PGCluster-II

Clustering system of PostgreSQL using Shared Data

Atsushi MITANI - mitani@sraw.co.jp

HA

First Italian PostgreSQL Day

PGDay 2007 – July 6,7 2007 – Prato, Italy

Agenda

Requirement

PGCluster

New Requirement

PGCluster-II

Structure and Process sequence

Pros & Cons

As a background

Requirement

PGCluster

New Requirement

PGCluster-II

Structure and Process sequence

Pros & Cons

Original requirement

- **Target application**
  - Web application
  - Heavy session load
- **High availability**
  - with ordinary servers
  - No down time
- **High performance for data read**
  - More than 90% sessions were data read query.

First step

Requirement

PGCluster

New Requirement

PGCluster-II

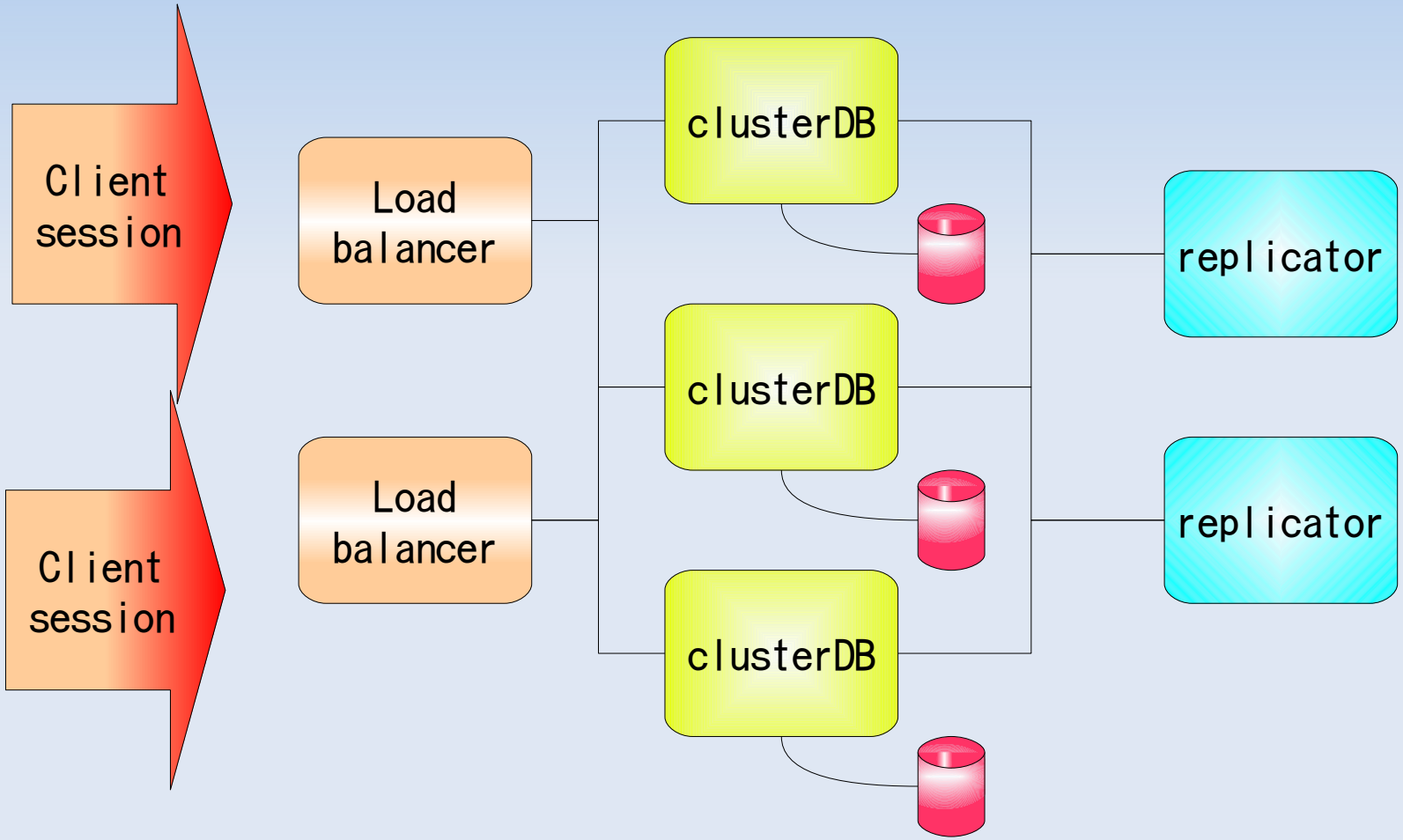
Structure and Process sequence

Pros & Cons

PGCluster(2002-)

- Synchronous & Multi-master Replication system
- Query based replication
  - DB node independent data can replicate
    - now(),random()
- No single point of failure
  - Multiplex load balancer, replication server and cluster DBs.
- Automatic take over
  - Restore should do by manually
- Add cluster DB and replication server on the fly.
  - Version upgrade as well

Structure of PGCLuster



Pros & Cons of PGCluster

- Enough HA
- Enough performance
  - for data reading load
- Cost
  - Ordinary PC servers
  - BSD license SW

- Performance issue
  - Very bad for data writing load
- Maintenance issue
- Document issue

5 years later

Requirement

PGCluster

New Requirement

PGCluster-II

Structure and Process sequence

Pros & Cons

Requirement is changed

- **Target application**
  - Web application
  - OLTP application
- **HA and HP**
  - HP is required even for data write
  - Service stop is not allowed

Coexistence of HA and HP

- **HA and HP conflict each other**
  - **HA** required **redundancy**
  - **HP** required **quick** response
- **Performance point of view**
  - **Replication** scales for data reading (not writing)
  - **Parallel query** has effect in both
    - However it is not easy to add redundancy (HA).
  - **Shared Data** Clustering also scales for both
    - However, it is not suitable for large data.
    - Shared Disk needs redundancy.

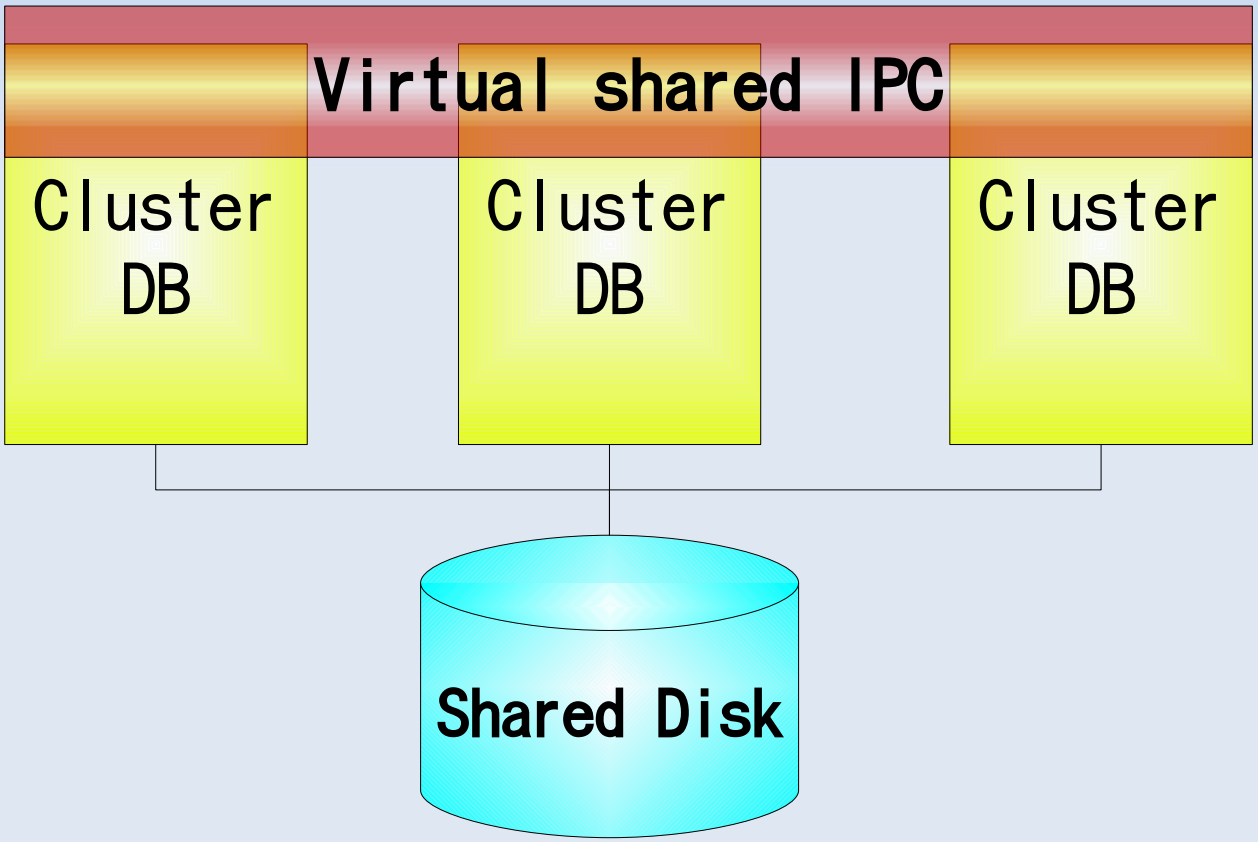
As a solution

- Requirement
- PGCluster
- New Requirement
- **PGCluster-II**
- Structure and Process sequence
- Pros & Cons

What is the PGCluster-II

- **Data shared clustering system**
  - Storage data shared by shared disk
    - NFS,GFS,GPFS(AIX) etc.
    - SAN/NAS
  - **Cache and lock status shared by Virtual IPC**
    - Detail as following slides

Concept of Shared Data



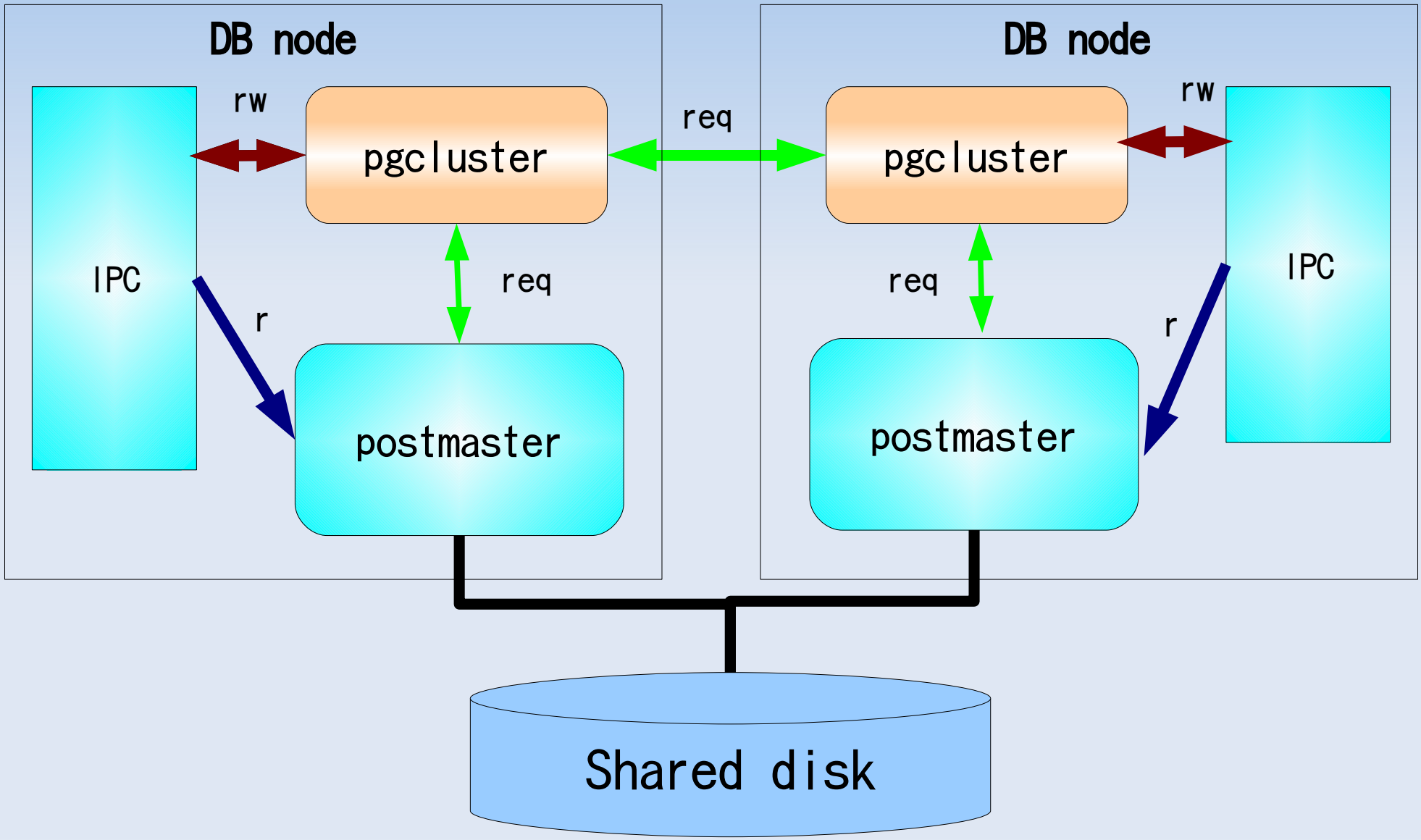
Inside of PGCluster-II

- Requirement
- PGCluster
- New Requirement
- PGCluster-II
- **Structure and Process sequence**
- Pros & Cons

Virtual IPC

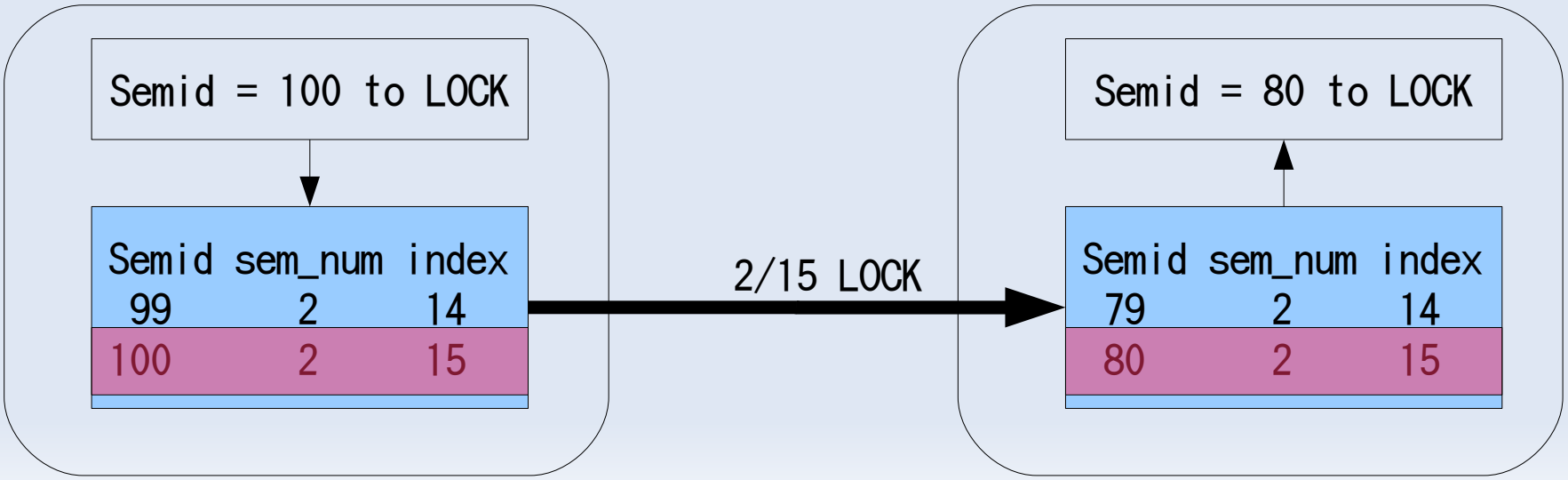
- Share semaphore and shared memory during DB nodes
  - Write it to remote nodes through cluster process
  - Read it from local node directory
- Signal and message queue are out of scope

Structure of PGCluster-II



Semaphore

- To Lock control
- How many semaphores are using?
  - Depends on the “max-connections” setting
  - In default, 7 x 16 semaphores are used.
- Mapping table is required



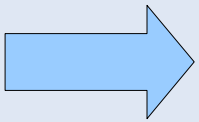
Shared Memory

- Communicate during each backend processes
- Store data of logs, caches, buffers and so on
- Single shared memory is allocated
  - But it is divided a number of peaces
  - more than 100 entry pointer are existing.

Issues of Shared Memory

- **Activity issue**
  - Size is not big but **update frequency** is very high
- **Contents issue**
  - It is including **memory/function address**
  - If copy shared memory to other server, other DB server may be **crashed** (depend on the OS).

Address	Data	Type	Label
&1000	&1004	Char *	Data
&1004	1	OID	Oid
&1008	&1012	Char *	Next
&1012	&1024	Char *	Data

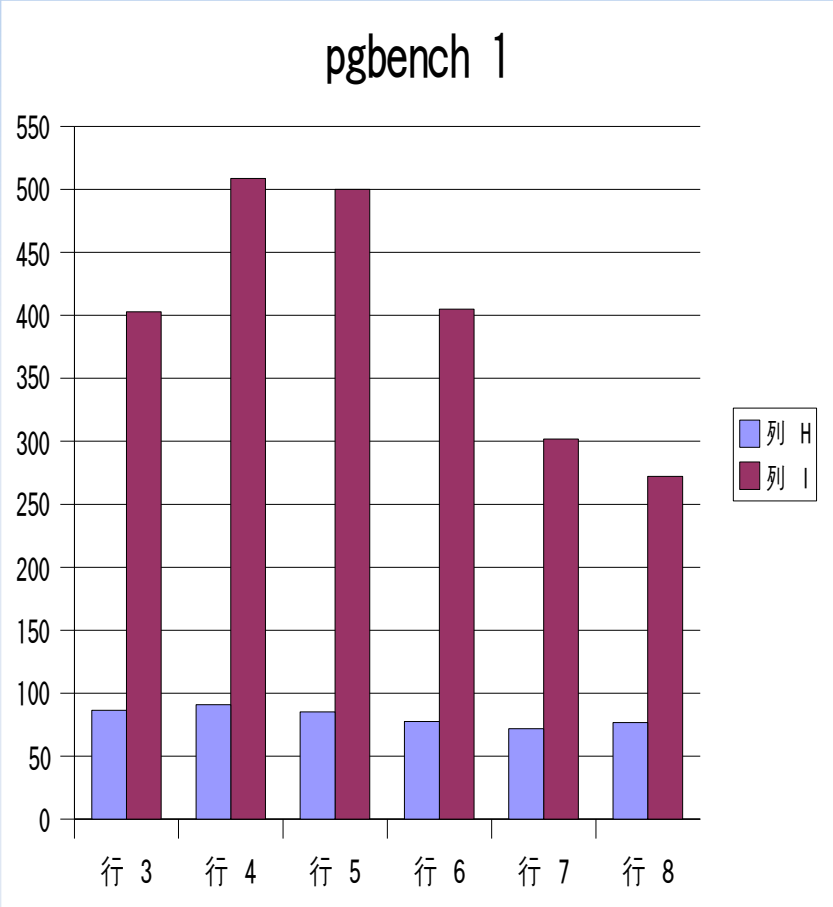


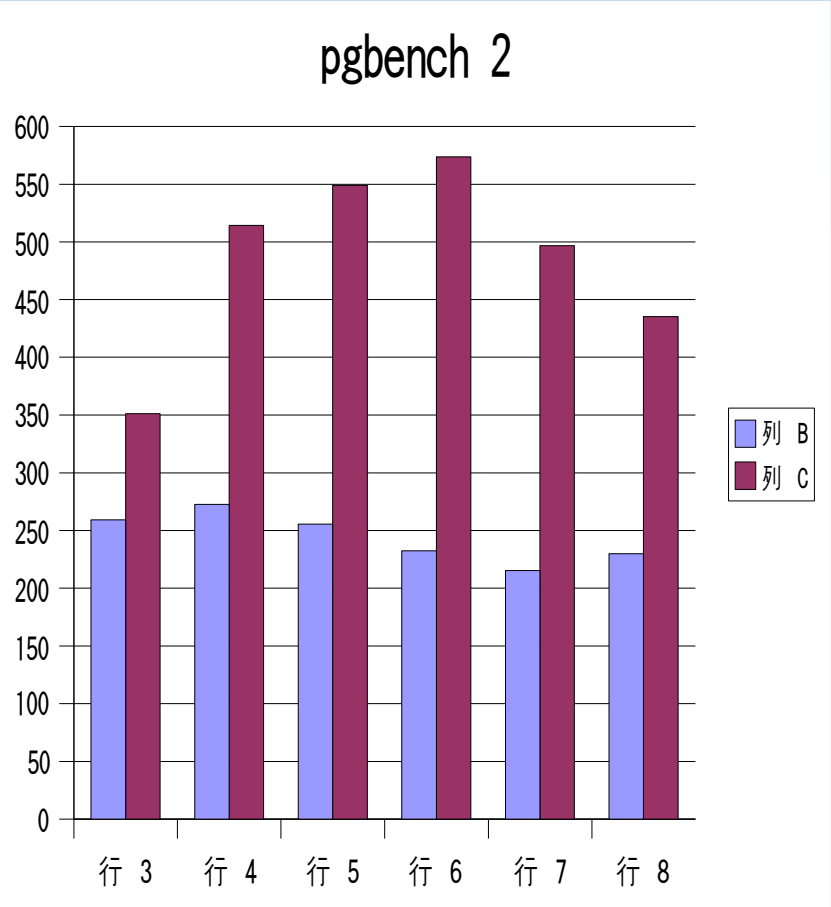
Address	Data	Type	Label
&2000	~~&1004~~	Char *	Data
&2004	1	OID	Oid
&2008	~~&1012~~	Char *	Next
&2012	&1024	Char *	Data

Solution

- **Mask table & localization table**
  - It worked, but very bad performance
- **Data changed to offset from address**
  - char * ptr -> Size offset
  - It's still over head, but better than before.

benchmark


pgbench 1
550
500
450
400
350
300
250
200
150
100
50
0
行 3
行 4
行 5
行 6
行 7
行 8
列 H
列 I


pgbench 2
600
550
500
450
400
350
300
250
200
150
100
50
0
行 3
行 4
行 5
行 6
行 7
行 8
列 B
列 C

As a result

- Requirement
- PGCluster
- New Requirement
- PGCluster-II
- Structure and Process sequence
- **Pros & Cons**

Pros & Cons

- Easy to add a node for redundancy / replace.
- Data writing performance does not slow by adding node.
- Big improve to data reading / many connection load.

- Required large RAM.
- Writing performance is not good yet.
- Nothing expands
- Cost
  - Shared disk system is expensive

TODO

- **Performance should more improve.**
  - Narrow down the target shared memory data.
  - It should send multi memory data at once.
- **Release source code**
  - ASAP
- **Documentation as well**

Thank you

- **Ask us about PGCluster**
  - pgcluster-general@pgfoundry.org
- **Ask me about PGCluster-II**
  - mitani@sraw.co.jp