PGECons
PostgreSQL Enterprise Consortium

Introducing PostgreSQL Enterprise Consortium activities

Hitoshi Hemmi / NIPPON TELEGRAPH AND TELEPHONE CORPORATION

Tatsuo Ishii / SRA OSS, Inc. Japan
Executive board member of PostgreSQL Enterprise Consortium

Abstract

- PostgreSQL Enterprise Consortium（PGEcons） is organized by leading IT/OSS companies in Japan last year aiming at promoting PostgreSQL in production use, especially in mission critical area.
- Two technical working groups（WG1/WG2） published and shared first collaborative achievements with PostgreSQL users and PGECons members.
  - WG1：Focuses on performance characteristics of PostgreSQL and replication/clustering software
  - WG2：Migration from other DBMSs to PostgreSQL
- Currently PGECons has 39 company members.

FUJITSU
HITACHI Inspire the Next
Empowered by Innovation NEC
hp
NTT
SRA OSS, INC.
SIOS
TIS
FUJITSU 富士通ソーシアルサイエンスラボラトリ
HITACHI Inspire the Next Hitachi Solutions
NEC NEC Soft, Ltd.
LPI-JAPAN OSS-DB
NTT SOFT
アシスト
IIM
QUALICA Inc.

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Major activities of PGECons

- **Collaborative verification**
  - PGECons performs necessary verification collaboratively by using resources provided by our member companies if enough information to apply PostgreSQL to production use in enterprise area is not available
- **Promoting PostgreSQL**
  - Through seminars PGECons presents technical reports created by the activities above. Also PGECons provides various case studies, which are important for those who are trying to adopt PostgreSQL.
- http://www.pgecons.org/en/about

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

WG1 activity themes (excerpt)

- We chose "scale up" and "scale out" as the first fiscal year's theme out of other domains of interest

PGECons is interested in following themes

Performance	Performance evaluation methods, performance enhancing, Database tuning
High availability	High availability clusters, BCP
Maintainability	Maintenance support, traceability
Serviceability	Monitoring, backing up
Security	Audit
Compatibility	Data, Schema, SQL, stored procedures
Connectivity	Connectivity to other software

Performance verification theme

Performance evaluation method	Performance model and sizing model for on-line or batch jobs
Scale up	Scale up characteristics on multi core CPU
Scale out	Scale out characteristics on load balance clusters
Performance enhancing	Query cache, partitioning, fast load etc.
Performance tuning	Performance tuning know-how, query planner control

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

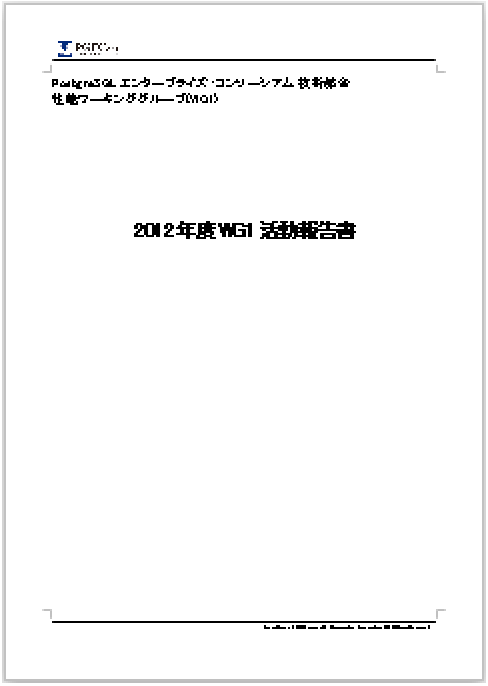
Themes chosen by WG1

- Scale up performance evaluation
  - Performance characteristics on many-core machine
  - How PostgreSQL 9.2 performs well?
    - Read query benchmark using pgbench
    - TCP-C like benchmark using JDBCrunner
- Scale out performance evaluation
  - We chose following OSS cluster/replication systems
    - PostgreSQL 9.2 cascading replication
      - Asynchronous replication
    - pgpool-II (replication mode) + PostgreSQL 9.2
      - Synchronous replication + read query load balance
    - Postgres-XC
      - Synchronous data distribution + write query load balance

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

First fiscal year's achievements

- Documents describing scale up, scale out evaluation steps and results
- Detailed description on hardware, software, deployment and results are public
- Documents and scripts can be copied distributed under CCL (Creative Commons License)
- Documents are published as over 70-page document titled "WG1 activity report in 2012" in April 2013



PostgreSQL エンタープライズ・コンソーシアム 技術部会
性能ワーキンググループ(WG1)

2012年度WG1 活動報告書

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

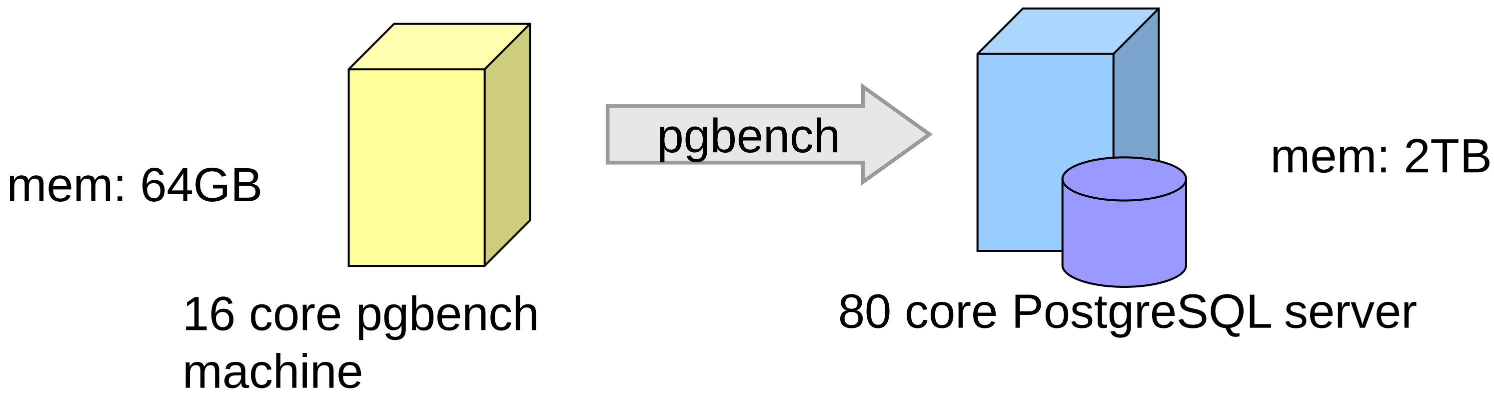
Report 1:Scale up performance evaluation

- 1.1 Read query scale up performance evaluation by using pgbench
  - PostgreSQL read query performance evaluated on 80-cores machine
  - We confirmed that PostgreSQL scales up to 80 clients

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Evaluation details

- Increase number of concurrent clients on many core machine
- pgbench -h [host] -p [port] [dbname] -c [c] -j [j] -T 30 -n -f custom.sql
  - Increase number of clients(-c)
  - Number of threads is ½ of -c
  - dedicated machine to run pgbench is prepared



Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

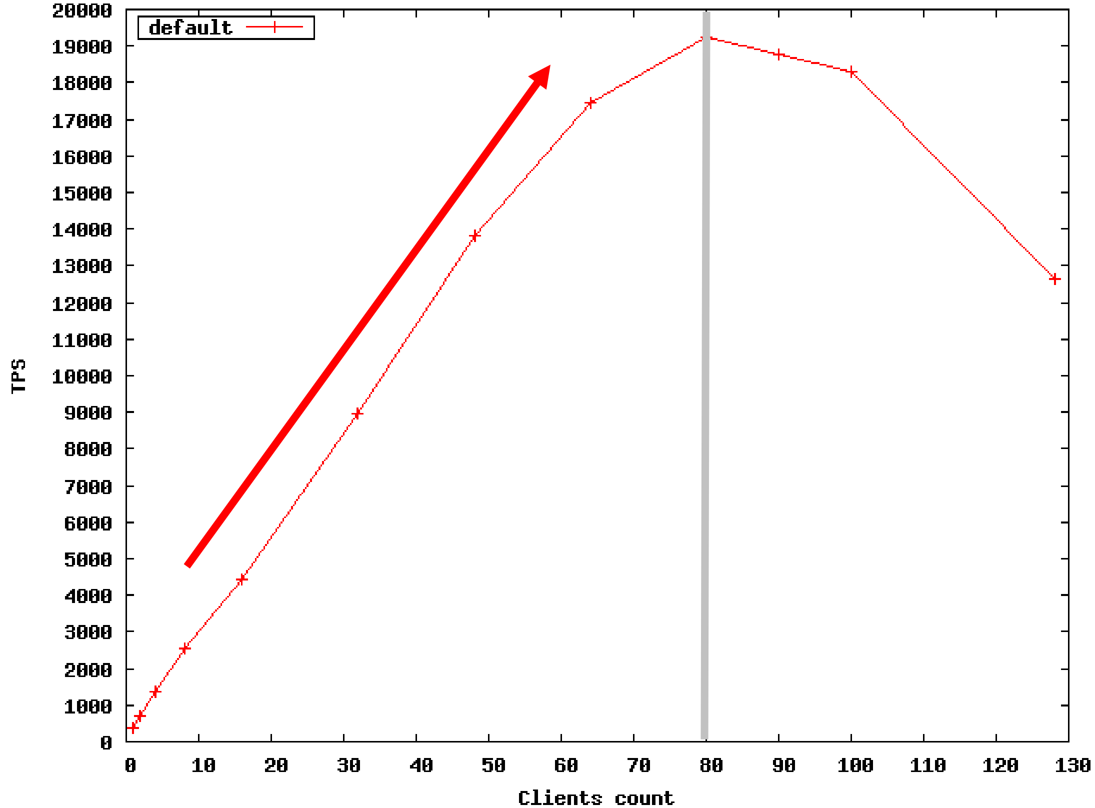
Custom.sql

```
¥set nbranches :scale
¥set ntellers 10 * :scale
¥set naccounts 100000 * :scale
¥set row_count 10000
¥set aid_max :naccounts - :row_count
¥setrandom aid 1 :aid_max

SELECT count(abalance) FROM pgbench_accounts WHERE aid
BETWEEN :aid and :aid + :row_count;
```

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

The result!



Scale factor: 1000
Number of rows:
a hundred million(15GB)

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Report 1:Scale up performance evaluation

1.2 Scale up evaluation using JdbcRunner

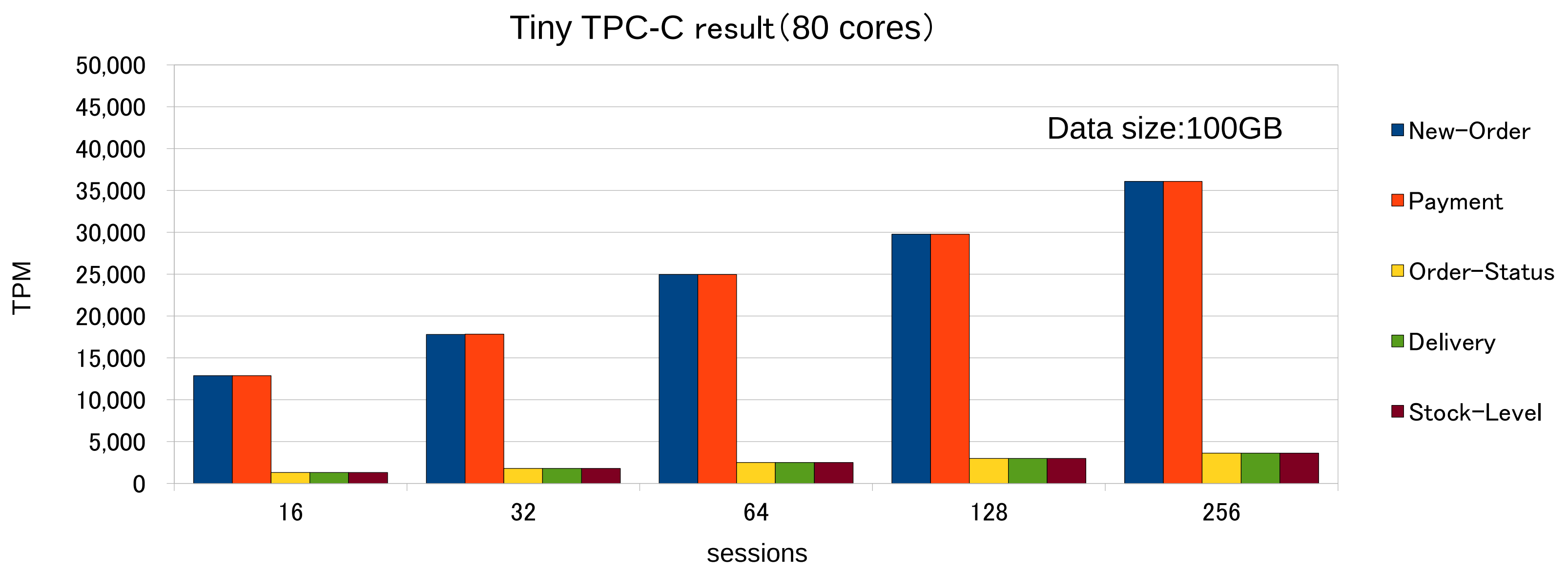
- We confirmed that the performance goes up as the number of sessions increased
- About JdbcRunner

Summary	･Java based benchmark tool ･New BSD License ･Ready for multiple DBMS（PostgreSQL, Oracle, MySQL) ･Scripts for Tiny SysBench, Tiny TPC-B, Tiny TPC-C are provided
Tiny TPC-C	･Simplified TPC-C Standard Specification 5.10.1 ･Implemented features in TPC-C 1 LOGICAL DATABASE DESIGN 2 TRANSACTION and TERMINAL PROFILES 2.4 The New-Order Transaction (except 2.4.1.1 and 2.4.3) 2.5 The Payment Transaction (except 2.5.1.1 and 2.5.3) 2.6 The Order-Status Transaction (except 2.6.1.1 and 2.6.3) 2.7 The Delivery Transaction (except 2.7.1.1, 2.7.2 and 2.7.3) 2.8 The Stock-Level Transaction (2.8.1 and 2.8.3) 4 SCALING and DATABASE POPULATION 4.3 Database Population 5 PERFORMANCE METRICS and RESPONSE TIME 5.2 Pacing of Transactions by Emulated Users 5.2.4 Regulation of Transaction Mix
Downloads	http://hp.vector.co.jp/authors/VA052413/jdbcrunner/

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Session scalability result

- Average TPM (Transaction Per Minute) increases as the number of session increases

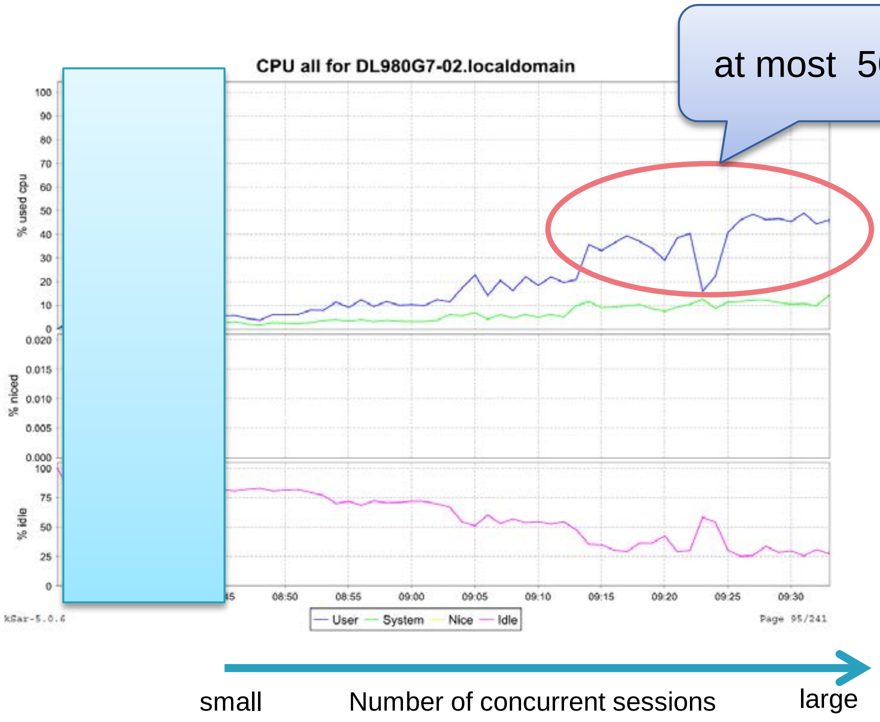


Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

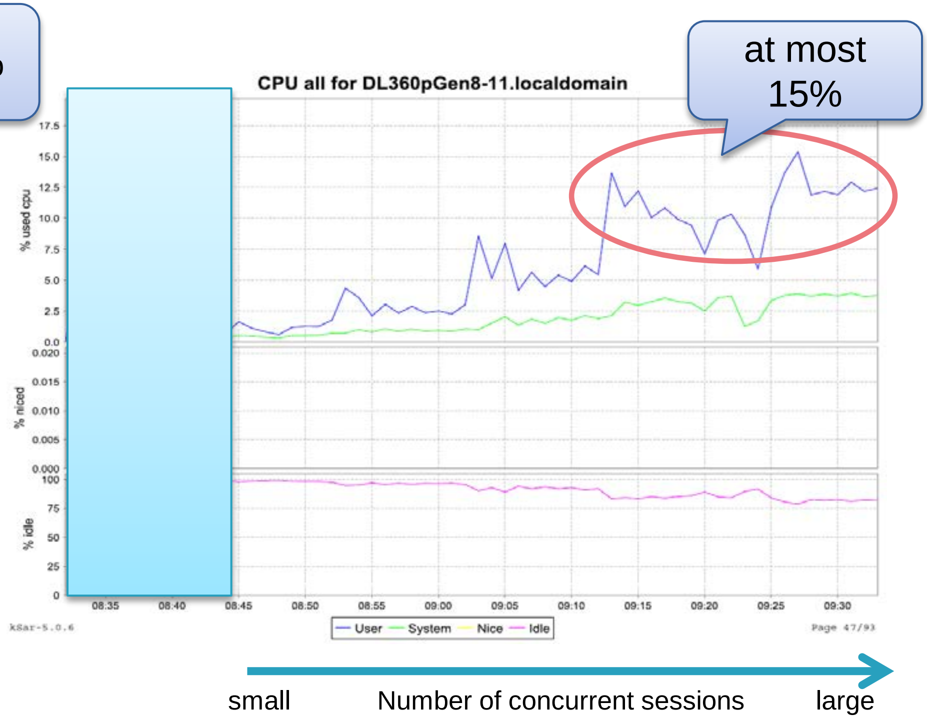
CPU utilization (40 cores)

- there is room for CPU utilization at both server and client

40core Server CPU utilization



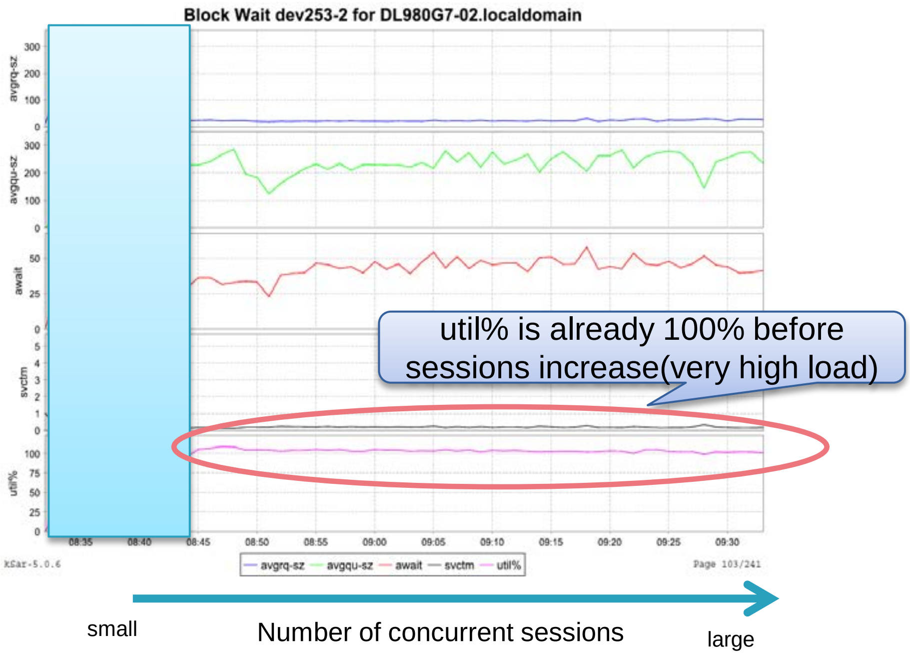
Client CPU utilization



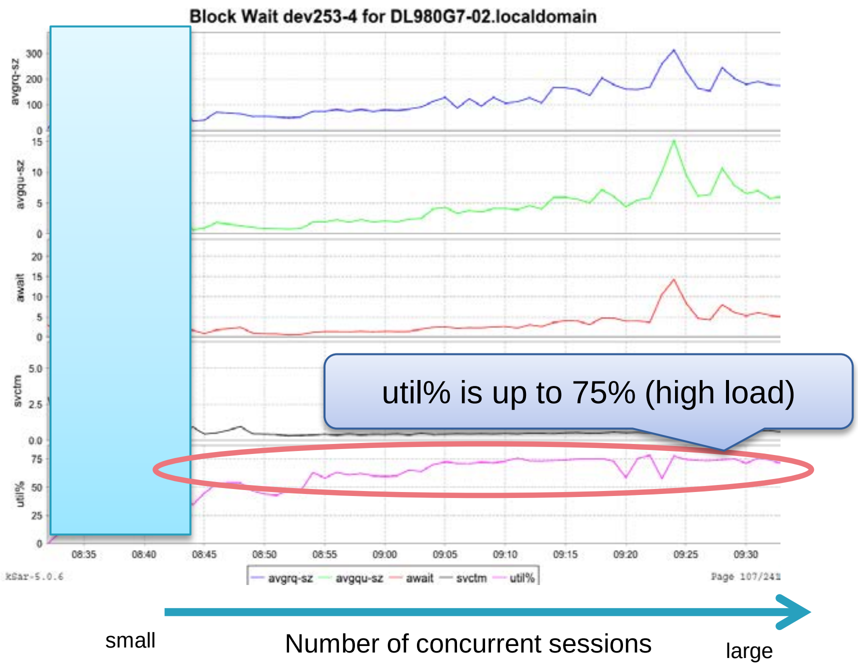
Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Disk utilization (40 cores)

I/O utilization of DB (base)



I/O utilization of pg_xlog



It is possible that the bottle neck is DB and XLOG I/O. Thus CPU resource is not well utilized

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

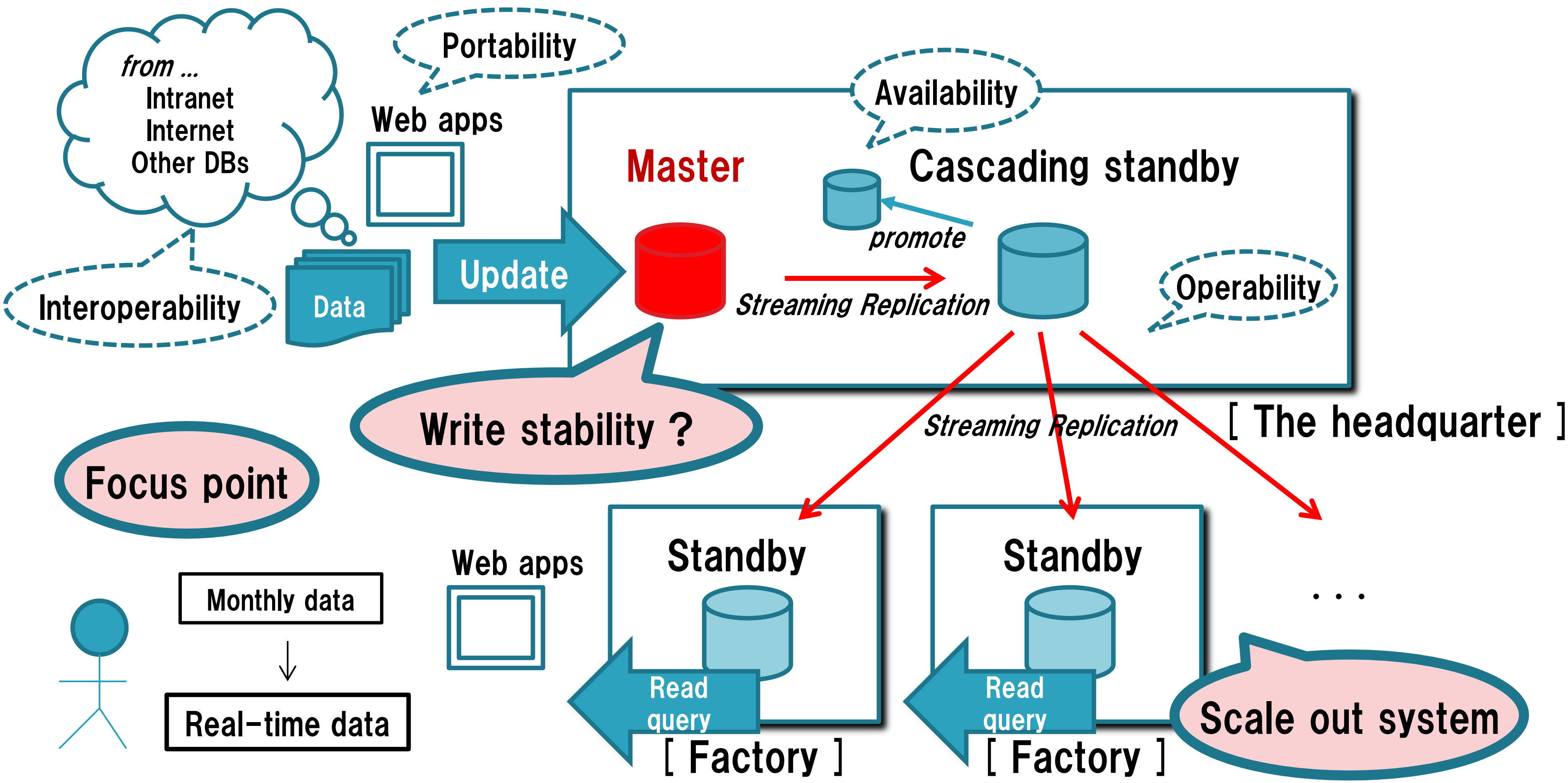
Report2: Scale out evaluation

- 2.1 PostgreSQL 9.2 cascading replication
  - Checking master DB performance while changing number of grandchild replication nodes
  - It is confirmed that the master DB performance is stable even if the number of replication nodes increase

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

The evaluation model

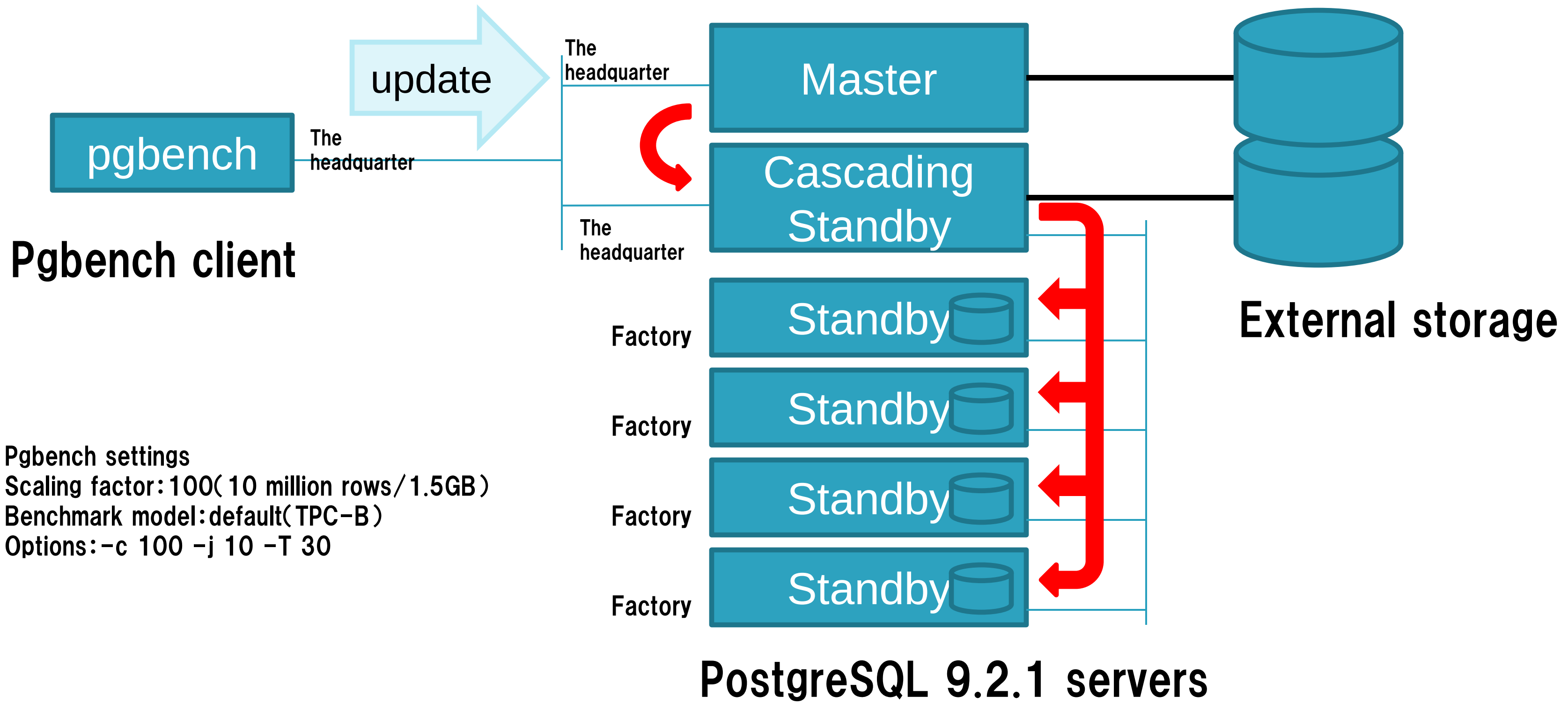
- The material information is delivered from the headquarter to factories



Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

The benchmark system for cascading replication

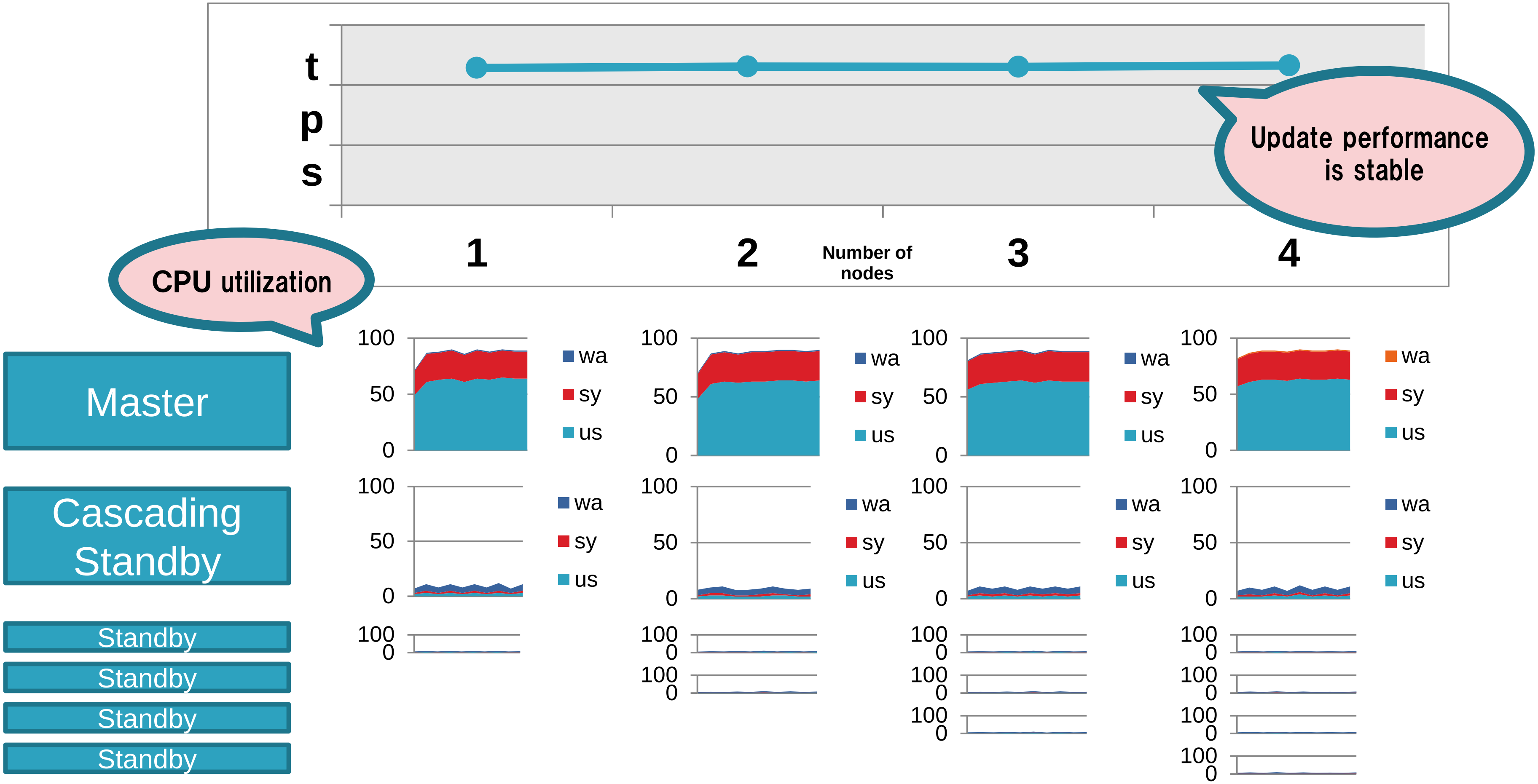
- Master and cascading standby are connected to an external storage. Standbys have their own storage
- Replication path: master→cascading standby→standby×4



- Pgbench settings
- Scaling factor:100(10 million rows/1.5GB)
- Benchmark model:default(TPC-B)
- Options:-c 100 -j 10 -T 30

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Cascading replication evaluation result



Update performance of master node is stable even if the number of standby nodes increase.

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

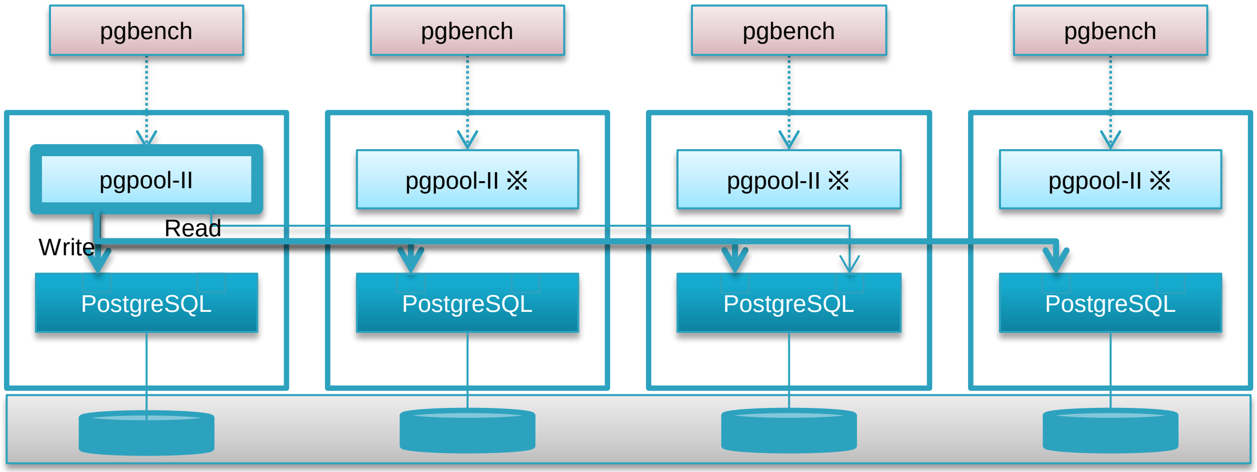
Report2: Scale out evaluation

- 2.2 pgpool-II
  - Write query performance decreases as the number of nodes increases
  - Read query performance increases as the number of nodes increases

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

The benchmark system for pgpool-II

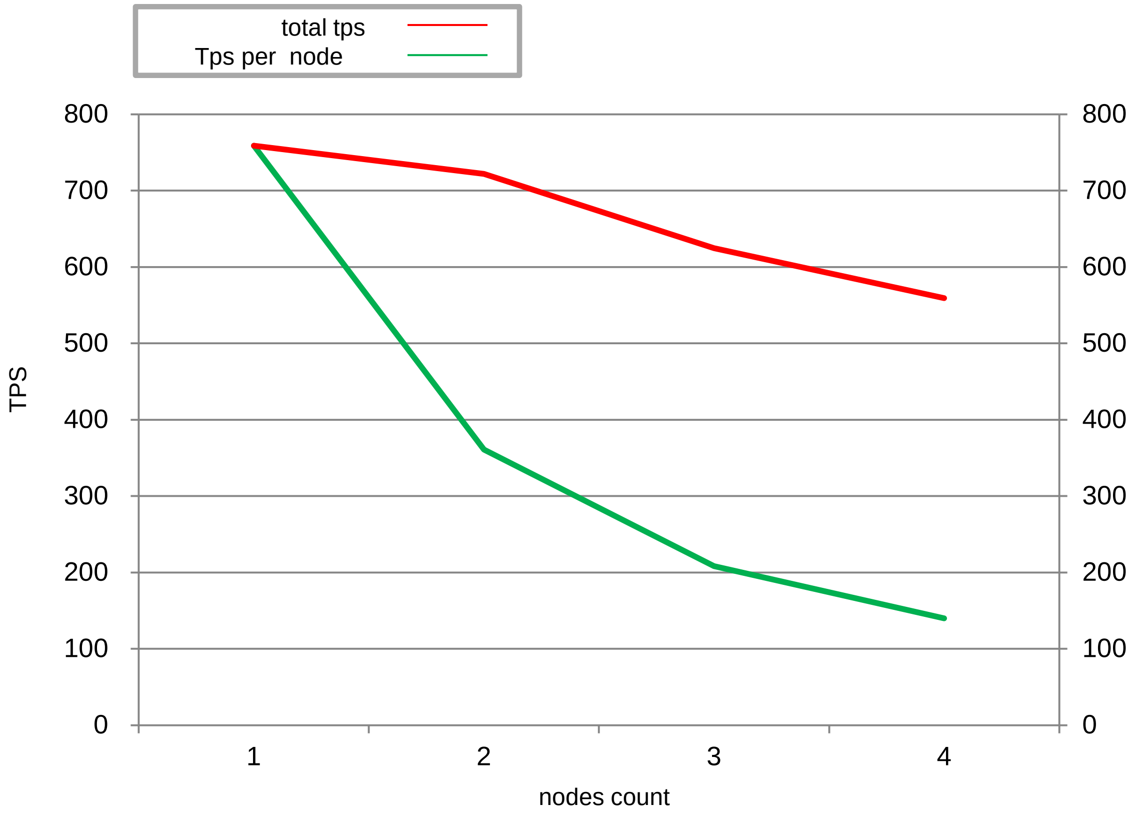
- **pgpool-II configuration**
  - Using pgpool-II (3.2.1) & PostgreSQL (9.2.1)
  - Pgpool-II is configured to use native replication mode (synchronous replication)
- **Pgbench configuration**
  - Up to 4 pgbench clients used. TPS is defined as the summary of each pgbench TPS
  - Pgbench default scenario is used for write query test
  - Read test uses custom scenario because default scenario (-S) is too subtle



※ Arrows for second node or above is omitted to avoid complexity of the figure

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Write query result

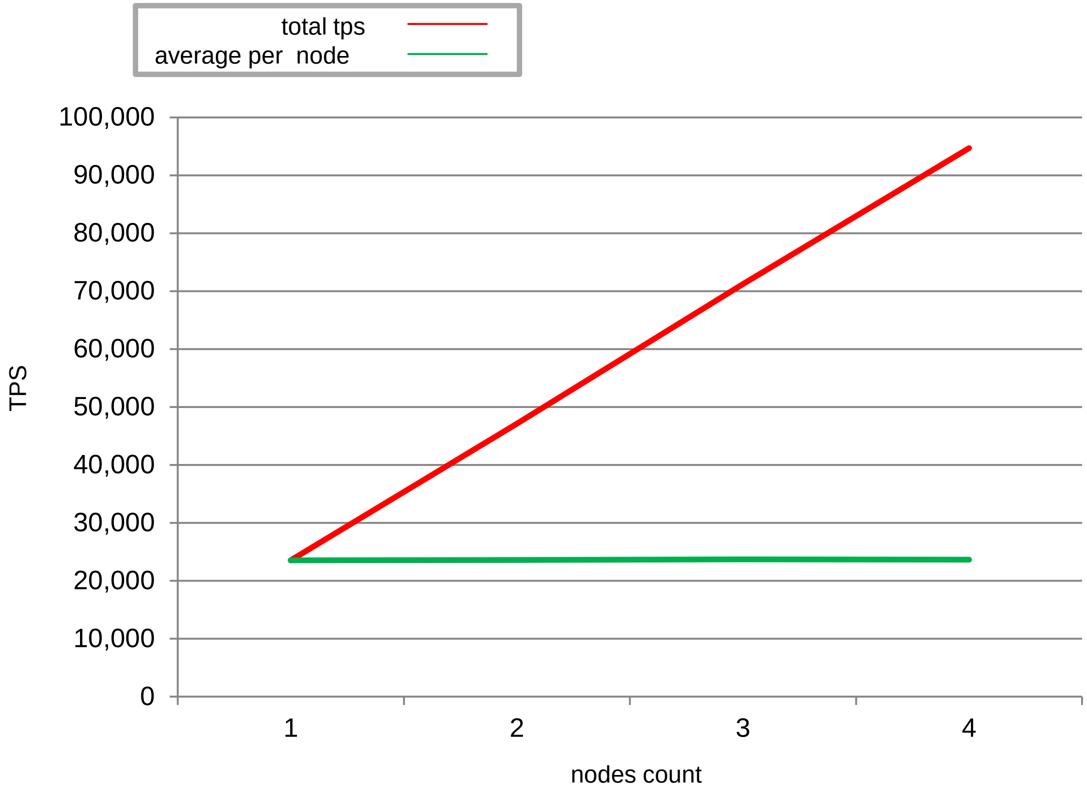


- Settings
  - PostgreSQL shared memory: 16GB
  - Pgbench scale factor: 1,000
  - Each duration(-T): 300 seconds
  - Number of concurrent sessions (-c): 100
  - Number of worker threads (-j): 20

- Total TPS decreases as the number of nodes increases
- This is due to the overhead of synchronous replication

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Read query result



- Settings
  - PostgreSQL shared memory: 32GB
  - Pgbench scale factor: 1,000
  - Each duration(-T): 300 seconds
  - Number of concurrent sessions (-c): 100
  - Number of worker threads (-j): 20

- The custom scenario. Randomly extracts 2,000 rows.

```
¥set nbranches :scale
¥set ntellers 10 * :scale
¥set naccounts 100000 * :scale
¥set range 2000
¥set aidmax :naccounts - :range
¥setrandom aid 1 :aidmax
¥setrandom bid 1 :nbranches
¥setrandom tid 1 :ntellers
¥setrandom delta -5000 5000
SELECT count(abalance) FROM
pgbench_accounts WHERE aid
BETWEEN :aid and :aid + :range;
```

- Total TPS increases as number of nodes increases(scale out)

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

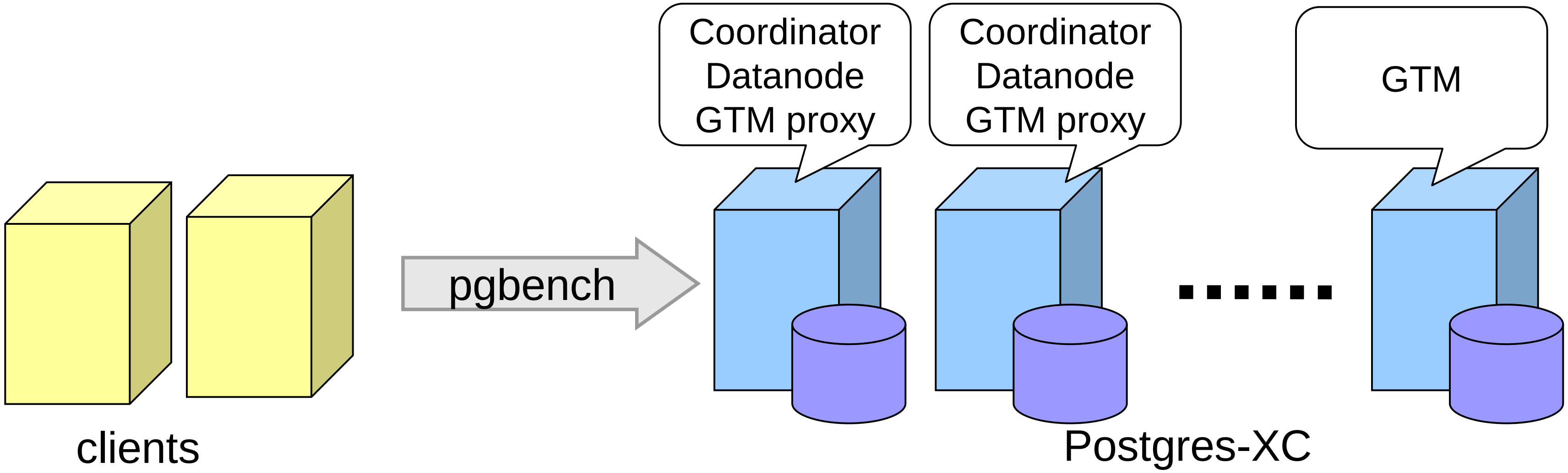
Report2: Scale out evaluation

- 2.3 Postgres-XC
  - Write query performance increases as the number of nodes increases
  - Read query performance increases when data size is larger

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

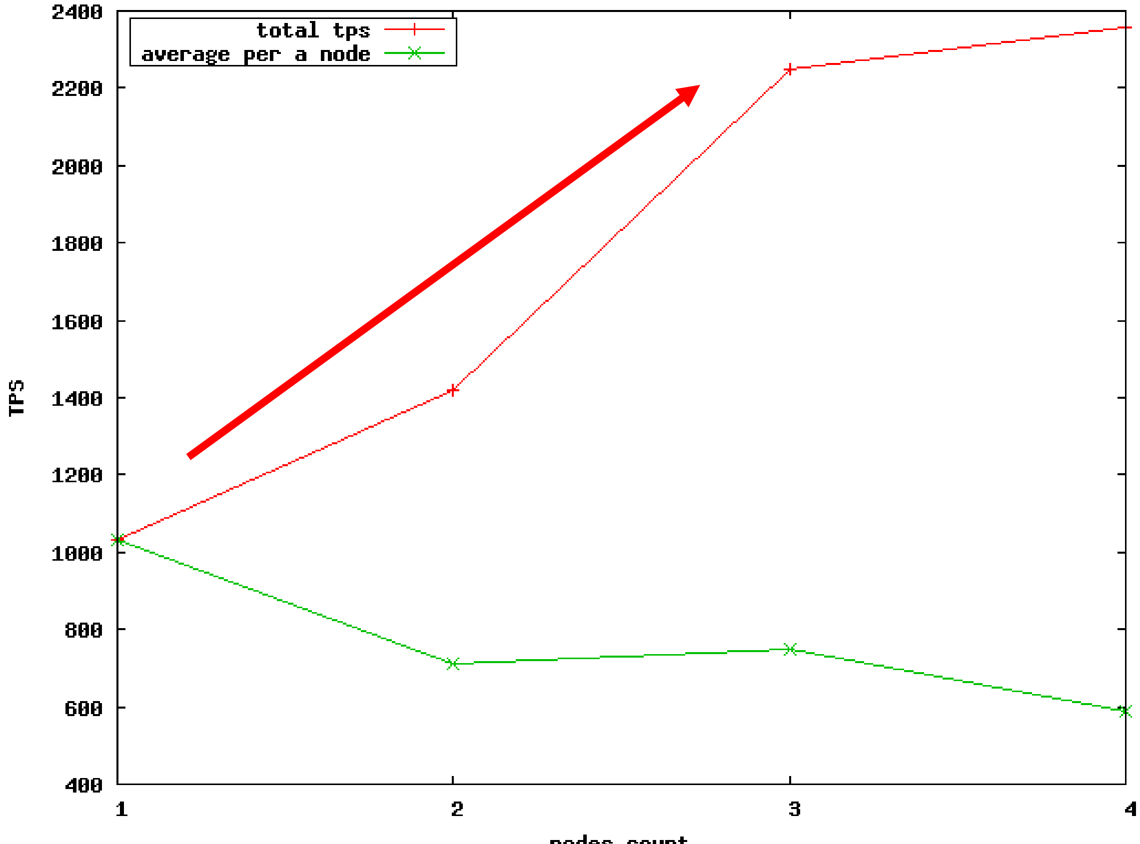
The benchmark system for Postgres-XC

- Postgres-XC configuration
  - Postgres-XC 1.0.1(based on PostgreSQL 9.1.5)
- Pgbench configuration
  - up to 4 pgbench clients used. TPS is defined as the summary of each pgbench TPS
  - Modified pgbench is used. It distributes data if “-k” is provided



Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Write query result



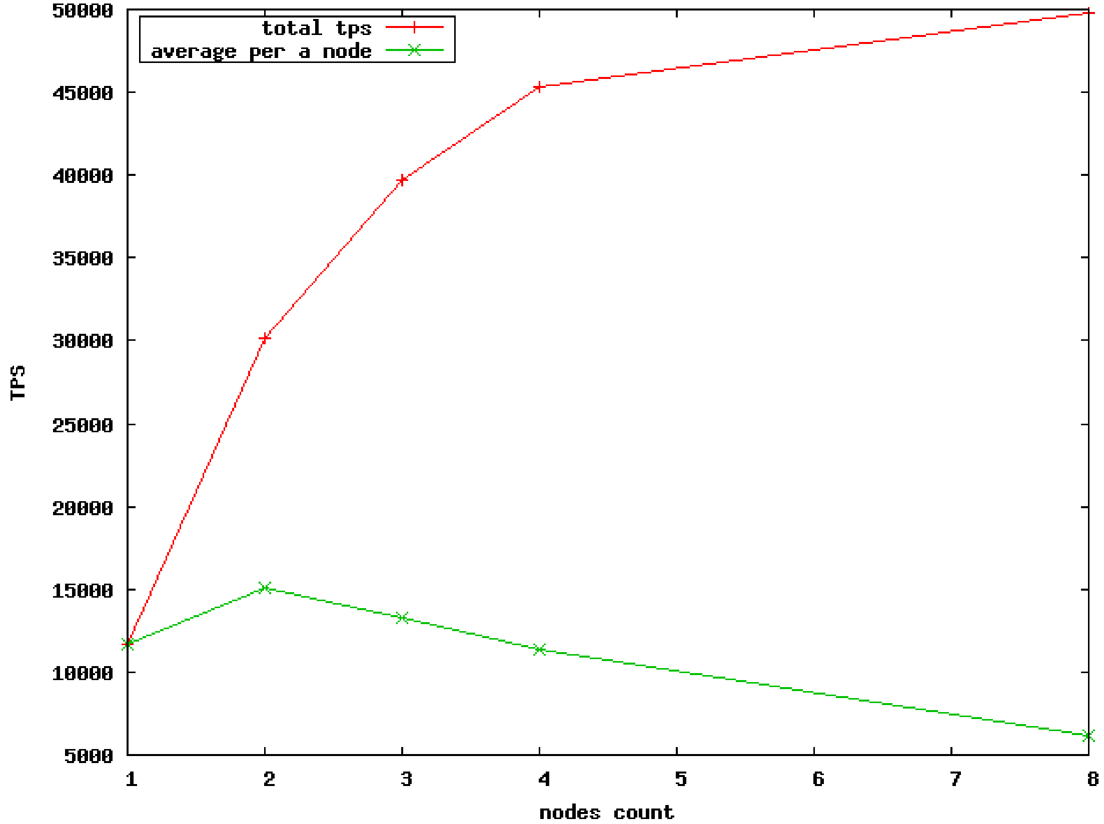
pgbench -i -k
-s 100
(total: 1.5GB)

pgbench
-c 100 -j 10
-n -k
-T 600

Sum of TPS increases as the number of Datanodes increase

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Read query result(pgbench -n -k -S)



pgbench -i -k
-s 1000
(total: 15 GB)

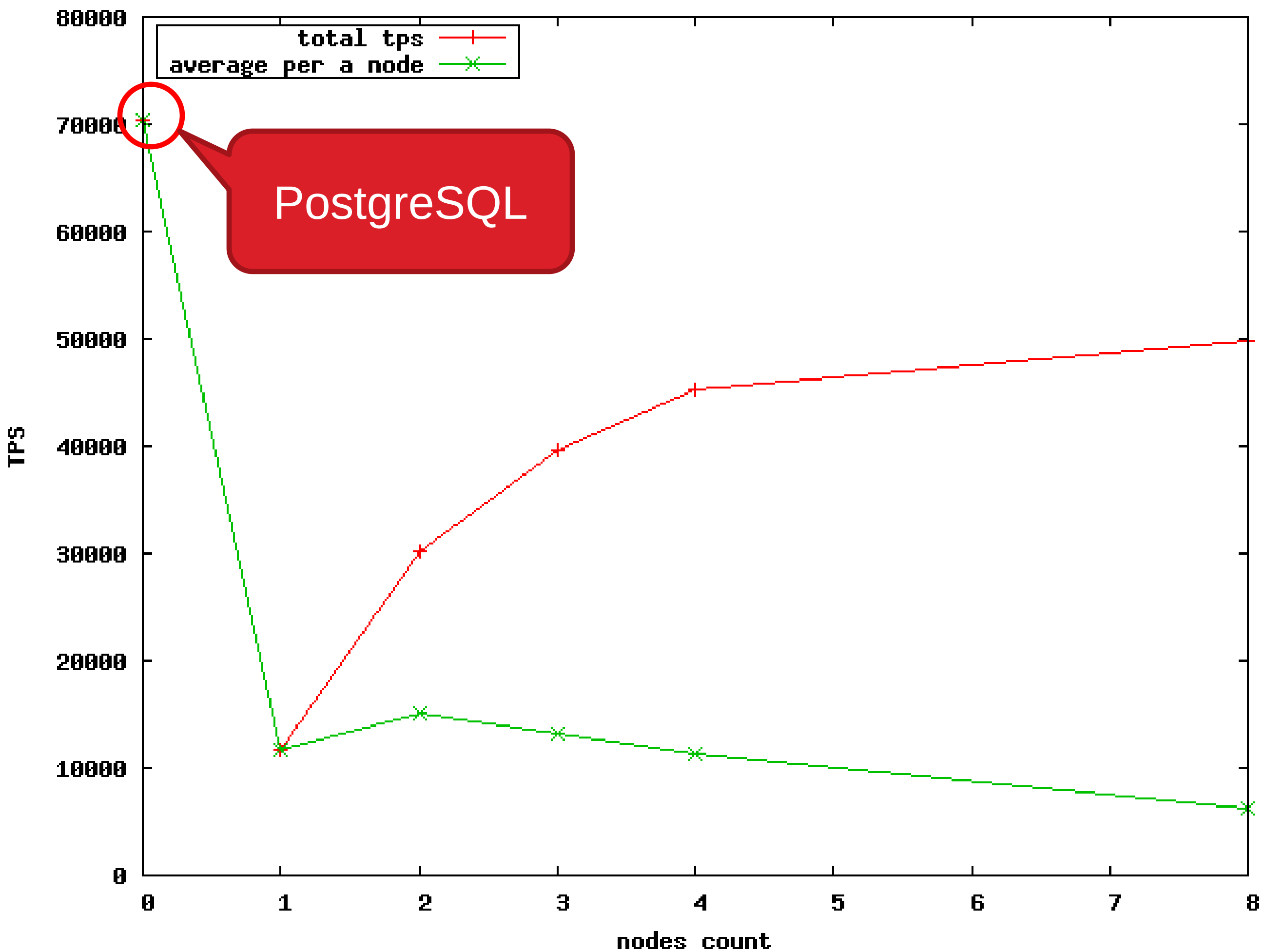
pgbench
-c 100 -j 10
-n -k -S
-T 300

Sum of TPS increases as the number of Datanodes increase

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Read query result compared with PostgreSQL

- Postgres-XC total TPS is lower than PostgreSQL



Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Summary of WG1 activities in 2012

- Scale up evaluation
  - Confirmed PostgreSQL scalability of up to 80 cores (64 cores scalability has been already published)
  - Confirmed read/write CPU scalability
- Scale out evaluation : PostgreSQL cascading replication, pgpool-II, and Postgres-XC
  - Cascading replication keeps stable write performance even if number of nodes increases
  - Pgpool-II is strong in read queries
  - Postgres-XC is strong in write queries

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Agenda（PGECons WG2）

- Why DBMS Migration?
- Highlights from FY2012 final report
- Concluding remarks: Crucial criteria for a successful migration project

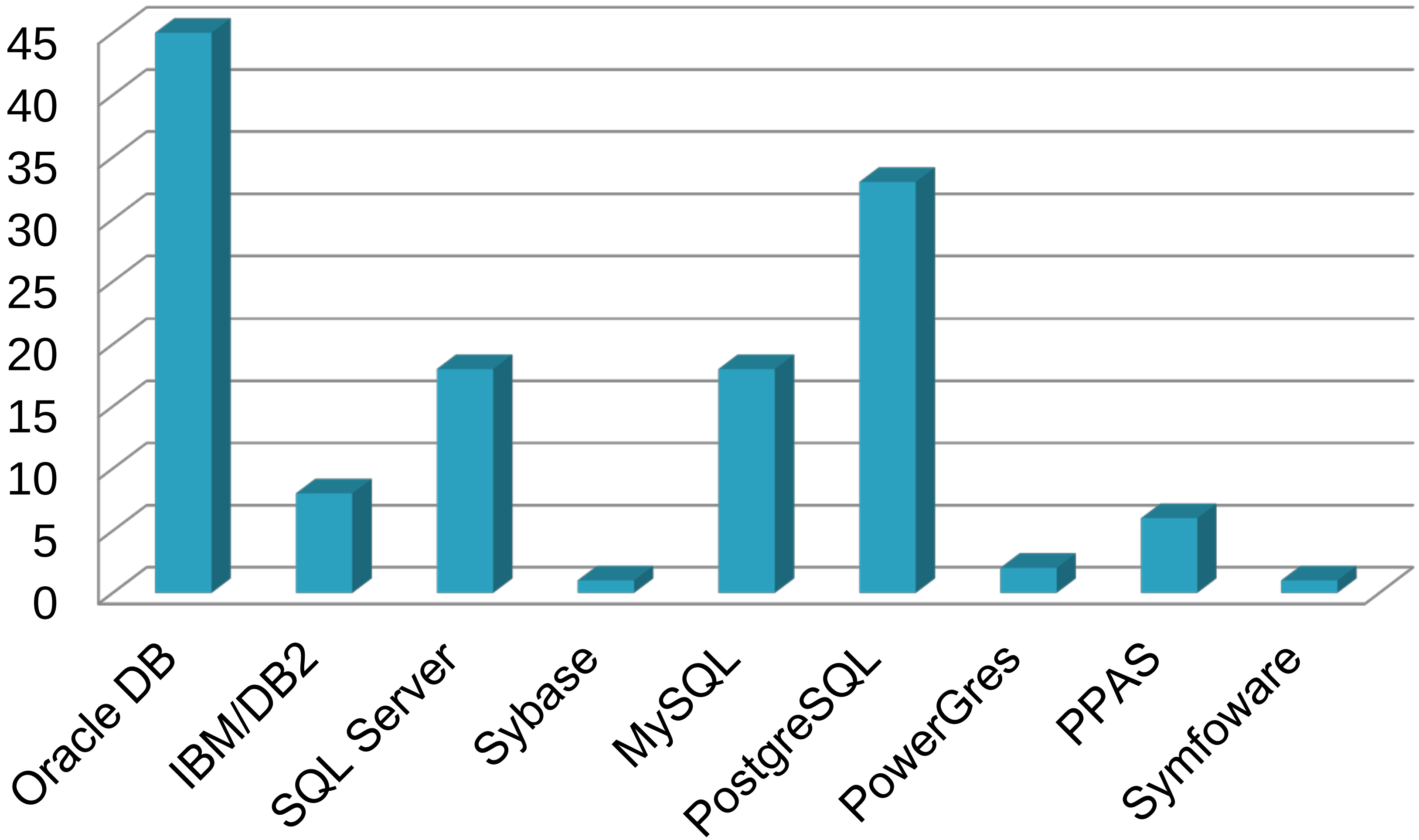
Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Why DBMS Migration?

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

From the result of questionnair in PGECons opening seminar（July 6th, 2012）

Q: Which DBMS are you using? (multiple answers allowed)



Users of Oracle, DB2, SQL Server, Sybase and MySQL are in total 2.5 times more than those of PostgreSQL.

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Why DBMS Migration?

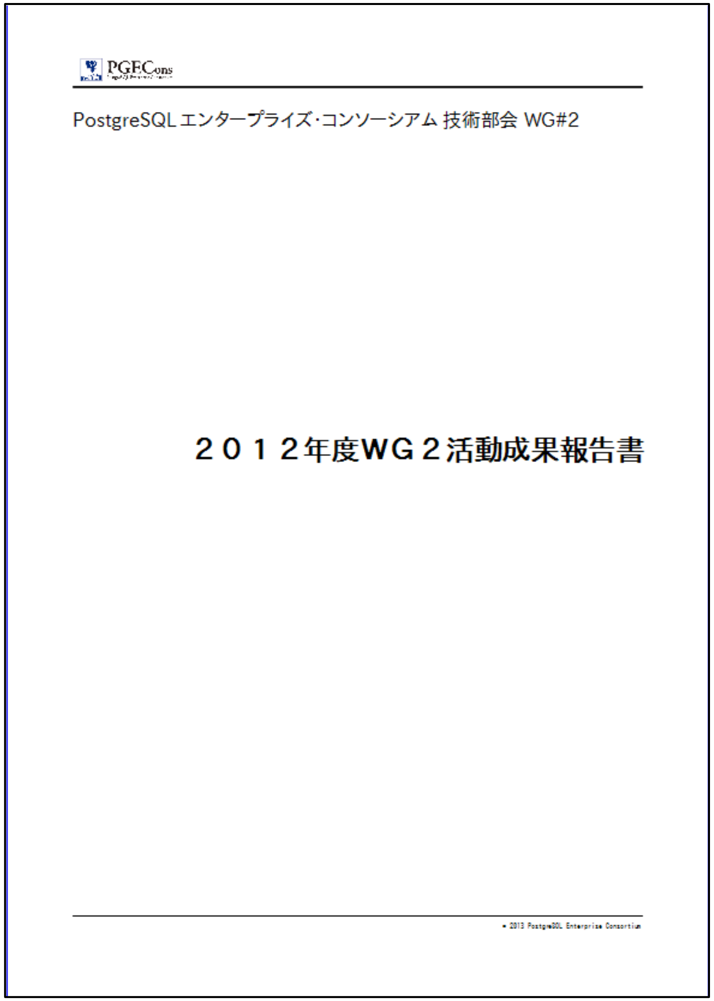
- To attract Enterprise users using other DBMS to PostgreSQL

No Migration, No Enterprise.
(PGECons WG2 slogan)

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

FY2012 WG2 Report

- Includes database migration guide as well as pilot migration trial reports
- More than 200 pages. Still growing in this year.
- Open to the public in PGECons sight.


PGECons

PostgreSQL エンタープライズ･コンソーシアム 技術部会 WG#2

２０１２年度ＷＧ２活動成果報告書

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Highlights from WG2 report

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

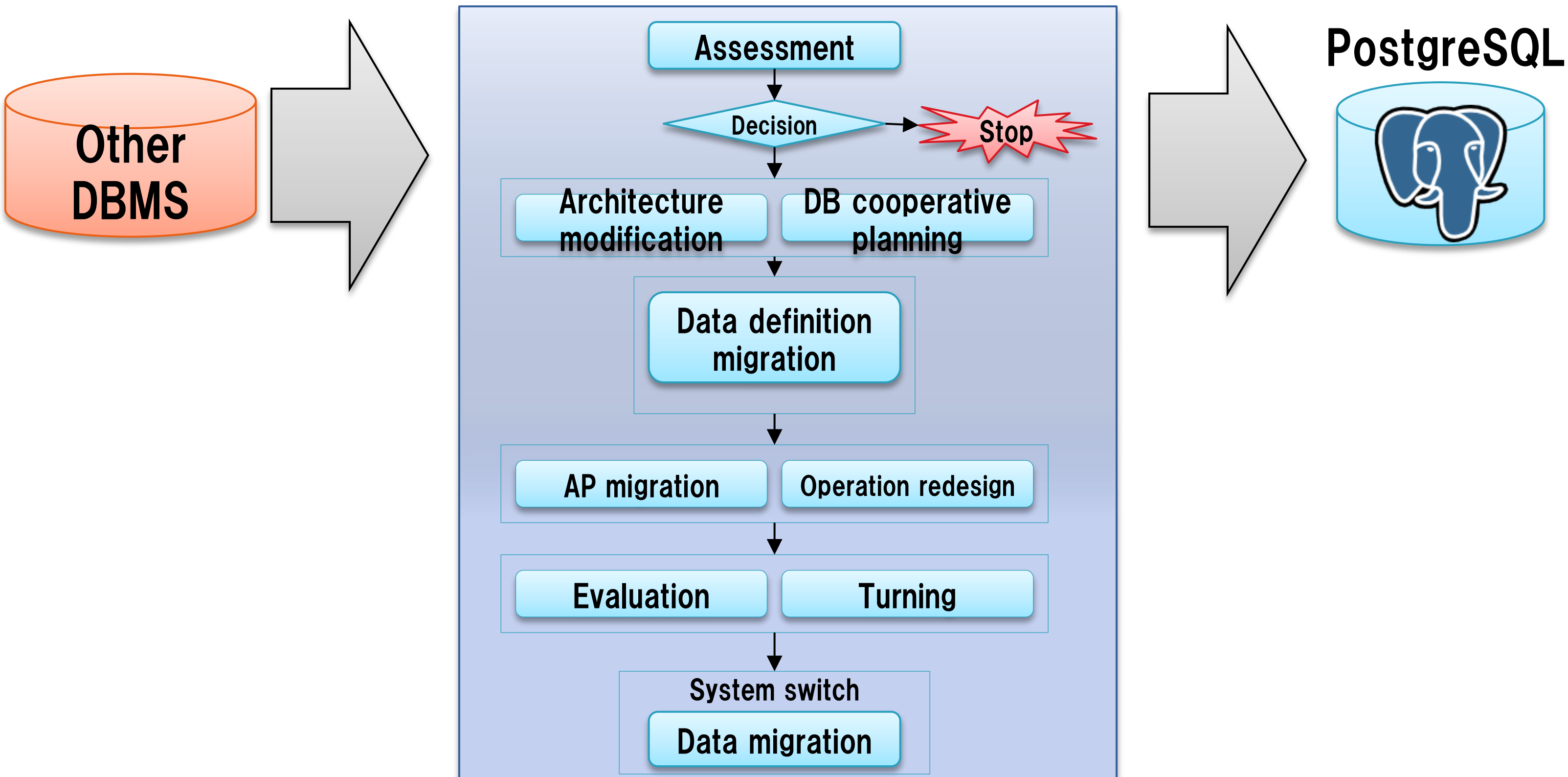
Organization of the migration guide

- Outlining a migration project
  - Database migration framework
- Study on each migration process
  - System architecture
  - Cooperative database systems
  - SQL migration
  - Stored procedure
  - Built-in function migration
  - Application migration
- Trial reports on migration works
  - Research on data migration and practice
  - Practice of application migration

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

DBMS Migration Framework

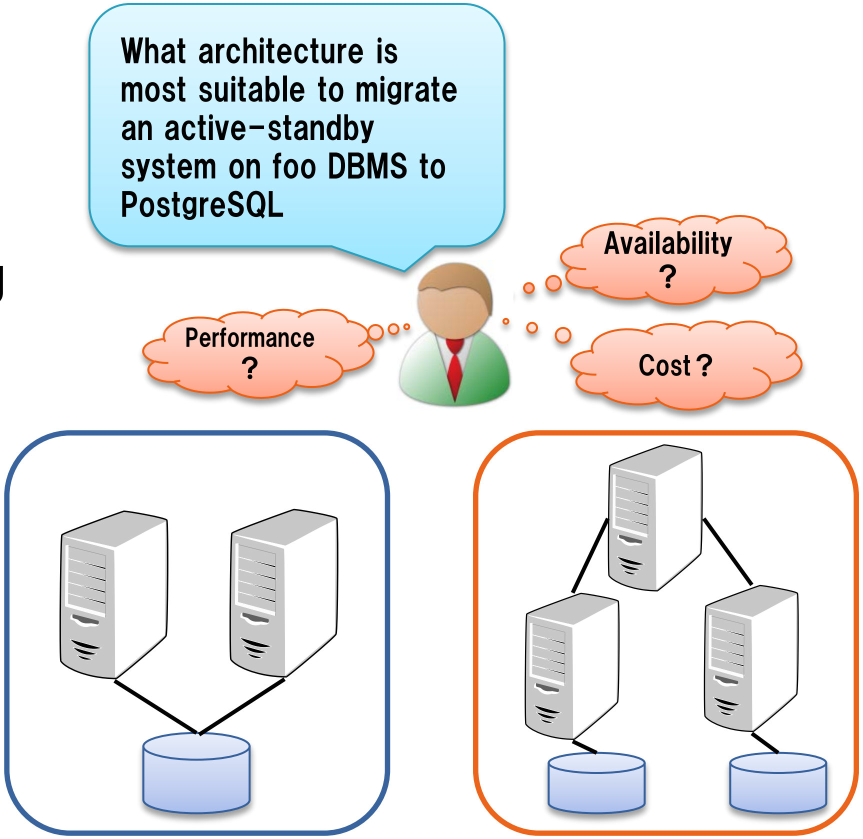
- Outlining the DBMS migration process to PostgreSQL
  - Establish common background of migration process
  - Flow chart of DBMS migration



Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

DB system architectures

- Investigate major DB-system architectures and provide an information to find correspondence in PostgreSQL
  - Major DB-system architectures
    - Single server
    - HA cluster
    - Replication
    - Multi-master load balancing
  - Points to consider
    - High availability
    - Performance (read/write)
    - Extensibility
    - Easy to design/operate
    - Initial cost



Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Cooperative database systems

- Purpose of cooperation
  - Stepwise migration of complex system with plural DB
  - Load reduction of master DB
  - Preparation for disaster
  - Data warehouse system
- Investigate some software allowing replication in different DBMSs
  - InfoFrame DataCoordinator （NEC）
  - DBMoto （Climb）
  - DataSpider Servista（Appresso）
  - xDB Replication Server（EnterpriseDB）

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Schema migration

- Investigate Schema of Oracle and PostgreSQL
- Study some manual intervention points using Ora2Pg
- Compile correspondence table for built in data type and guideline to migrate schema

属性	Oracle			PostgreSQL			
	データ型	単位	備考	データ型	サイズ	単位	備考
文字	VARCHAR2	byte	4000 バイト	varchar(n)			上限付き可変長
		char	4000 バイト	varchar(n)			上限付き可変長
	NVARCHAR2	char	4000 バイト	varchar(n)			上限付き可変長
	CHAR	byte	2000 バイト	char(n)			空白で埋められた固定長
		char	2000 バイト	char(n)			空白で埋められた固定長
	NCHAR	char	2000 バイト	char(n)			空白で埋められた固定長
	LONG （下位互換）		2G-1 バイト	text			可変長(最大1GB)
	CLOB		4G-1 バイト	text			可変長(最大1GB)
	NCLOB		4G-1 バイト	text			可変長(最大1GB)
真数	NUMBER		精度(10進数38桁)、位取り(10進数-84～127桁)	decimal	可変長	固定小数点	小数点前までは131072桁、小数点以降は16383桁
				numeric	可変長	固定小数点	小数点前までは131072桁、小数点以降は16383桁
				smallint	2バイト	整数データ	整数(-32768～+32767)
				integer	4バイト	整数データ	整数(-2147483648～+2147483647)
				bigint	8バイト	整数データ	整数(-9223372036854775808～9223372036854775807)
概数				real	4バイト	単精度浮動小数点	6桁精度
				double precision	8バイト	倍精度浮動小数点	15桁精度
	FLOAT		精度(2進数126桁)	float	4、8バイト	浮動小数点	精度（2進数53桁）
	BINARY_FLOAT		単精度浮動小数点数	real	4バイト	単精度浮動小数点	6桁精度
	BINARY_DOUBLE		倍精度浮動小数点数	double precision	8バイト	倍精度浮動小数点	15桁精度
日時	DATE	日～秒	-4712/01/01 ～ 9999/12/31	timestamp	8 バイト	1μ秒単位	日付と時刻両方（時間帯なし） 4713 BC～294276 AD（1μ秒、14桁）
				date	4 バイト	日単位	日付のみの場合はPostgreSQLのdate型で表現可能
	TIMESTAMP		DATE 型に加えてミリ秒、最小でナノ秒単位	timestamp	8 バイト		日付と時刻両方（時間帯なし） 4713 BC～294276 AD（1μ秒、14桁）
	TIMESTAMP WITH TIMEZONE		タイムスタンプ型に加えてタイムゾーン情報	timestamp [(p)] with time zone	8バイト		日付と時刻両方、時間帯付き 4713 BC～294276 AD（1μ秒、14桁）

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

SQL migration

- Investigate SQL compatibility of Oracle, SQL Server, and PostgreSQL
- Compile conversion table and guideline

SQL機能	標準SQL	説明	PostgreSQL 9.2.1		Oracle 11g R2		SQL Server 2008 R2	
			対応	備考	対応	備考	対応	備考
SELECT								
WITH句	○	WITH 問い合わせ(共通テーブル式)	○	WITH 句の中で更新系コマンドが使用可能	○		○	列の別名に独自の構文あり
DISTINCT	○	重複している行を取り除く	○	DISTINCT ON は PostgreSQL 特有	○		○	
UNIQUE	×	重複している行を取り除く	×		○	Oracle 特有	×	
TOP句	×	取得する行数の指定	×		×		○	SQL Server 特有
FROM句	○	テーブルの指定	○	FROM 句を省略可能	○	FROM 句は省略不可 サブクエリの別名を省略可	○	FROM 句を省略可能
CONNECT BY / START WITH 句	×	階層問い合わせ	×		○	Oracle 特有	×	
JOIN 句	○	テーブルの結合	○		○	独自の結合演算子(+)あり	○	
WINDOW句	○	OVER 句で参照するウィンドウ	○		×		×	
LIMIT 句	×	取得する行数などの指定	○		×		×	
FETCH 句	○	取得する行数などの指定	○		×		×	
SELECT FOR UPDATE	○	更新ロックを取得する (標準 SQL ではカーソルのオプションとしてのみ有効)	○	任意の SELECT で使用可能	○	サブクエリでは使用不可	○	カーソルでのみ使用可能
SELECT FOR SHARE	×	共有ロックを取得する	○	PostgreSQL特有	×		×	
更新系								
INSERT		行の挿入		RETURNING句を使用可能				TOP句を使用可能

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Stored procedure migration

- Some part of procedure can be translated automatically
- Difficult to migrate if the procedure is involved in transaction control
- For some stored procedures, it is better to rewrite them to application logics than to convert them to corresponding stored functions.

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

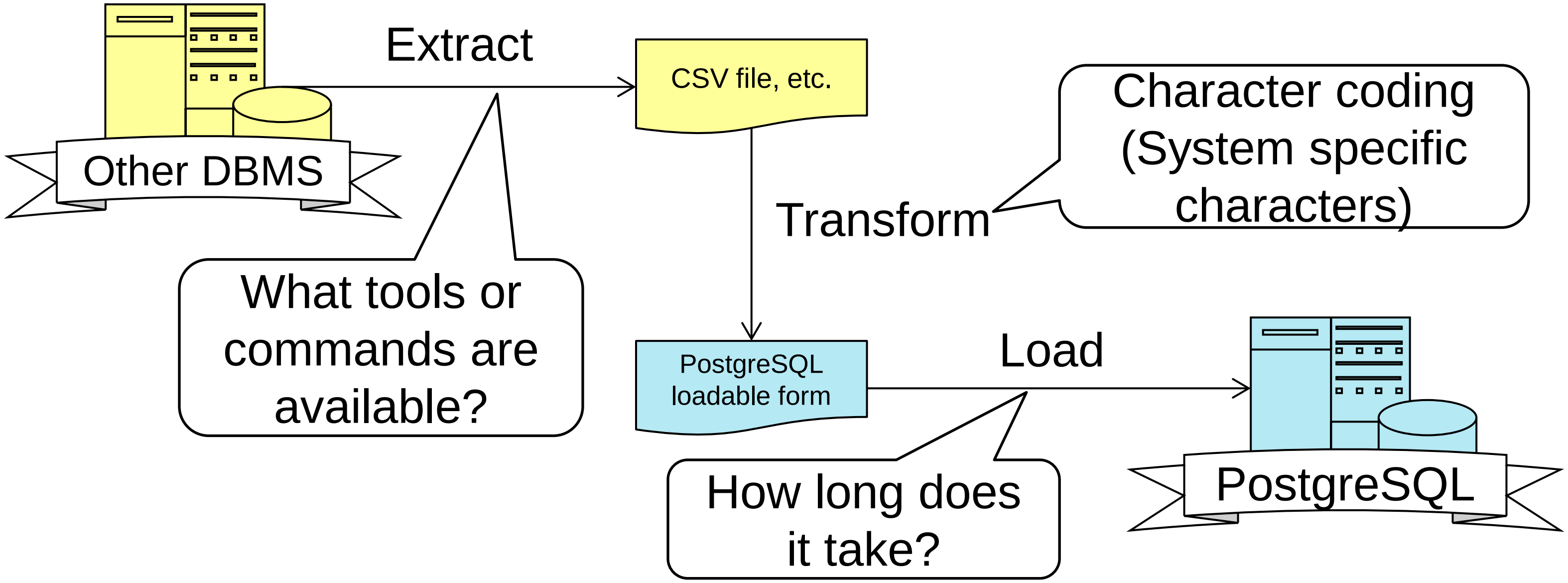
Built-in function migration

- Investigate built-in function difference between Oracle and PostgreSQL
- Compile built-in function comparison table

Oracle関数	説明	PostgreSQL関数	対応可否
日時ファンクション			
ADD_MONTHS (date, integer)	日付dateに月数integerを加えて戻す。	+演算子を使って書き換え可能 例: select date '2013-03-22' + interval '1 months'	○
CURRENT_DATE	セッション・タイムゾーンの現在の日付を戻す。	current_date、current_timestamp	○
CURRENT_TIMESTAMP	セッション・タイムゾーンの現在の日付および時刻をTIMESTAMP WITH TIME ZONEデータ型の値で戻す。	current_timestamp	○
DBTIMEZONE	データベースのタイムゾーンの値を戻す。		×
EXTRACT (element FROM date)	日時式または期間式から指定された日時フィールドの値を抽出して戻す。	extract(field from timestamp)	○
FROM_TZ(timestamp, time_zone_value)	タイムスタンプ値およびタイムゾーンをTIMESTAMP WITH TIME ZONE値に変換する。		×
LAST_DAY(d)	d を含む月の最後の日付を戻す。	last_date(date)	○
LOCALTIMESTAMP	セッション・タイムゾーンの現在の日付および時刻をTIMESTAMPデータ型の値で戻す。	localtimestamp	○
MONTHS_BETWEEN(d1, d2)	d1 とd2 の間の月数を戻す。	months_between(d1, d2)	○
NEW_TIME(d, z1, z2)	時間帯z1の日時がdの時点の時間帯z2の日時を戻す。		×
NEXT_DAY(d, char)	charで指定した曜日で日付d以降の最初の日付を戻す。	next_day(date, text)	○
NUMTODSINTERVAL(n, 'char_expr')	n をINTERVAL DAY TO SECOND リテラルに変換する。		×
NUMTOYMINTERVAL(n, 'char_expr')	n をINTERVAL YEAR TO MONTH リテラルに変換する。		×
ROUND(d, fmt)	d を書式 fmt で指定した単位に丸めた結果を戻す。	round(date, text)	○

Data migration: study and trial

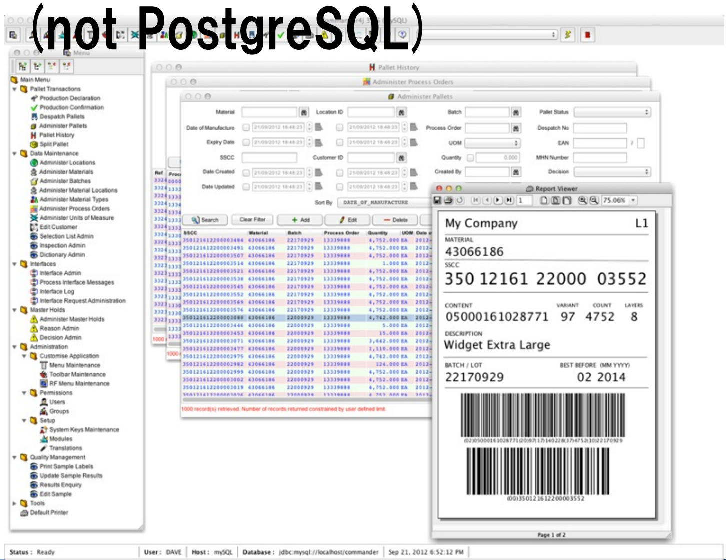
- Study data migration process from other DBMS to PostgreSQL
- Apply acquired information and knowhow to a practical data migration project



Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Application migration trial (1/4)

- To apply acquired information and knowhow to a trial migration project of a practical application on other DMBS to PostgreSQL (9.2.2)
- Migration target application
  - Commander4J
    - Open source Java application to create bar code labels
    - Runs on:Oracle, SQLServer, MySQL (not PostgreSQL)
    - Size:71K steps
    - Number of tables:39
    - Number of SQL statements:3390
    - Stored procedure:No



Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Application migration trial (2/4)

- Utilize a tool to extract necessary modification points in SQL
- db_syntax_diff
  - A migration aid tool from Oracle Database to PostgreSQL
  - Open source software developed by NTT
  - The PostgreSQL License
  - https://github.com/db-syntax-diff
  - Provide syntax difference dictionary and operate pattern matching on SQL program
  - Dictionary entries are written in regular expression and user customizable

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Application migration trial (3/4)

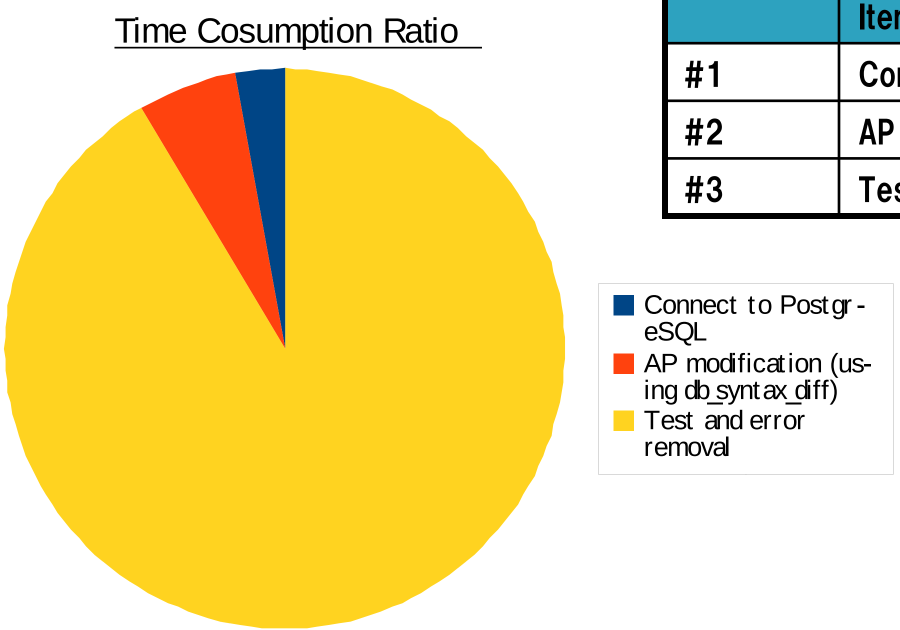
- Migration process
  - Apply a migration tool “db_syntax_diff” to Java source code, schema DDL and data loading DML
  - Modify calling part of Oracle JDBC driver class
  - Modify SQL statements according to db_syntax_diff output
  - Test of the application operation
    - Basic normal functionality of Commander4j

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Application migration trial (4/4)

- **Trial conclusion**
  - More than 90% of time was spent in test phase
    - SQL modification was rapid due to utilization of migration tool (db_syntax_diff)
    - All SQL were tested regardless of modification.
    - Oversights of the tool were corrected in test phase.

	Item	Time ratio
#1	Connect to PostgreSQL)	2.8%
#2	AP modification (using db_syntax_diff)	5.8%
#3	Test and error removal	91.4%



Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Concluding remarks: crucial criteria for a successful migration project

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

What is a successful migration?

- Even if the system itself operates in PostgreSQL, it is not a success if the migration cost ends up very huge.
- A successful migration we consider is migration whose final cost is very near to initial estimation.

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Points for successful migration to PostgreSQL

- Accuracy of assessment
  - Initial assessment accuracy is most important.
  - Understand accurately the requirements of the original system and the migrated new system
  - Understand accurately the difference between the original DBMS and PostgreSQL
  - Utilize migration aid tools cleverly
- Prepare testing time sufficiently
  - After all, works tend to be increase
  - In our trial, more than 90% of time is spent in test phase.
  - Modification works can be reduced using migration tools.
  - Tools are not perfect. Prepare for manual modification.

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.

Thank you.

Copyright © PostgreSQL Enterprise Consortium, All Rights Reserved.