

DB Ops

*Easy and Effective Operation for production systems
with PostgreSQL*

Tetsuo SAKATA &
Masahiko SAKAMOTO
NTT Open Source Software Center
17th May 2012

Agenda

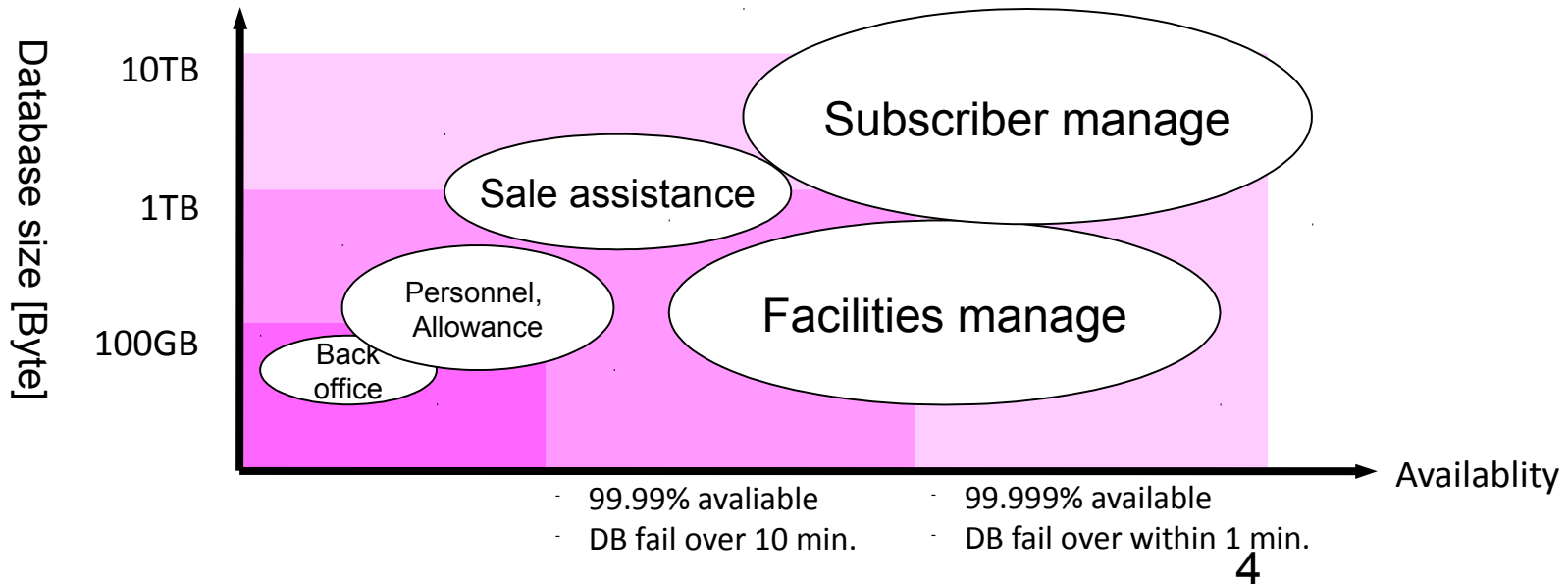
- **Introduce ourselves**
- **Needs to introduce PostgreSQL**
- **Map of peripheral tools**
- **Demonstraion**

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

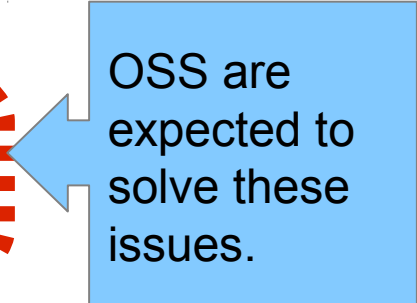
View of NTT's Production systems

- 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



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



OSS are expected to solve these issues.

Understand user needs;

How to introduce PostgreSQL?

- Information on performance
 - Show good and stable performance
 - Availability/reliability
 - Sizing Info. for hardwares (HDDs, CPUs etc.)
- **Operation capability**
 - Usability
 - compatibility with other operation tools
- Technical support
 - Q and A, trouble shooting, consultation, etc.

Essentials on Operations

- Tools support house-keeping tasks
 - temporal; (initial) **data loading, data migration**
 - daily; **back-up, performance observation, audit, maintenance**
- Skills to cope with such tasks
 - back-up via PITR
 - observe DB activities via statistics

not only tools but skills are required

Episode on data loading

- PostgreSQL COPY was not fast
 - it took 2 or 3 times that Oracle SQL*Loader did. PostgreSQL unable to introduce.
- COPY is too restrictive
 - even an error fails (huge) data loading. more flexible error handling desired.
- We need **fast and flexible data loader!**

We often consulted; migrate data from other DB and they may contain some invalid data. we have to filter out invalid data from the source, and correct them into valid data.

Episode on DB observation

- Elusive performance problem
 - reproduction takes long time
 - **data logger for *post-mortem analysis*** required.
 - which queries? how they planned?
resource consumption?

PostgreSQL's statistics are useful to know these facts. *Read* them and fix problems requires know-hows on PostgreSQL internals, with which production engineer is unfamiliar.

Episode on DB back-up

■ PITR is powerful. but...

■ procedures are (relatively) complex.

- Which archived logs are required to recover DB?
- When we can expire (huge) bunch of archive logs?
- How we can recover (crashed) DB?
(how to use the latest log files)

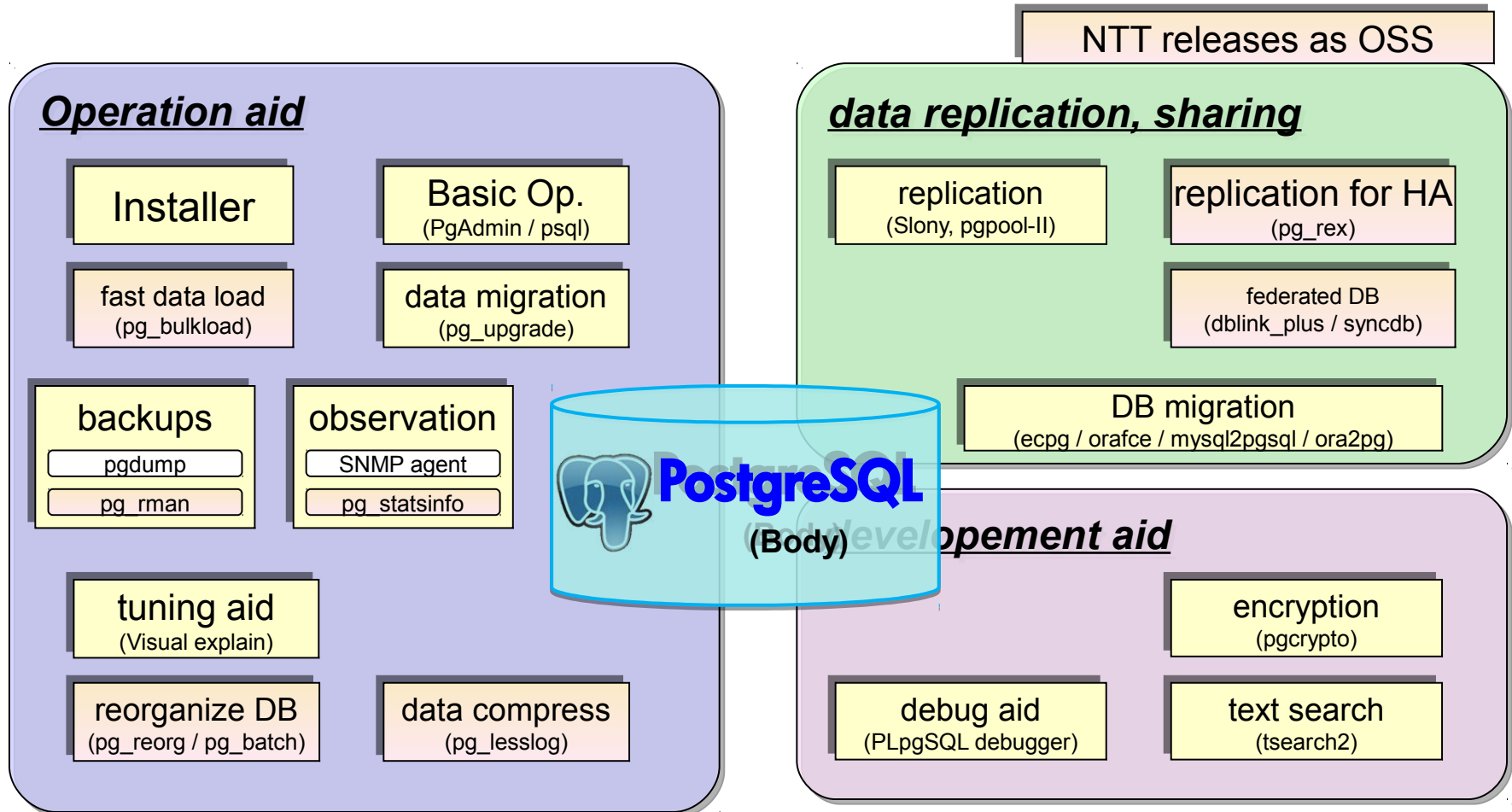
We often ask these questions.
Especially recovery procedure is complex and

Lessons from our episodes.

- Only User knows real issues
 - data loading include **invalid data**
 - DB activity data logger for **post-mortem analysis**
 - **Management tool** for multiple back-up data
- not only Simple operation but *Operation know-how* are required
 - which statistics should be logged?
 - which archive log data recover production data?
- user community share know-how to make tools (and ourselves :-) mature.

Overview of PostgreSQL Peripheral Tools

- Many peripheral tools are provided from 3rd party
- NTT OSS Center focuses on Op. tools



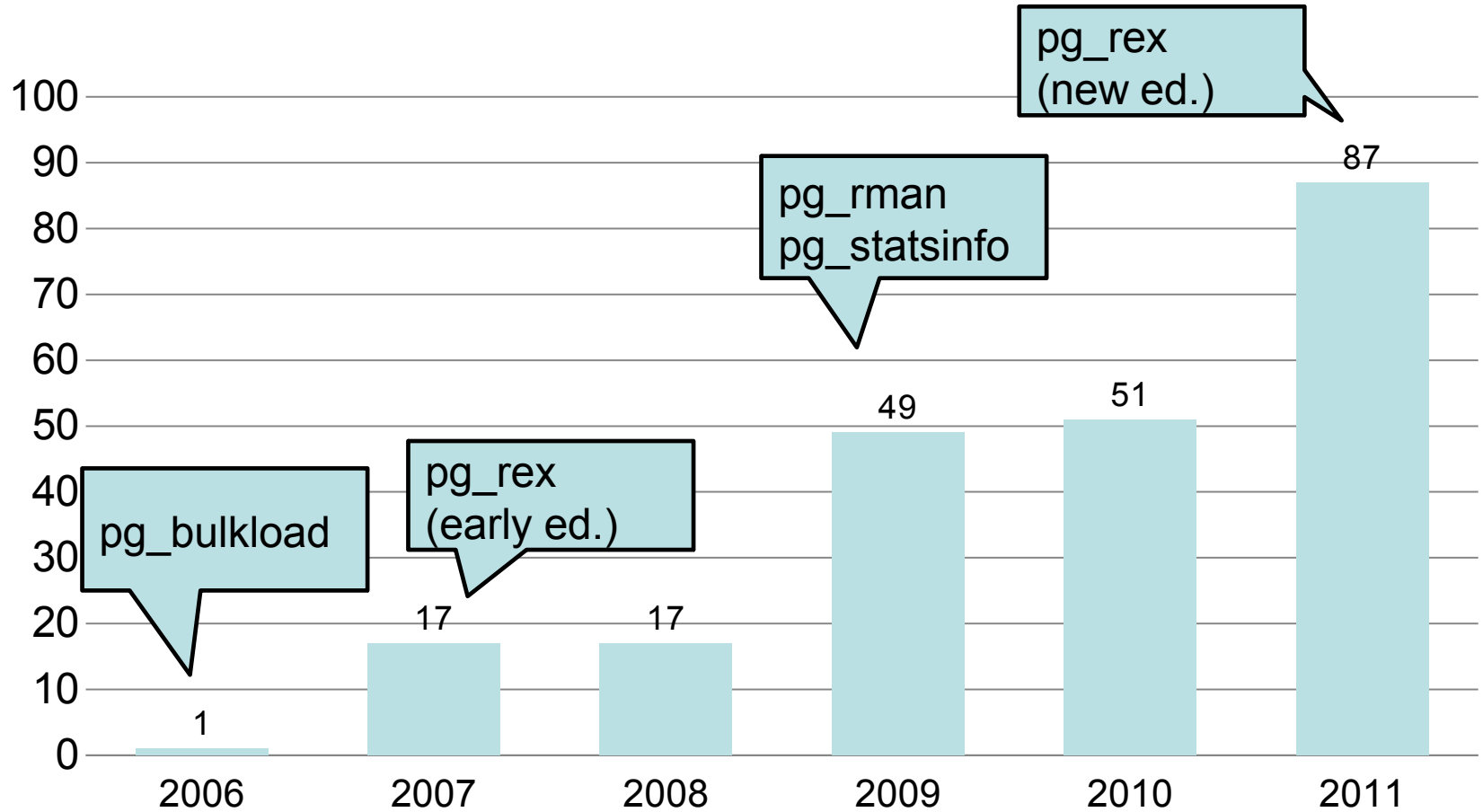
PostgreSQL introduction with Op. tools

- Most systems accompanied with Op tools below;
 - pg_statsinfo to shoot performance troubles
 - pg_rman to simplify backup procedures.
 - pg_reorg and bg_bulkload to reduce operation efforts.

Tools above have been developed to meet needs from our production engineers;
Must-have items for system integration.

PostgreSQL introduction with Op. tools

PostgreSQL introduced each year



Database Ops –demos–

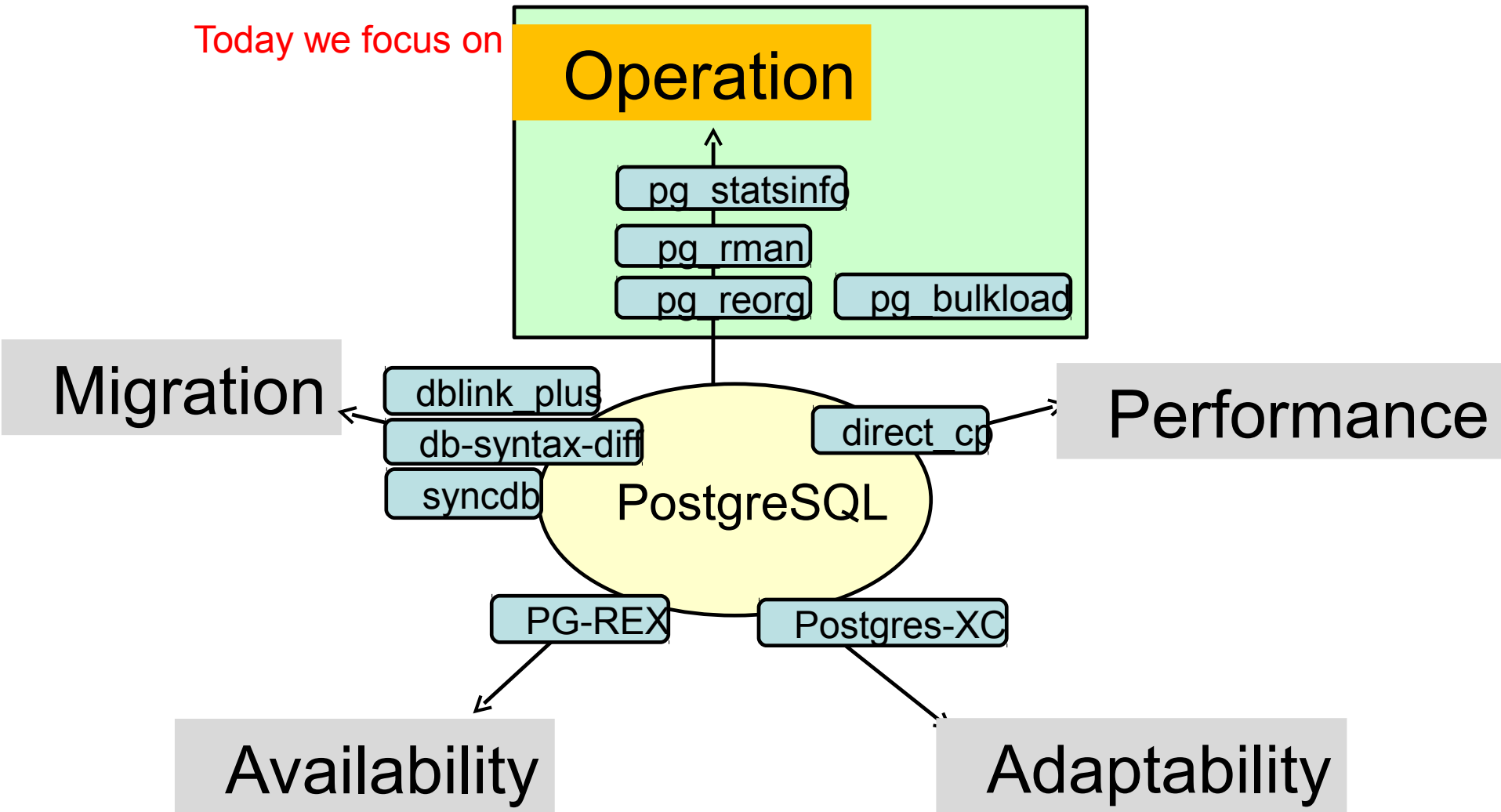
**PostgreSQL Conference 2012
17 May**

M.Sakamoto, NTT

(Recap) What we are working on

NTT Groups have been developing numbers of tools to enhance PostgreSQL, almost of which are available as open source software.

Today we focus on

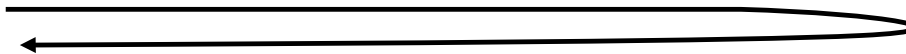


pg_statsinfo visualizes databases' statistics.

You can get **visualized summary** of your instances via web browser. **No special knowledge on system catalogs, and no maintenance tasks** are required to use pg_statsinfo.



Web access



dynamical HTML report

pg_statsinfo
(pg_reporter)

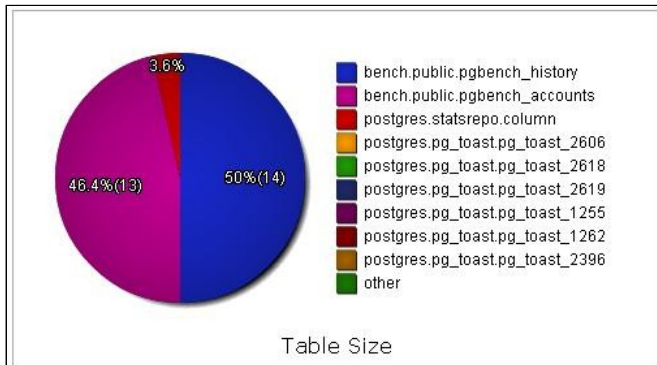
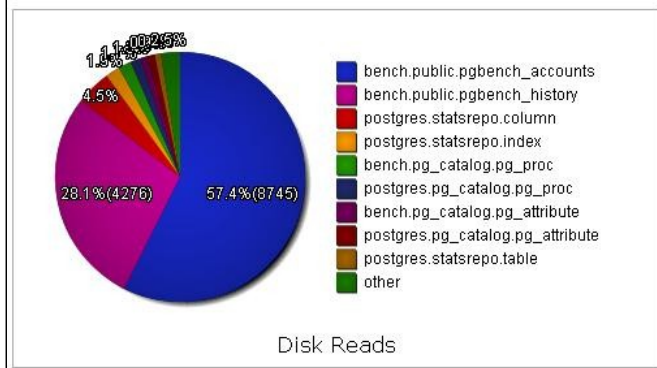
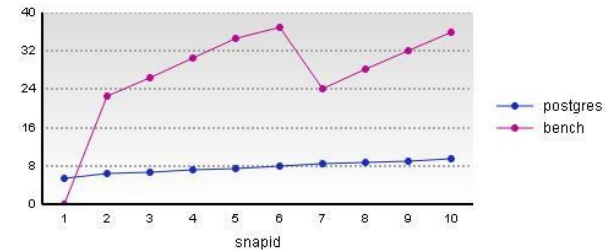


Table Size



Disk Reads

Database Size

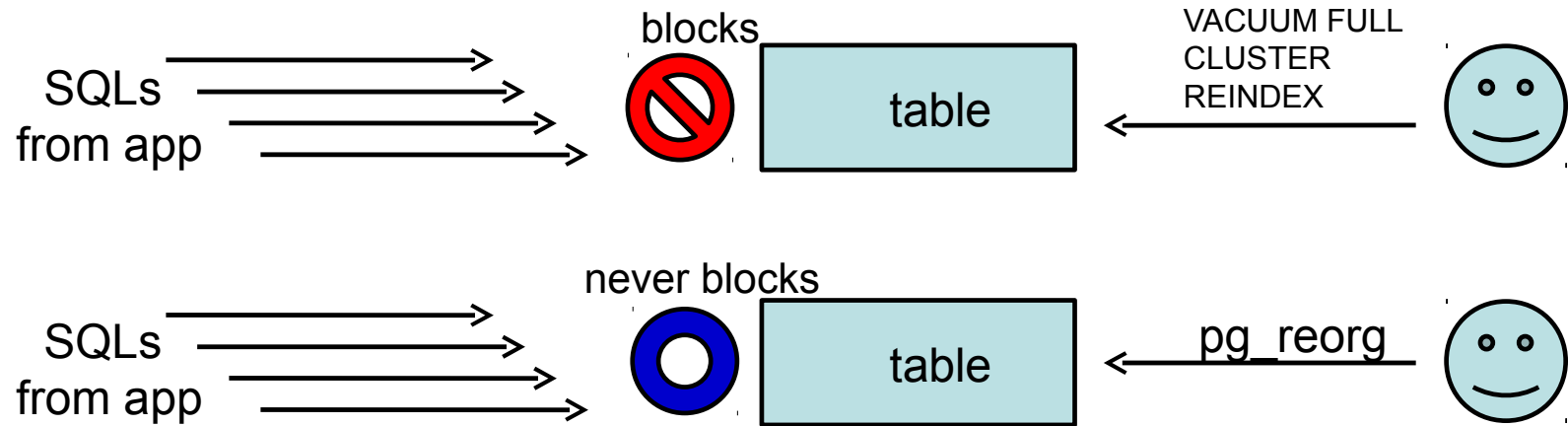


Report includes:

- most issued queries.
- checkpoint statistics
- autovacuum statistics
- and more.

pg_reorg reorganizes a table without blocking.

VACUUM FULL blocks other concurrent queries. On the other hand, pg_reorg doesn't block!

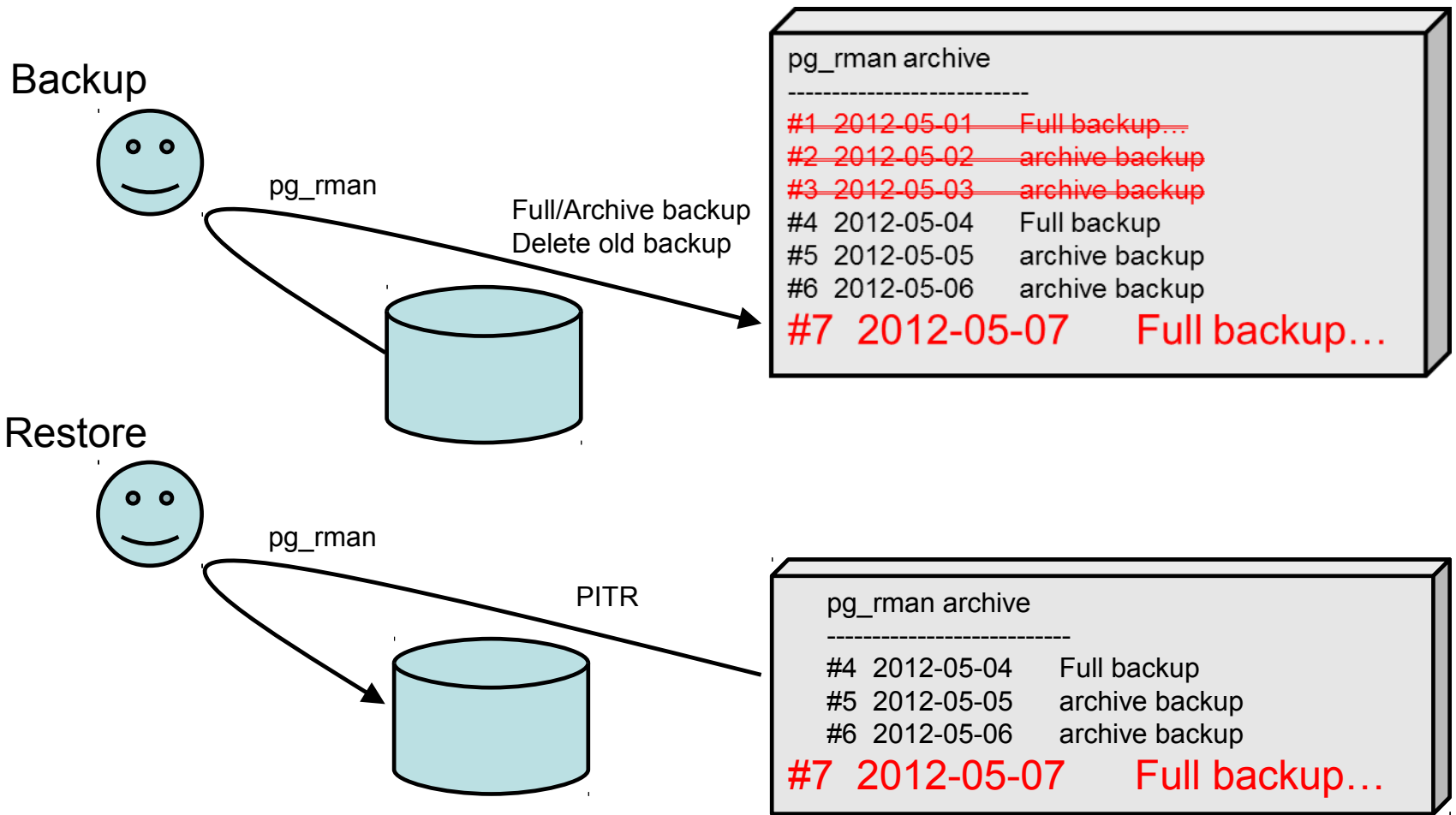


pg_reorg is a [good alternative of VACUUM FULL / CLUTER](#) in 365/24 systems.

pg_rman simplifies backup and PITR ops.

Everything related to backup is just a one-line operation when using pg_rman.

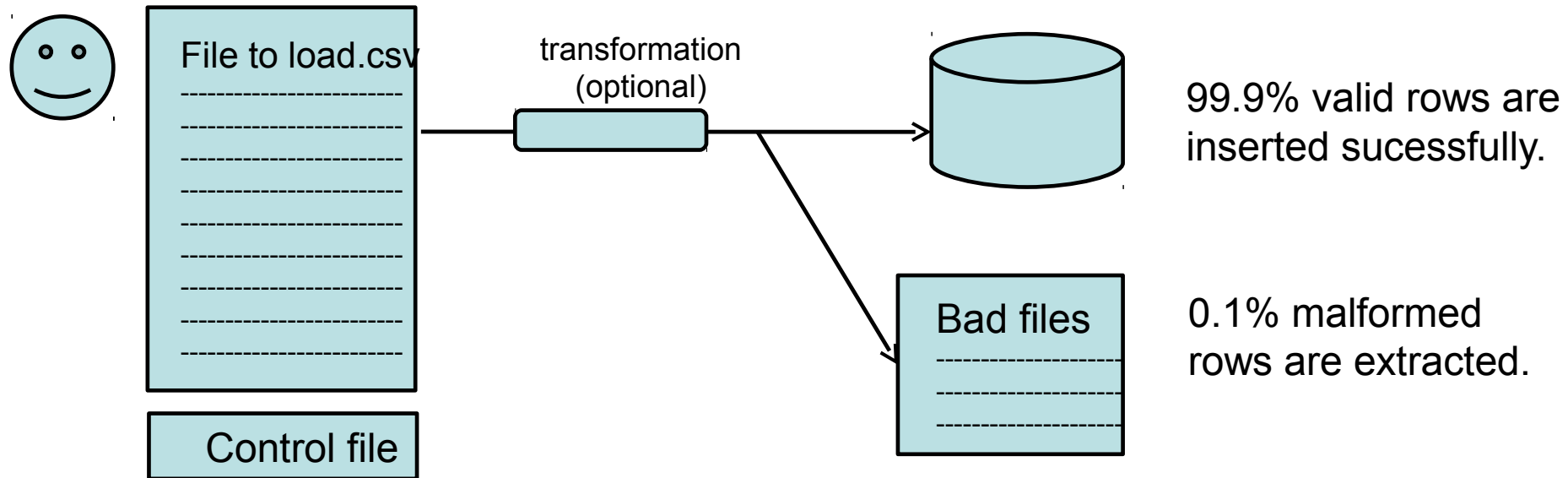
Managing generations and Incremental backups are outstanding features!



pg_bulkload can load skipping malformed rows.

COPY command just succeeds or not, which makes it very difficult to load a huge number of rows with a little portions of malformed rows.

When pg_bulkload finds some malformed rows, it puts them to bad files and proceeds loading.



Demos

Thanks for listening.

Softwares NTT contributes

Software	Available at
pg_statsinfo	http://pgstatsinfo.projects.postgresql.org/index.html
pg_reorg	http://reorg.projects.postgresql.org/index.htm
pg_rman	http://code.google.com/p/pg-rman
pg_bulkload	http://pgbulkload.projects.postgresql.org/index.html
dblink_plus	http://sourceforge.net/projects/interdbconnect/
syncdb	http://sourceforge.net/projects/interdbconnect/
db-syntax-diff	https://github.com/db-syntax-diff/
Postgres-XC	http://postgres-xc.sourceforge.net/
PG-REX	http://sourceforge.jp/projects/pg-rex/
direct_cp	Coming soon