PGCluster-II

Clustering system of PostgreSQL using Shared Data

PGCon 2007

Atsushi MITANI
AGENDA

- Introduction
- Requirement
- PGCluster
- New Requirement
- PGCluster-II
- Structure and Process sequence
- Pros & Cons
- Conclusion
As a background

- Introduction
- Requirement
- PGCluster
- New Requirement
- PGCluster-II
- Structure and Process sequence
- Pros & Cons
- Conclusion
Status of DB

- **Broken**
  - Data would be lost. sorry...

- **Stop**
  - Out of service, but data is remained.

- **Run**
  - Perfect! You can read and write data.

- **Between Run and Stop**
  - Hey, it's not working.
  - Huum, I can connect it.
What is DBA

- **Not good DBA**
  - Break DB by wrong patch / restore wrong data

- **Ordinary DBA**
  - monitors, patches, backups of DB
  - Stop DB before data broken

- **Good DBA**
  - Stop use such a funky DB

- **Joke ?**
High Availability (HA)

- **What is required?**
  - Short down time as much as possible
  - Even if hardware failure, power down and DB maintenance

- **Why it is required?**
  - Prevent data lost / service stop

- **Who needs?**
  - Data owner
  - Service user
What is required?
- Short response time as much as possible

Why it is required?
- User dislikes waiting
- Many processing data is the value of system

Who needs?
- Service user
At the beginning

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

- Target was Web application
- High Availability
  - Scheduled maintenance only
- High Performance
  - More than 200 accesses / sec
    - 700,000/hr, 1,500,000/day
  - 99.9% are data reading queries
As a solution

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

- Enough HA
- Enough performance
  - for data reading load
- Cost
  - Normal PC servers
  - BSD license SW
- Performance issue
  - Very bad for data writing load
- Maintenance issue
- Document issue
Demand changes with a time

- Introduction
- Requirement
- PGCluster
- New Requirement
- PGCluster-II
- Structure and Process sequence
- Pros & Cons
- Conclusion
Current requirement

- High Availability
  - 24/7 non stop
- High Performance
  - Not only read but write
- Reduce cost
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.
Suitable solution for HA and HP

- Synchronous replication
- Asynchronous replication
- Shared data clustering
- Parallel query
Assumption of the performance

Request type
write
read

Connection num
many
few

Data instance size
small
large

Connection num

Request type

PGCLuster
pgpool
Slony
PGCLuster-II
pgpool-II
As a solution

- Introduction
- Requirement
- PGCluster
- New Requirement
- PGCluster-II
- Structure and Process sequence
- Pros & Cons
- Conclusion
What is the PGCluster-II

Data shared clustering system

- Storage data shared by shared disk
  - NFS, GFS, GPFS (AIX) etc.
  - NAS

- Cache and lock status shared by Virtual IPC
  - Detail as following slides
Concept of Shared Data

Virtual shared IPC

Cluster DB

Cluster DB

Cluster DB

Shared Disk
Inside of PGCluster-II

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

- **DB node**
  - `pgcluster`
  - `postmaster`
- **IPC**
- **Shared disk**

Connections:
- `rw` from `IPC` to `pgcluster` and from `pgcluster` to `IPC`.
- `req` from `postmaster` to `pgcluster` and from `pgcluster` to `postmaster`.
- `req` from `DB node` to `postmaster` and from `postmaster` to `DB node`.
- `r` from `IPC` to `postmaster` and from `postmaster` to `IPC`.

*PGCon 2007*
Semaphore

- **To Lock control**
- **How many semaphores are using?**
  - Depends on the “max-connections” setting
  - By default, 7 x 16 semaphores are used.
- **Mapping table is required**
- 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 places
  - more than 100 entry pointer are existing.
Shared Memory usage

90% of usage is BufferBlocks
Issues of Shared Memory

- **Activity issue**
  - Size is not big but *update frequency* is very high

- **Contents issue**
  - It is including memory *address* it self
  - If copy shared memory to other server, other DB server may be *crashed*.

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
<th>Type</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;1000</td>
<td>&amp;1004</td>
<td>Char *</td>
<td>Data</td>
</tr>
<tr>
<td>&amp;1004</td>
<td>1</td>
<td>OID</td>
<td>Oid</td>
</tr>
<tr>
<td>&amp;1008</td>
<td>&amp;1012</td>
<td>Char *</td>
<td>Next</td>
</tr>
<tr>
<td>&amp;1012</td>
<td>&amp;1024</td>
<td>Char *</td>
<td>Data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
<th>Type</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;2000</td>
<td>&amp;1004</td>
<td>Char *</td>
<td>Data</td>
</tr>
<tr>
<td>&amp;2004</td>
<td>1</td>
<td>OID</td>
<td>Oid</td>
</tr>
<tr>
<td>&amp;2008</td>
<td>&amp;1012</td>
<td>Char *</td>
<td>Next</td>
</tr>
<tr>
<td>&amp;2012</td>
<td>&amp;1024</td>
<td>Char *</td>
<td>Data</td>
</tr>
</tbody>
</table>
Solution

- **All address data should not copy**
  - Copy mask table is required

- **All address data should translate to each local address**
  - Data address Offset is required in each address data
### Mask & Transrate Sequence

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
<th>Type</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;1000</td>
<td>‘+12’</td>
<td>Int</td>
<td>data_offset</td>
</tr>
<tr>
<td>&amp;1004</td>
<td>‘+20’</td>
<td>Int</td>
<td>next_offset</td>
</tr>
<tr>
<td>&amp;1008</td>
<td>&amp;1012</td>
<td>Char *</td>
<td>Data</td>
</tr>
<tr>
<td>&amp;1012</td>
<td>1</td>
<td>OID</td>
<td>Oid</td>
</tr>
<tr>
<td>&amp;1016</td>
<td>&amp;1020</td>
<td>Char *</td>
<td>Next</td>
</tr>
<tr>
<td>&amp;1020</td>
<td>‘+32’</td>
<td>Int</td>
<td>data_offset</td>
</tr>
</tbody>
</table>

- **Address offset added**
- **Address data masked**

**Copy with mask**

**Change offset to local address**

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
<th>Type</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;2000</td>
<td>‘+12’</td>
<td>Int</td>
<td>data_offset</td>
</tr>
<tr>
<td>&amp;2004</td>
<td>‘+20’</td>
<td>Int</td>
<td>next_offset</td>
</tr>
<tr>
<td>&amp;2008</td>
<td>&amp;2012</td>
<td>Char *</td>
<td>Data</td>
</tr>
<tr>
<td>&amp;2012</td>
<td>1</td>
<td>OID</td>
<td>Oid</td>
</tr>
<tr>
<td>&amp;2016</td>
<td>&amp;2020</td>
<td>Char *</td>
<td>Next</td>
</tr>
<tr>
<td>&amp;2020</td>
<td>‘+32’</td>
<td>Int</td>
<td>data_offset</td>
</tr>
</tbody>
</table>
Shared Disk

- Each node shares all db cluster
  - base/, global/, pg_clog/, pg_multixact/, pg_subtrans/, pg_tblspc/, pg_twophase/, pg_xlog/
- Each node has own configuration files
  - pg_hba.conf, pg_ident.conf, postgresql.conf, pgcluster.conf
- Each node should have same setup values
  - Connections (max_connections)
  - Resource usage(memory, Free Space Map)
Pgcluster table description

- Hostname/IP & port
- Multiple servers can be described
- Top described server may be master.

Self node description

- hostname/IP & port
- Only one node can be described
## Startup Sequence

### Node 1

<table>
<thead>
<tr>
<th>Postgres</th>
<th>Pgcluster</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start up</strong>&lt;br&gt;Create SEM&lt;br&gt;Create SHM</td>
<td>Begin req&lt;br&gt;Search other nodes&lt;br&gt;Create node table</td>
</tr>
<tr>
<td>Listen</td>
<td>Begin ans</td>
</tr>
<tr>
<td></td>
<td>Add new node&lt;br&gt;Send SEM</td>
</tr>
<tr>
<td></td>
<td>Sync req&lt;br&gt;Sync SEM req&lt;br&gt;Sync SEM ans&lt;br&gt;Sync SHM req&lt;br&gt;Sync SHM ans&lt;br&gt;Sync SYS req&lt;br&gt;Sync SYS ans&lt;br&gt;Sync SEM ans</td>
</tr>
<tr>
<td></td>
<td>Send SHM</td>
</tr>
<tr>
<td></td>
<td>Send node table</td>
</tr>
</tbody>
</table>

### Node 2

<table>
<thead>
<tr>
<th>Pgcluster</th>
<th>Postgres</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start up</strong>&lt;br&gt;Create SEM&lt;br&gt;Create SHM</td>
<td>Begin req&lt;br&gt;Search other nodes&lt;br&gt;Create node table</td>
</tr>
<tr>
<td></td>
<td>Copy SEM</td>
</tr>
<tr>
<td></td>
<td>Copy SHM</td>
</tr>
<tr>
<td></td>
<td>Copy node table</td>
</tr>
<tr>
<td></td>
<td>Begin ans&lt;br&gt;Listen</td>
</tr>
</tbody>
</table>
Stop sequence

Node1

Postgres
Pgcluster

Update node table

End req

End ans

Node2

Pgcluster
Postgres

Search other nodes

Stop req

Stop ans

Delete IPC
As a result

- Introduction
- Requirement
- PGCluster
- New Requirement
- PGCluster-II
- Structure and Process sequence
- Pros & Cons
- Conclusion
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.
- Data writing does not become fast by adding node.
- Writing performance is not good.
- Nothing expands except CPU & network I/O

Cost
- Shared disk
Suitable place

- It will be one of solutions the system which has high CPU load and network load.
  - Most of WEB system, a part of the Online Transaction Processing (OLTP) system

Combination of PGCluster-II and pgpool-II

- PGCluster-II might get performance with large data.
From now

- Introduction
- Requirement
- PGCluster
- New Requirement
- PGCluster-II
- Structure and Process sequence
- Pros & Cons
- Conclusion
Performance should more improve.

- Some write (and erase) memory data is not need to sync.
- The conversion methods (from offset to local address) should improve.

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

- **You can download this slide from**
  - http://pgfoundry.org/docman/?group_id=1000072