (Let‘s Postgres 記事)
  http://lets.postgresql.jp/documents/technical/pgpool/3/

- 商標
  - AMDは，Advanced Micro Devices, Inc.の商標です。
  - Intel Xeonは，アメリカ合衆国およびその他の国におけるIntel Corporationの商標です。
  - Linuxは，Linus Torvalds氏の日本およびその他の国における登録商標または商標です。
  - PostgreSQLは，PostgreSQL Global Development Groupが提唱する，オープンソースのオブジェクトリレーショナルデータベース管理システムの名称です。
  - Red Hatは，米国およびその他の国でRed Hat, Inc. の登録商標もしくは商標です。



END

PostgreSQL 9.0
ストリーミングレプリケーションの実力

2011年2月25日

株式会社日立製作所 ソフトウェア事業部
福岡博
NECソフト株式会社 PFシステム事業部
岩浅晃郎