# 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

# High Performance (HP)

- 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

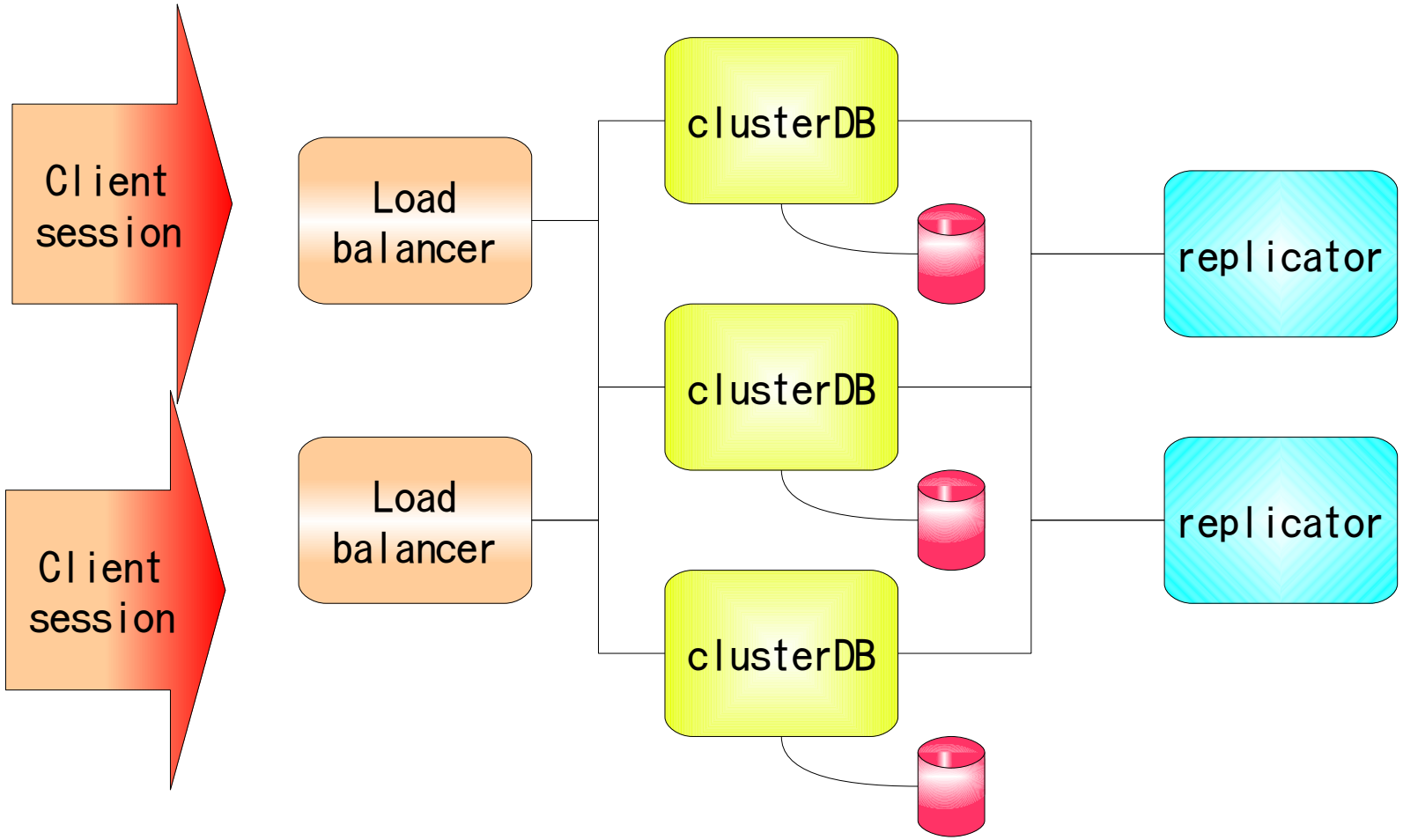
# As a solution

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

# 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
  - 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

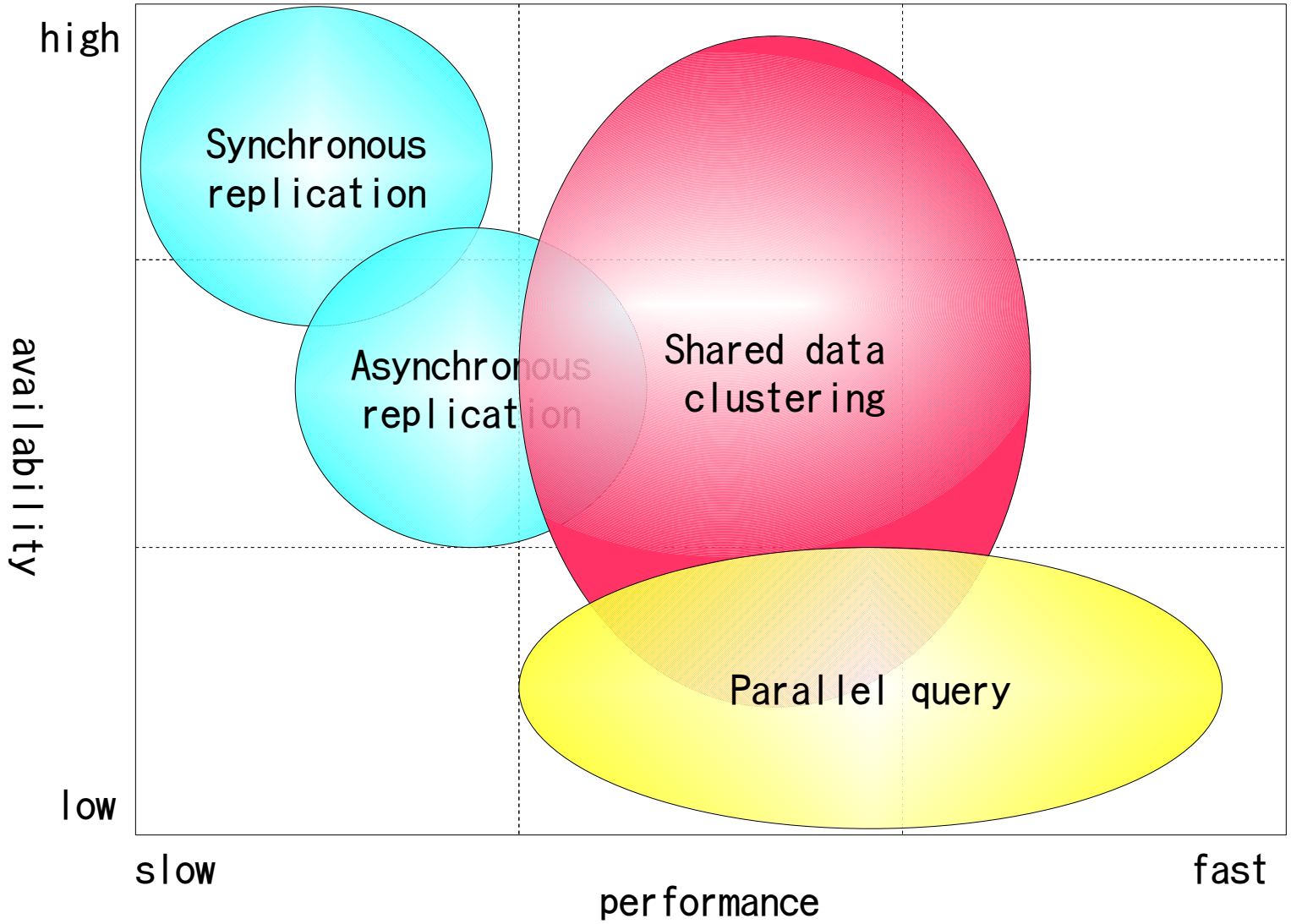
# Current requirement

- High Availability
  - 24/7 non stop
- High Performance
  - Not only read but write
- Reduce cost

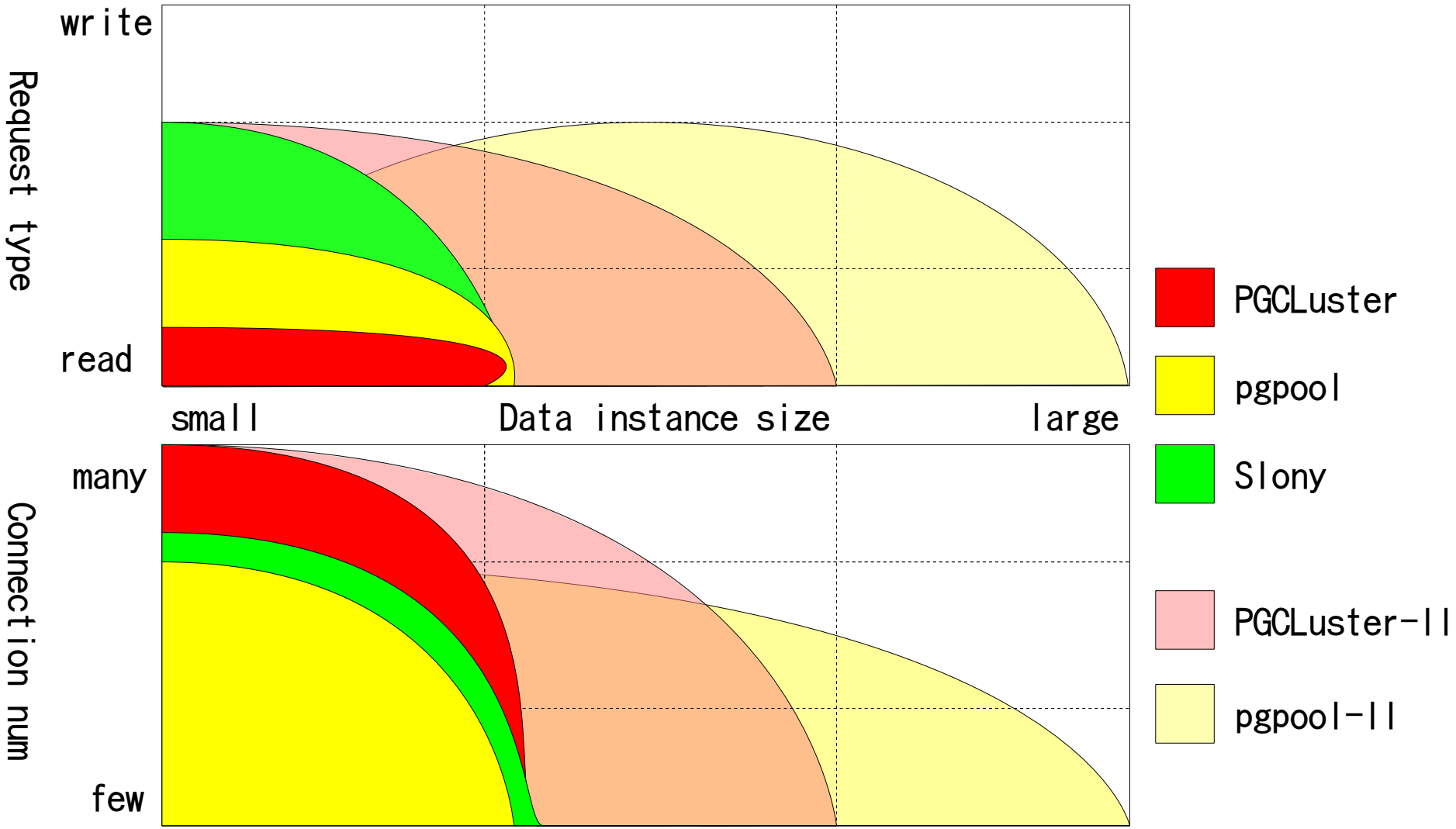
# Coexistence of HA and HP

- 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



# Assumption of the performance



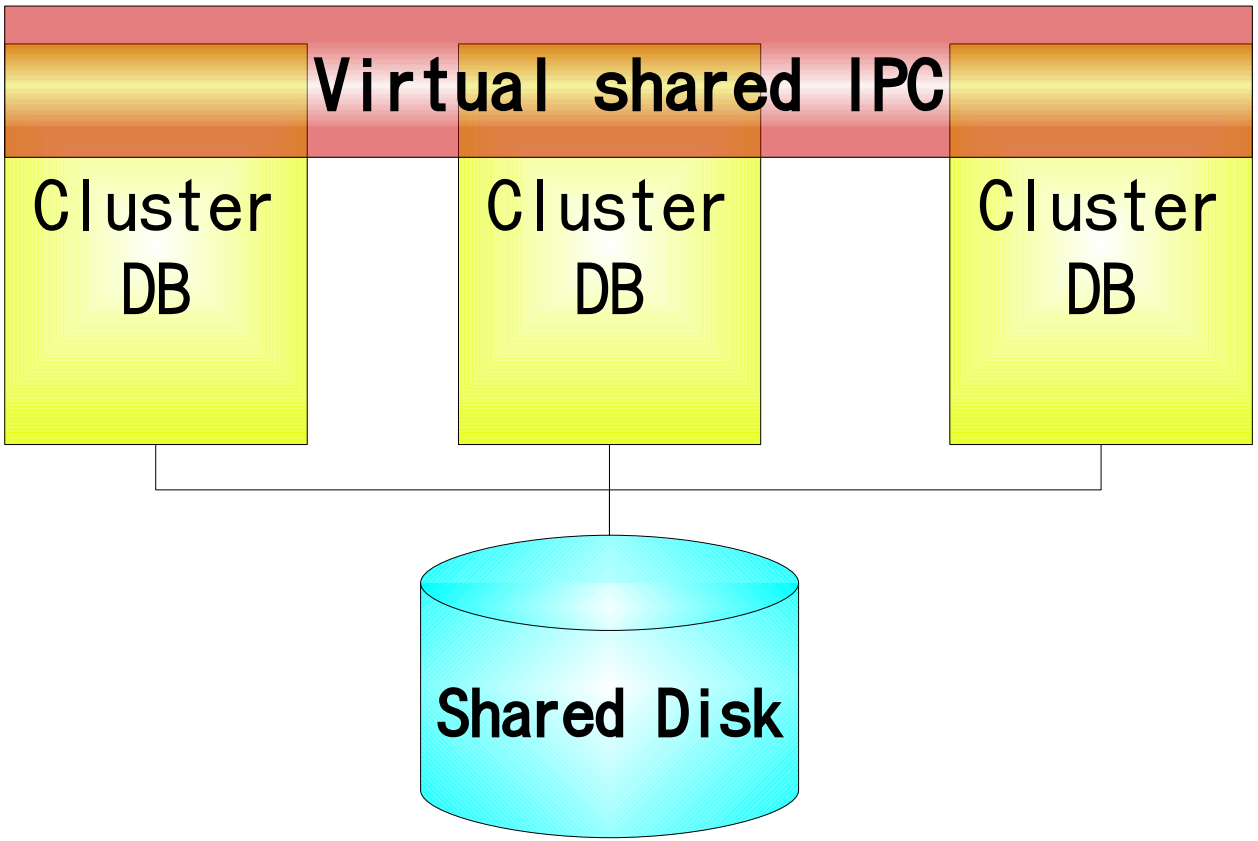
# As a solution

- 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



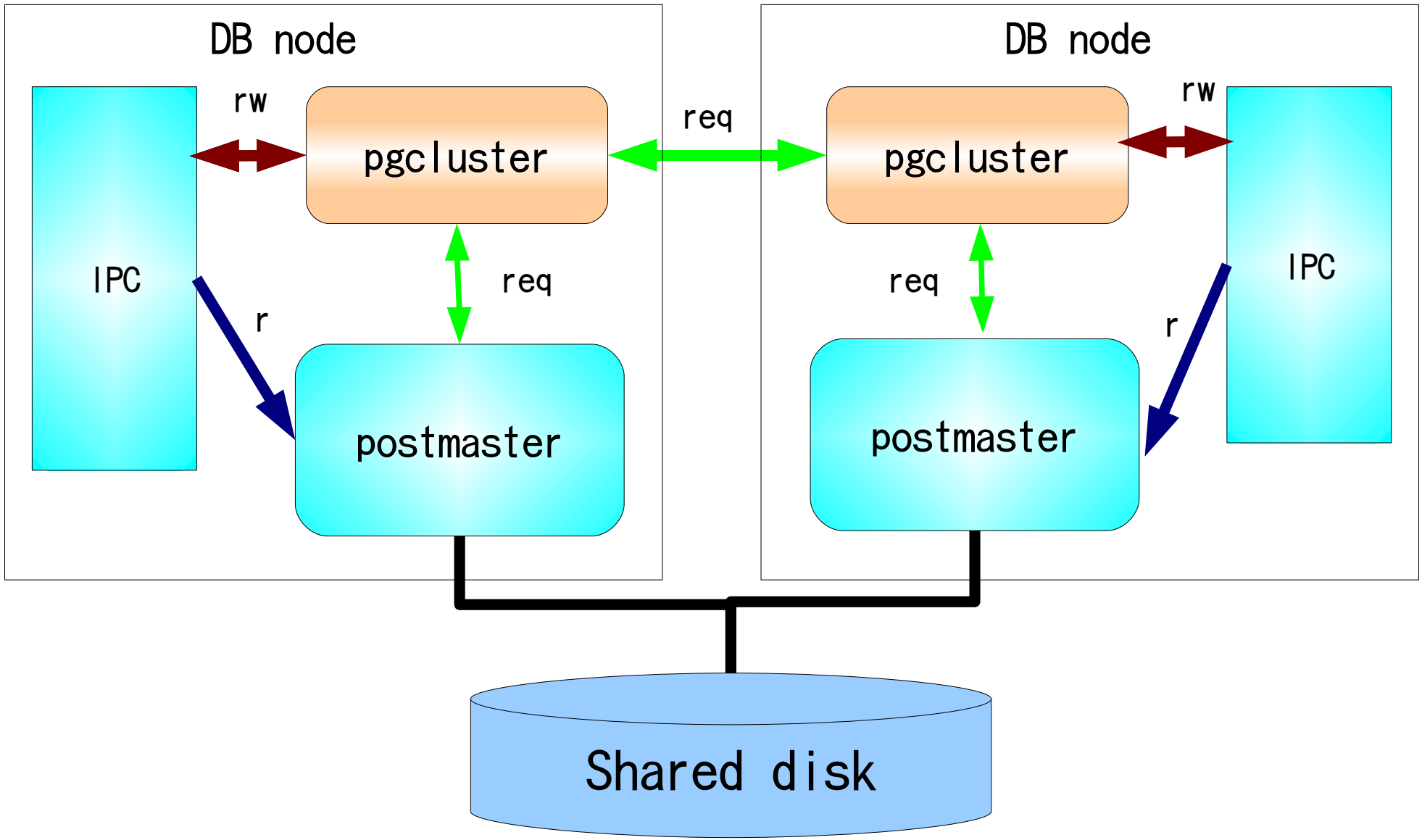
# Inside of PGCluster-II

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

# Virtual IPC

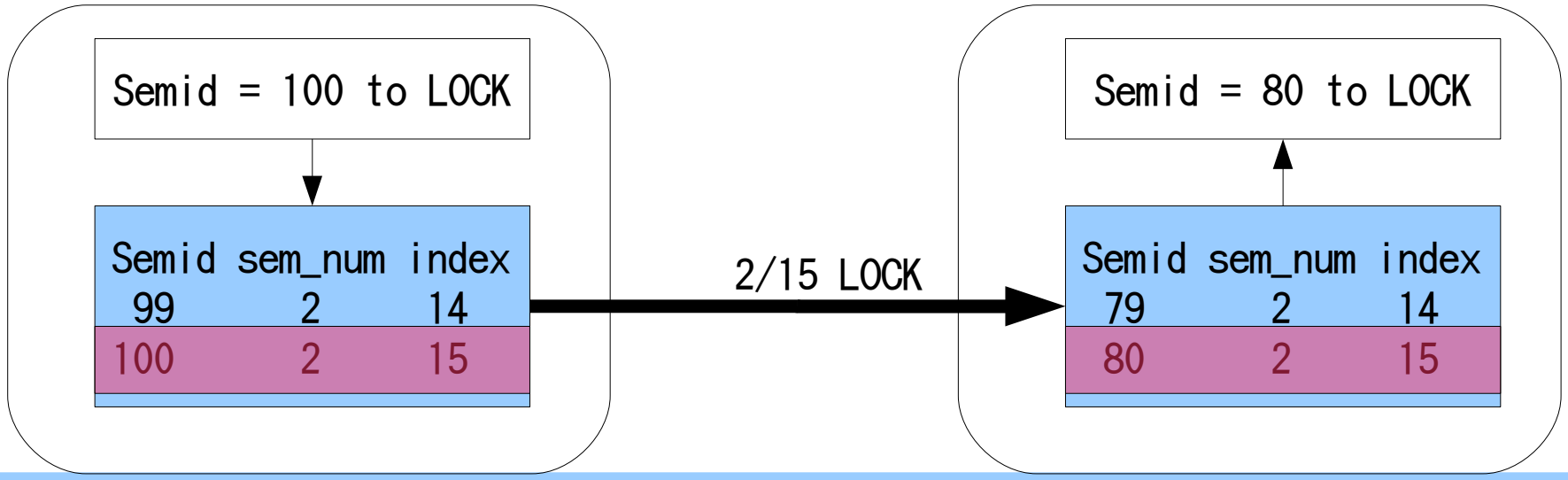
- Share semaphore and shared memory during DB nodes
  - Write it to remote nodes through cluster process
  - Read it from local node directory
- Signal and message queue are out of scope

# Structure of PGCluster-II



# Semaphore

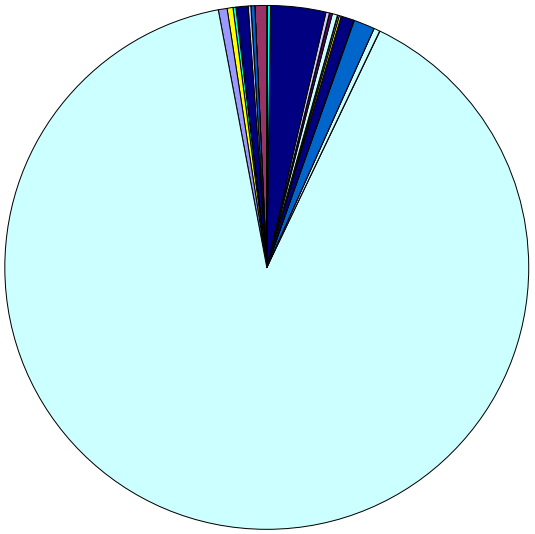
- To Lock control
- How many semaphores are using?
  - Depends on the “max-connections”setting
  - By default, 7 x 16 semaphores are used.
- Mapping table is required



# 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.

# Shared Memory usage



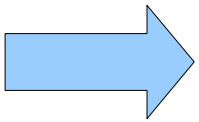
90% of usage
is BufferBlocks

| | | | |
|---|---|---|---|
| ShmemVariableCache" | LWLockArray | ShmemIndex | newSpace |
| newSpace | ControlFile | XLogCtl | CLOG Ctl" |
| SUBTRANS Ctl | TwoPhaseState | MultiXactOffset Ctl | MultiXactMember Ctl |
| BufferDescriptors | BufferBlocks | Shared Buffer Lookup Table | newSpace |
| StrategyControl | LOCK hash | newSpace | PROCLOCK hash" |
| newSpace | ProcGlobal | DummyProcs | newSpace |
| procs | ProcStructLock | procArray | BackendStatusArray |
| shmInvalBuffer | FreeSpaceMap | Free Space Map Hash | newSpace |
| FreeSpaceMap->arena" | PMSignalFlags | BgWriterShmem | btvacinfo |

# 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**.

| 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

- 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

| Address | Data | Type | Label |
|---|---|---|---|
| &1000 | '+12' | Int | data_offset |
| &1004 | '+20' | Int | next_offset |
| &1008 | &1012 | Char * | Data |
| &1012 | 1 | OID | Oid |
| &1016 | &1020 | Char * | Next |
| &1020 | '+32' | Int | data_offset |

Address offset added

Address data masked

Copy with mask

| Address | Data | Type | Label |
|---|---|---|---|
| &2000 | '+12' | Int | data_offset |
| &2004 | '+20' | Int | next_offset |
| &2008 | | Char * | Data |
| &2012 | 1 | OID | Oid |
| &2016 | | Char * | Next |
| &2020 | '+32' | Int | data_offset |

Change offset
to local address

| Address | Data | Type | Label |
|---|---|---|---|
| &2000 | '+12' | Int | data_offset |
| &2004 | '+20' | Int | next_offset |
| &2008 | &2012 | Char * | Data |
| &2012 | 1 | OID | Oid |
| &2016 | &2020 | Char * | Next |
| &2020 | '+32' | Int | data_offset |

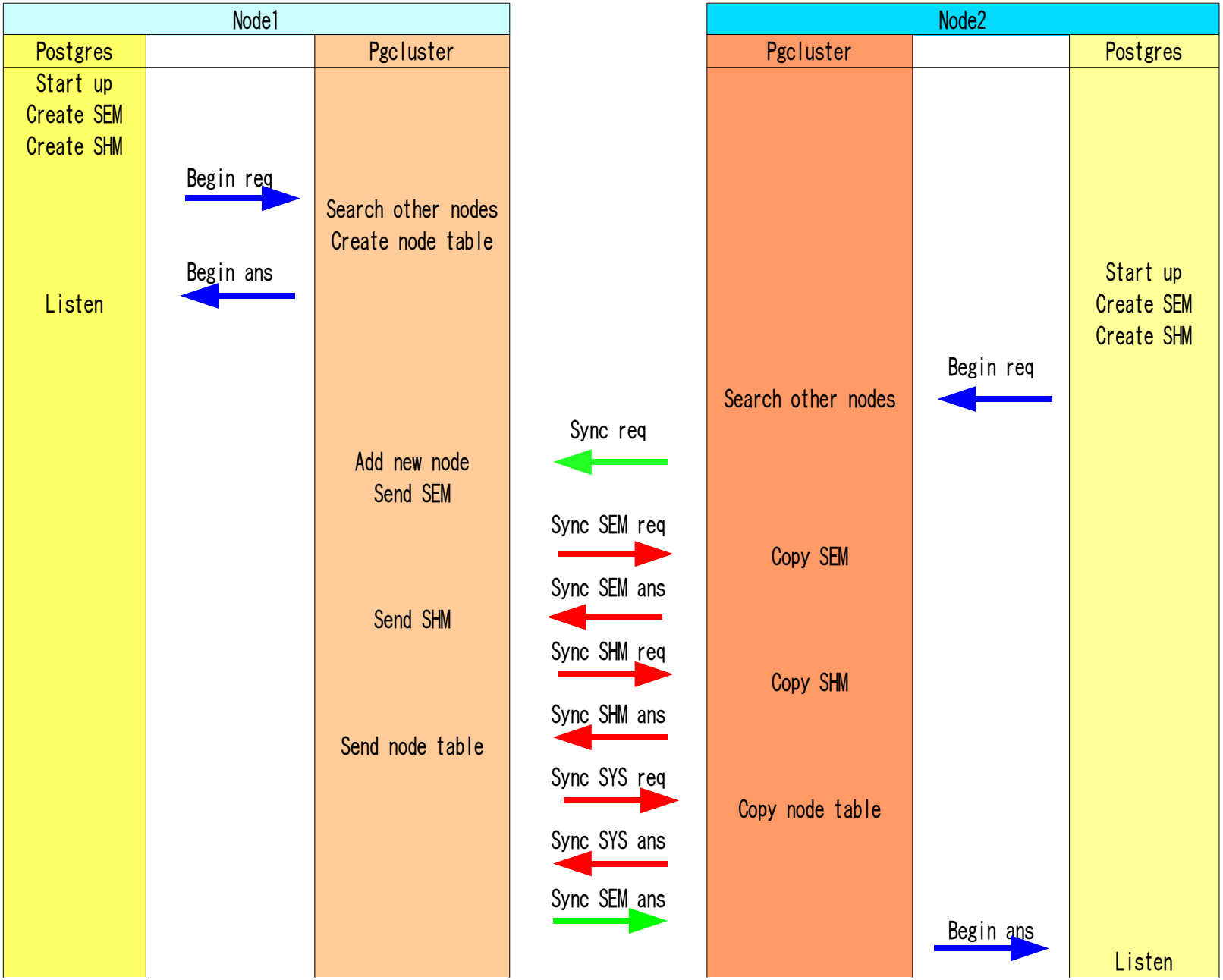
# Shared Disk

- 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)

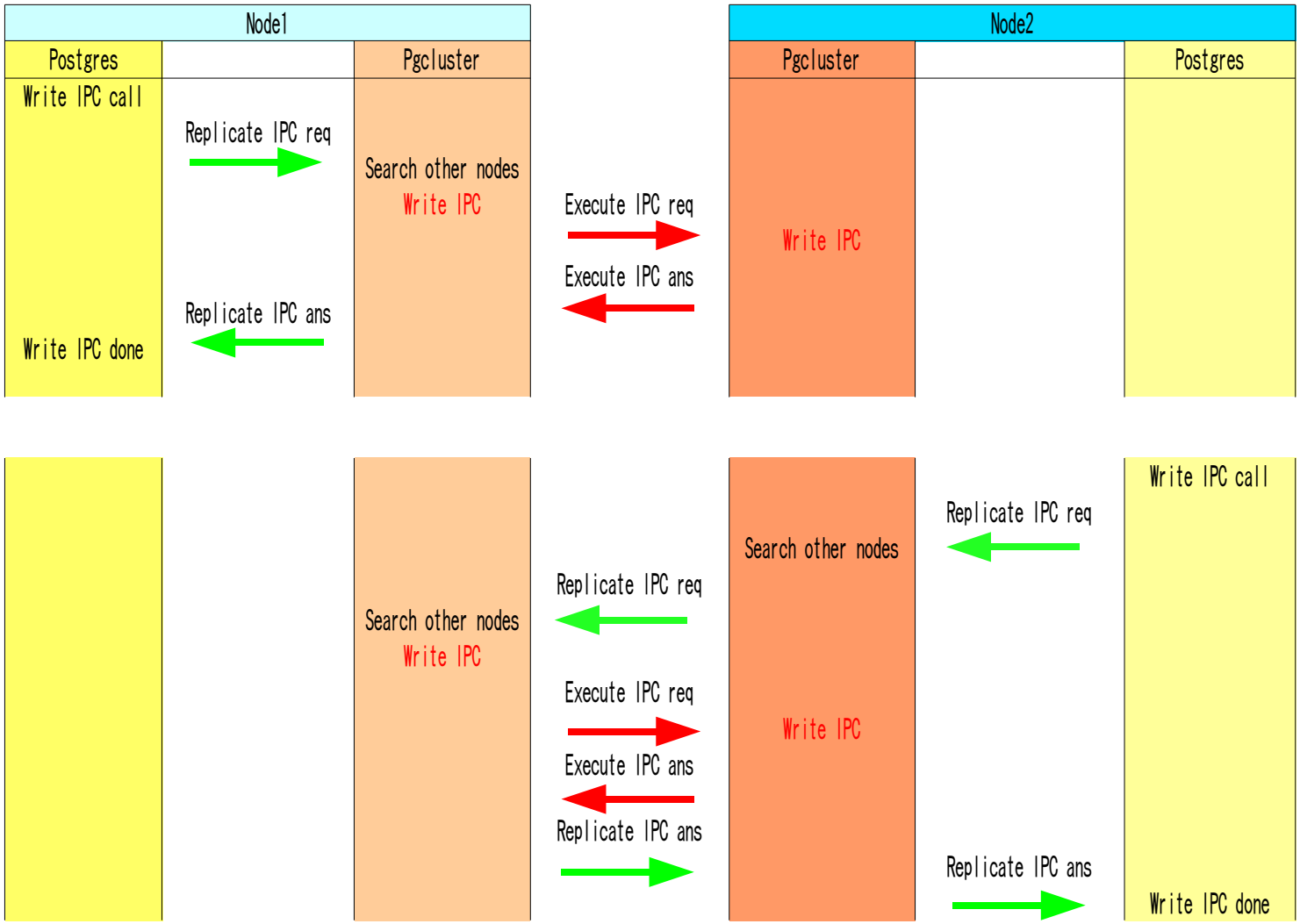
# pgcluster.conf

- 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

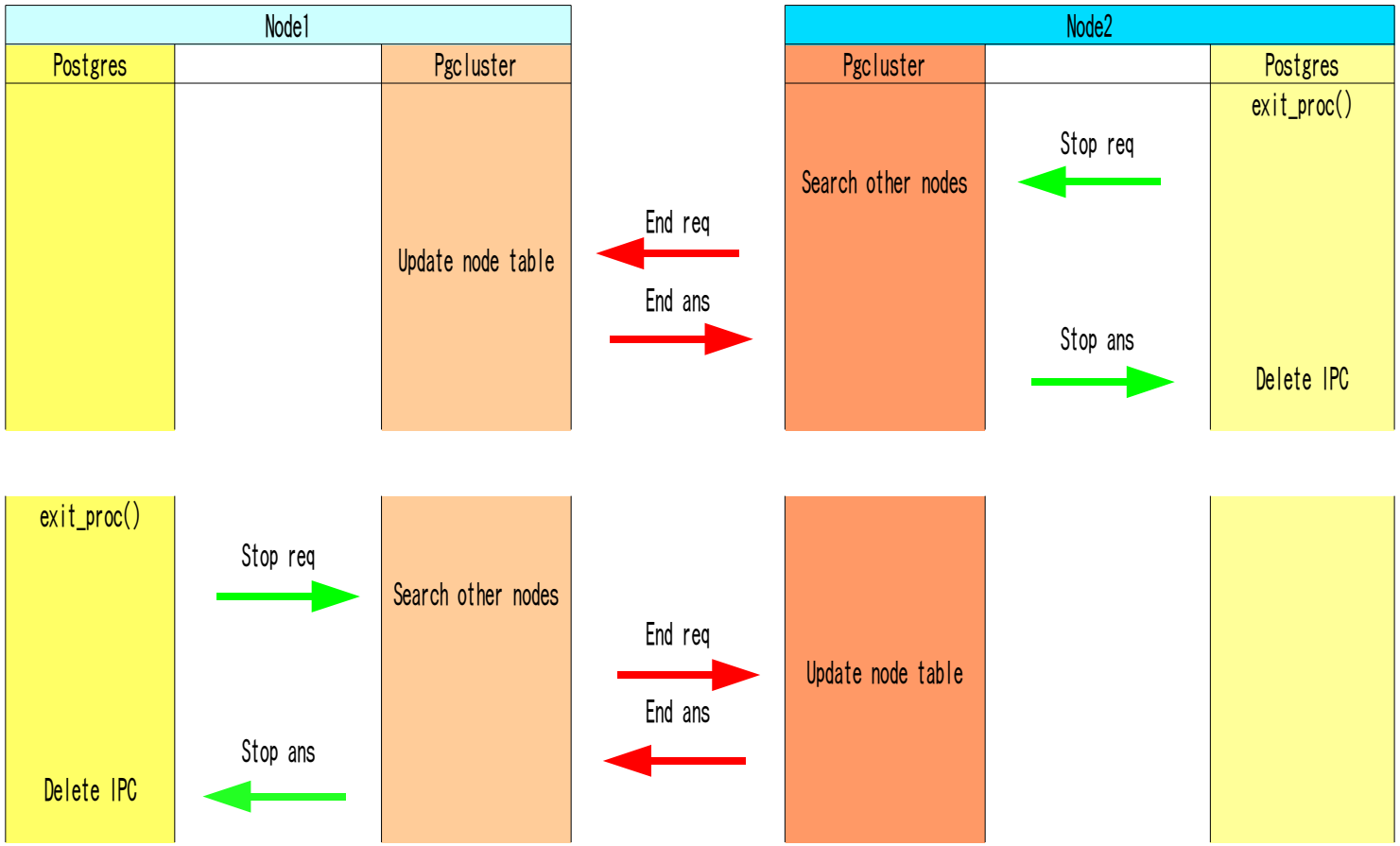
# Startup sequence



# IPC sync sequence



# Stop sequence



# 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

# Possibility

- 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

# TODO

- 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