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

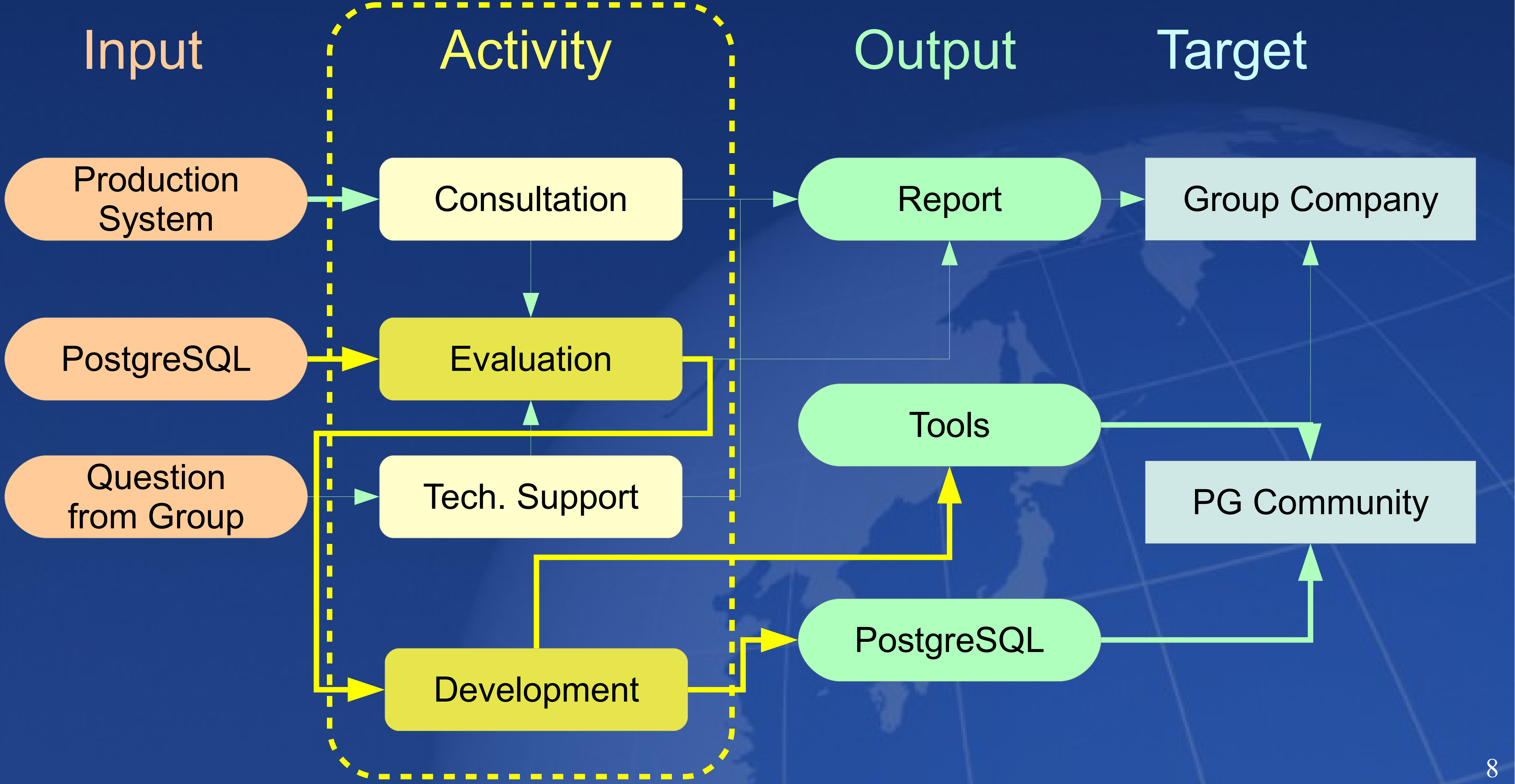
- Mission:
  - Reduce TCO with OSS; replacing proprietary software
    - Support NTT Group companies' OSS usage
      - Q and A
      - Consultation
    - Develop / improve OSS
  - Center of OSS competence in NTT Group.
- Established in Apr. 2006.
- Location: Shinagawa Tokyo.

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



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 through put

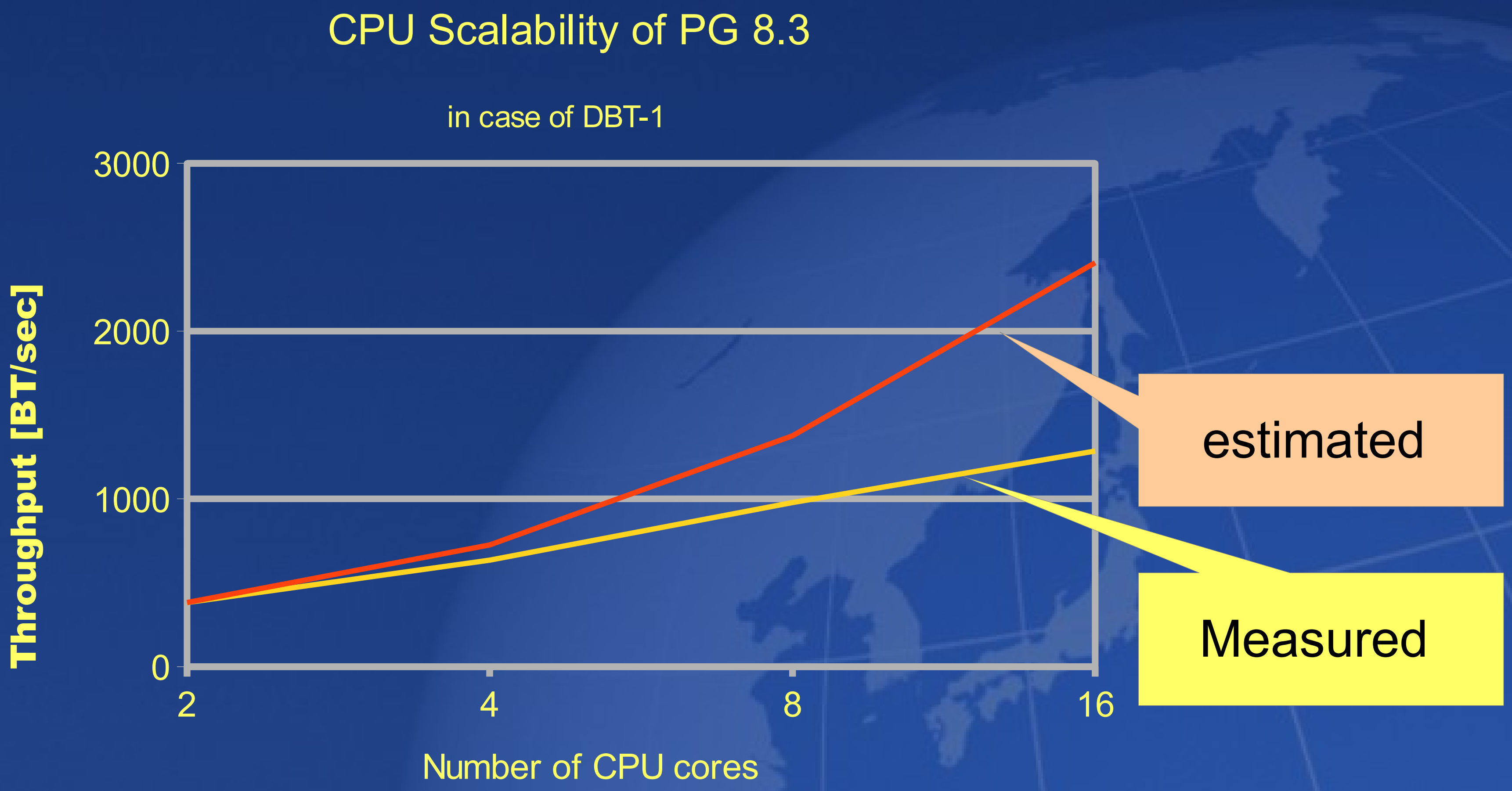
- Results of PostgreSQL and other DBMS.
  - Help adapting PostgreSQL for production systems having particular population and frequent requests.

	8.2	**8.3**
TPC-W WIPS rd:wrt = **8:2**	1700tps	2100tps
TPC-W WIPSo rd:wrt = **5:5**	1100tps	2100tps
TPC-C rd:wrt = **1:9**	123tps	165tps

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.



Results on through put

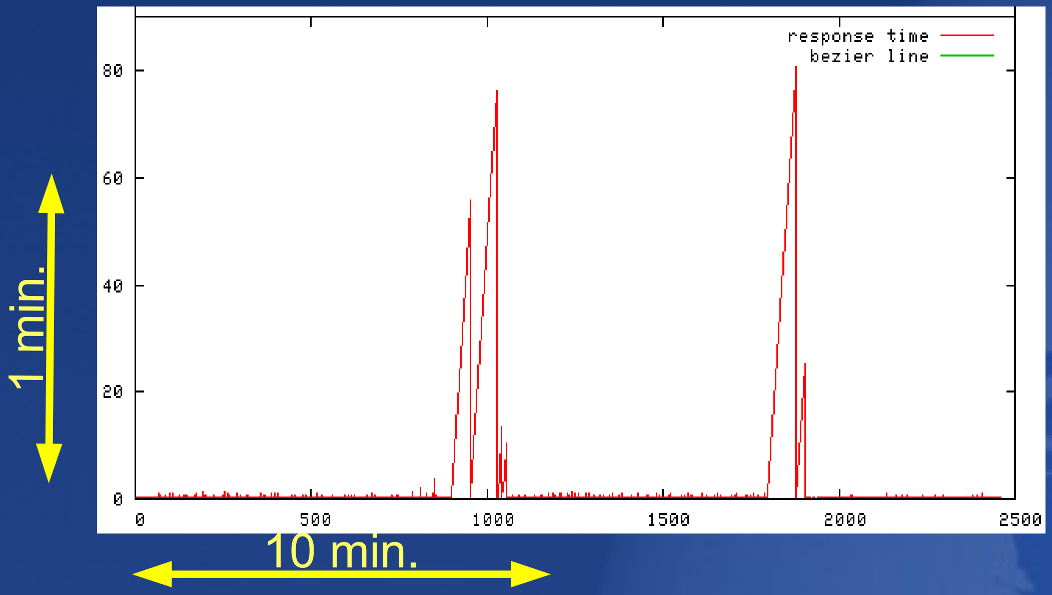
- Show the results on PostgreSQL and other DBMS.
  - Help choosing PostgreSQL for production systems having particular population and frequent requests.
  - PostgreSQL usable to replace proprietary DB
- Average performance sufficient
  - How about transitional performance ?
    - Stability of performance

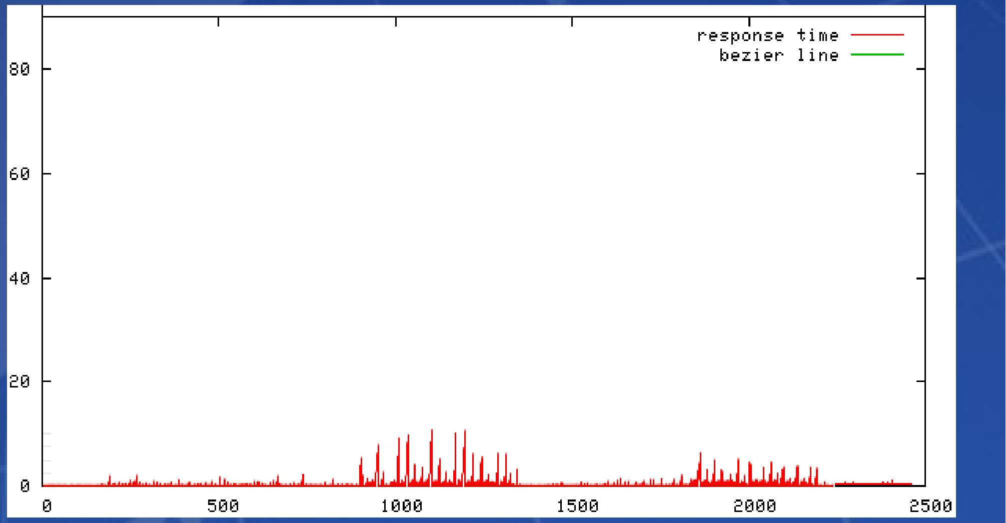
Significance of Perfomance Stability

- 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)

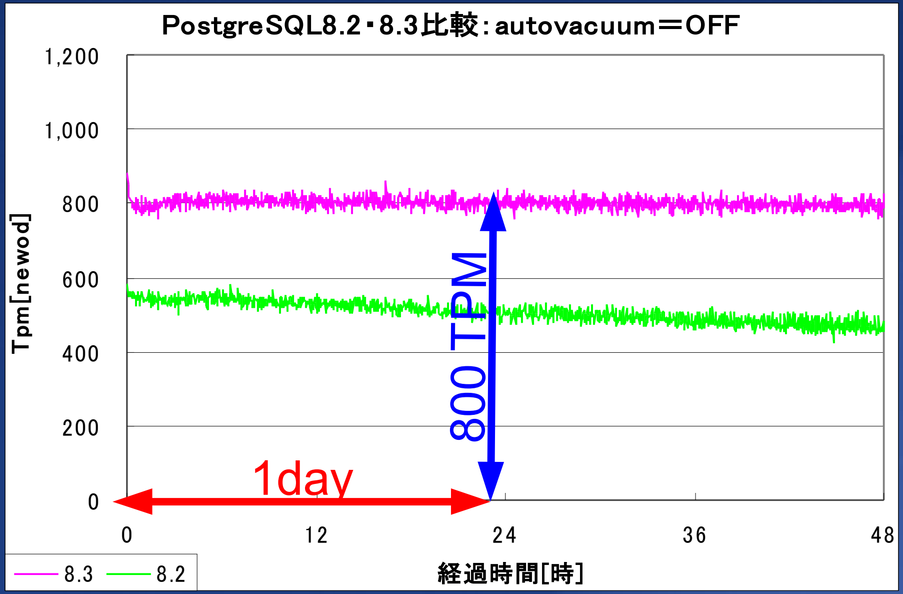
- Response stabilized in 8.3
  - 8.2 (Left) glitches caused by checkpoints
  - 8.3 (Right) glitches reduced 20% of 8.2
- Glitches in 8.2 concerned to be obstacle for production systems.

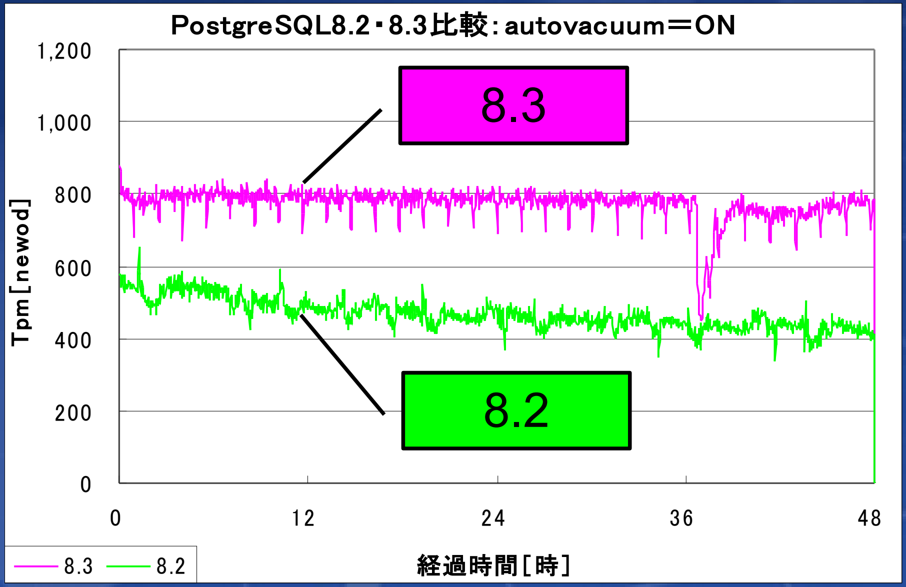




Results on Stability test (2)

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

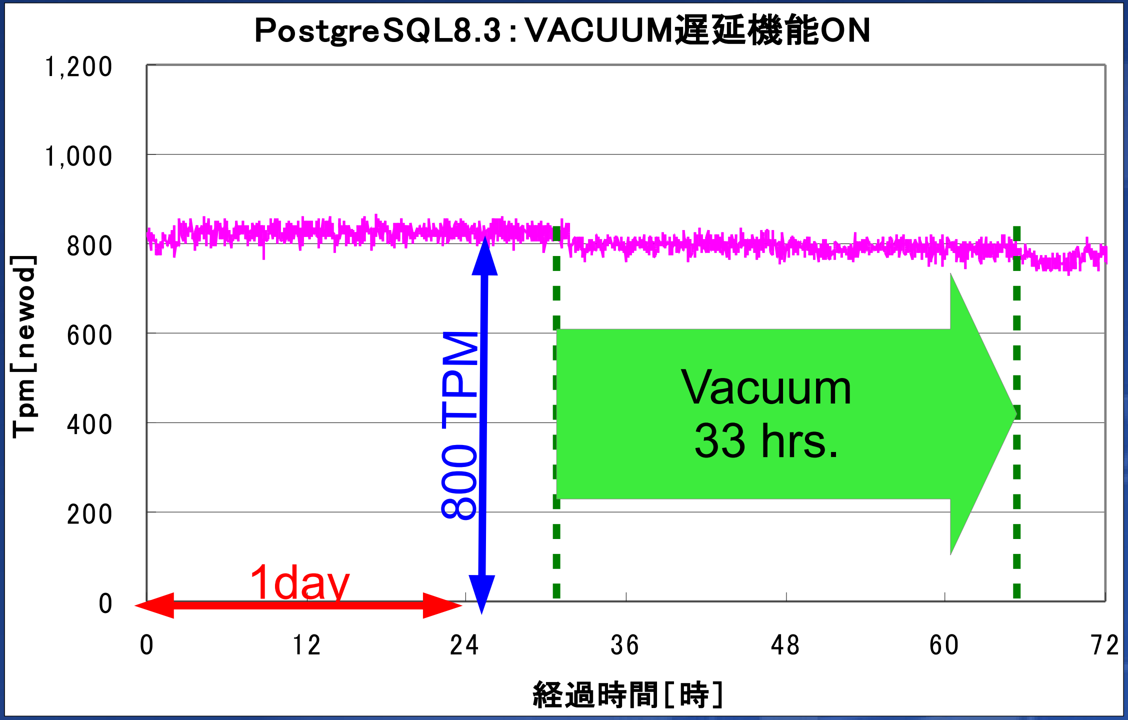




* 2 figures above are referred from 'Let's Postgres'
http://lets.postgresql.jp/documents/case/ntt_comware/2

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

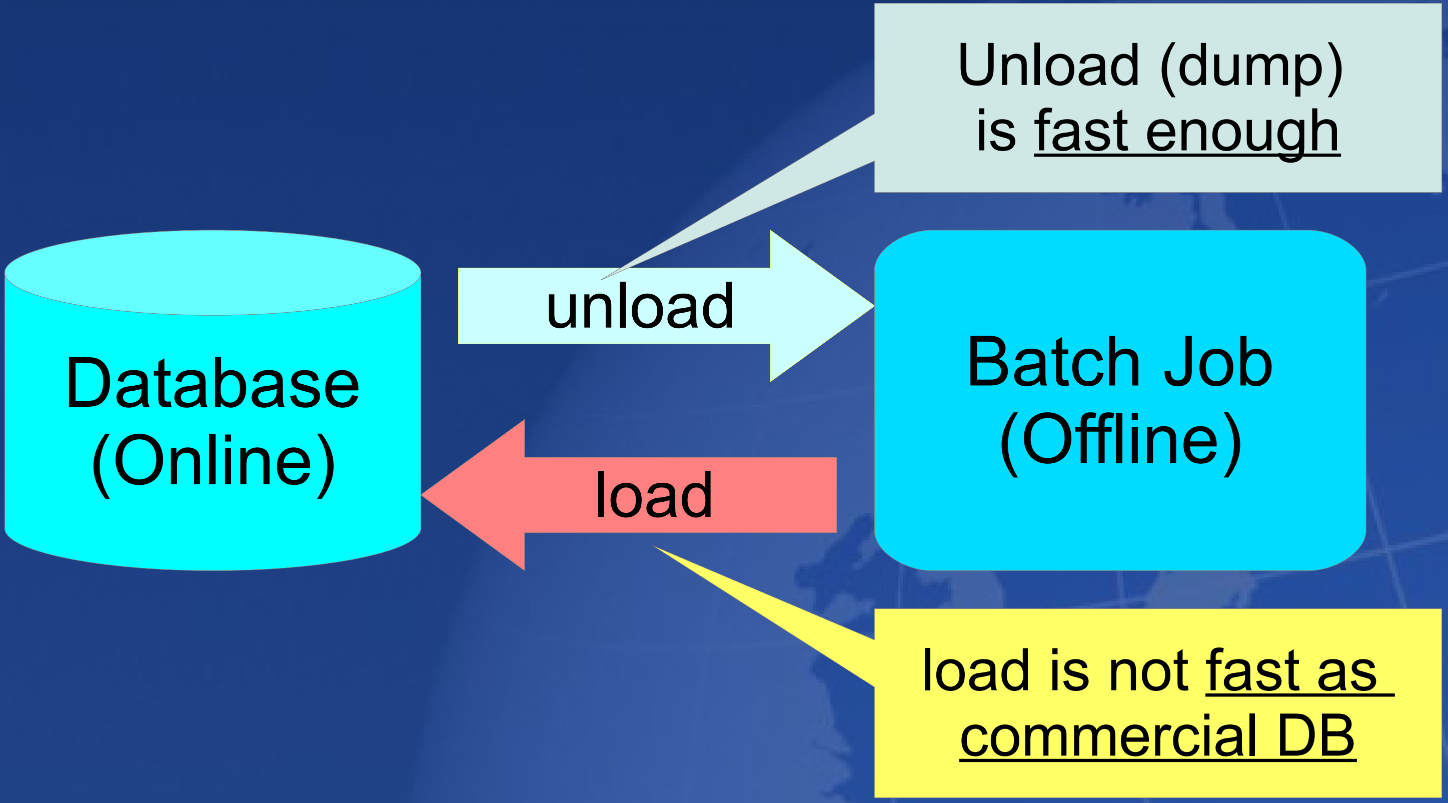
- How to evaluate Operation feature?
  - Interview: Operating companies have OSS dept. , which we interview their needs.
  - Tech. Support: FAQs hint improvement requests.
    - e.g. PITR operations (setting, take backups, erase dated archive files etc)
- What to evaluate about ?
  - Data Handling: backup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



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.

Type	Usage	Means	Status
Living	Fail over Cluster	Process id check	OK
Slow query	Trouble shoot	Operation logs	OK
Internal statistics	Trouble shoot	Query to PostgreSQL	Need external monitoring tool

Development

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

For performance stability

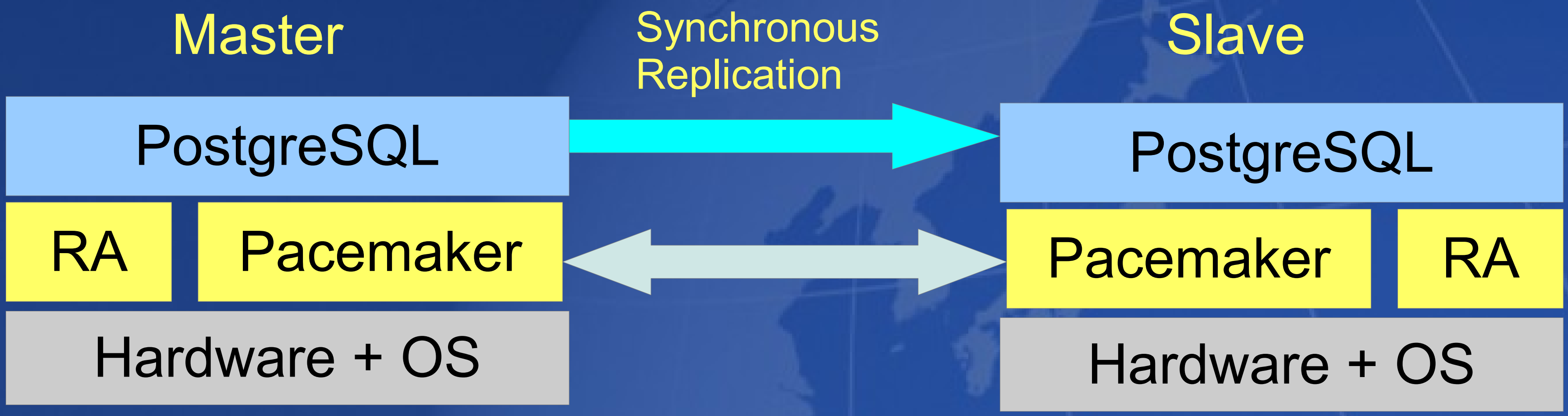
- NTT OSS Center donated some functionality for Vacuum and Checkpoints
  - Most of them were accepted to PostgreSQL core
    - Cost-based vacuum
    - multiple concurrent autovacuum processes
    - Checkpoints spread out (smooth checkpoint)
  - These help PostgreSQL performance stability, which accelerate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

Improve Availability

- About 1/3 NTT systems require fail over within 1 min.
  - Fail over cluster with shared disk requires fsck when swiching, 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

Improve Availability (2)

- 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



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?

Many know-hows

- Solution
  - Tool embedded operation know-hows
- Pg rman

http://code.google.com/p/pg-rman/

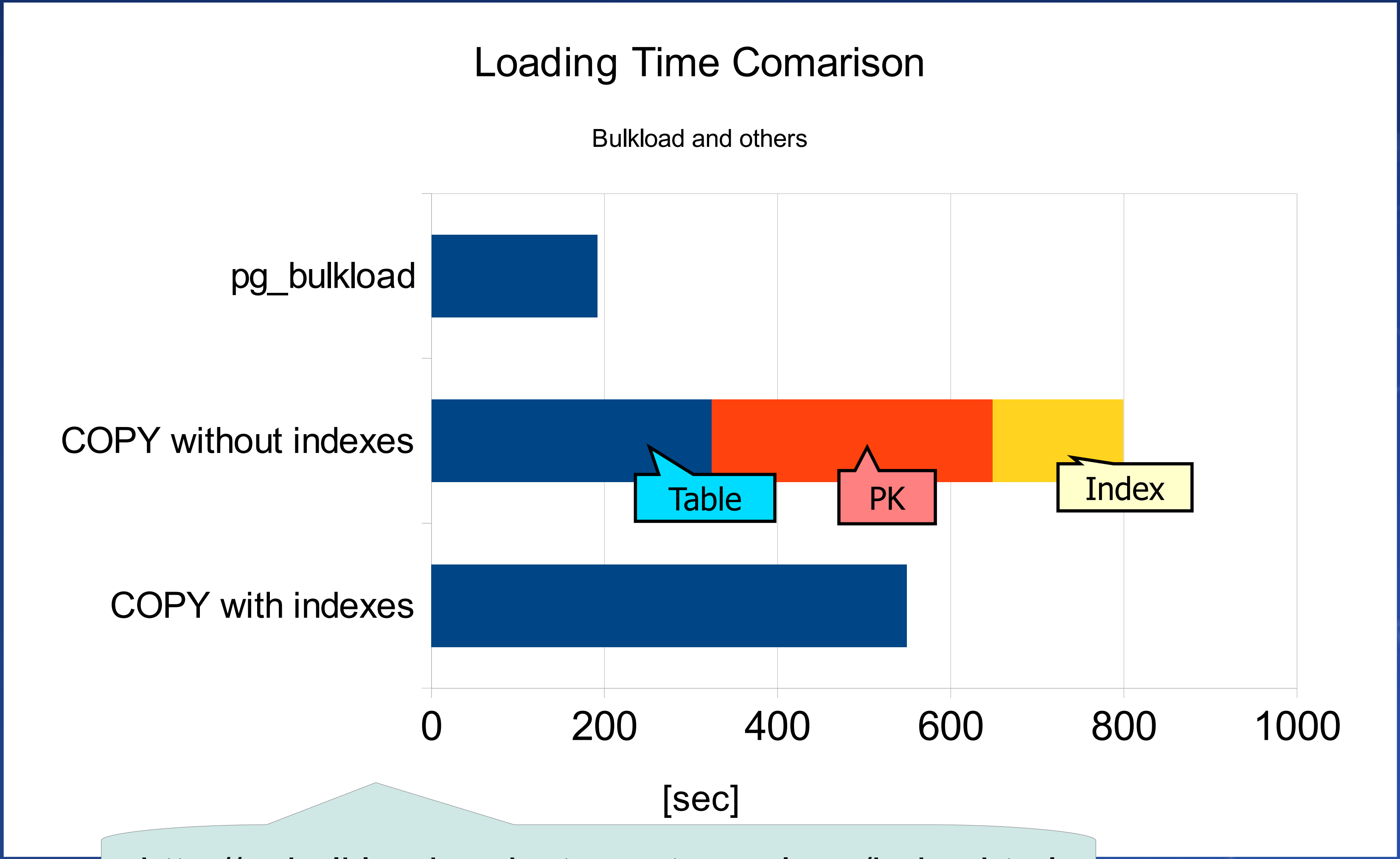
  - Takes and restores all necessary files to recover with one command
  - Back-up files are cataloged.

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



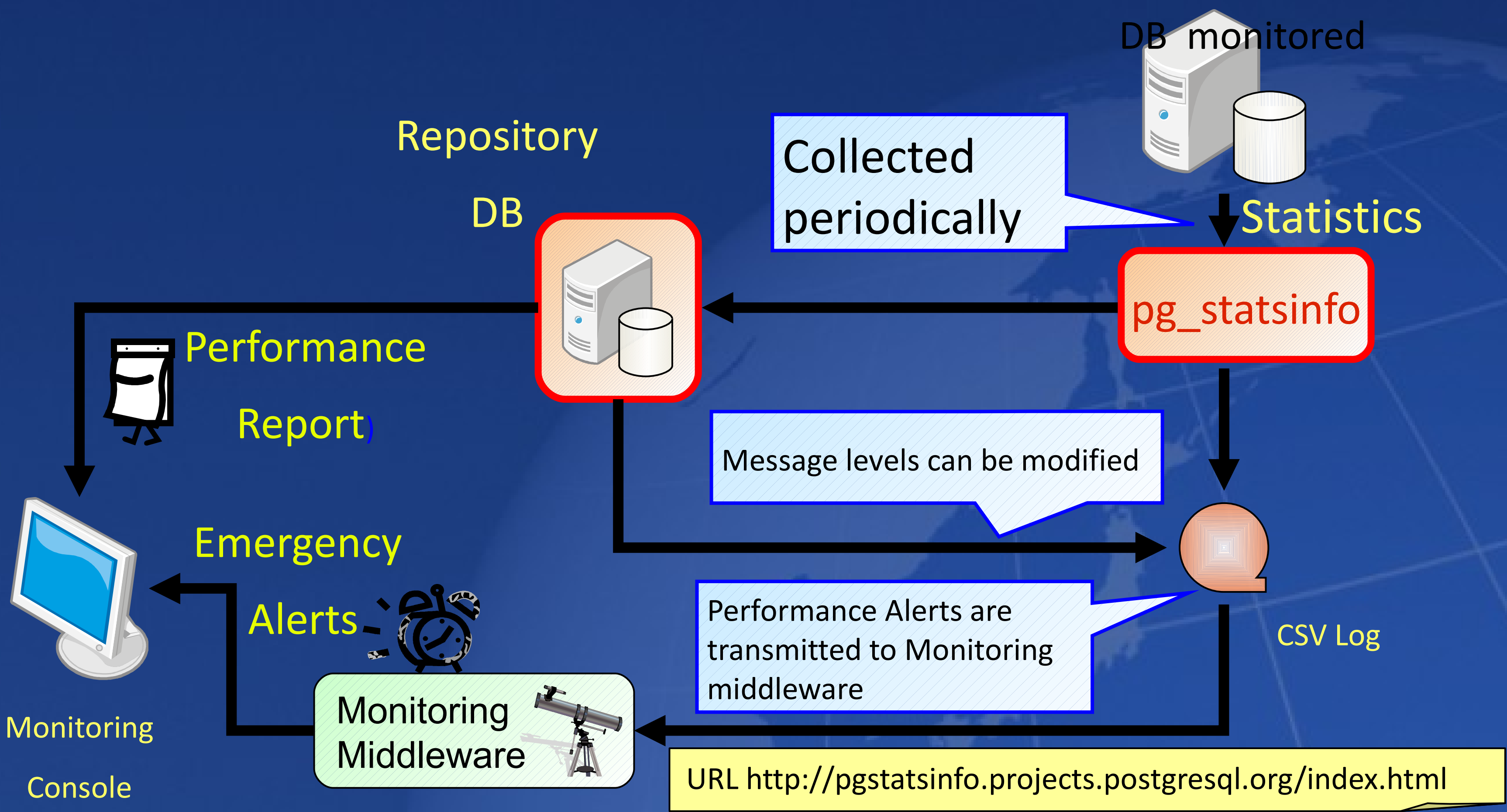
http://pgbulkload.projects.postgresql.org/index.html

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

pg_statsinfo; schematic diagram

- Collected data generate 'Report' and 'Alert'
  - Configuration: statistics collector + message filter for alert
  - Lower consumption: overhead < 3%



URL http://pgstatsinfo.projects.postgresql.org/index.html

Support Activities

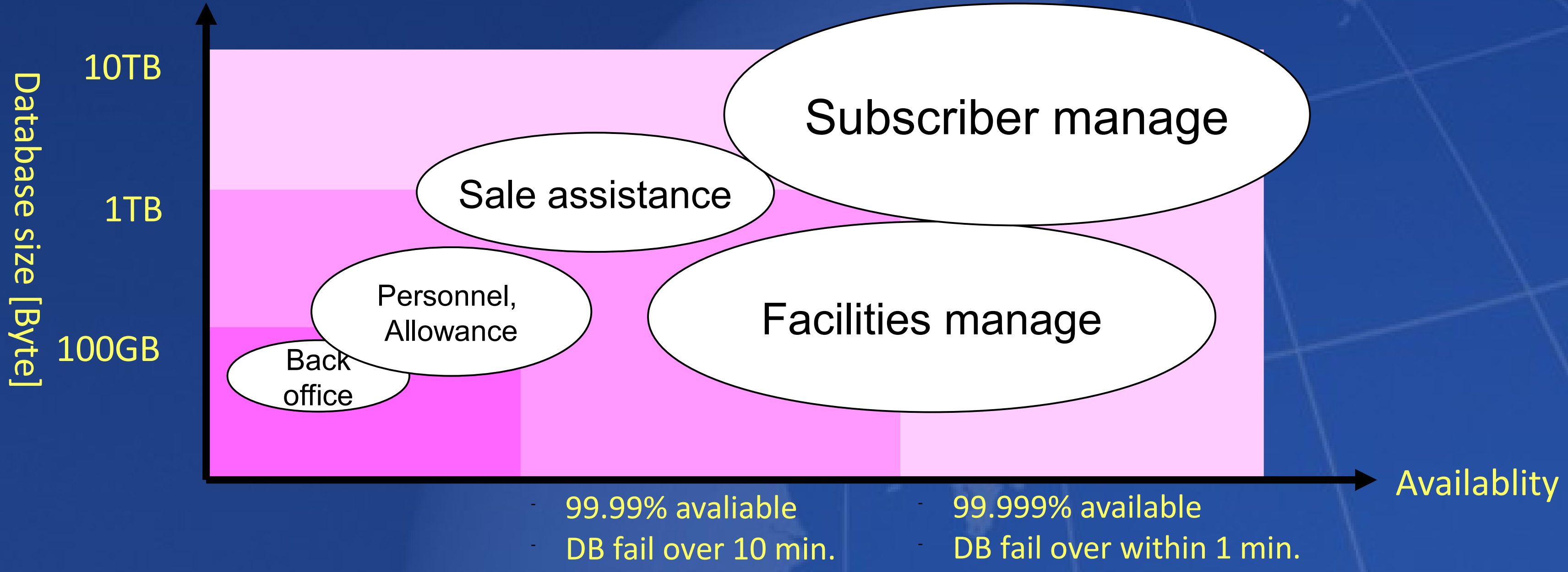
- Technical Q and A
  - A few hundreds questions answered a year within 3 business days
  - Various questions
    - From usages to trouble issues
- Consultation
  - Migrate from Proprietary DBMS
    - Migration know-hows are cataloged (ca. 50 items; “how to rewrite synonym in Oracle”)
  - Performance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

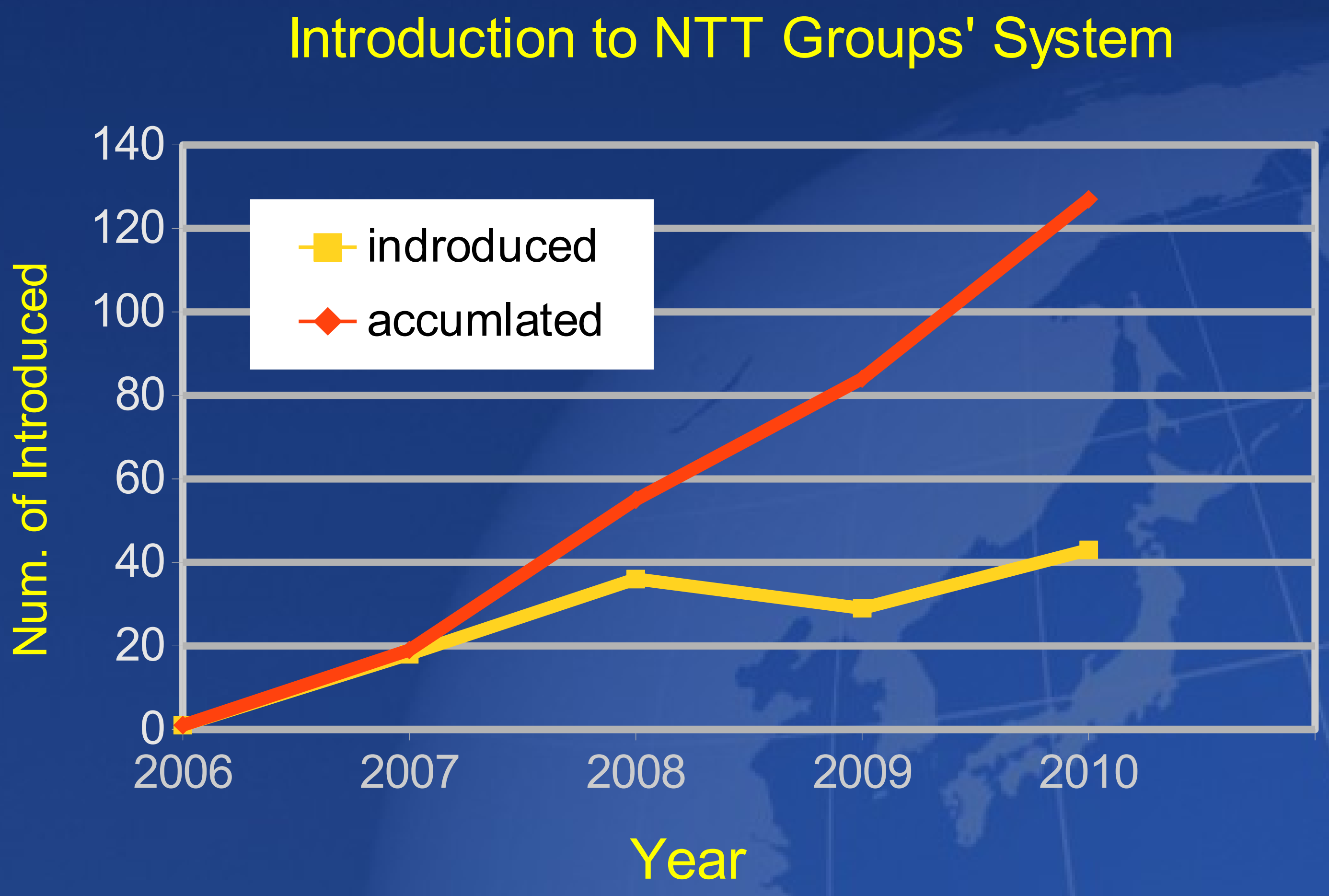
View of NTT's Production systems

- Target of OSS introduction in NTT in-house system
  - NTT runs several hundreds systems
  - Survay 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



Trend of PostgreSQL Introduction

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



Expectation

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

End

Thank you for your attention