



# NTT OSSセンターにおける PostgreSQLへの取り組み

- NTT OSSセンターの活動紹介
- NTTグループにおける事例紹介

2014. 9.18  
NTT OSS センタ  
原野 鉄也

## ■NTT OSSセンタの活動紹介

- ①OSS適用推進
- ②ソフトウェア基盤技術力向上
- ③OSSトータルサポート
- ④技術開発

## ■NTTグループにおける事例紹介

- ⑤PG-REX導入事例
- ⑥マイグレーション事例
- ⑦トラブルシューティング事例

# NTT OSSセンタ設立の目的

- ◆ OSS活用によるNTTグループ社内システムのTCO削減を目的に2006年4月設立
- ◆ 事業現場におけるOSS導入阻害要因の解消をめざす

情報収集を一元的にできないか

プロダクトの保守サポート体制は大丈夫なのか

構築ノウハウ(ミドルウェアの選定・構成方法等)を結集できないか

エンタープライズ用途にはOSSの機能強化(高信頼化・高可用化)が必要ではないか

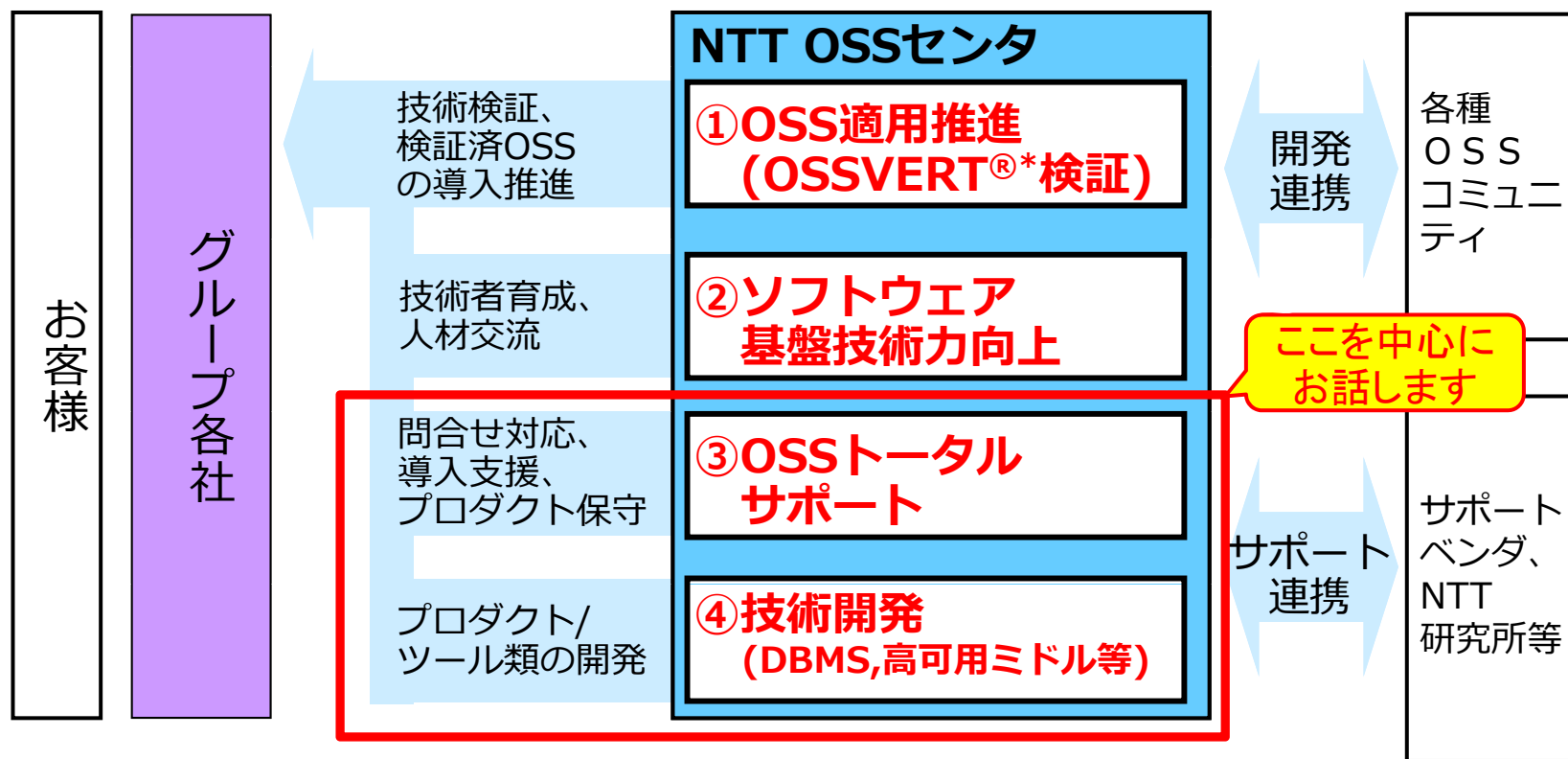
など

# NTT OSSセンタの位置づけ



◆ 下記の4つ（①～④）のミッションでグループ事業に貢献

## NTT OSSセンタのミッションと位置づけ



\*)OSSVERT® : OSs Suites VERified Technically (技術検証済みOSS組合せ)

## ■NTT OSSセンタの活動紹介

### ①OSS適用推進

### ②ソフトウェア基盤技術力向上

### ③OSSトータルサポート

### ④技術開発

## ■NTTグループにおける事例紹介

### ⑤PG-REX導入事例

### ⑥マイグレーション事例

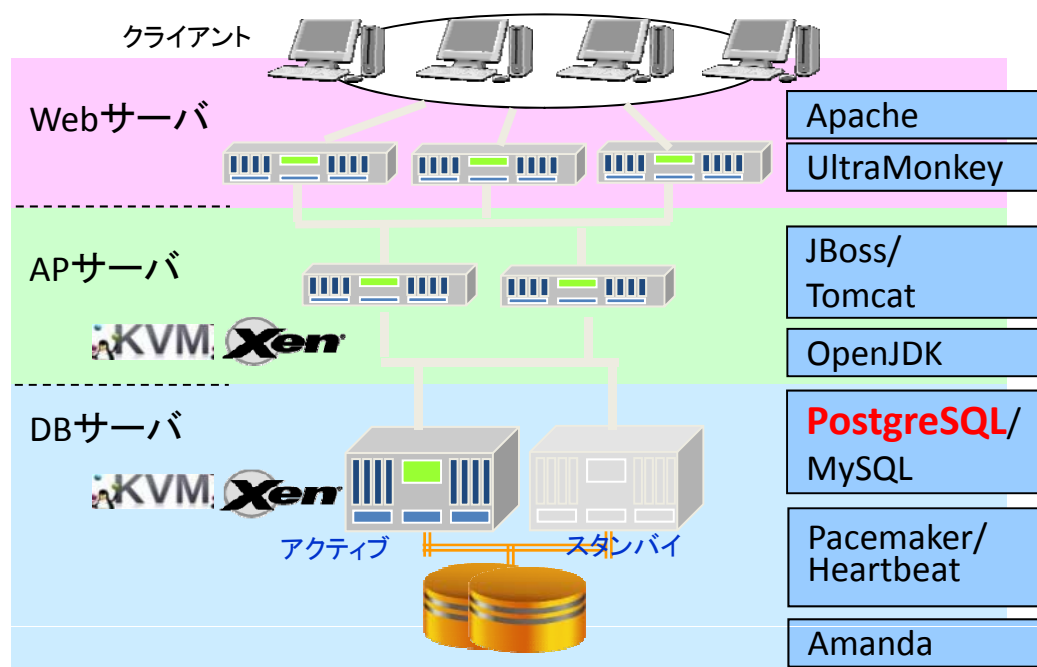
### ⑦トラブルシューティング事例

# OSSVERT®モデル検証



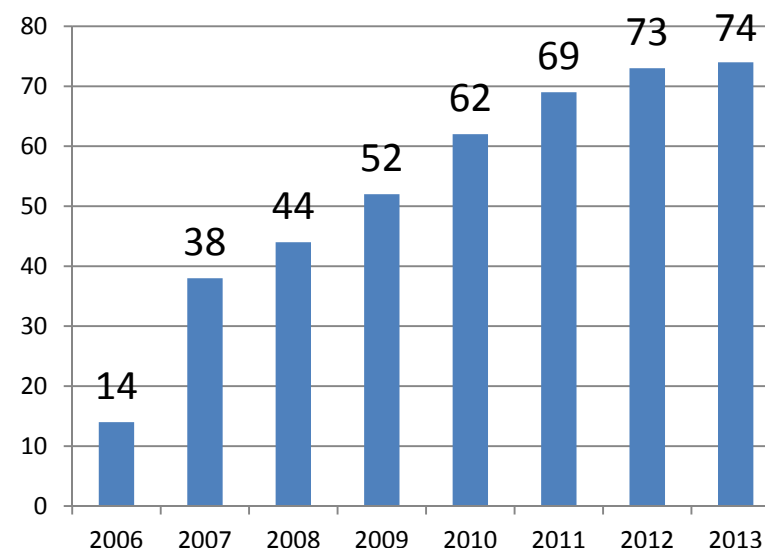
◆ Web3層モデルを中心とした**OSSミドル組合せ検証**により  
品質・性能確認、各種パラメータ値決定、適用領域明確化

- 失敗しない製品選定
- 手戻りのない設計・開発
- 性能向上、迅速なトラブル解決



OSSVERT®モデル構成イメージ

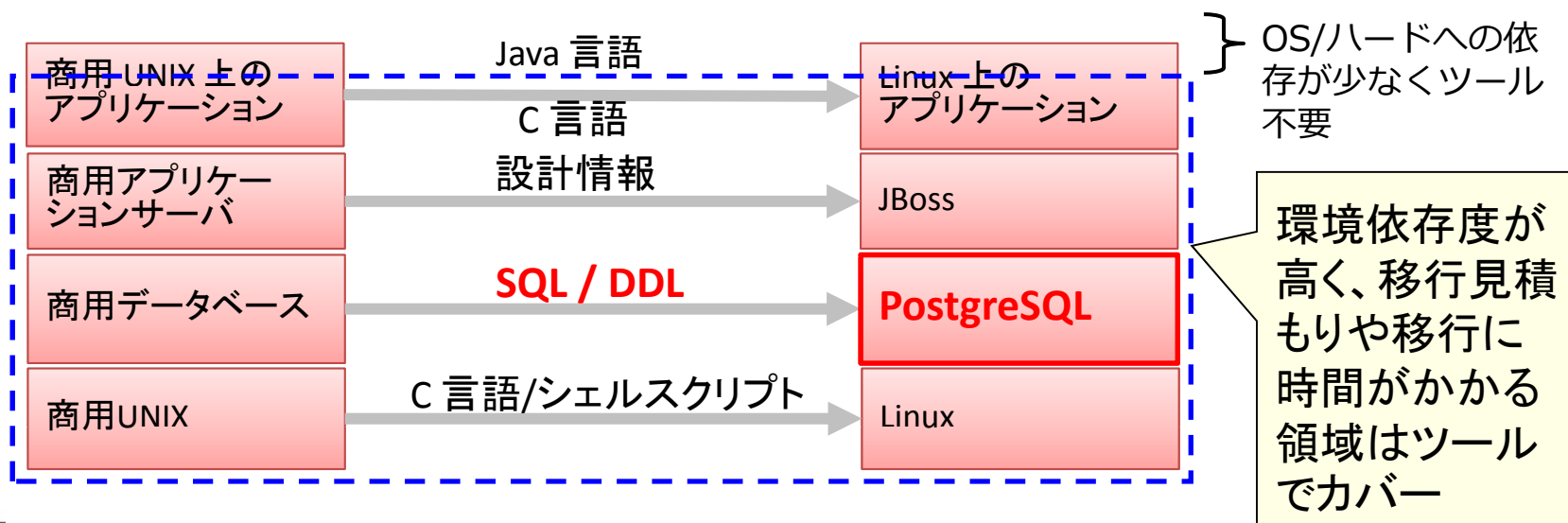
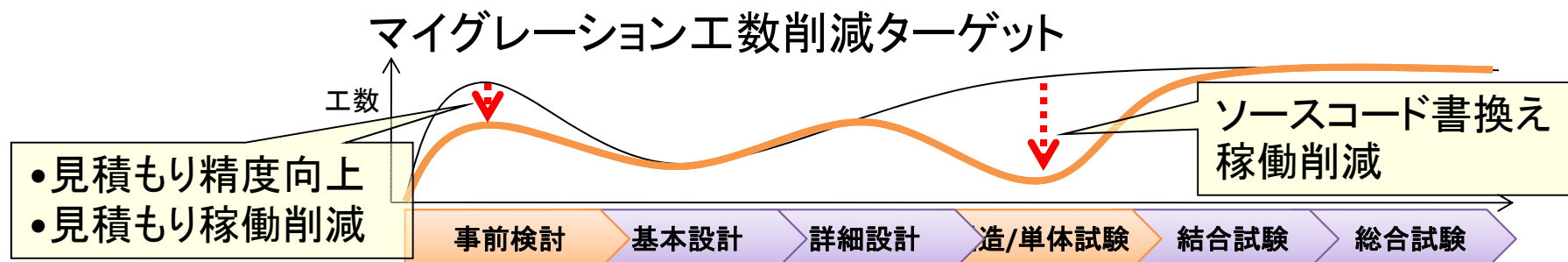
社内システムへのOSSVERT®導入数



# OSSマイグレーション



## ◆ マイグレーション化に向けたノウハウ蓄積・支援ツール整備 (OSSマイグレーション支援サービス)



## ■NTT OSSセンタの活動紹介

- ①OSS適用推進
- ②ソフトウェア基盤技術力向上**
- ③OSSトータルサポート
- ④技術開発

## ■NTTグループにおける事例紹介

- ⑤PG-REX導入事例
- ⑥マイグレーション事例
- ⑦トラブルシューティング事例

# コミュニティ活動

## ◆ 主要プロダクトのOSSコミュニティに積極的に参画

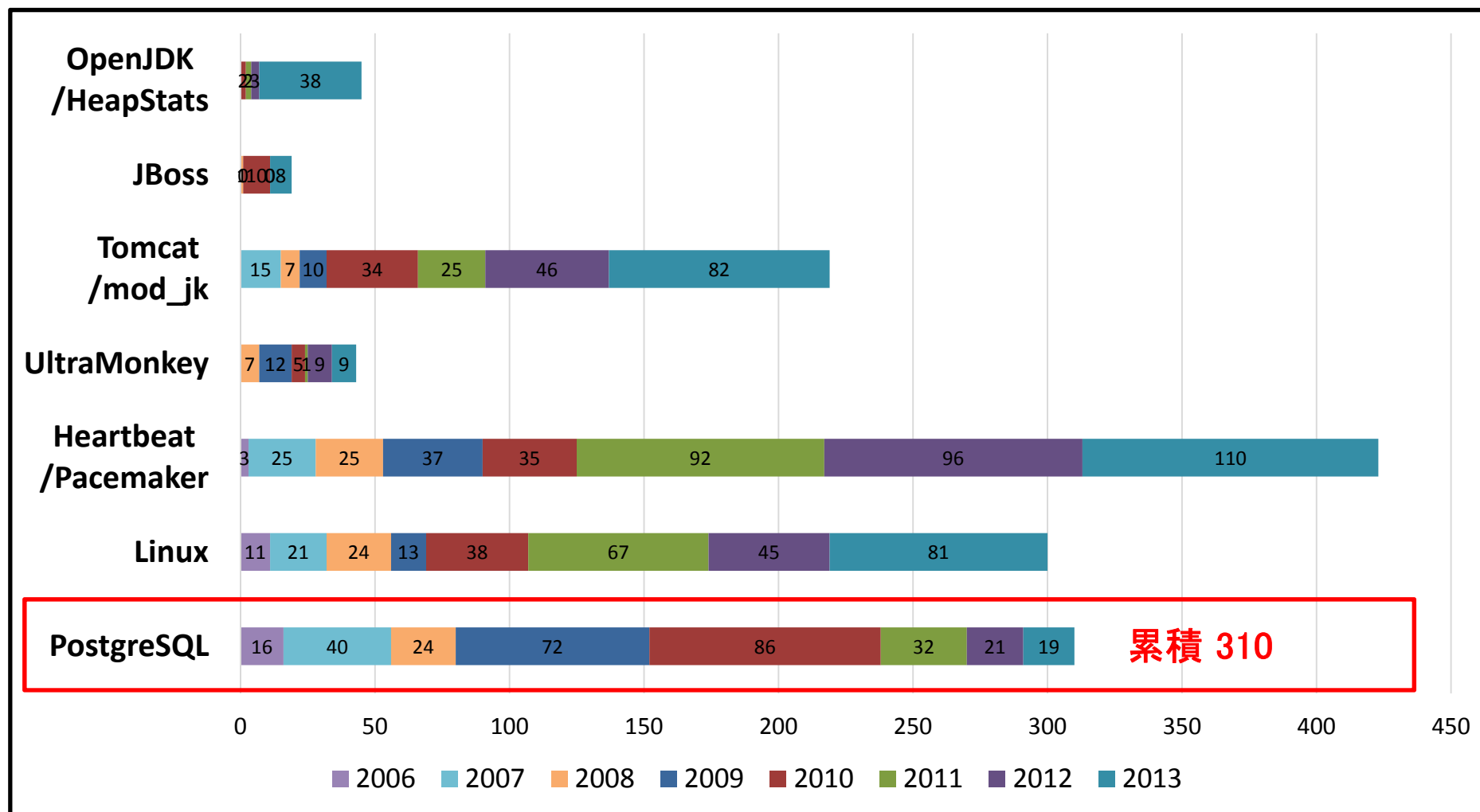
### 主なコミュニティ活動状況

| プロダクト                | 主なコミュニティ活動                                                                                                                                                                                                                    | 主要メンバ（OB含む）                                                                                                                               |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| PostgreSQL           | <ul style="list-style-type: none"> <li>○PostgreSQL 開発・改善へ継続的にパッチ貢献</li> <li>○PostgreSQLのレプリケーション機能開発への貢献で藤井が「日本OSS奨励賞」受賞(2010.10)</li> <li>○企業ユーザ会PGECons立ち上げ(2012.4)</li> </ul>                                              | <ul style="list-style-type: none"> <li>●コミッタ</li> <li>●JPUG理事</li> </ul>                                                                  |
| Linux                | <ul style="list-style-type: none"> <li>○Linux Kernel 開発・改善へ継続的にパッチ貢献</li> <li>○カーネルクラッシュダンプ機能開発等でフェルナンドが「日本OSS貢献者賞」(2009.10)、TOMOYO Linux開発等で半田が「日本OSS貢献者賞」(2013.2)、KVM開発等で吉川が「日本OSS奨励賞」(2013.2)受賞</li> </ul>                 | <ul style="list-style-type: none"> <li>●LKDTT (Linux Kernel Dump Test Tool)メンテナ</li> <li>●TOMOYO Linuxメンテナ</li> <li>●NILFSメンテナ</li> </ul> |
| Heartbeat /Pacemaker | <ul style="list-style-type: none"> <li>○コミュニティのステアリングコミッティメンバとしてHeartbeatからPacemakerへの移行に貢献</li> <li>○Heartbeat/Pacemaker開発貢献で森が「日本OSS貢献者賞」受賞(2012.3)</li> <li>○Pacemaker-PostgreSQL連携機能開発で松尾が「日本OSS奨励賞」受賞(2014.2)</li> </ul> | <ul style="list-style-type: none"> <li>●Linux-HAステアリングコミッティメンバ/コミッタ</li> <li>●Pacemakerメンテナ</li> </ul>                                    |
| UltraMonkey          | <ul style="list-style-type: none"> <li>○コムウェア発、負荷分散ソフトウェアのコミュニティを牽引</li> </ul>                                                                                                                                                | <ul style="list-style-type: none"> <li>●コミッタ</li> </ul>                                                                                   |
| TOMCAT/mod_jk        | <ul style="list-style-type: none"> <li>○コミッタとしての活動が認められ、藤野が上級コミッタに昇格(2012.7)</li> <li>○TOMCAT開発貢献で藤野が「日本OSS貢献者賞」受賞(2014.2)</li> </ul>                                                                                         | <ul style="list-style-type: none"> <li>●上級コミッタ</li> <li>●コミッタ</li> </ul>                                                                  |
| JBossEAP             | <ul style="list-style-type: none"> <li>○米国Red Hat社とのJBoss品質改善TFに参画し、品質改善に貢献</li> <li>○EAP6β版評価プログラムに参画、品質改善に貢献</li> </ul>                                                                                                     | <ul style="list-style-type: none"> <li>●HornetQコミッタ</li> <li>●TUBAMEコミッタ</li> </ul>                                                       |
| OpenJDK HeapStats    | <ul style="list-style-type: none"> <li>○Java障害解析支援ツールHeapStatsをIcedTeaコミュニティに公開(2013.4)</li> </ul>                                                                                                                            | <ul style="list-style-type: none"> <li>●OpenJDKオーサ</li> <li>●HeapStatsコミッタ</li> </ul>                                                     |

# コミュニティ活動



## ◆パッチ貢献件数



## ■NTT OSSセンタの活動紹介

- ①OSS適用推進
- ②ソフトウェア基盤技術力向上
- ③OSSトータルサポート**
- ④技術開発

## ■NTTグループにおける事例紹介

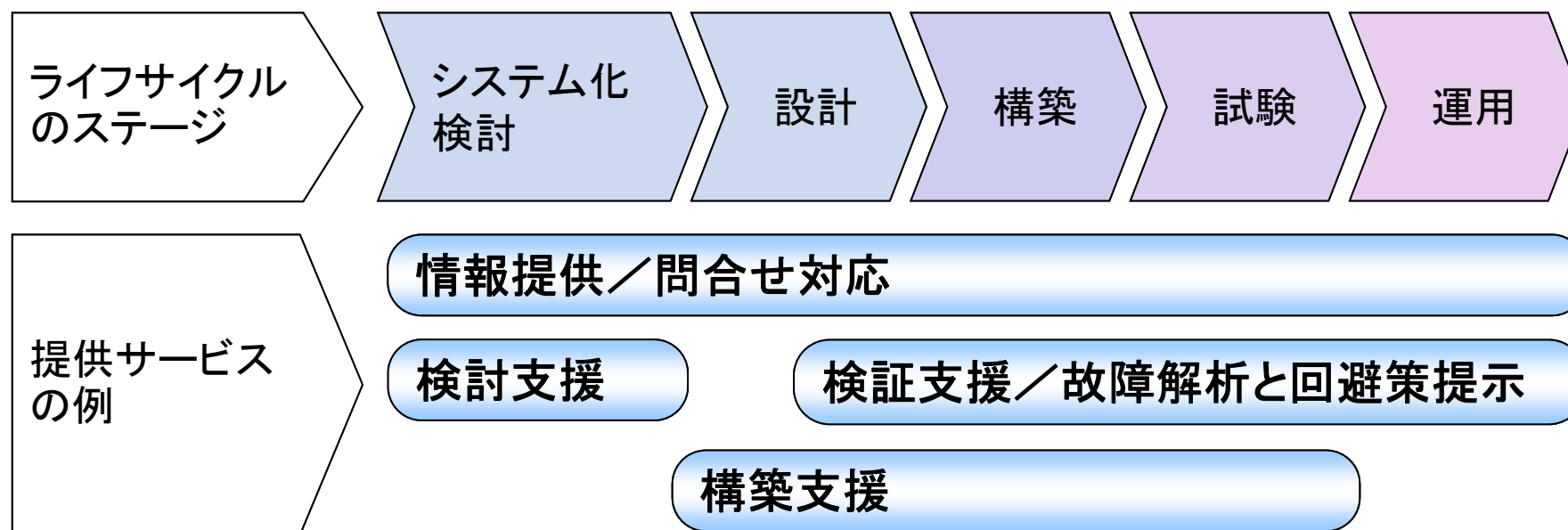
- ⑤PG-REX導入事例
- ⑥マイグレーション事例
- ⑦トラブルシューティング事例

# OSSトータルサポートとは



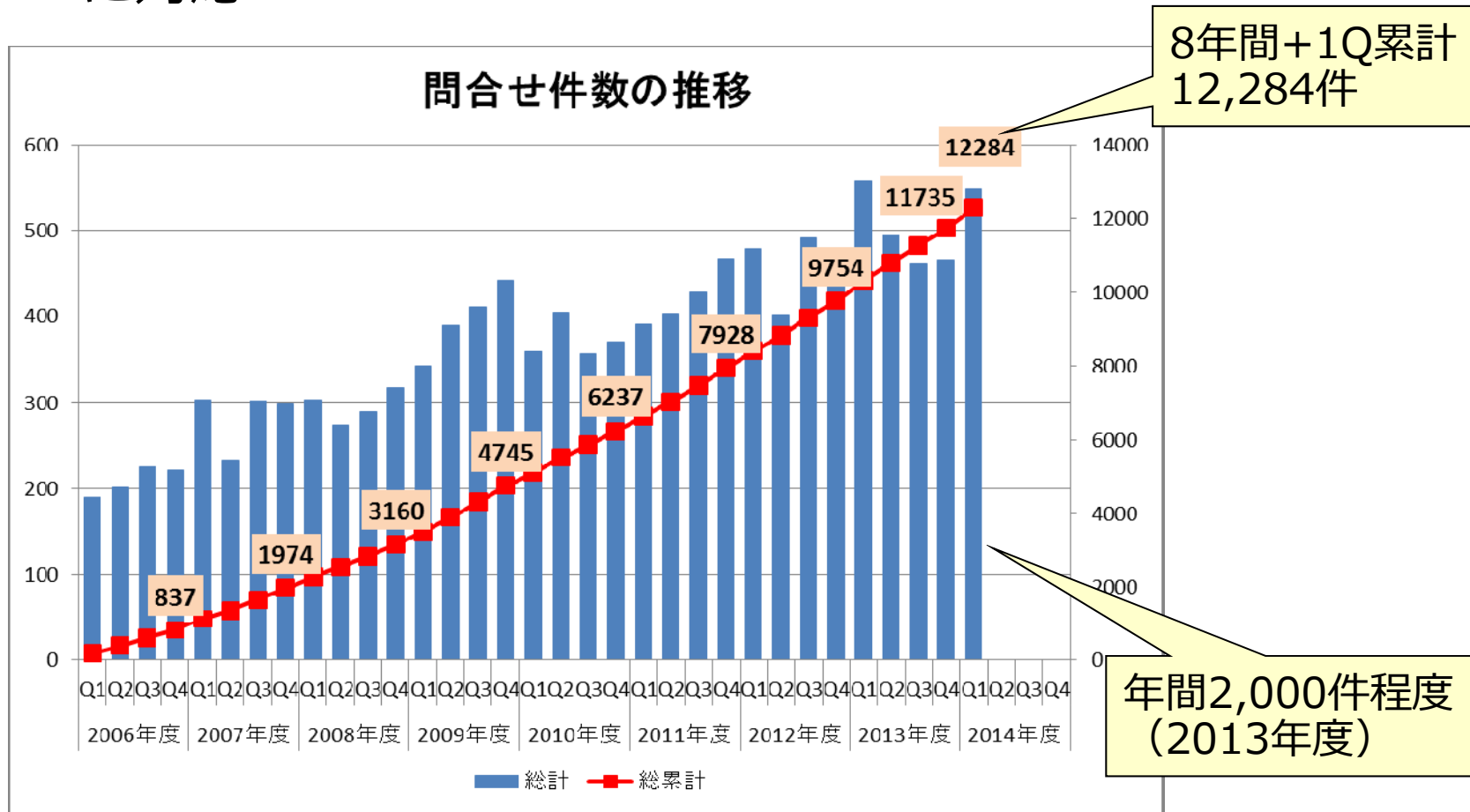
- ◆ システムライフサイクルの各ステージでOSS化をサポート  
システム化の検討段階での製品選定、性能検証から構築支援、運用時の  
トラブル対応まで

## OSSトータルサポートのスコープ



# 問い合わせ対応 (1/2)

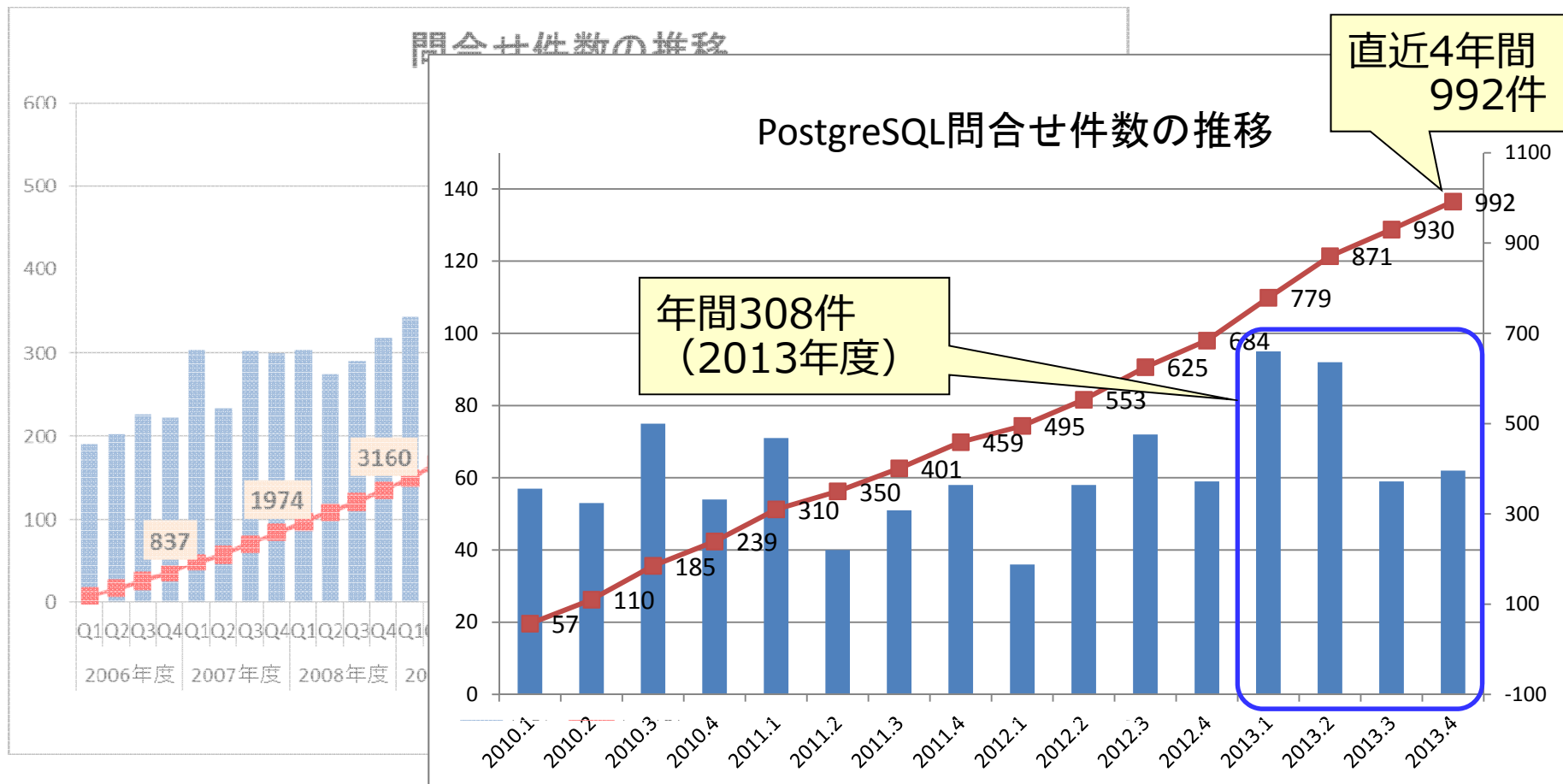
◆ ポータルサイト経由の問合せ累計 **約12,000件** (2014年Q1)  
 に対応



# 問い合わせ対応 (1/2)

◆ 直近4年間でPostgreSQLの問合せ 約1,000件

➤ 2013年度問い合わせで約300件あり、利用数は年々増加している。



# 問い合わせ対応 (2/2)

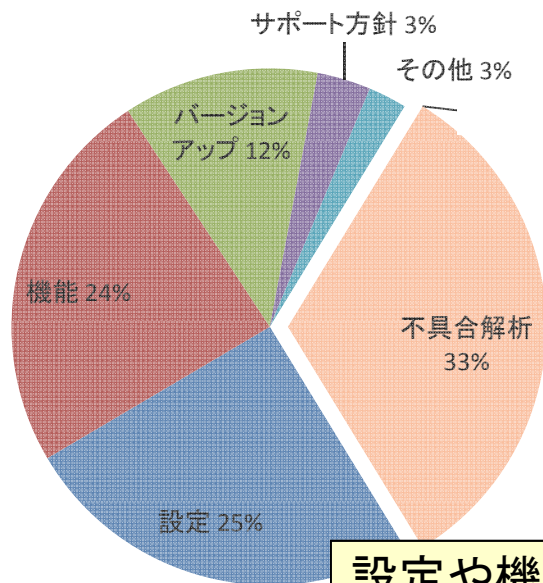
## ◆ 2013年度におけるPostgreSQLの問合せ 約300件

- 設定や機能に関する問合せが多い (約6割)
- 不具合解析 26% → そのうち『仕様通り』が約4割
- PostgreSQLのバグ起因 1%、新規バグ 0%

ノウハウ整備  
とその展開が  
重要

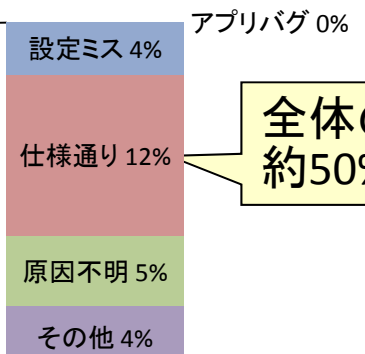
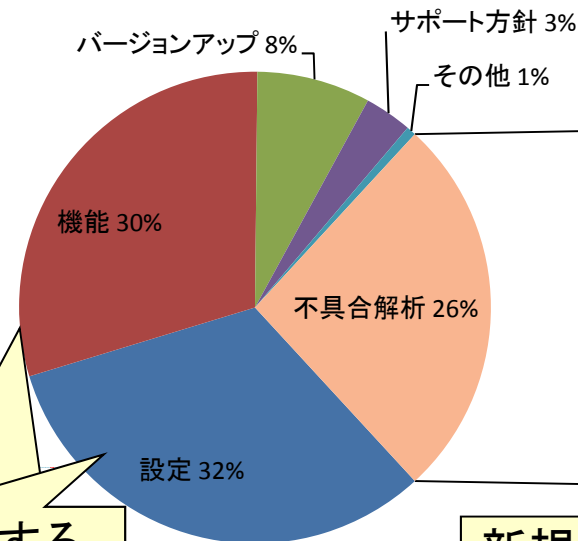
→品質面に問題なく、安定した運用が行えている

問合せ全体-傾向分析(件)



設定や機能に関する  
問合せ約6割

PostgreSQL 傾向分析(件)-2013年度



全体の  
約50%

新規は0

DBMSバグ 1%

## ■NTT OSSセンタの活動紹介

- ①OSS適用推進
- ②ソフトウェア基盤技術力向上
- ③OSSトータルサポート
- ④技術開発**

## ■NTTグループにおける事例紹介

- ⑤PG-REX導入事例
- ⑥マイグレーション事例
- ⑦トラブルシューティング事例

# PostgreSQLへの取り組み～Background



- ◆ NTT研究所において、OSSデータベースの研究開発に着手した2004年当時、機能と開発体制から選定

## オープンソースDBMSの比較(2004年当時)

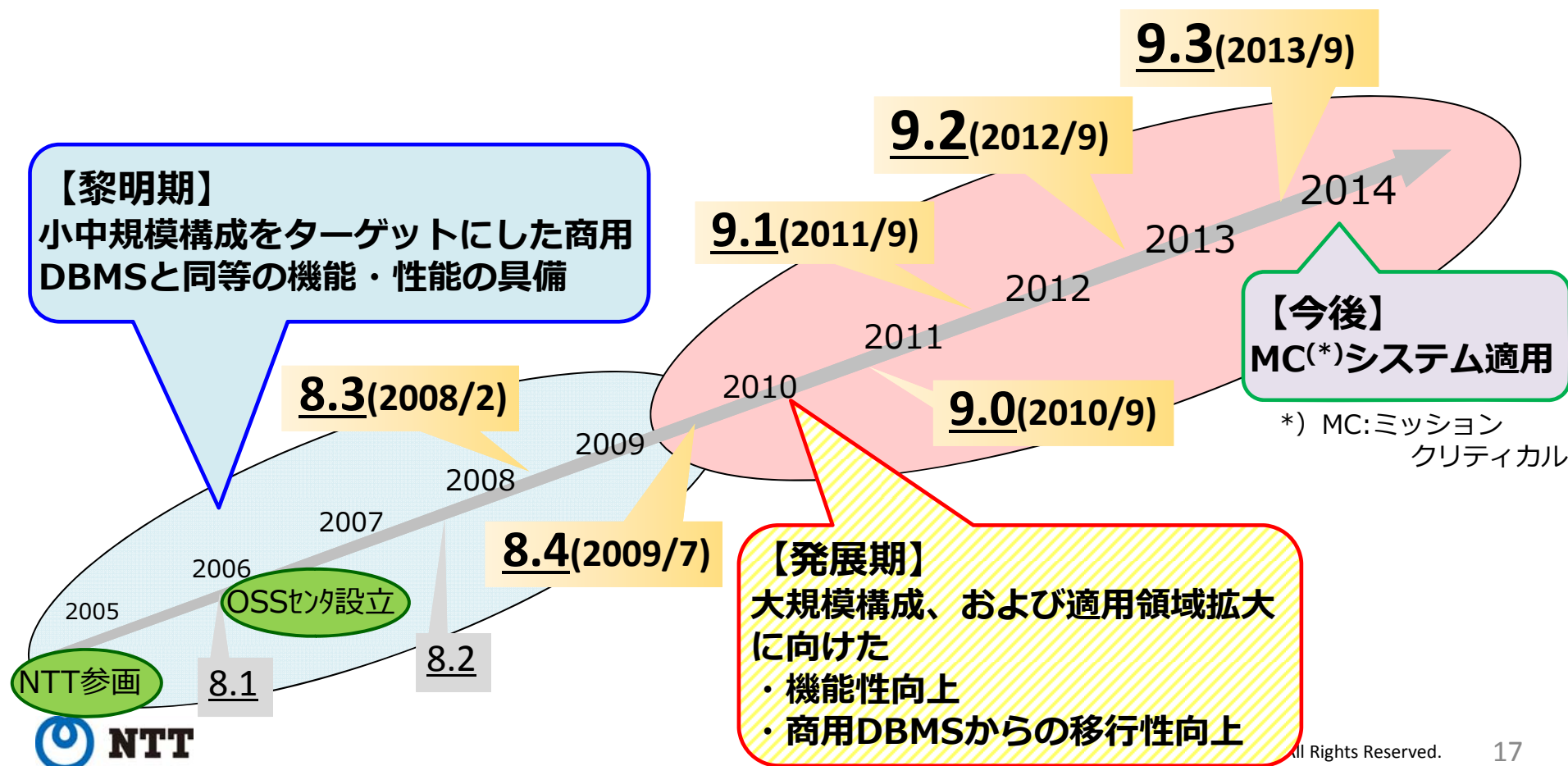
|      |          | PostgreSQL (7.4)                                | MySQL(4.0)                                    |
|------|----------|-------------------------------------------------|-----------------------------------------------|
| 機能   | SQLの対応範囲 | SQL92を <b>ほぼカバー</b>                             | 副照会とカーソルの <b>一部に制約があった</b>                    |
|      | 日本語文字の対応 | JIS, EUC等主要コードを <b>利用可能</b>                     | 種々の <b>制約があった</b>                             |
| 開発体制 | 開発主体     | ベンダ独立のコミュニティが開発しており、パッチの受け入れ等について <b>中立性が高い</b> | 特定ベンダのプロダクトであり、パッチの受け入れ等は <b>ベンダの意向に左右される</b> |
|      | 海外コミュニティ | <b>開発コミュニティを含め活発</b>                            | <b>ユーザコミュニティのみ活発</b>                          |
|      | 国内コミュニティ | ユーザコミュニティが <b>活発</b>                            | ユーザコミュニティが <b>立ち上がりつつあった</b>                  |

# PostgreSQLの進化とOSSセンタの関わり



## ◆ 中長期的な取り組みとして、PostgreSQL開発にも参画

- 開発方針議論への参画、ユーザ会JPUGポータルサイト運営支援
- 9.xにおいて国内トップのパッチ貢献

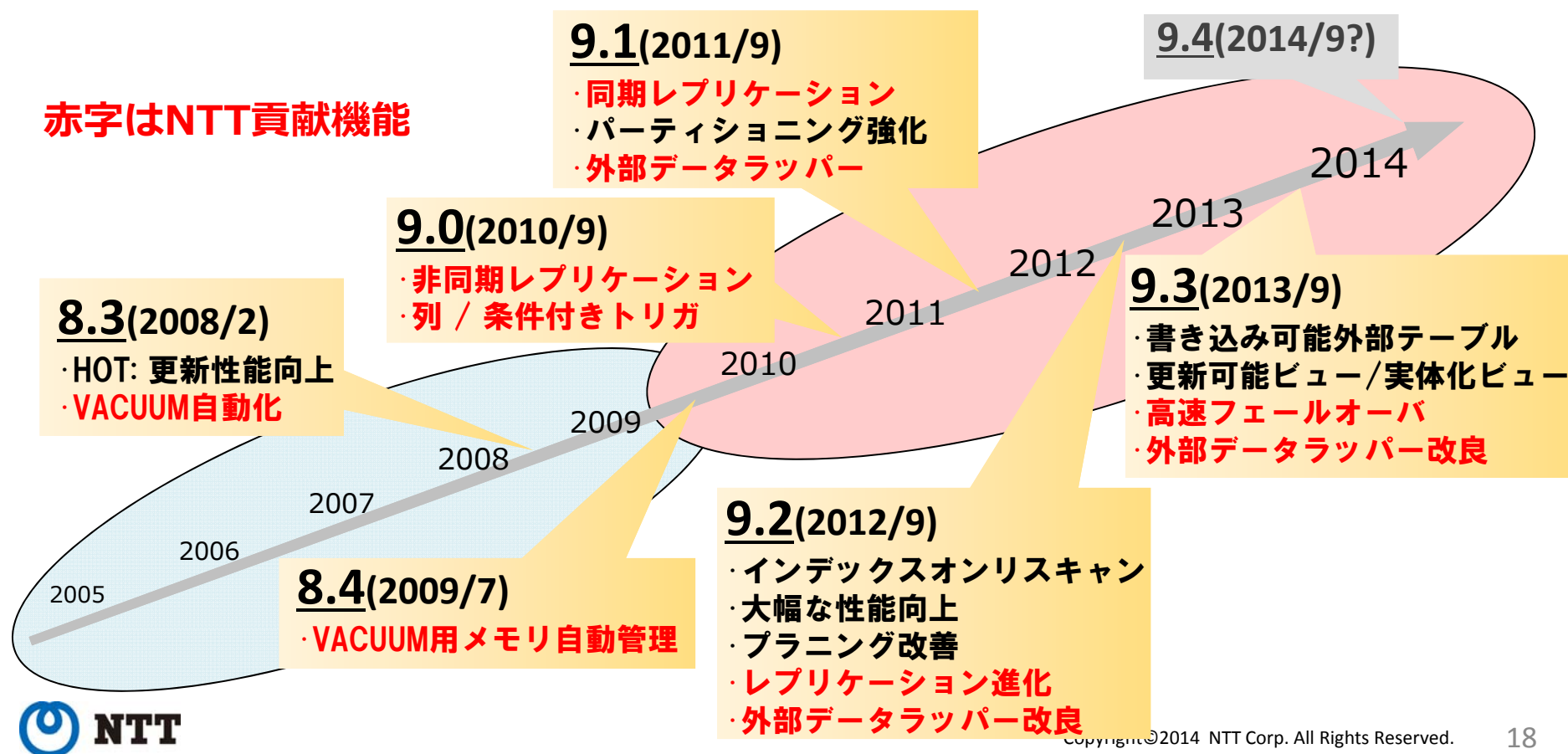


# PostgreSQLの進化とOSSセンタの関わり



## ◆ エンタープライズ用途観点で新機能や性能改善の開発に貢献

- NTTグループでは8.3からシステムへの導入が大きく加速
- 最近ではレプリケーション機能とHAを組合わせたPG-REX構成も...



## ◆ NTTが機能開発に取り組んだ機能のご紹介

- 開発に取り組んだ背景や課題
- 開発した機能の概要

### 1. レプリケーション (PG-REX)

### 2. 外部データラッパー

### 3. 移植の効率化

# PostgreSQL開発 - レプリケーション



## ◆ NTTが取り組んだ背景・課題 ～OSSセンタ設立の頃より…

### TCOの徹底的な削減

- ・ 厳しい競争により **システム投資は抑制傾向**  
HWも安価なものを組合わせてシステム要件を実現したい  
→特に小規模システムで要望が強い … その上一定の信頼性も求められる

### 可用性の向上

- ・ **サービス継続性** に対する要求レベルの上昇  
ダウンタイム時間の短縮化 99.99% (52.6分) → 99.999% (5.26分)  
→共有ディスク方式の冗長構成では実現出来ないケースが…

### 性能のスケールアウト

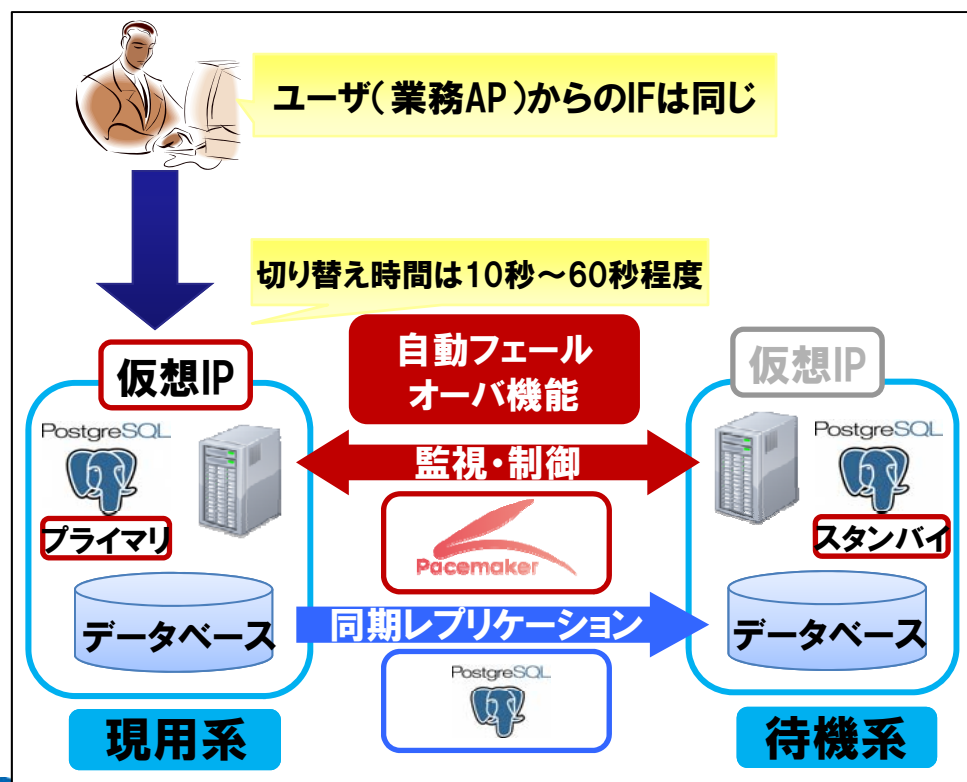
- ・ DBサーバの処理性能向上では、**スケールアップが一般的**  
→スケールアップは、HWやSWの初期コストも大きくなりがち  
また  
サーバ分割 (スケールアウト) は設計見直しなどの開発コスト増に…

# PostgreSQL開発 - レプリケーション

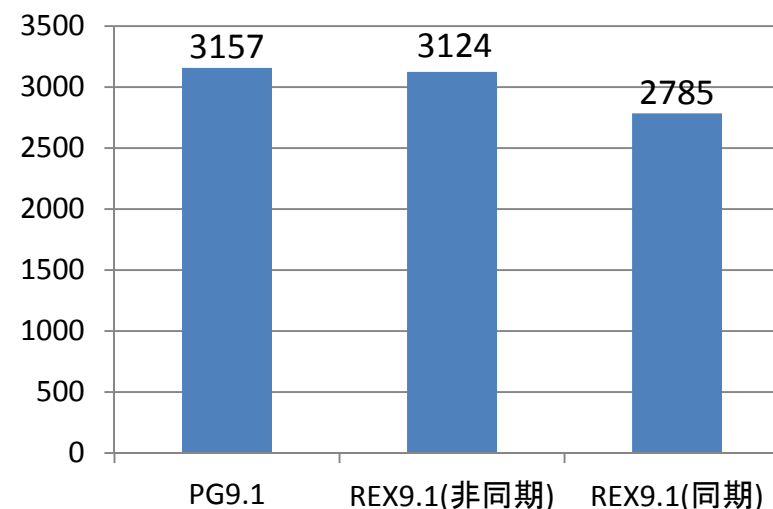


## ◆ PG-REX (<http://sourceforge.jp/projects/pg-rex/>) 最新版はPG-REX9.3

- PostgreSQLとPacemakerを組み合わせた高可用ソリューション
- 共有ディスク不要：低コストでHAクラスタを実現
- 待機系の有効利用：待機系でも参照SQLを処理可能



NOTPM DBT-2 を用いた更新性能の検証



### 検証環境条件

- H/W
  - CPU Xeon 2.66GHz × 4core, MEM 18GB
  - Storage 146GB × 4本 (RAID 1+0)
- S/W
  - RHEL5.5, PostgreSQL9.1

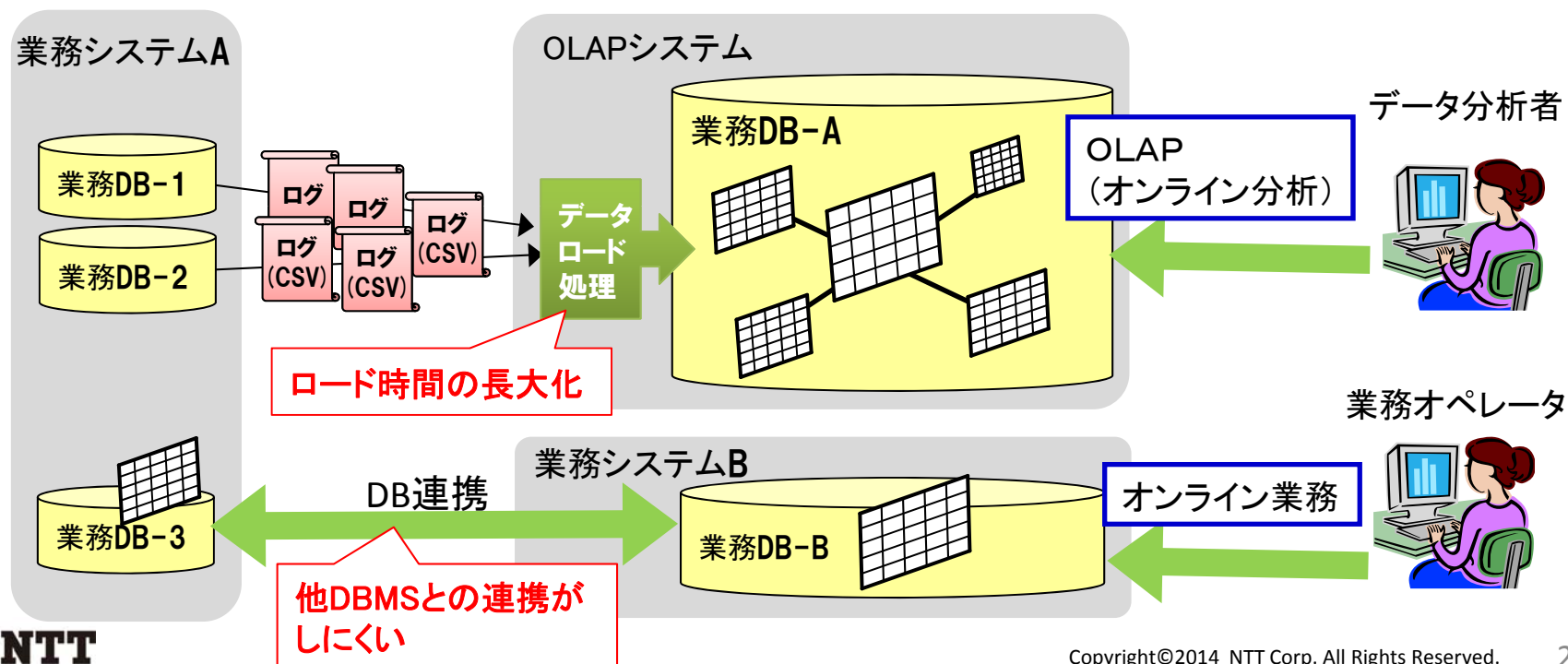
# PostgreSQL開発 - 外部データラッパー



## ◆ NTTが取り組んだ背景・課題

### 外部データとの関係

- ・ 販売履歴・応対履歴、課金履歴といった大量のログ情報を取り扱う  
→ RDBMSへのロード時間を短縮できないか…
- ・ 他システムとのDBMS連携を行う  
→ より柔軟、より簡単に他のDBMSと連携できないか…

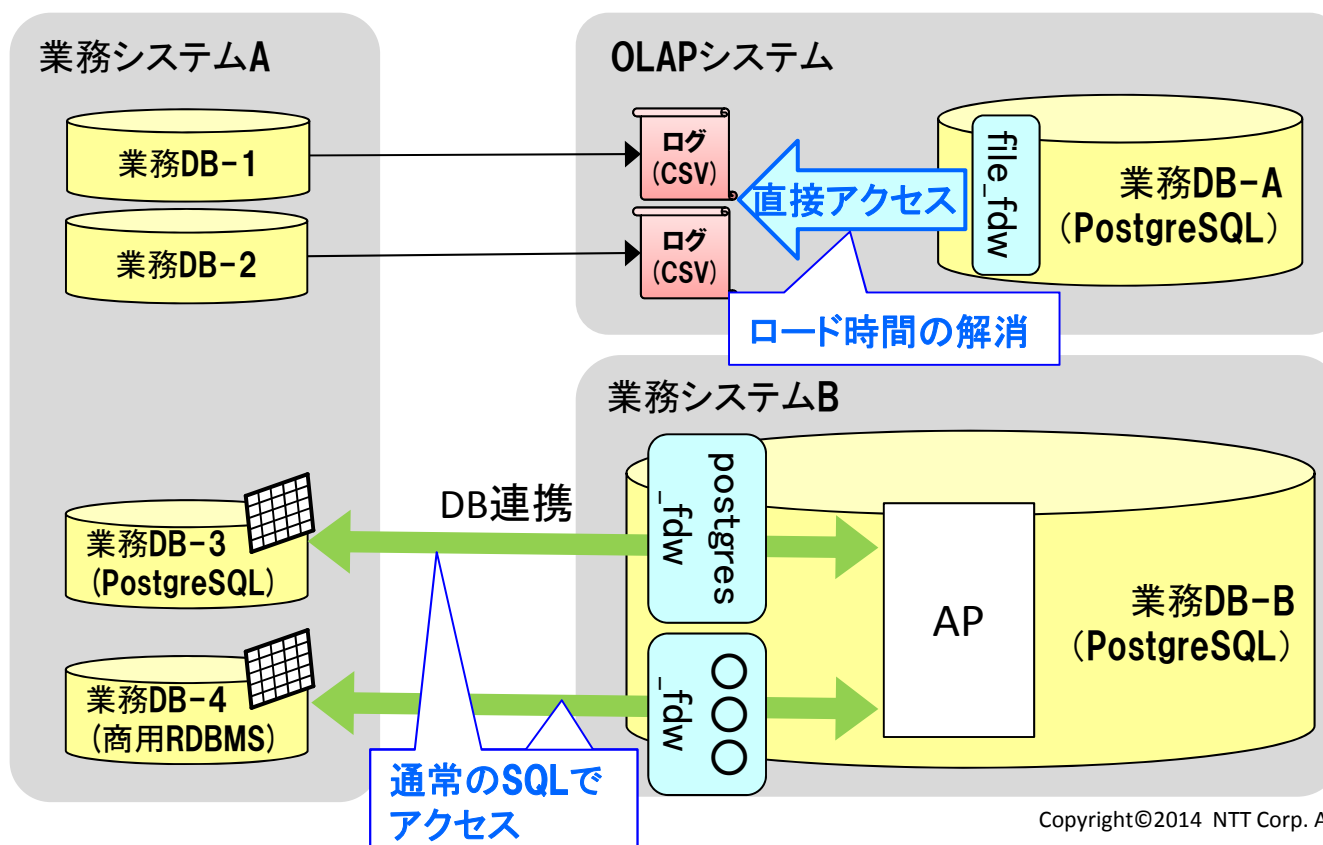


# PostgreSQL開発 - 外部データラッパー



## ◆ FDW ( Foreign data wrappers )

- 9.3 からcontribモジュールとして追加
- 外部データへアクセスする仕組み(テキストファイル、DBMSなど)
- SQL中で通常のテーブルと同じように記述できる



# PostgreSQL開発 - 移植の効率化

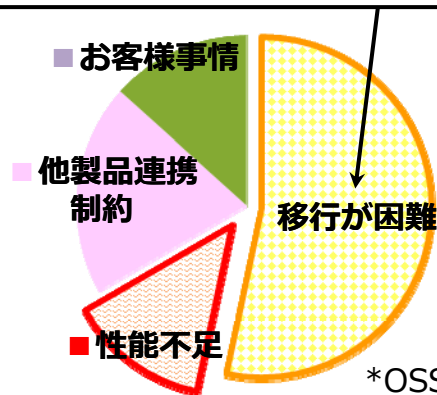
## ◆ NTTが取り組んだ背景・課題

### 移植コストの低減

- ・ システム開発の多くは更改案件（OSSセンタへの相談の **8割以上**）
- ・ 商用DBMSからPostgreSQLへの移行  
→ 既存APの流用率を高めることが重要なポイント

### PostgreSQL不採用理由(\*)

- 商用DBMS固有機能の移植コスト
- 短期間 & 高精度での見積りができない



## ★移行検討時にチェックするポイント

### 適用性

#### OSSがシステム要件への充足確認

- ・ 機能、性能、システム間連携
- ・ 運用期間、サポートレベル、・・・etc

### 移植性

#### 既存APの修正量や修正難易度を確認

- ・ 既存APの記述言語や流用規模
- ・ 非互換SQLの修正箇所、難易度、修正方法
- ・ 商用DBMSの独自実装機能

例) パーティショニング、PL/SQL、シノニム など

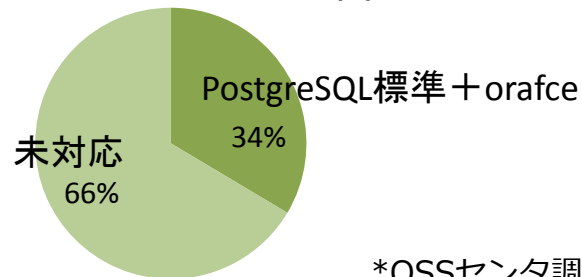
# PostgreSQL開発 - 移植の効率化



## ◆ orafce (<https://github.com/orafce>)

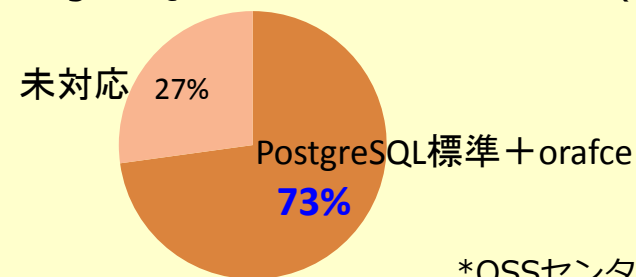
- Oracle に含まれる関数や機能を PostgreSQLで実行可能にするツール
- PGECons WG2の成果物に参考となるドキュメントが…  
→組み込み関数移行調査編 (<https://www.pgecons.org/downloads/62>)

PostgreSQL+orafce がサポートしている  
パッケージの関数カバー率(\*)



\*OSSセンタ調べ

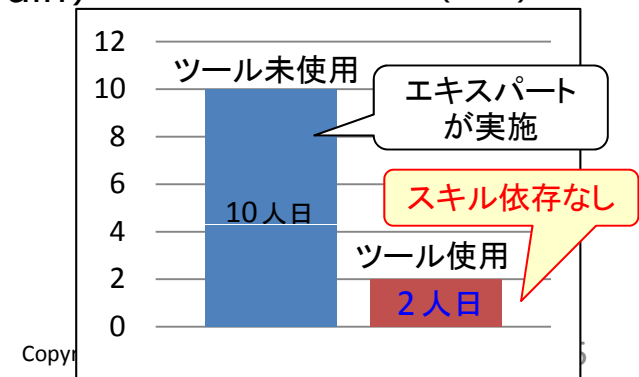
実案件 (Javaアプリ) での  
PostgreSQL+orafce の関数カバー率(\*)



\*OSSセンタ調べ

## ◆ db\_syntax\_diff (<https://github.com/db-syntax-diff>) 影響箇所調査稼働(人日)

- APをPostgreSQLへ移植する際の影響箇所を検出するツール
- 影響箇所抽出作業の短縮化
- 抽出の網羅性向上、品質の均一化



# PostgreSQL開発 - その他周辺ツール



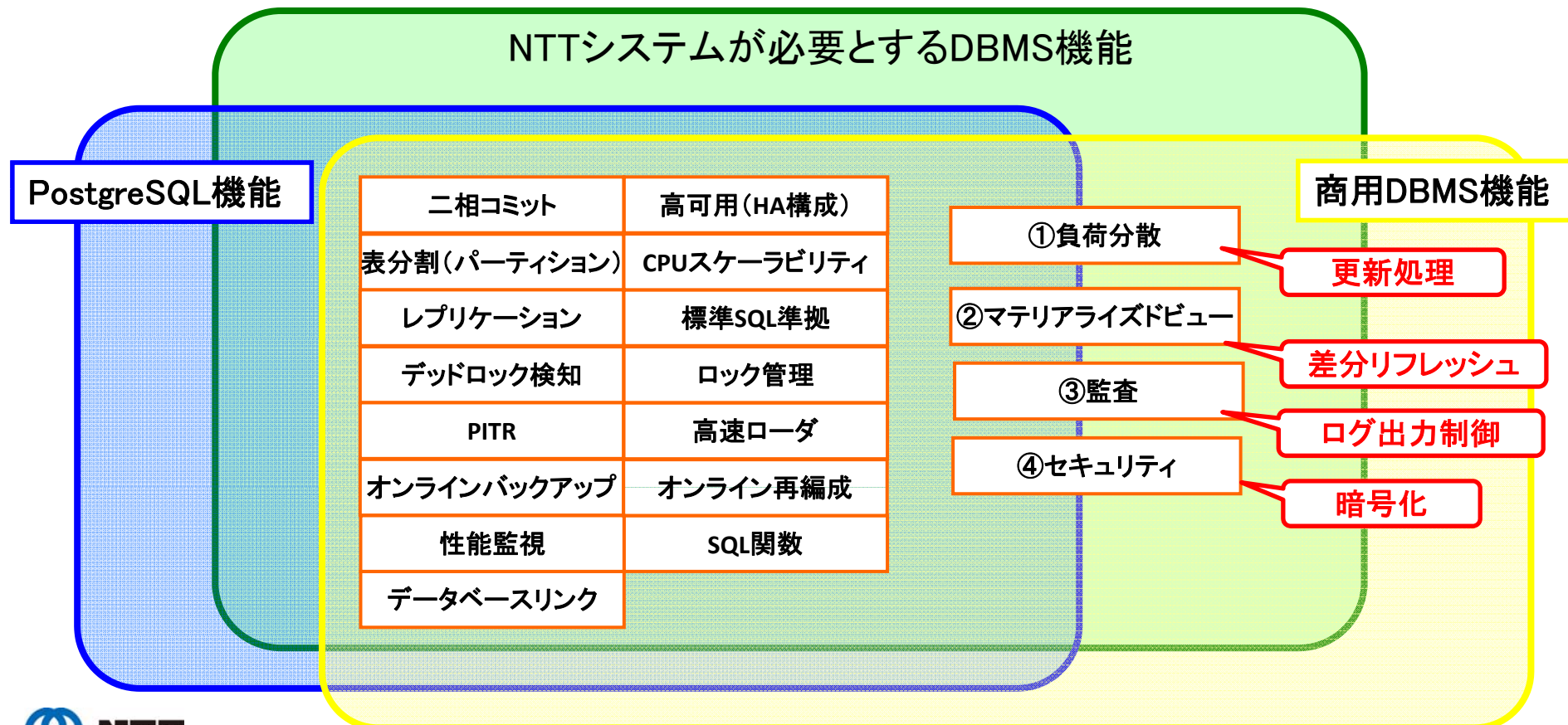
## ◆ その他、NTTが開発に関わったOSS（一部）

- pg\_statsinfo / pg\_stats\_reporter
  - 統計情報を定期的に収集・蓄積し、稼働状況をレポートに出力
    - [http://pgstatsinfo.projects.pgfoundry.org/pg\\_statsinfo-ja.html](http://pgstatsinfo.projects.pgfoundry.org/pg_statsinfo-ja.html)
- pg\_bulkload
  - 高速データロードユーティリティ
    - <http://pgbulkload.projects.postgresql.org>
- pg\_reorg
  - オンラインで PostgreSQL のテーブルを再編成するツール
    - <http://reorg.projects.pgfoundry.org>
- pg\_rman
  - バックアップやリストアを簡単に実行できるツール
    - <http://sourceforge.net/projects/pg-rman/>
- dblink\_plus
  - 他のデータベースへ接続するためのモジュール
    - <http://sourceforge.net/projects/interdbconnect/>

# NTTシステムとのFit & Gap

## ◆ NTTシステム要件とPostgreSQLの機能マッピング

- 機能面について、基本的にはPostgreSQLで問題なし
- ただし、①～④の要件がある場合は詳細な検討が必要！



## ■NTT OSSセンタの活動紹介

- ①OSS適用推進
- ②ソフトウェア基盤技術力向上
- ③OSSトータルサポート
- ④技術開発

## ■NTTグループにおける事例紹介

- ⑤PG-REX導入事例
- ⑥マイグレーション事例
- ⑦トラブルシューティング事例

# NTT グループにおける事例紹介



## ◆ PostgreSQL適用事例のご紹介

### ⑤ PG-REX導入事例

高可用システムへのPG-REX導入事例

### ⑥ OSSマイグレーション事例

レガシーシステムの更改案件における商用RDBMSから  
PostgreSQLへのマイグレーション事例

### ⑦ トラブルシューティング事例

PostgreSQLでよく陥るトラブル事例

## ■NTT OSSセンタの活動紹介

- ①OSS適用推進
- ②ソフトウェア基盤技術力向上
- ③OSSトータルサポート
- ④技術開発

## ■NTTグループにおける事例紹介

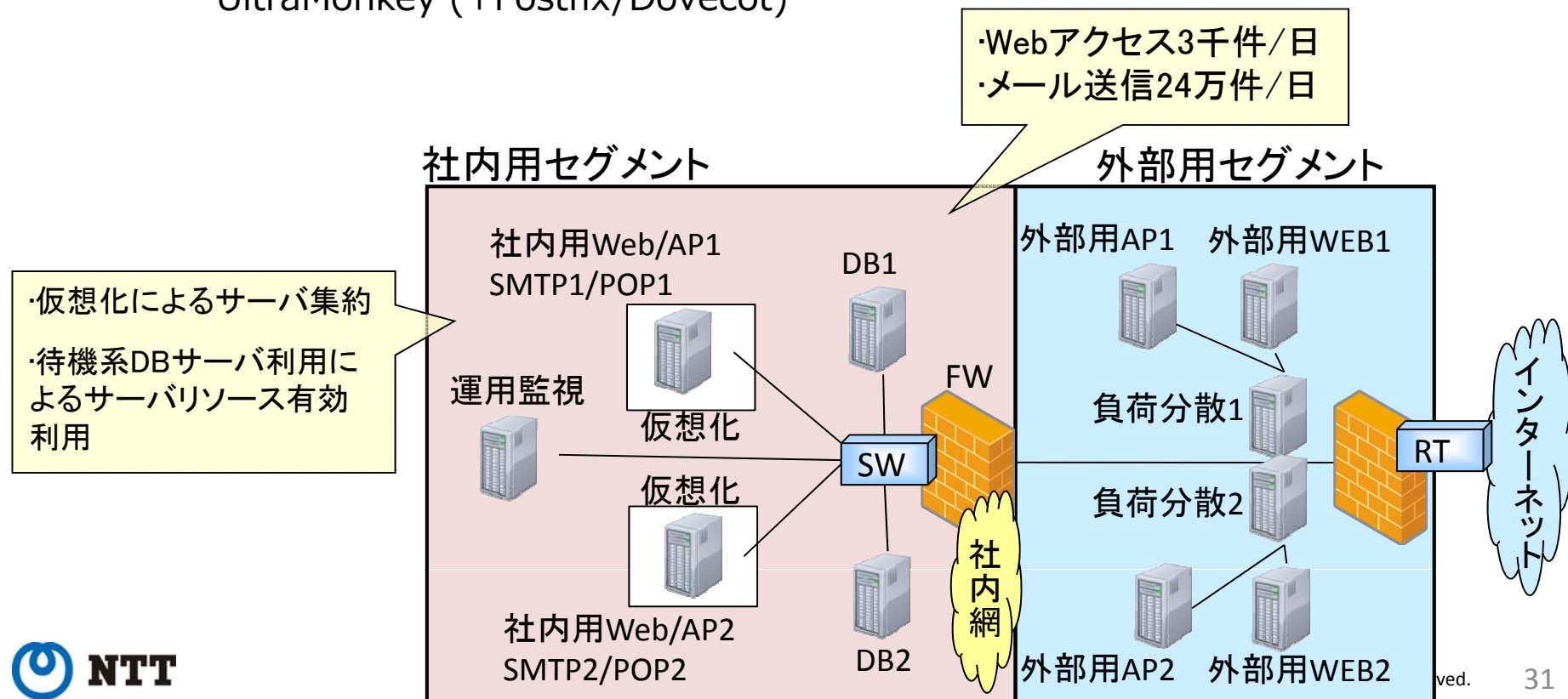
- ⑤**PG-REX導入事例**
- ⑥マイグレーション事例
- ⑦トラブルシューティング事例

# PG-REX 導入事例 概要

## ◆ NTT網情報を利用者に対してプロアクティブにメールやWebで情報配信するKシステム

➤ OSSVERTフル活用 → システム開発・運用コストを低減

- RHEL / KVM / Apache / mod\_jk / Tomcat / Pacemaker / **PostgreSQL** / UltraMonkey (+Postfix/Dovecot)



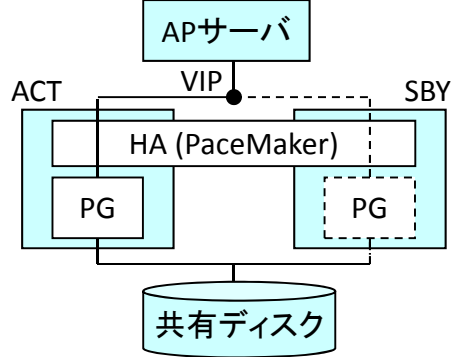
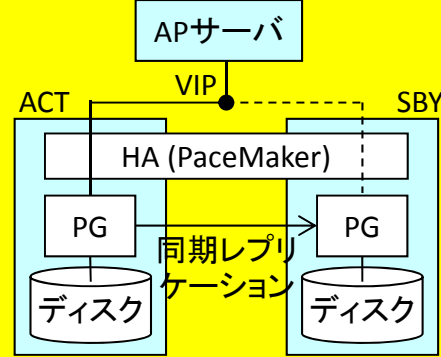
# PG-REX 導入事例 概要

## ◆ システム要件（DBに関わる部分のみ抜粋）

- 24H365日運用
- クラスタ構成によるサービス継続性の確保
- サービス停止時間（クラスタが切り替わる時間）の最小化

## ◆ DBサーバ構成 メリット・デメリット比較

**採用！**

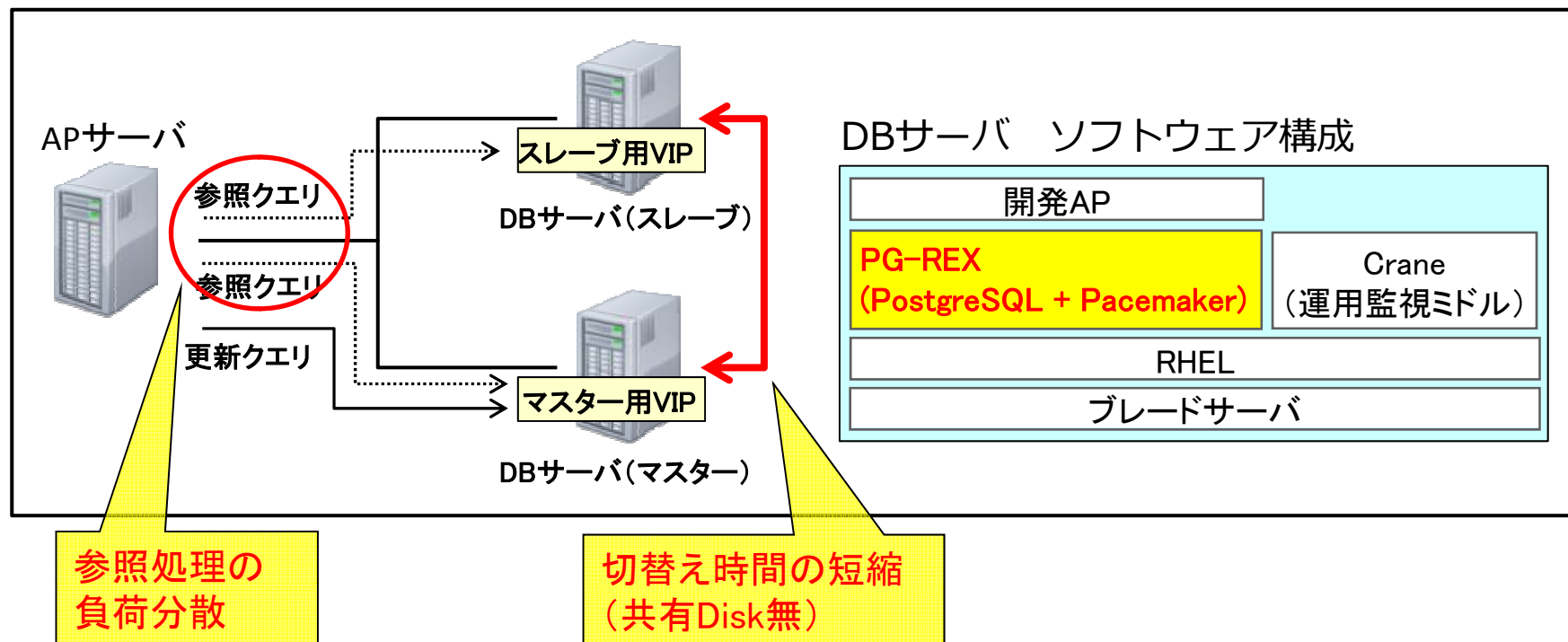
| 方式         | 【案1】HAクラスタ(共有ディスク)                                                                  | 【案2】PG-REX                                                                           |
|------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 概略図        |  |  |
| データ保全性     | ○: データ損失なし(単一障害)                                                                    | ○: データ損失なし(単一障害)                                                                     |
| フェイルオーバー時間 | ○: 数十秒～数分(リブート時間)                                                                   | ◎: 10～60秒程度(検知後即座に切替え)                                                               |
| 復旧手順       | ◎: 容易                                                                               | ○: スレーブの再構築(手順は確立済み)                                                                 |
| 性能         | ◎: オーバヘッド微小                                                                         | △: 単体時より10～20%ダウン                                                                    |
| サポート、実績    | ◎: 過去に多数の実績あり                                                                       | ○                                                                                    |



# PG-REX 導入事例 DBサーバ構成

## ◆ 本システムで採用したDBサーバ構成

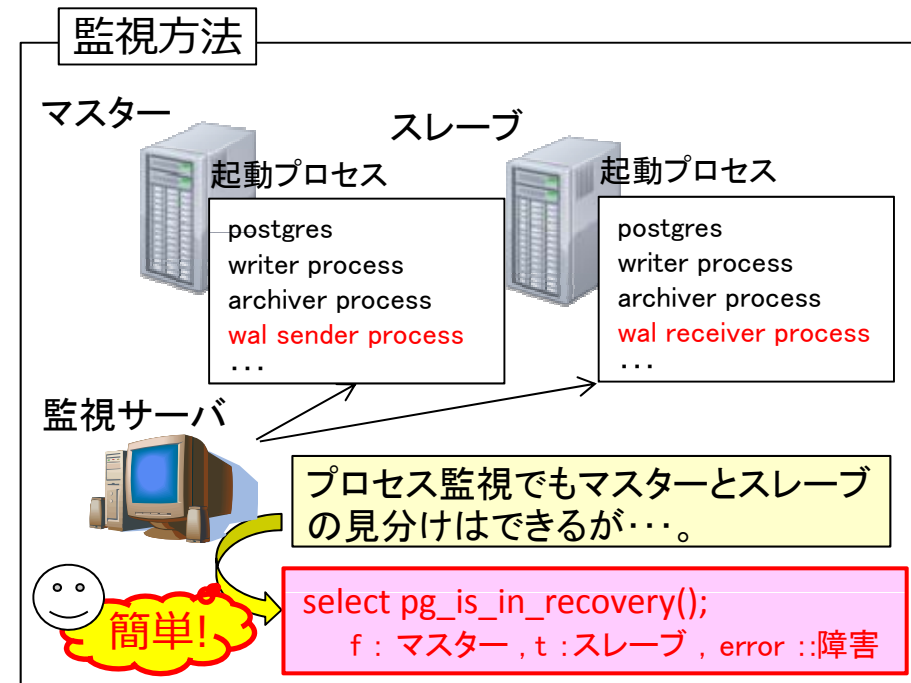
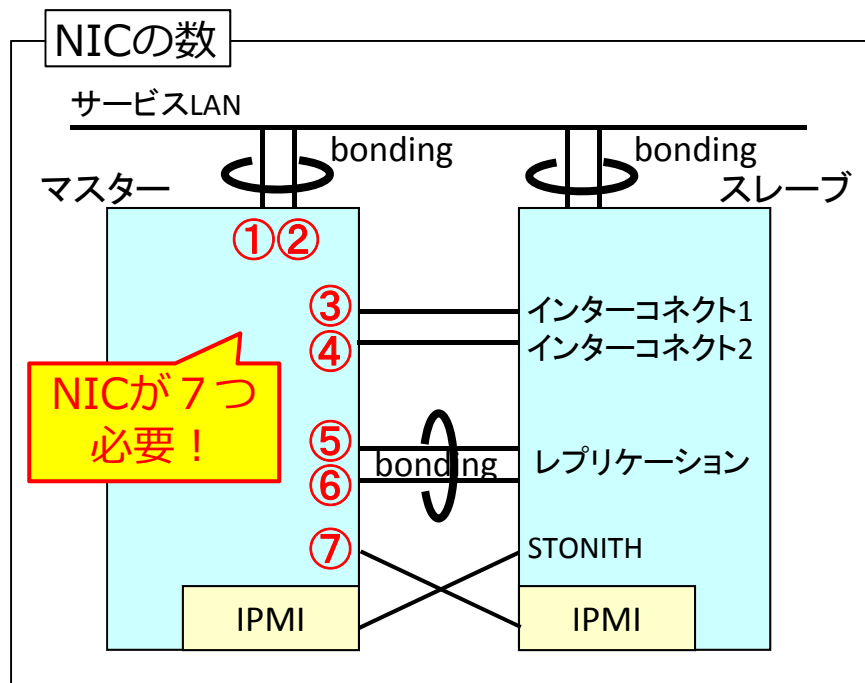
- DBサーバは性能を考慮し、物理サーバ上に構築
- PG-REXによるマスタースレーブ構成
  - ただし、スレーブ側も参照クエリの負荷分散先として利用



# PG-REX 導入事例 留意事項

## ◆ PG-REX導入時の設計や運用で注意すべきポイント

- NICの数
- レプリケーション再構築手順の確立
  - ・スレーブサーバ故障、スレーブサーバのマスター昇格からの切り戻しなど
- 監視方法



## ■NTT OSSセンタの活動紹介

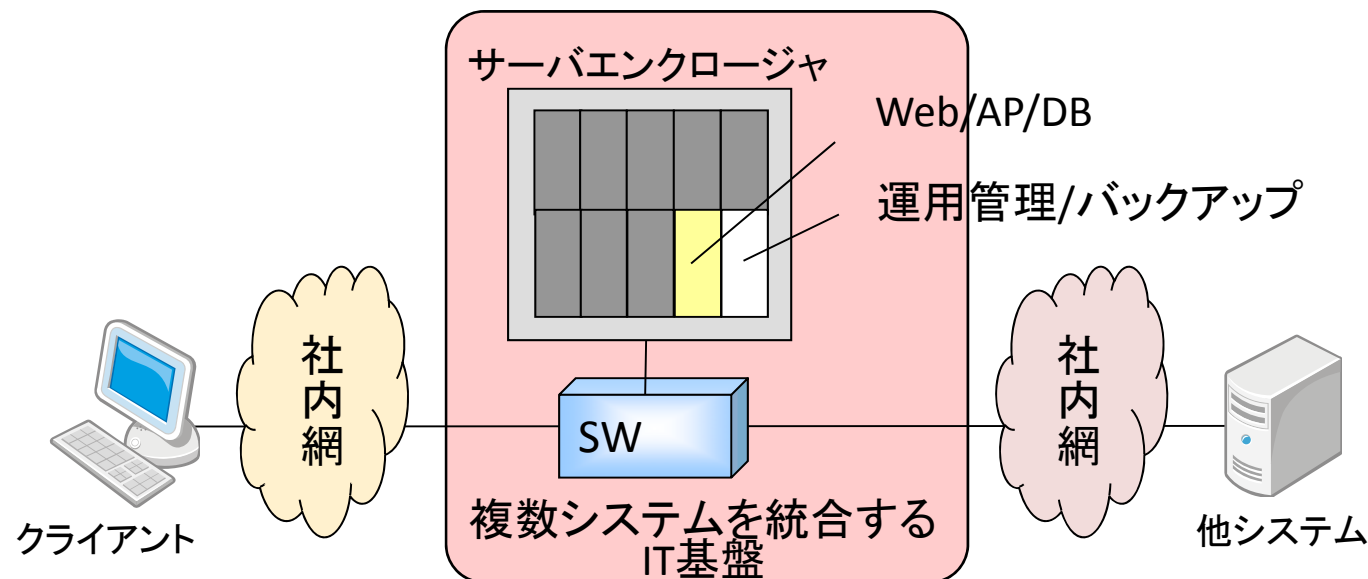
- ①OSS適用推進
- ②ソフトウェア基盤技術力向上
- ③OSSトータルサポート
- ④技術開発

## ■NTTグループにおける事例紹介

- ⑤PG-REX導入事例
- ⑥マイグレーション事例**
- ⑦トラブルシューティング事例

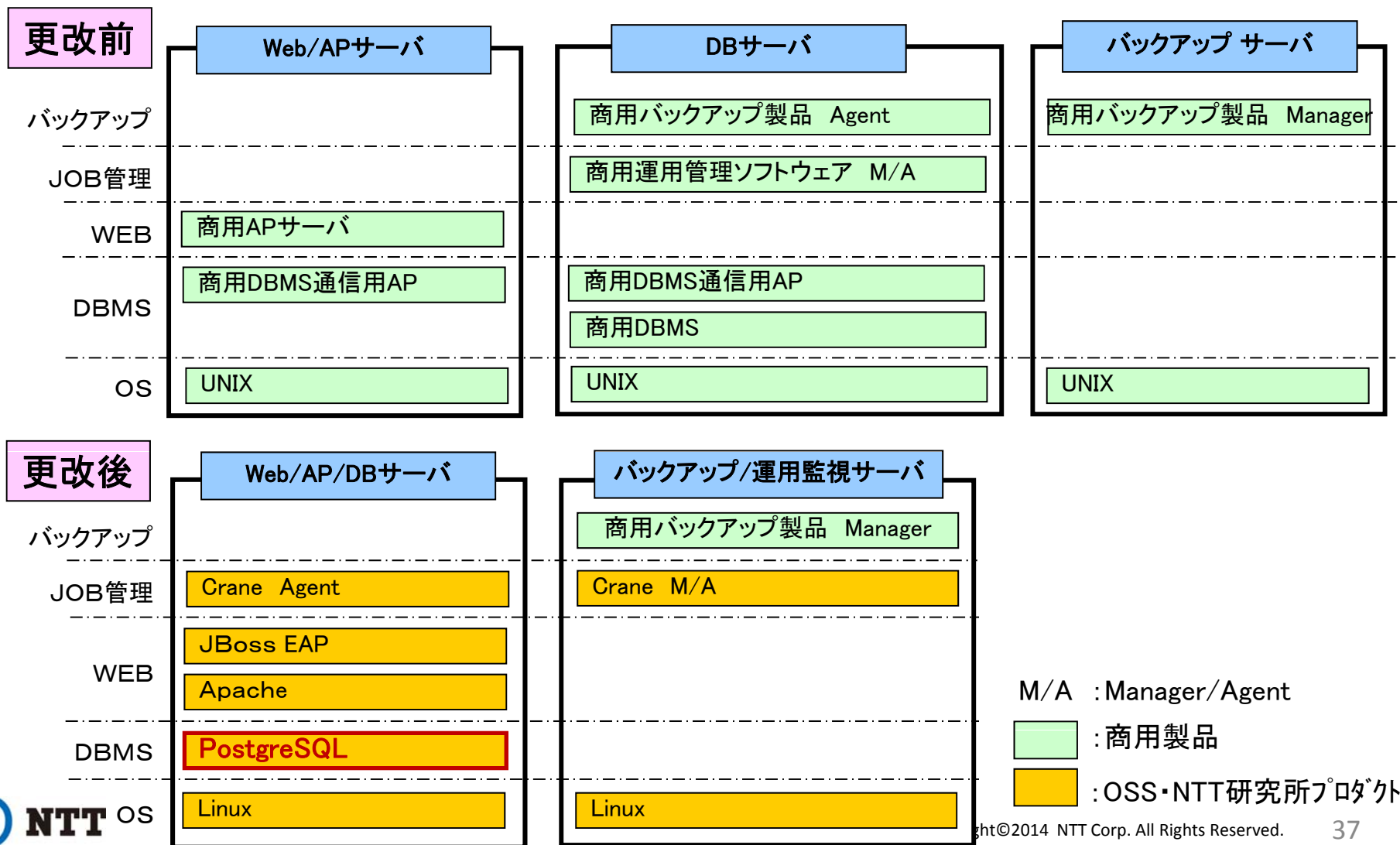
# OSSマイグレーション事例 概要

- ◆ 特定サービスに関する契約情報(申込情報や顧客情報等)を管理して営業を支援するMシステム
  - 商用製品からRHEL/Apache/mod\_jk/JBoss EAP/**PostgreSQL**に移行し、システム開発・運用コストを低減



# OSSマイグレーション事例 概要

## ◆ 更改前後のソフトウェア構成

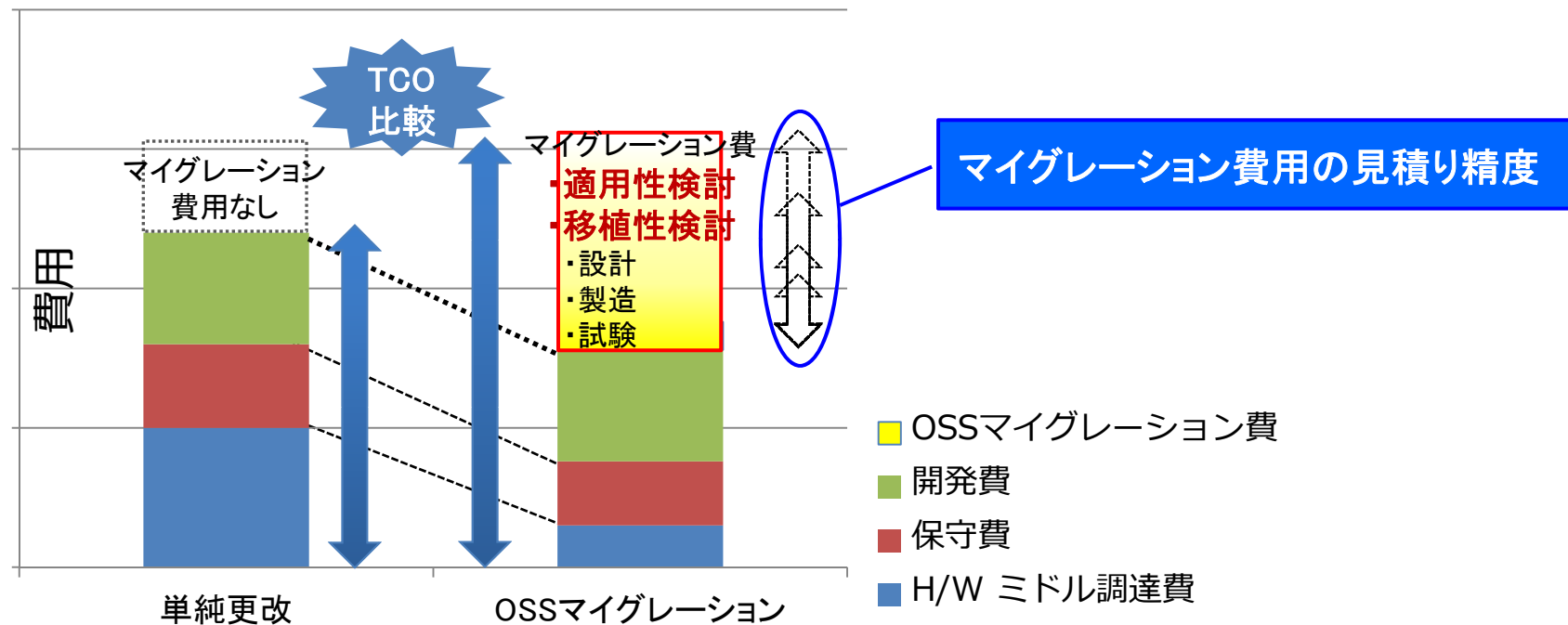


# OSSマイグレーション事例 概要

## ◆ マイグレーション実施可否は、最終的にはTCO比較で判断

- 適用性検討 : 機能、性能に関する充足性の確認
- 移植性調査 : マイグレーションによる影響範囲・修正難易度の確認

### 単純更改・OSSマイグレーション 費用比較



# OSSマイグレーション事例 適用性



## ◆ 適用性検討におけるDBMS課題

- 既存システムではパーティショニングを利用
- オンライン・バッチ性能要件を満足出来るかが不安...

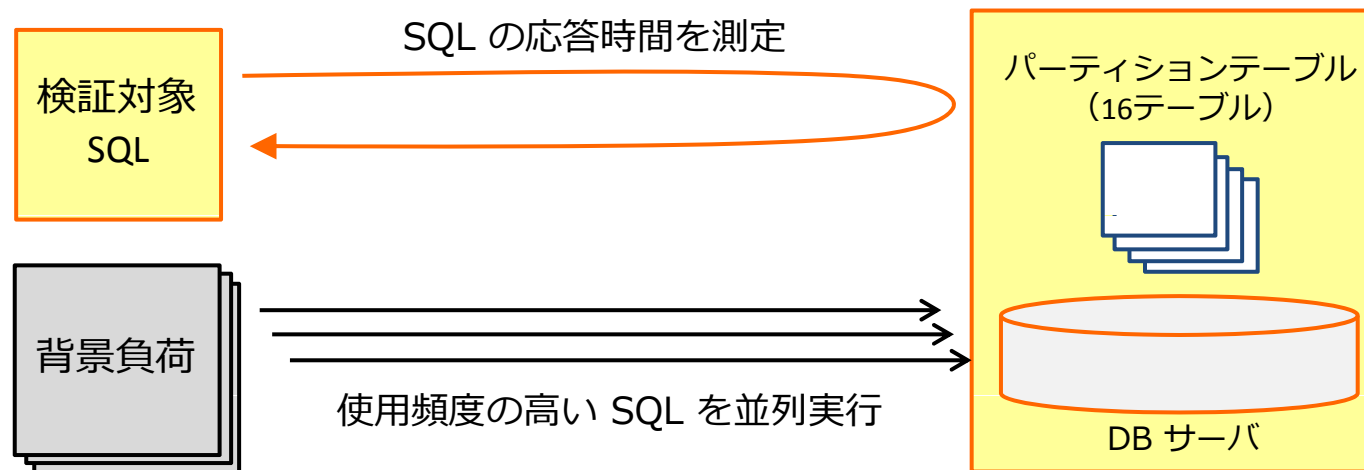
### 実機検証

#### ■ 評価観点

オンライン業務 SQL およびバッチ SQL の応答時間が、目標値を下回ること

#### ■ 検証条件

繁忙期と同程度の背景負荷上で、オンライン・バッチSQLを発行する.



# OSSマイグレーション事例 適用性



## ◆ 実機検証結果

- PostgreSQLで目標性能を満足できることが確認できた！
- パーティショニングも不要！！

| No | 業務分類  | 対象業務 | 性能目標値     | 結果<br>(パーティション有) | 結果<br>(パーティション無) |
|----|-------|------|-----------|------------------|------------------|
| 1  | オンライン | A業務  | 4.9 sec以下 | ○ 4.69 sec       | ○ 2.98 sec       |
| 2  | オンライン | B業務  | 0.1 sec以下 | ○ 0.095 sec      | ○ 0.08 sec       |
| 3  | オンライン | C業務  | 3.8 sec以下 | ○ 3.20 sec       | ○ 0.86 sec       |
| 4  | バッチ   | E業務  | 850 sec以下 | ○ 787 sec        | ○ 767 sec        |

### 検証環境条件

#### ● H/W

- ・CPU Intel Xeon 2.66GHz × 4core
- ・MEM 24GB
- ・Storage 146GB × 12本 (VRAID10)

#### ● S/W

- ・RHEL6.2
- ・PostgreSQL9.1 (パーティショニング数16)

パーティショニング無しの方が性能が良いSQLが存在している。

分割キー以外の列を条件としてパーティションをまたがる検索を行っているSQLであった。

# OSSマイグレーション事例 移植性

## ◆ 移植性検討におけるDBMS課題

- 開発工数が想定よりも増大することによるリスク
- 既存APの修正範囲の大きさや修正難易度、対処方法に不安が...

## ◆ 移植性調査結果 その1

- 母体規模の4%弱の修正でDBMSの移植が可能！

### 修正規模レポート

| 調査箇所    | ソースファイル数 | ソース行数  | 修正箇所数 | 目視確認箇所数 | 修正ユニーク行数 |
|---------|----------|--------|-------|---------|----------|
| SQL/DDL | 約740個    | 約250kL | 約11kL | 約5kL    | 約9kL     |
| Java    |          |        |       |         |          |
| Pro*C   |          |        |       |         |          |

母体規模

db\_systax\_diffで  
算出

母体規模の4%弱

# OSSマイグレーション事例 移植性

## ◆ 移植性調査結果 その2

- 移植難易度も、全体の **99%** が低1、低2
- 調査結果その1 と合わせ、AP移行に課題がないことが確認できた！

### 修正難易度レポート

| # | 難易度 | SQL/DDL | Java | Pro*C |
|---|-----|---------|------|-------|
| 1 | 低1  | 4,186   | 1    | 11    |
| 2 | 低2  | 5,361   | 798  | 94    |
| 3 | 中   | 20      | 0    | 0     |
| 4 | 高   | 0       | 0    | 0     |
|   | 計   | 9,567   | 799  | 105   |

全体の99%

修正例（本事例で最も多いもの）

- ・ VARCHAR2 → VARCHAR に置き換え
- ・ ただし、バイト数と文字数の差異あり

- ・ 外部結合演算子 (+) → LEFT JOIN, RIGHT JOIN, FULL JOIN に置き換え

- ・ rtrim関数における空白文字の扱いに差異があり、単純置き換えができない

#### 難易度のレベル

- ・ 低1: 機械的に直せるもの
- ・ 低2: 機械的修正は出来ないが修正がきわめて容易なもの
- ・ 中: 機械的修正が出来ず、周辺処理を確認しながら修正が必要なもの
- ・ 高: 機械的修正が出来ず、PostgreSQLに代替機能がないために検討が必要なもの

## ■NTT OSSセンタの活動紹介

- ①OSS適用推進
- ②ソフトウェア基盤技術力向上
- ③OSSトータルサポート
- ④技術開発

## ■NTTグループにおける事例紹介

- ⑤PG-REX導入事例
- ⑥マイグレーション事例
- ⑦トラブルシューティング事例**

# トラブルシューティング事例



我々が数多くの障害解析を実施して行く中で、よく見かける PostgreSQLに関わるトラブルを事例としてご紹介します。

## ◆運用中における突然のSQL性能低下

運用中に突如、PostgreSQLの性能低下が発生したトラブル事例をご紹介します。

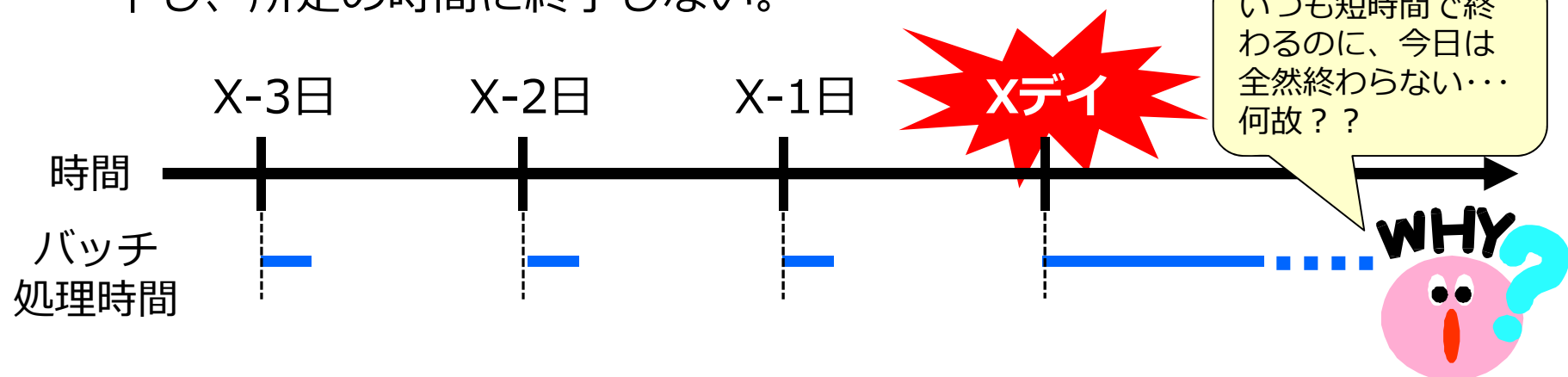
PostgreSQLトラブルの未然防止や早期解決の糸口として活用頂ければ幸いです。

# 運用中における突然のSQL性能低下



## ◆ トラブル内容

- 毎日、あるテーブルの全件レコードを処理するバッチが、突然性能低下し、所定の時間に終了しない。



## ◆ トラブル原因（キーワード）

- ロングトランザクション
  - 長時間 Commit/Rollback がなされないトランザクション
- MVCC、VACUUM
  - MultiVersion Concurrency Control: 多版型同時実行制御
  - 不要領域の回収（ゴミレコードの再利用化）

# 運用中における突然のSQL性能低下

## ◆ トラブル発生メカニズム

テーブル

|    | 追加XID | 削除・更新XID | 主キー | データ |
|----|-------|----------|-----|-----|
| 追加 | 10    |          | 1   | 北海道 |
| 更新 | 20    | 25       | 2   | 東京  |
|    | 25    |          | 2   | 東京西 |
| 削除 | 30    | 35       | 3   | 名古屋 |
| 削除 | 40    | 45       | 4   | 大阪  |
| 追加 | 50    |          | 5   | 広島  |
| 更新 | 60    | 65       | 6   | 福岡  |
|    | 65    |          | 6   | 福岡南 |
| 追加 | 70    |          | 7   | 鹿児島 |
|    |       |          | ⋮   |     |
| 削除 | 80    |          | △△  | 沖縄  |

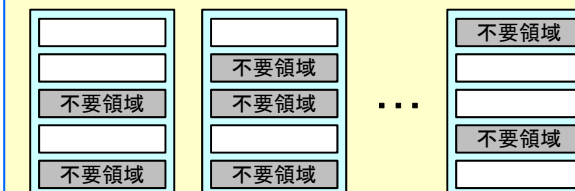
: 有効レコード  
 : 不要レコード

ロングトランザクション

TR-1

TR-2

### PostgreSQLの物理ページ



ページ単位辺りの有効行が減少  
→ ページ数が増大

AutoVacuum

テーブル肥大化!!

ロングトランザクション  
のため回収不可

業務バッチ

処理時間 増大

全件検索処理  
シーケンシャルスキャン

# 運用中における突然のSQL性能低下



## ◆ 対処方法

### ① 障害事象の確認

- 対象テーブルにVACUUM VERBOSEを行い、以下のメッセージが出るかどうか確認する。

```
DETAIL: 200004 dead row versions cannot be removed yet.
```

### ② ロングトランザクションの確認

- ロングトランザクションの存在を xact\_start の時刻で確認する。

```
SELECT * FROM pg_stat_activity ORDER BY xact_start;
```

### ③ ロングトランザクションを終了させる

- クライアント強制終了, pg\_terminate\_backend()関数, killコマンド

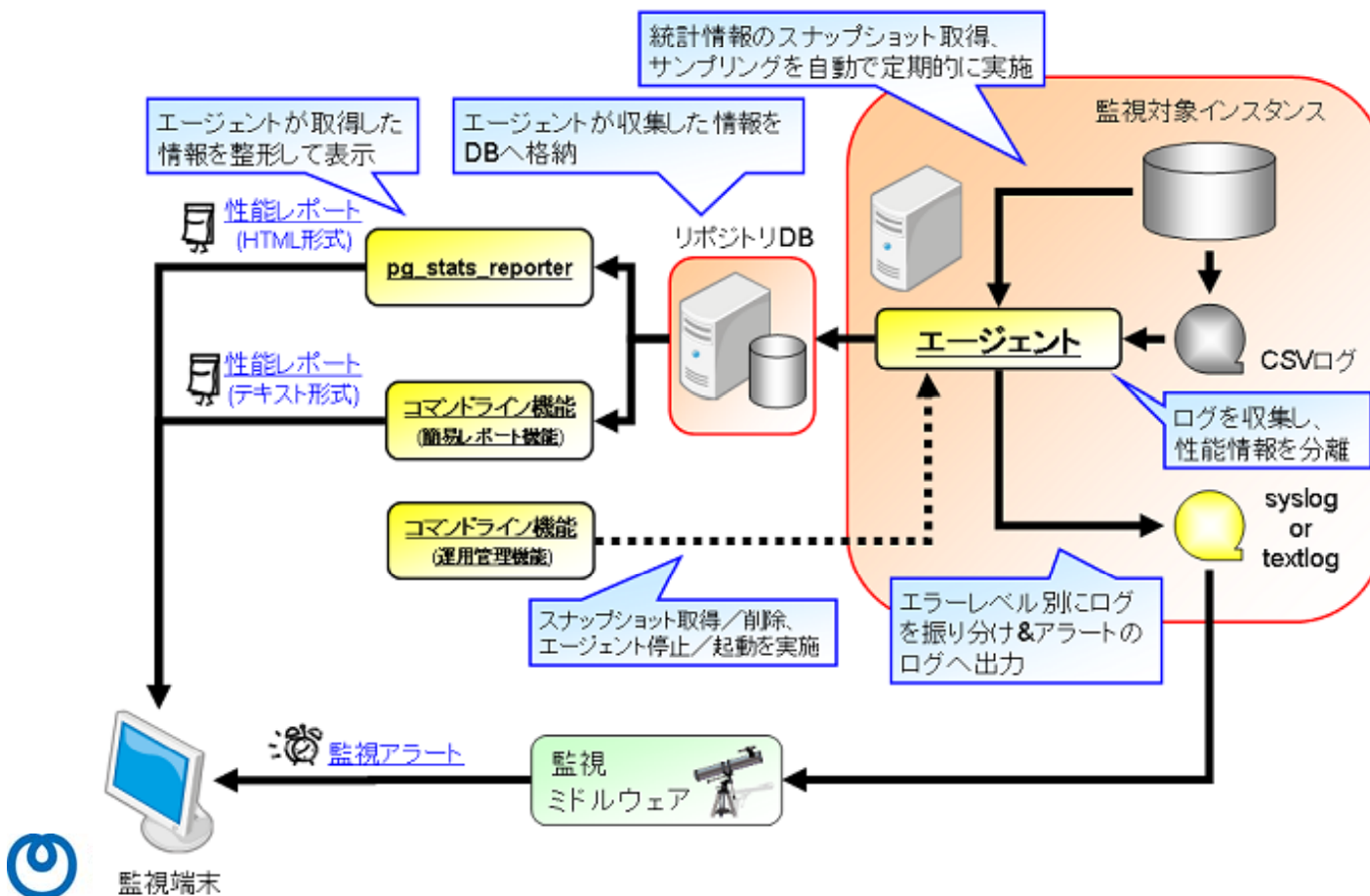
### ④ テーブル, インデックスの再編成（必要に応じて）

- ファイルが肥大化している場合は再編成を実施

# 運用中における突然のSQL性能低下

## ◆ 防止策（ロングトランザクション検知）

- pg\_statsinfo / pg\_stats\_reporter
  - PostgreSQLやOSの統計情報を定期的に自動で取得します。



# 運用中における突然のSQL性能低下



## ◆ 防止策（ロングトランザクション検知）

- pg\_statsinfo / pg\_stats\_reporter
  - PostgreSQLやOSの統計情報を定期的に自動で取得します。

統計情報のスナップショット取得。

pg\_stats\_reporter 2.0.0

性能レポート (HTML形式)

性能レポート (テキスト形式)

監視アラート

監視端末

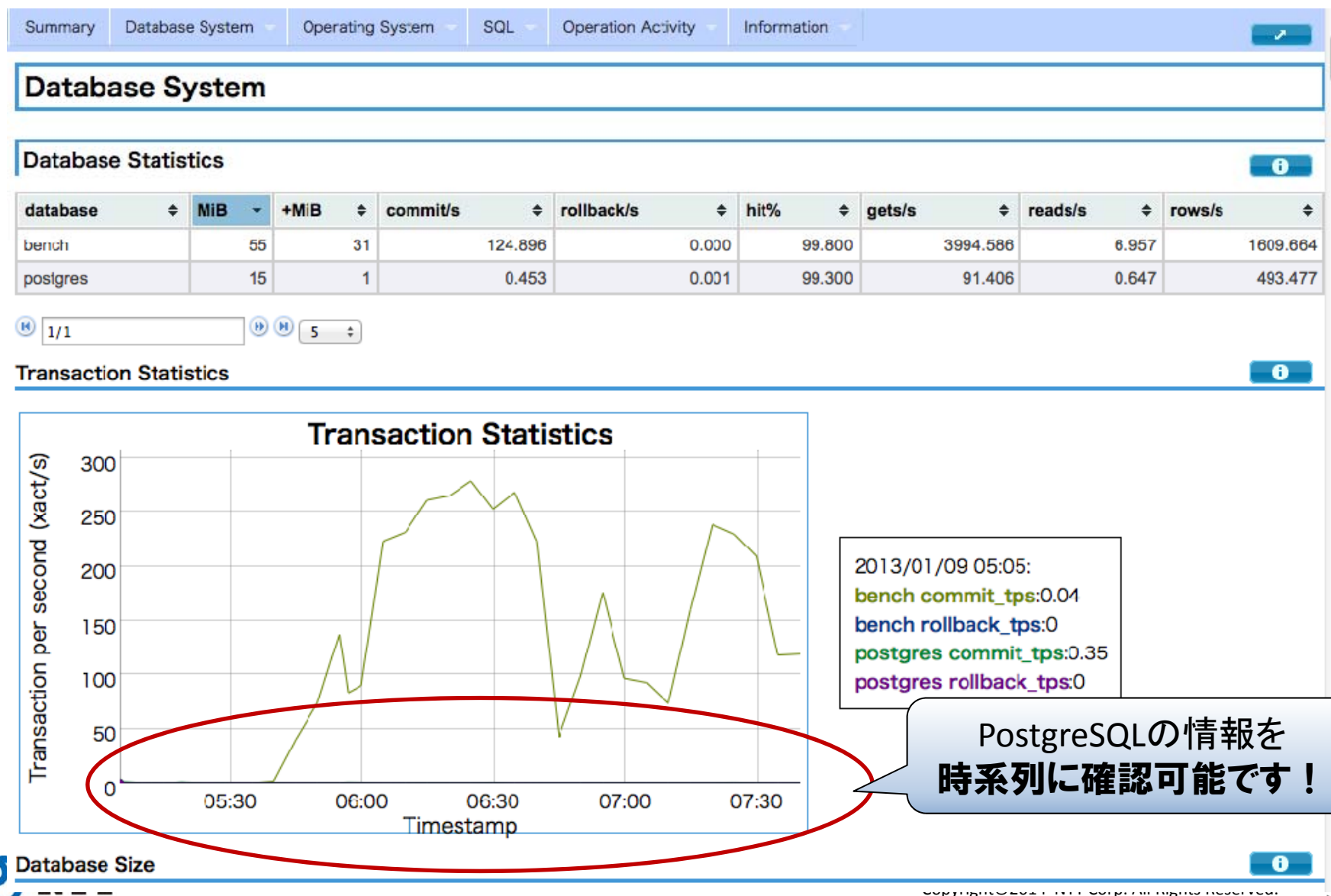
ロングトランザクションを捕捉  
アラートとしてログ出力！

Long Transactions

| pid  | client address | when to start       | duration (sec) | query                                                                       |
|------|----------------|---------------------|----------------|-----------------------------------------------------------------------------|
| 9199 |                | 2014-09-10 11:24:58 | 46.565         | autovacuum: ANALYZE public.pgbench_accounts                                 |
| 9262 |                | 2014-09-10 11:31:41 | 44.079         | autovacuum: ANALYZE public.pgbench_accounts                                 |
| 9155 |                | 2014-09-10 11:19:47 | 2.537          | UPDATE pgbench_accounts SET abalance = abalance + -2938 WHERE aid = 358856; |

ITT Corp. All Rights Reserved. 49

# (参考)pg\_stats\_reporter



# トラブルシューティング手法



## ◆ 他のDBMSと同じような切り分け手法でOK！

### ➤ 解析に必要な情報の収集

- 障害事象の確認
- PostgreSQLのバージョン（JDBCなども）
- PostgreSQLの設定ファイル
- ログ
- 実行計画、システム構成、スキーマ情報・・・など

### ➤ 障害発生前後におけるDBMSの挙動

- 発生前に何か作業を行っていないか？。
- pg\_statsinfo/pg\_stats\_reporter であれば、発生前後の差異を見つけることができる・・・かも

# 本日のまとめ

# 本日のまとめ

## ◆ OSSセンタの活動紹介

- サポートサービスからみたPostgreSQLの品質 …… OK !
  - バグ起因のトラブルが少なく、非常に安定した運用
- エンタープライズ用途でのPostgreSQLの機能 …… OK !
  - バージョンアップ毎に着実に進化
- 商用DBMSからPostgreSQLに移行する …… OK !
  - AP移植ツール

## ◆ NTTグループにおける事例紹介

- PG-REX導入やPostgreSQL移行も実績あり！
- トラブルも怖くない！？
  - PostgreSQLに合わせた設計、試験
  - 運用を楽にするツールも活用

最後に… ぜひ一度PostgreSQLに触ってみてください！

**皆様の今後のPostgreSQL利用の  
参考になれば幸いです。**

**ご清聴ありがとうございました。**