NTT's Case Report

Introduce PostgreSQL into reliable and large-scale telecommunication support system

Tetsuo SAKATA
NTT Open Source Software Center
19th May 2011
Agenda

- Introduce ourselves
- Understand Needs
- Evaluation
- Development
- Technical supports
- NTT Cases
- Expectation
Introduce myself

- **Name:** Tetsuo SAKATA
- **Job:** Software engineer / manager at NTT OSS center.
- **Community**
  - director of JPUG (Japan PostgreSQL User Group)
Introduce NTT

- Nippon Telegram and Telephone Group profile
  - Revenue: 10.2 trillion yen ($113 billion)
    - Second largest telecommunication company.
  - Number of employees: 200,000.
  - Businesses
    - Number of Consolidated Subsidiaries: 536
    - Telecommunication
      - Subscribers: 93 million (incl. regional, long distance, mobile)
    - System Integration
      - Large company and government systems
    - Others
      - Construction, hospital, publishing, florists etc.
Character of NTT system

- Telecommunication operation system (OpS)
  - Large-scale
    - Each DB is large (e.g. 100GB) and some communicate each other.
  - High availability and reliability
    - Telephone system is available more than 99.999%.
  - Long-lived
    - Expected lifetime is 7 year's

- Issues
  - Proprietary DBMS are widely used.
    - High-cost, supports are short
    - Vendor lock-in.

OSS are expected to solve these issues.
Introduce Open Source Software Center

- Mission:
  - Reduce TCO with OSS; replacing proprietary software
    - Support NTT Group companies' OSS usage
      - Q and A
      - Consultation
    - Develop / improve OSS
  - Center of OSS competence in NTT Group.

- Established in Apr. 2006.
- Location: Shinagawa Tokyo.
Understand user needs;
*How to introduce PostgreSQL?*

- Information on performance
  - Show good and stable performance
  - Availability/reliability
    - downtime to recovery (e.g. 5' for five-9s)
  - To prepare equipment (HDDs, CPUs etc.)
- Operation capability
  - compatibility with other operation tools
  - Usability
- Improve performance and usability
- Technical support
OSSC's Activities

- Input, Activity, Output and Target

Input
- Production System
- PostgreSQL
- Question from Group

Activity
- Consultation
- Evaluation
- Tech. Support
- Development

Output
- Report
- Tools
- PostgreSQL

Target
- Group Company
- PG Community
Evaluations

What characters to know?

- Most systems are OLTP not OLAP
- Types of Transactions; read/write intensive
- TPC C and TPC W models are used
  - C model (DBT-2): write, I/O intensive
  - W model (DBT-1): read, CPU intensive
  - Other models: pgbench, DBT-3

Thru-put and stability

- Peak performance test (3Hr. Workload > 90%)
  - CPU scalability evaluated.
- Long-run test (72Hr. 70% workload)
  - observe stability during vacuum and checkpoint
Results on throughput

- Results of PostgreSQL and other DBMS.

- Help adapting PostgreSQL for production systems having particular population and frequent requests.

<table>
<thead>
<tr>
<th></th>
<th>8.2</th>
<th>8.3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TPC-W WIPS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rd:wrt = <strong>8:2</strong></td>
<td>1700tps</td>
<td>2100tps</td>
</tr>
<tr>
<td><strong>TPC-W WIPSo</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rd:wrt = <strong>5:5</strong></td>
<td>1100tps</td>
<td>2100tps</td>
</tr>
<tr>
<td><strong>TPC-C</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rd:wrt = <strong>1:9</strong></td>
<td>123tps</td>
<td>165tps</td>
</tr>
</tbody>
</table>

Equiments used for evaluations;
[TPC-W] Server: HP DL380G5 (Xeon 5160 3GHe, 12GB memory), Storage HP MSA500
[TPC-C] Server: DL580G4(Xeon DC 3.4 GHz 4 core, 24GB memory), Storage HP MSA 1000
[OS] Redhat Enterprise Linux 5 update 1
Values are gotten from 48 hours execution and displayed in average.
Results on CPU scalability

- Many cores CPU be commodity
  - 4-8 for middle-scale, 32 for large-scale.
  - Good scalability up to 8 cores for 8.3 and after.

CPU Scalability of PG 8.3 in case of DBT-1
Results on throughput

- Show the results on PostgreSQL and other DBMS.
  - Help choosing PostgreSQL for production systems having particular population and frequent requests.
  - PostgreSQL usable to replace proprietary DB
  - Average performance sufficient
  - How about transitional performance?
    - Stability of performance
Significance of Performance Stability

- If performance is not stable,
  - Query not answered for a long time → trouble
  - Difficult to guarantee minimum performance (e.g. longest response time)
- Observe stability with long-run test.
  - Vacuums and checkpoints done many times
  - Long-run stability evaluated with TPC-W
    - Workload itself stable against time
    - TPC-C increases data population and (in result) workload as time passes.
Results on Stability test (1)

- Response stabilized in 8.3
  - 8.2 (Left) glitches caused by checkpoints
  - 8.3 (Right) glitches reduced 20% of 8.2
- Glitches in 8.2 concerned to be obstacle for production systems.
Results on Stability test (2)

- Influence of dead tuples and vacuum op.
  - autovacuum=off (Left) 8.2 reduces performance
  - autovacuum=on (Right) both cause glitches

* 2 figures above are referred from 'Let's Postgres'
  http://lets.postgresql.jp/documents/case/ntt_comware/2
Results on Stability test (3)

- Improvement by cost-bases vacuum
  - Cost-based vacuum smooths through put
    - Vacuum prolonged to 33 hrs from 2 hrs prev. case

* the figure above is referred from 'Let's Postgres' http://lets.postgresql.jp/documents/case/ntt_comware/2
Summary on Evaluation

- PostgreSQL 8.3 shows enough good performance for our production systems having middle scale DB.
  - Since 8.3, introduction has been accelerated.
  - Vacuum with HOT and cost-based, time-spread checkpoint are important improvements.
    - Improved vacuum reduces operation design.
  - Remaining issues...(including other evaluations)
    - Scalable CPU handling (e.g. for 64 cores)
    - More efficient I/O handling (an evaluation on I/O bandwidth shows that of PostgreSQL is 4 times as commercial DBMS)
    - Shorter recovery time.
Evaluations on Operation

- How to evaluate Operation feature?
  - Interview: Operating companies have OSS dept., which we interview their needs.
  - Tech. Support: FAQs hint improvement requests.
    - e.g. PITR operations (setting, take backups, erase dated archive files etc)
- What to evaluate about?
  - Data Handling: backup (restore), data-load
  - Monitoring: slow queries, statistics etc.
- This process gives us important insights.
  - Information is qualitative not quantitative as throughput, it gives us insights for improvements.
Evaluations on Data Operation

- Backups:
  - Logical: pg_dump itself is good enough but not widely used because it doesn't guarantee committed transactions (by nature).
  - Physical: PITR method furnished since 8.0, but not easily used because its complex operation.

- Data loading:
  - COPY is useful but not enough fast.
    - In old versions, COPY was not fast enough comparing commercial DBMS.
  - Data loading used daily to speed batch jobs partly done by offline.
Evaluations on Data Operation

- Usage of fast Data loading:
  - DB migration for production system done limited time.
  - Speed batch jobs partly done by offline (below)

- **Unload (dump) is fast enough**
- **Load is not fast as commercial DB**
Evaluations on monitoring

- Importance of various Monitoring:
  - PostgreSQL provides useful data for tuning and trouble shoot via queries, we need external tool that get and collect PostgreSQL's internal statistic data.
    - Some trouble difficult to reproduce, acquired data used for post-mortem analysis by OSSC staff.

<table>
<thead>
<tr>
<th>Type</th>
<th>Usage</th>
<th>Means</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Living</td>
<td>Fail over Cluster</td>
<td>Process id check</td>
<td>OK</td>
</tr>
<tr>
<td>Slow query</td>
<td>Trouble shoot</td>
<td>Operation logs</td>
<td>OK</td>
</tr>
<tr>
<td>Internal statistics</td>
<td>Trouble shoot</td>
<td>Query to PostgreSQL</td>
<td>Need external monitoring tool</td>
</tr>
</tbody>
</table>
Development

- improvement to PostreSQL core
  - Stability
  - Availability
- development of peripheral tools
  - Backup
  - Data loading
  - Monitoring tool
For performance stability

- NTT OSS Center donated some functionality for Vacuum and Checkpoints
  - Most of them were accepted to PostgreSQL core
    - Cost-based vacuum
    - multiple concurrent autovacuum processes
    - Checkpoints spread out (smooth checkpoint)
  - These help PostgreSQL performance stability, which accelerate introduction.
Improve Availability

- About 1/3 NTT systems require fail over within 1 min.
  - Fail over cluster with shared disk requires fsck when switching, which takes several minutes.
  - Replication clusters using query replication guarantee loss-less fail over, however impose incompatibilities with original PostgreSQL.

- We start to develop stream replication about 2006.
  - At first non OSS product, changed OSS in 2008.
  - Proposal at 2008 PG Con (Mr. Fujii)
  - Streaming replication was implemented in 9.0 (2010)
  - Synchronous mode will be in 9.1
Peripheral software for HA has been developed

- To switch server when failure, Linux-HA (Pacemaker) is used
  - NTT OSSC also uses Pacemaker for High-availability system
- Pacemaker's Resource Agents

**Diagram:**
- **Master**
  - PostgreSQL
  - RA
  - Pacemaker
  - Hardware + OS
- **Slave**
  - PostgreSQL
  - RA
  - Pacemaker
  - Hardware + OS

**Synchronous Replication**
Application of HA Cluster

- HA Cluster including PostgreSQL with synchronous Replication expected to be introduced to more reliable systems;
  - Telecommunication support systems
  - Trading systems
  - Web commerce with high-availability
pg_rman ; backup tool

- Motivation ; FAQ.
  - PITR is powerful but complex
    - When expire old archival files?
    - How and from which archives to restore?
- Solution
  - Tool embedded operation know-hows
  - Pg rman
    - Takes and restores all necessary files to recover with one command
    - Back-up files are cataloged.
  Many know-hows

http://code.google.com/p/pg-rman/
pg_bulkload; data loader

- **Motivation**: Data migration speed up.
  - Data migration in production systems should complete scheduled time
    - Data migration duration dominates DB size limit for PostgreSQL
    - COPY was not enough quick (ca. 2005)

- **Solution**
  - Dedicated Loading Tool; pg_bulkload
    - Initial and append modes
    - Direct and parallel load
    - Fast index creation
pg_bulkload; data loader

- Pg bulkload is as 2-3 times fast as COPY

![Loading Time Comparison Chart]

- Bulkload and others
- pg_bulkload
- COPY without indexes
  - Table
  - PK
  - Index
- COPY with indexes

[http://pgbulkload.projects.postgresql.org/index.html](http://pgbulkload.projects.postgresql.org/index.html)
pg_statsinfo; monitoring Tool

- **Motivation**
  - **Effective support activity**
    - Post-mortem analysis
  - **Handy performance monitor**
    - Predict performance trouble beforehand

- **Features**
  - **Statistics collector with low power-consumption**
    - Monitoring system runs (partially) on the Production system.
  - **Visualize statistics**
  - **Programmable alert**
Collected data generate 'Report' and 'Alert'

- Configuration: statistics collector + message filter for alert
- Lower consumption: overhead < 3%
Support Activities

- Technical Q and A
  - A few hundreds questions answered a year within 3 business days
  - Various questions
    - From usages to trouble issues
- Consultation
  - Migrate from Proprietary DBMS
    - Migration know-hows are cataloged (ca. 50 items; “how to rewrite synonym in Oracle”)
  - Performance tuning aids
    - Evaluate particular workloads and suggest tuning methods.
NTT Cases

- OSS Center has introduced PostgreSQL more than 100 systems; High light specs as follows
  - **DB Size:** Largest 3TB.
  - **Frequency:** 1000 TPS (or more)
  - **HA:** fail over takes less than 1 min. (15” measured)
- Statistical Facts expressed
  - **Individual cases are not allowed to open**
View of NTT's Production systems

- Target of OSS introduction in NTT in-house system
  - NTT runs several hundreds systems
  - Survey shows 80% of system can be introduced PostgreSQL

- Trend of PostgreSQL introduction
  - From small-scale and less available system to large-scale and high available ones

Database size [Byte] vs. Availability

- 100GB: 99.99% available, DB fail over 10 min.
- 1TB: 99.999% available, DB fail over within 1 min.
- 10TB: 99.999% available, DB fail over within 1 min.

System categories:
- Subscriber manage
- Facilities manage
- Sale assistance
- Personnel, Allowance
- Back office

- 289x56 - 303x36
- 448x56 - 462x36

- 99.99% available
- DB fail over 10 min.
- 99.999% available
- DB fail over within 1 min.
Trend of PostgreSQL Introduction

- About 130 systems introduced PostgreSQL
  - 30-40 systems a year.

Introduction to NTT Groups' System

<table>
<thead>
<tr>
<th>Year</th>
<th>Introduced</th>
<th>Accumulated</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>2007</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>2008</td>
<td>40</td>
<td>70</td>
</tr>
<tr>
<td>2009</td>
<td>50</td>
<td>120</td>
</tr>
<tr>
<td>2010</td>
<td>70</td>
<td>190</td>
</tr>
</tbody>
</table>
Expectation

- Federated DB
  - Large DB system consists of many databases.
- Performance for 'internal cloud'
  - Efficient processing is essential
    - CPU scalable
    - I/O bandwidth
- More installation via community
  - Many installations improve quality
  - Many use cases accelerate introduction
End

Thank you for your attention