

*Data Management*

# DB2 *pureScale* : A Technology Preview

Oct 21, 2009

[ibm.com/db2/labchats](http://ibm.com/db2/labchats)

## > Executive's Message



**Sal Vella**

**Vice President, Development,  
Distributed Data Servers and Data Warehousing**

**IBM**



## > Featured Speaker



**Matt Huras**

**Distinguished Engineer,  
DB2 for Linux, UNIX, and Windows**

**IBM**



## > Featured Speaker



**Aamer Sachedina**

**Senior Technical Staff Member,  
DB2 for Linux, UNIX, and Windows**

**IBM**



# Agenda



## Introduction

- ▶ Goals & Value Propositions
- ▶ Technology Overview

## ■ Technology In-Depth

- ▶ Key Concepts & Internals
- ▶ Efficient scaling
- ▶ Failure modes & recovery automation
- ▶ Stealth Maintenance

## ■ Configuration, Monitoring, Tooling

- ▶ Cluster configuration and operational status
- ▶ Monitoring data
- ▶ Client configuration and load balancing
- ▶ Solution Packaging



# DB2 pureScale : Goals

- **Unlimited Capacity**
  - ▶ Any transaction processing or ERP workload
  - ▶ Start small
  - ▶ Grow easily, with your business
  
- **Application Transparency**
  - ▶ Avoid the risk and cost of tuning your applications to the database topology
  
- **Continuous Availability**
  - ▶ Maintain service across planned and unplanned events

## DB2 pureScale

Unlimited capacity, transparent to applications.



DB2 pureScale reduces the risk and cost of business growth by providing unlimited capacity, continuous availability, and application transparency. DB2 pureScale on IBM Power Systems incorporates [PowerHA pureScale technology](#) to deliver levels of database scalability and availability unmatched on Unix or x86 systems. This complements DB2 for z/OS and System z, the undisputed leader in total system availability, scalability, security and reliability.

### Unlimited Capacity

DB2 pureScale provides practically unlimited capacity for any transactional workload. Scaling your system is simply a matter of connecting a new node and issuing two simple commands. DB2 pureScale's cluster-based, shared-disk architecture reduces costs through efficient use of system resources.

### Application Transparency

With DB2 pureScale, you don't need to change your application code to efficiently run on multiple nodes. Thanks to a proven, scalable architecture, you can grow your application to meet the most demanding business requirements. You can also run applications written for other database software with little or no changes; DB2 offers native support for commonly used syntax and PL/SQL procedure language, making it easier than ever to move from Oracle database to DB2.

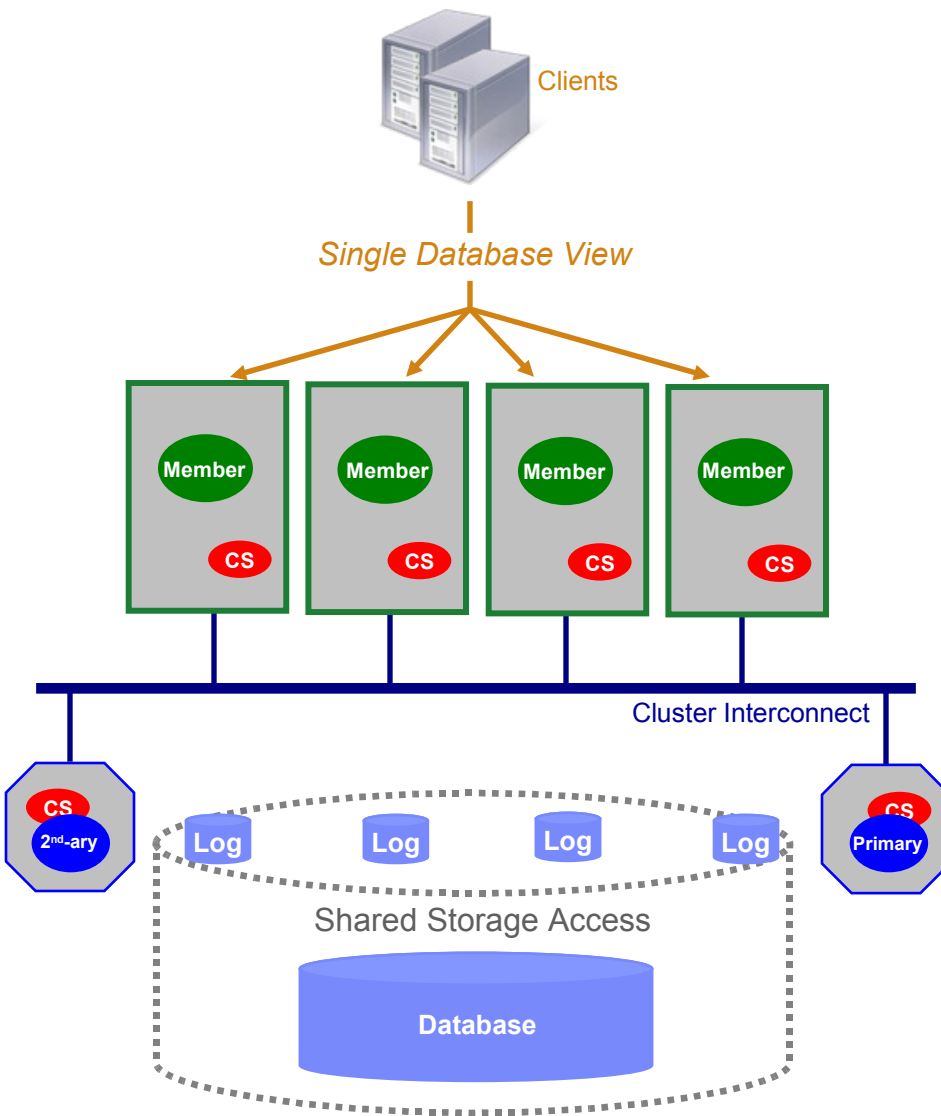
### Continuous Availability

DB2 pureScale provides continuous availability through the use of highly reliable IBM PowerHA pureScale technology on IBM Power systems and a redundant architecture. The system recovers nearly instantaneously from node failures, immediately redistributing the workload to surviving nodes.



# DB2 pureScale : Technology Overview

Leverage IBM's System z Sysplex Experience and Know-How



## Clients connect anywhere, ... ... see single database

- ▶ Clients connect into any member
- ▶ Automatic load balancing and client reroute may change underlying physical member to which client is connected

## DB2 engine runs on several host computers

- ▶ Co-operate with each other to provide coherent access to the database from any member

## Integrated cluster services

- ▶ Failure detection, recovery automation, cluster file system
- ▶ In partnership with STG (GPFS, RSCT) and Tivoli (SA MP)

## Low latency, high speed interconnect

- ▶ Special optimizations provide significant advantages on RDMA-capable interconnects (eg. Infiniband)

## PowerHA pureScale technology from STG

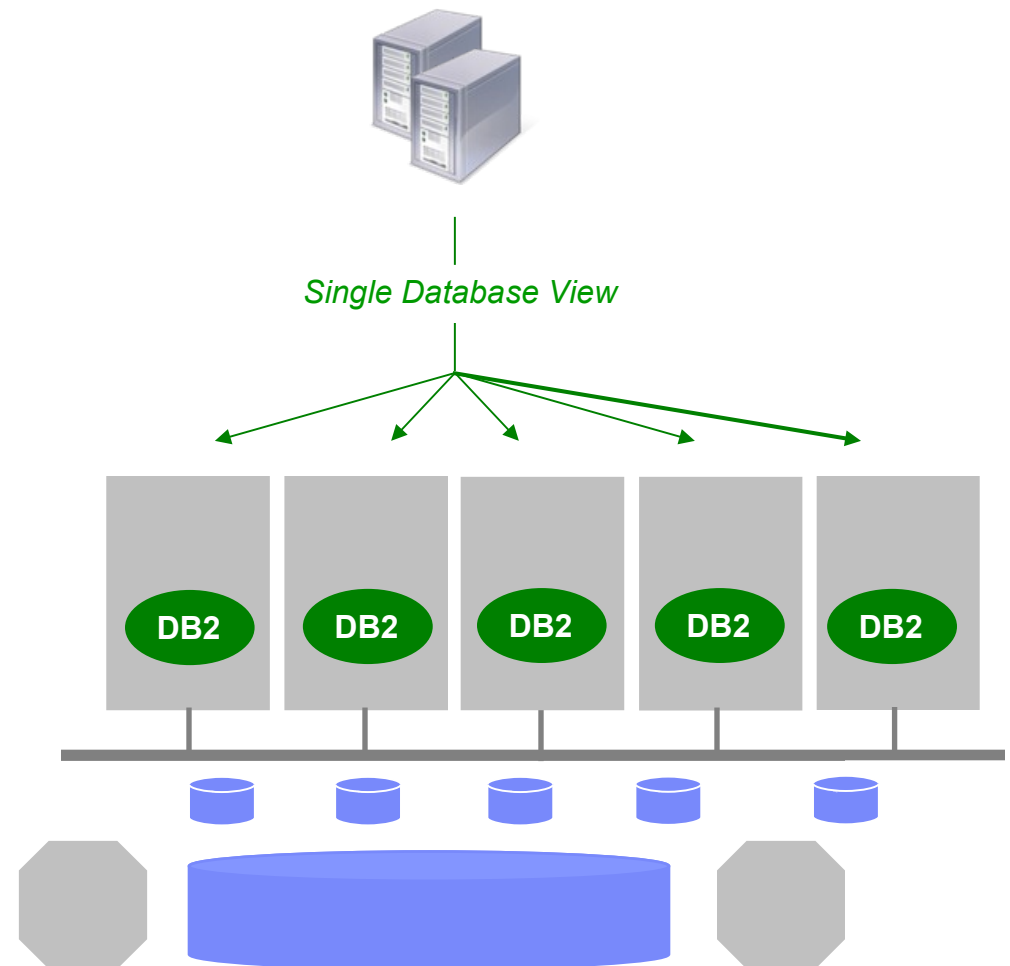
- ▶ Efficient global locking and buffer management
- ▶ Synchronous duplexing to secondary ensures availability

## Data sharing architecture

- ▶ Shared access to database
- ▶ Members write to their own logs
- ▶ Logs accessible from another host (for recovery)

# Scale with Ease

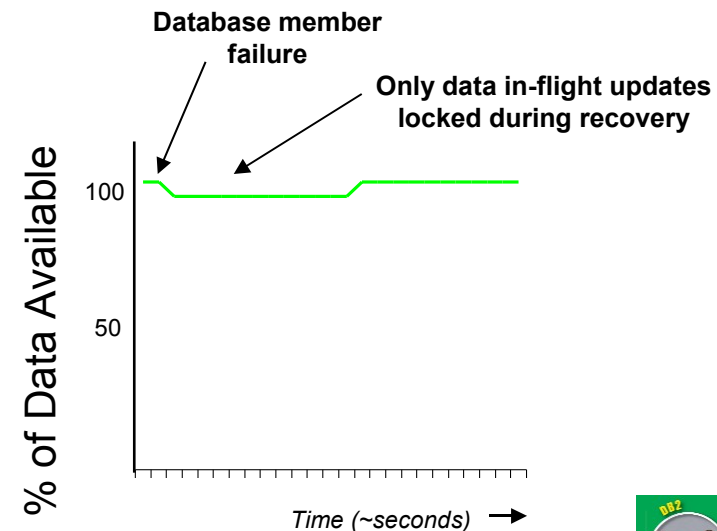
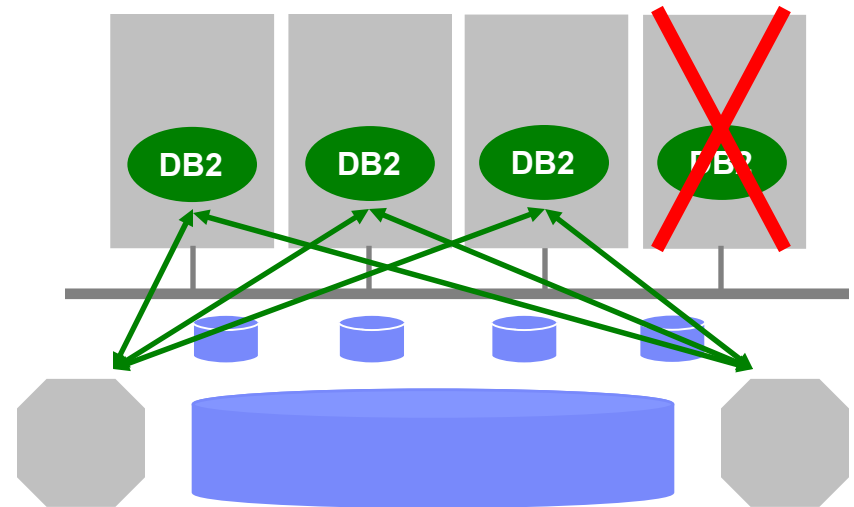
- Without changing applications
  - ▶ Efficient coherency protocols designed to scale without application change
  - ▶ Applications automatically and transparently workload balanced across members
- Without administrative complexity
  - ▶ No data redistribution required
- To 128 members in initial release





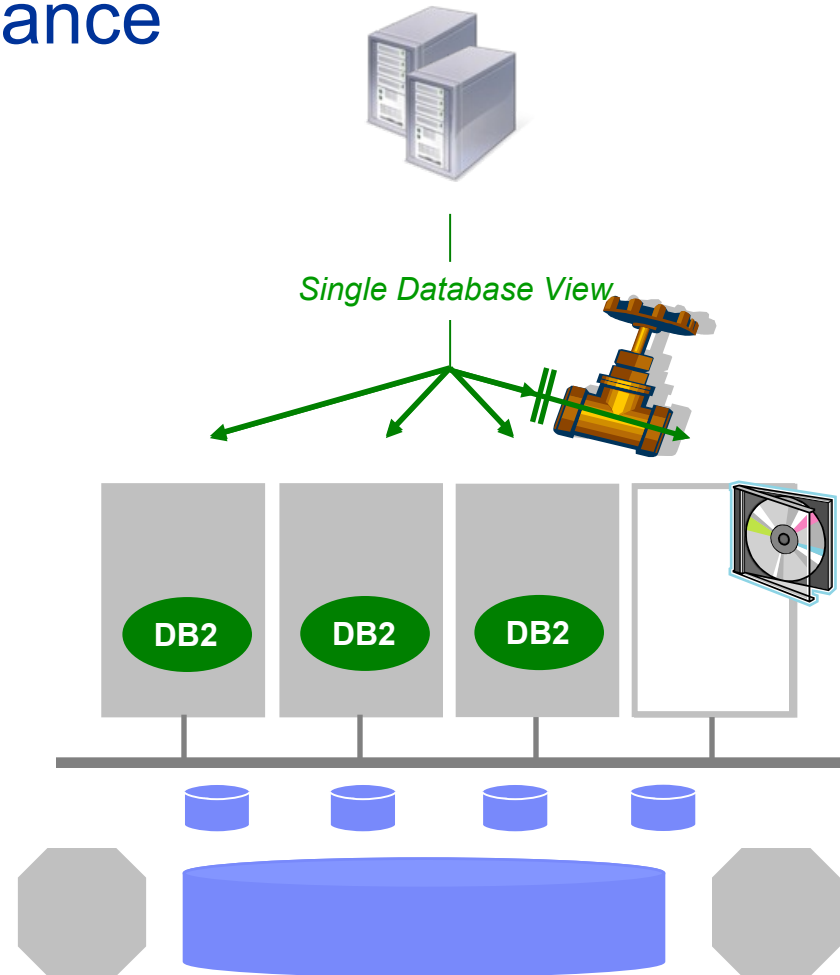
# Online Recovery

- A key DB2 pureScale design point is to maximize availability **during** failure recovery processing
- When a database member fails, only data *in-flight* on the failed member remains locked during the automated recovery
  - ▶ In-flight = data being updated on the member at the time it failed



# Stealth System Maintenance

- Goal: allow DBAs to apply system maintenance without negotiating an outage window
- Procedure:
  - ▶ Drain (aka Quiesce)
  - ▶ Remove & Maintain
  - ▶ Re-integrate
  - ▶ Repeat until done



# Agenda

- Introduction

- ▶ Goals & Value Propositions
- ▶ Technology Overview

- ▶  Technology In-Depth

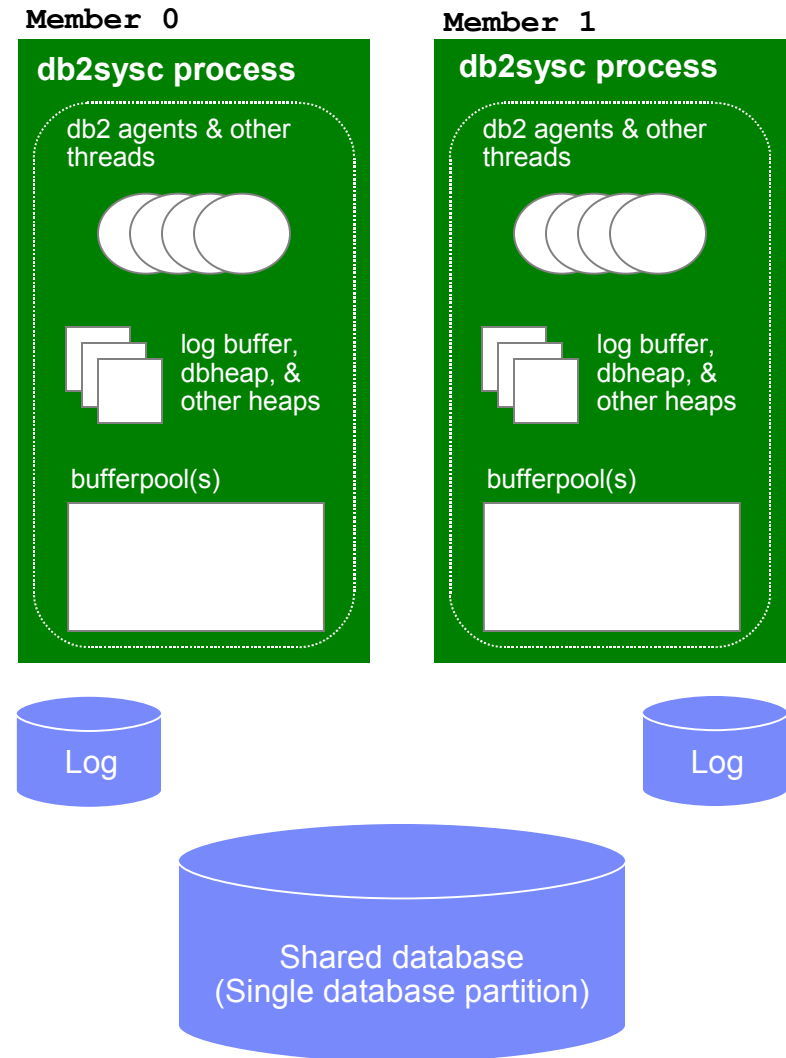
- ▶ Key Concepts & Internals
- ▶ Efficient scaling
- ▶ Failure modes & recovery automation
- ▶ Stealth Maintenance

- Configuration, Monitoring, Tooling

- ▶ Cluster configuration and operational status
- ▶ Monitoring data
- ▶ Client configuration and load balancing
- ▶ Installation

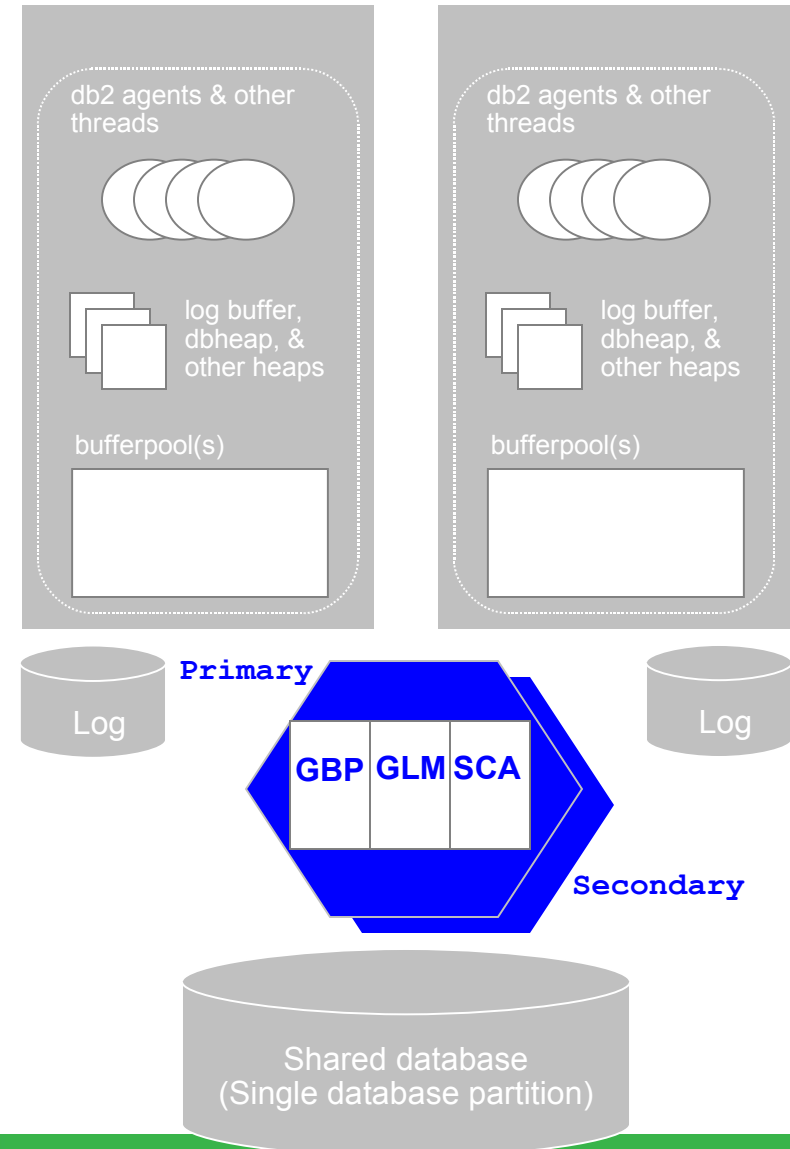
# What is a Member ?

- A DB2 engine address space
  - ▶ i.e. a db2sysc process and its threads
- Members Share Data
  - ▶ All members access the same shared database
  - ▶ Aka “Data Sharing”
- Each member has it's own ...
  - ▶ Bufferpools
  - ▶ Memory regions
  - ▶ Log files
- Members are logical. Can have ...
  - ▶ 1 per machine or LPAR (recommended)
  - ▶ >1 per machine or LPAR (not recommended)



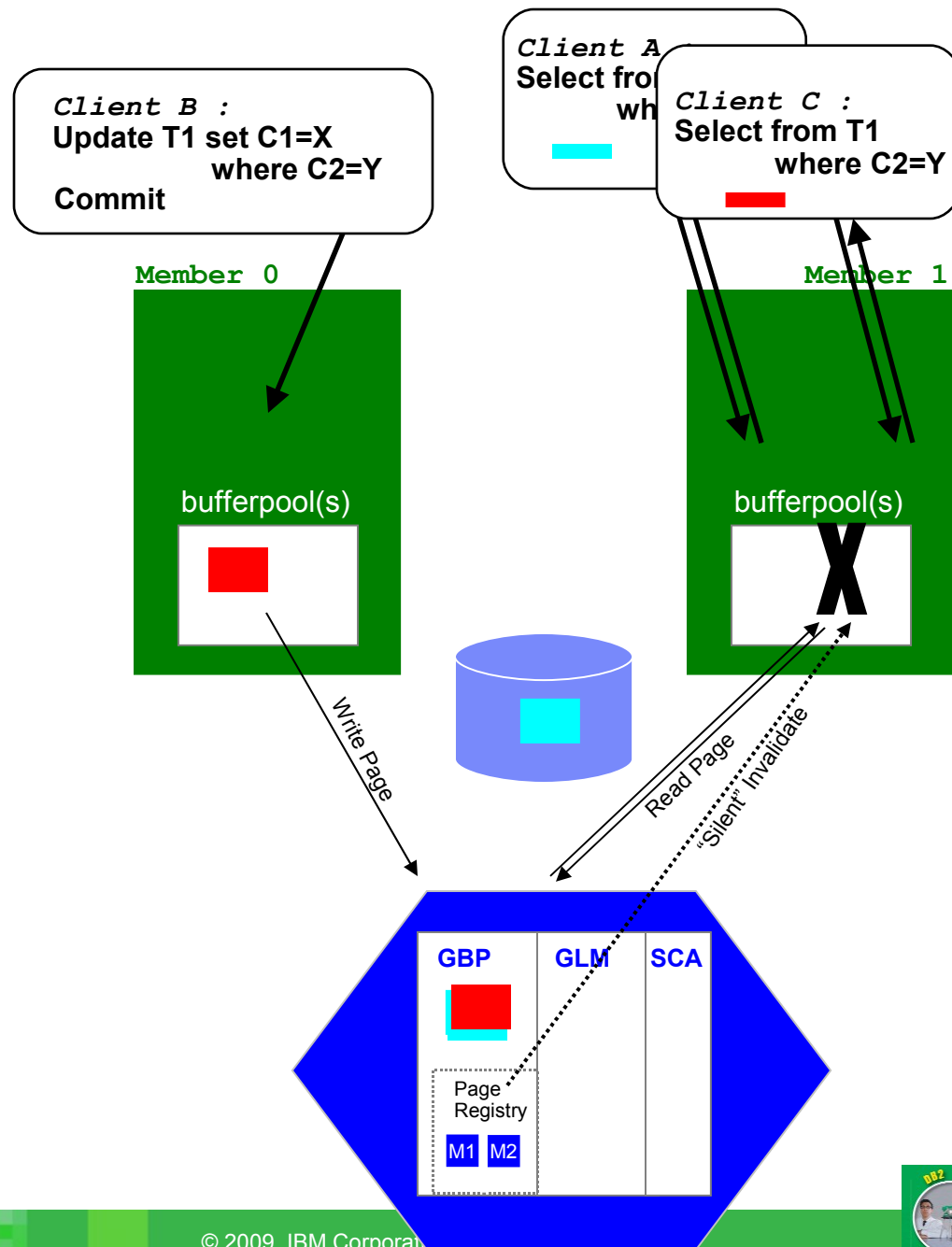
# What is a *PowerHA pureScale* ?

- Software technology that assists in global buffer coherency management and global locking
  - ▶ Derived from System z Parallel Sysplex & Coupling Facility technology
  - ▶ Software based
  
- Services provided include
  - ▶ Group Bufferpool (GBP)
  - ▶ Global Lock Management (GLM)
  - ▶ Shared Communication Area (SCA)
  
- Members duplex GBP, GLM, SCA state to both a primary and secondary
  - ▶ Done synchronously
  - ▶ Duplexing is optional (but recommended)
  - ▶ Set up automatically, by default



# The Role of the GBP

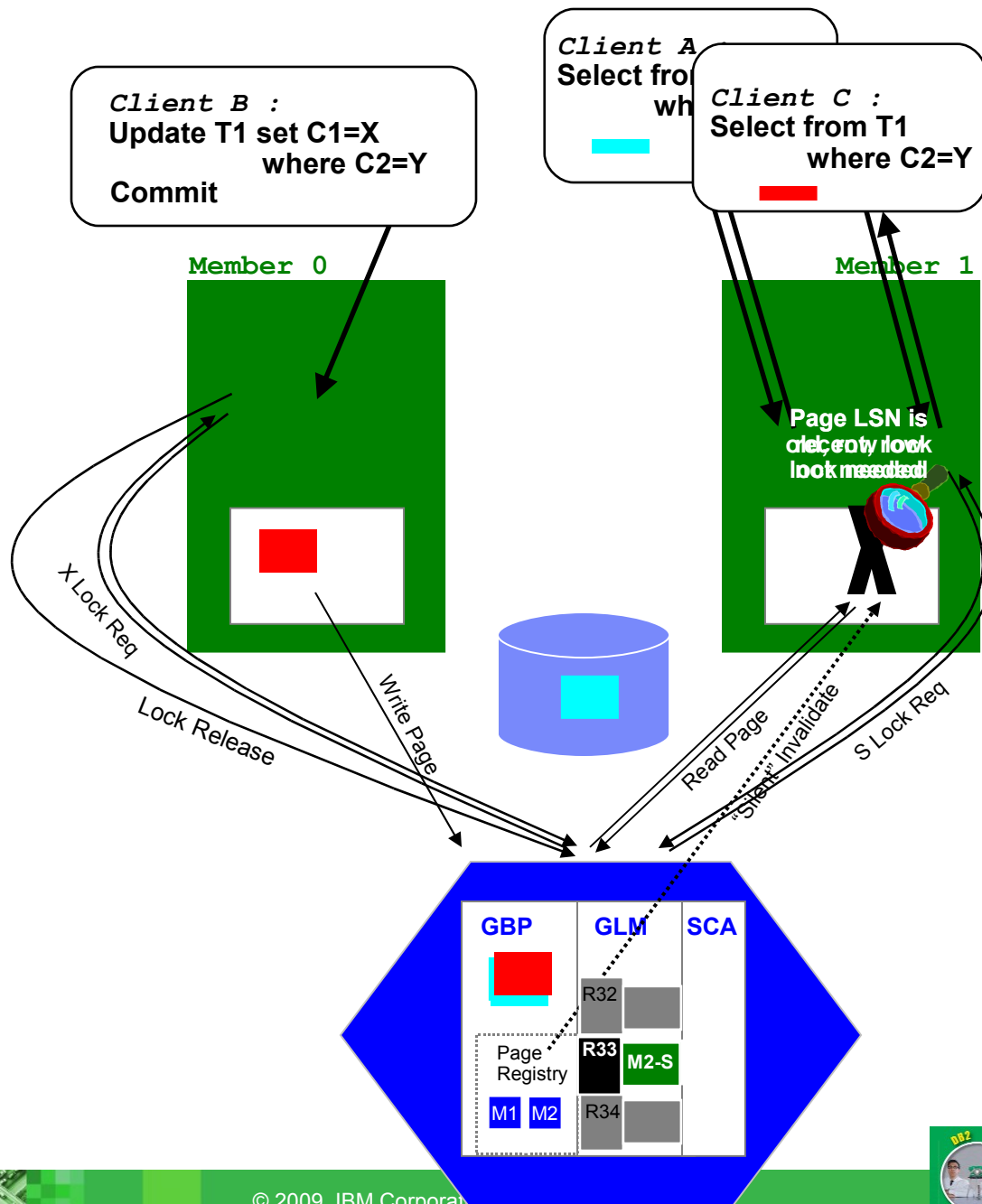
- GBP acts as fast disk cache
  - ▶ Dirty pages stored in GBP, then later, written to disk
  - ▶ Provides fast retrieval of such pages when needed by other members
  
- GBP includes a “Page Registry”
  - ▶ Keeps track of what pages are buffered in each member and at what memory address
  - ▶ Used for fast invalidation of such pages when they are written to the GBP
  
- Force-at-Commit (FAC) protocol ensures coherent access to data across members
  - ▶ DB2 “forces” (writes) updated pages to GBP at COMMIT (or before)
  - ▶ GBP synchronously invalidates any copies of such pages on other members
    - New references to the page on other members will retrieve new copy from GBP
    - In-progress references to page can continue





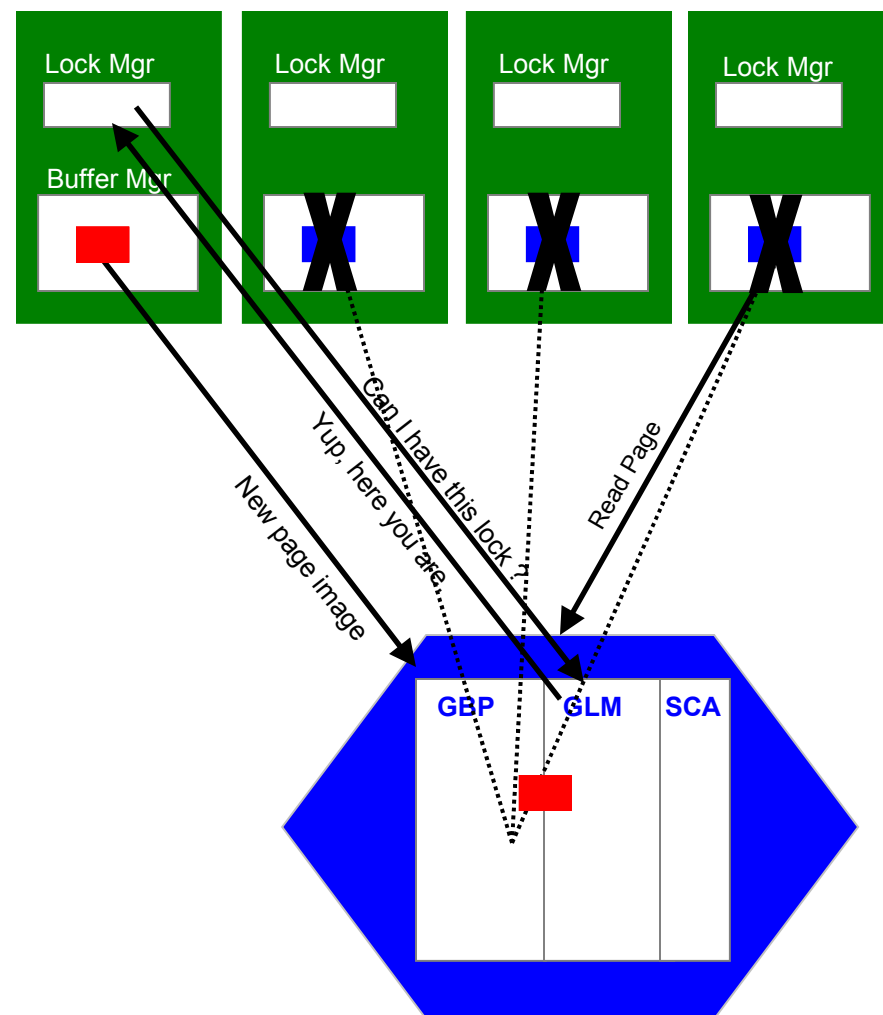
# The Role of the GLM

- Grants locks to members upon request
  - ▶ If not already held by another member, or held in a compatible mode
- Maintains global lock state
  - ▶ Which member has what lock, in what mode
  - ▶ Also - interest list of pending lock requests for each lock
- Grants pending lock requests when available
  - ▶ Via asynchronous notification
- Notes
  - ▶ When a member owns a lock, it may grant further, locally
  - ▶ "Lock Avoidance" : DB2 avoids lock requests when log sequence number in page header indicates no update on the page could be uncommitted



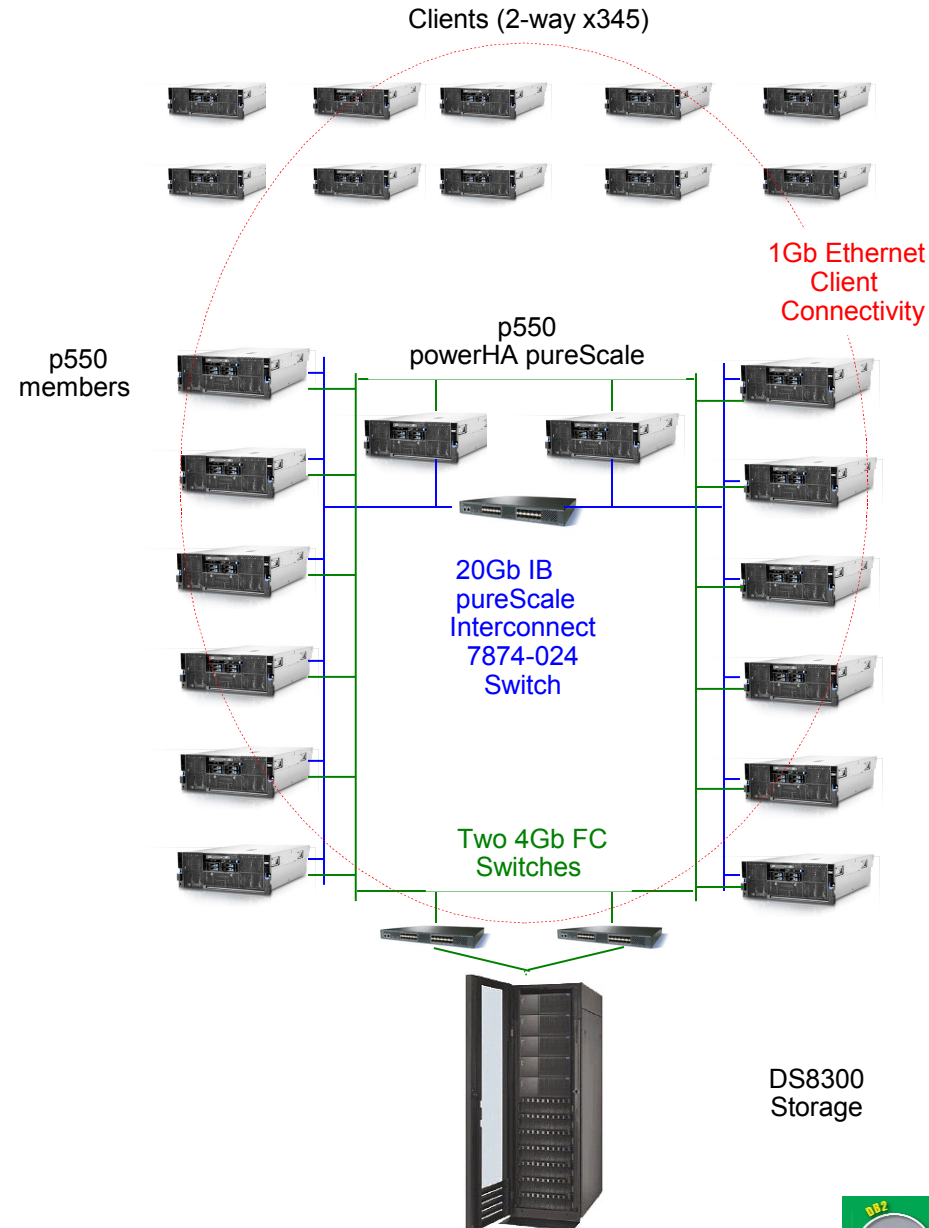
# Achieving Efficient Scaling : Key Design Points

- Deep RDMA exploitation over low latency fabric
  - ▶ Enables round-trip response time **~10-15 microseconds**
- Silent Invalidation
  - ▶ Informs members of page updates requires **no CPU cycles** on those members
  - ▶ No interrupt or other message processing required
  - ▶ Increasingly important as cluster grows
- Hot pages available without disk I/O from GBP memory
  - ▶ RDMA and dedicated threads enable read page operations in **~10s of microseconds**

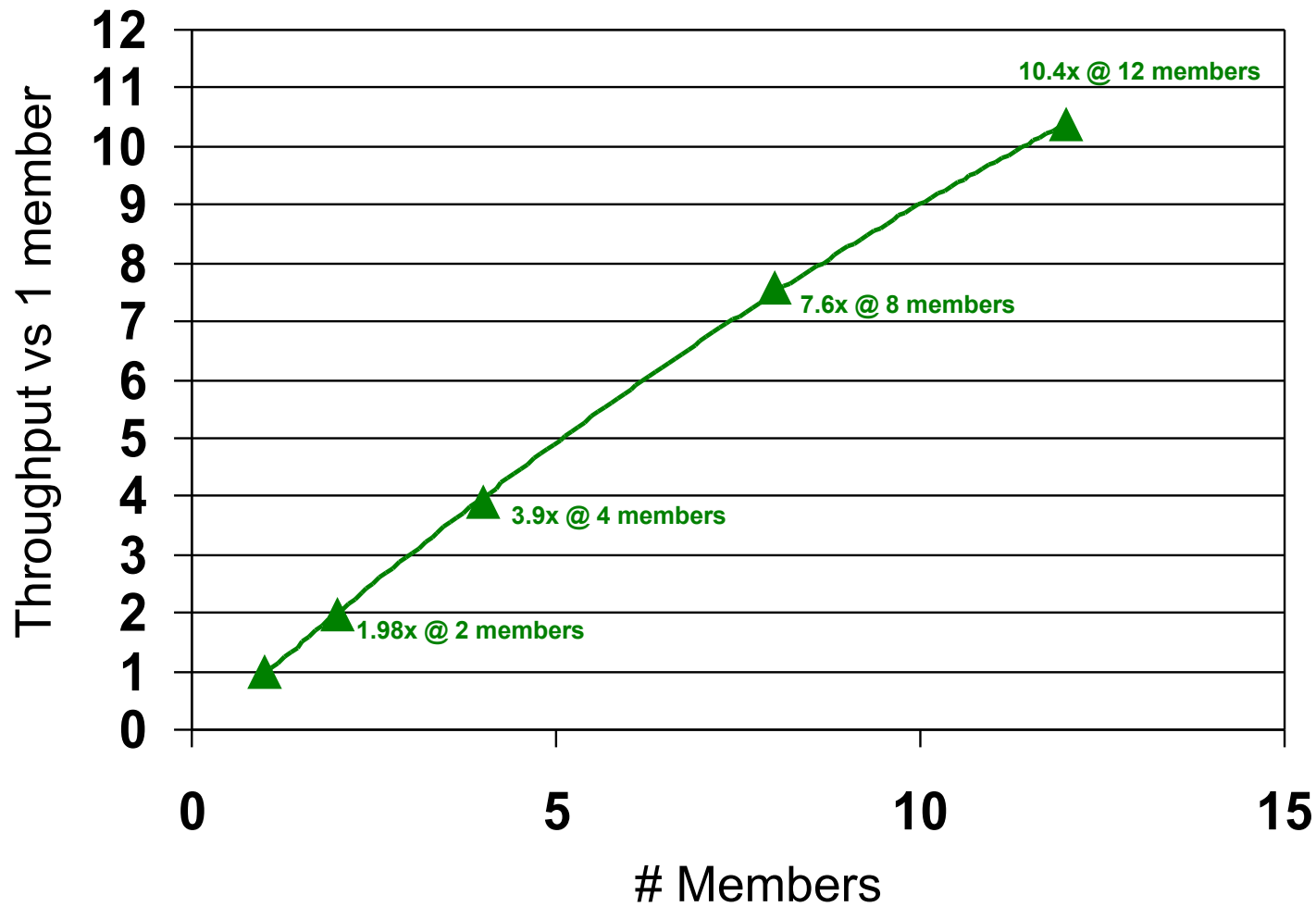


# Scalability : Example

- Transaction processing workload modeling warehouse & ordering process
  - ▶ Write transactions rate to 20%
  - ▶ Typical read/write ratio of many OLTP workloads
  
- No cluster awareness in the application
  - ▶ No affinity
  - ▶ No partitioning
  - ▶ No routing of transactions to members
  - ▶ Testing key *DB2 pureScale* design point
  
- Configuration
  - ▶ 12 8-core p550 members
    - 64 GB, 5 GHz each
  - ▶ Duplexed PowerHA pureScale across 2 additional 8-core p550s
    - 64 GB, 5 GHz each
  - ▶ DS8300 storage
    - 576 15K disks, Two 4Gb FC Switches
  - ▶ IBM 20Gb/s IB HCAs
    - 7874-024 IB Switch



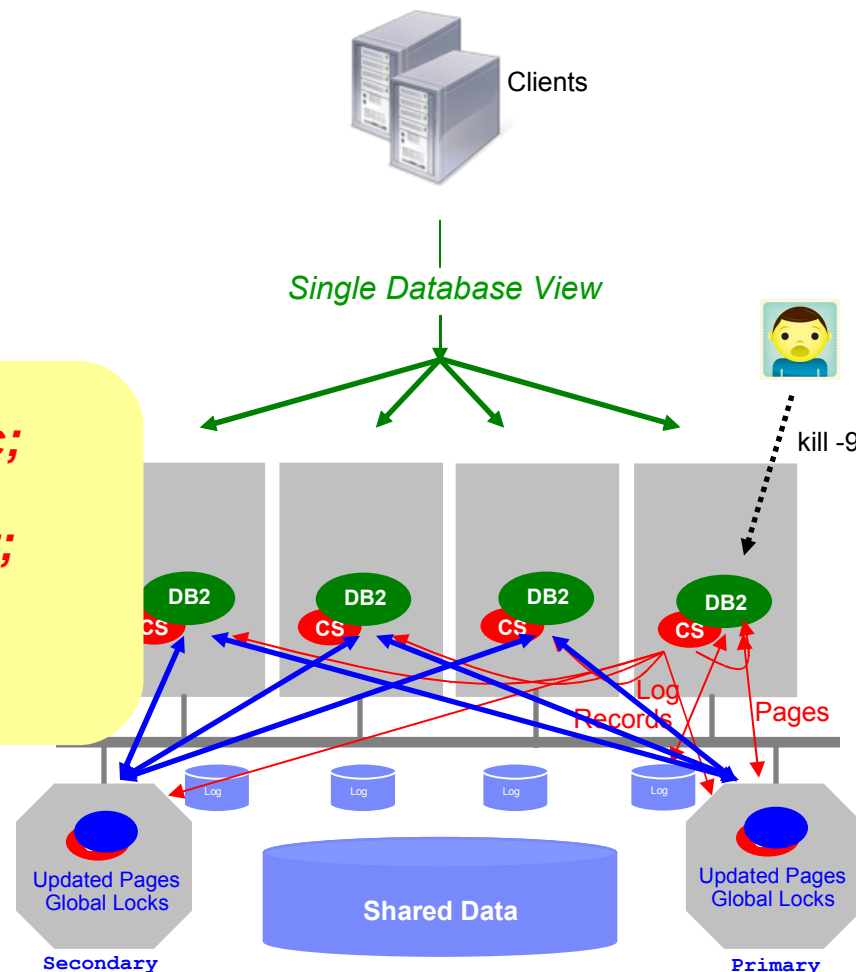
# Scalability : Example



# Member SW Failure : “Member Restart on Home Host”

- kill -9 erroneously issued to a member
- DB2 Cluster Services automatically detects member’s death
  - ▶ Informs other members & *powerHA pureScale* servers
  - ▶ Initiates automated member restart on same (“home”) host
  - ▶ Member restart is like a database crash recovery in a single system database, but is much faster
    - Redo limited to in-flight transaction
    - Benefits from page cache in GE
- In the mean-time, client connections transparently re-routed to healthy members
  - ▶ Based on least load (by default)
  - ▶ Pre-designated failover member
- Other members remain fully available throughout – “**Online Failover**”
  - ▶ Primary retains update locks held by member at the time of failure
  - ▶ Other members can continue to read and update data not locked for write access by failed member
- Member restart completes
  - ▶ Retained locks released and all data fully available

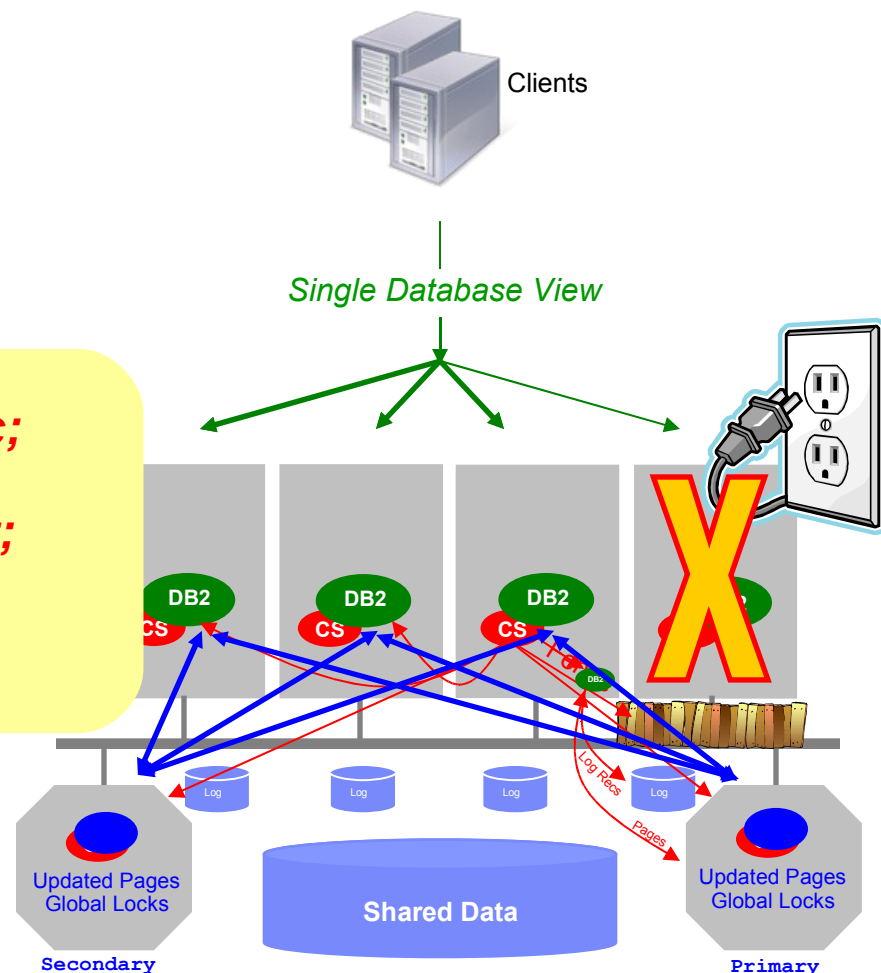
Automatic;  
Ultra Fast;  
Online



# Member HW Failure : “Member Restart on Guest Host (aka Restart Light)”

- Power cord tripped over accidentally
- DB2 Cluster Services loses heartbeat and declares member down
  - ▶ Informs other members & *PowerHA pureScale* servers
  - ▶ Fences member from logs and data
  - ▶ Initiates automated member restart on another (“guest”) host
    - Using reduced, and pre-allocated memory model
  - ▶ Member restart is like a database crash recovery in a single system database, but is more efficient
    - Redo limited to inflight transactions
    - Benefits from page cache in Primary
- In the mean-time, client connections are automatically re-routed to healthy members
  - ▶ Based on least load (by default)
  - ▶ Pre-designated failover member
- Other members remain fully available throughout – “**Online Failover**”
  - ▶ Primary retains update locks held by member at the time of failure
  - ▶ Other members can continue to read and update data not locked for write access by failed member
- Member restart completes
  - ▶ Retained locks released and all data fully available

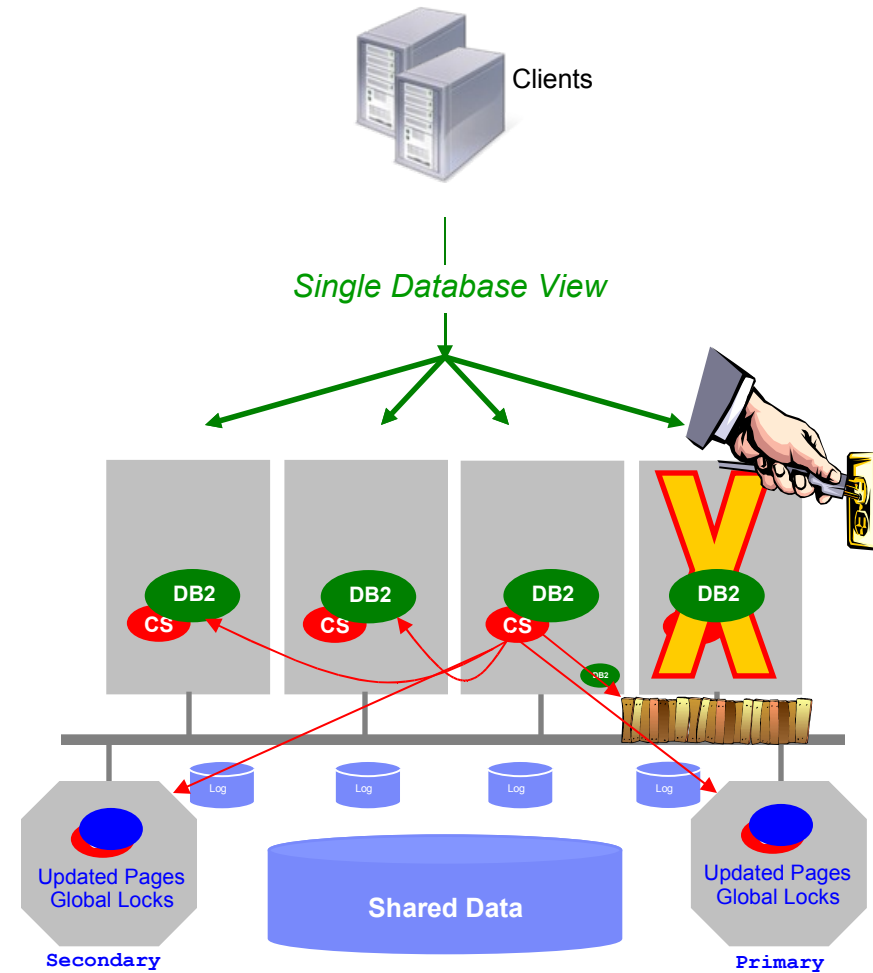
Automatic;  
Ultra Fast;  
Online





# Member Failback

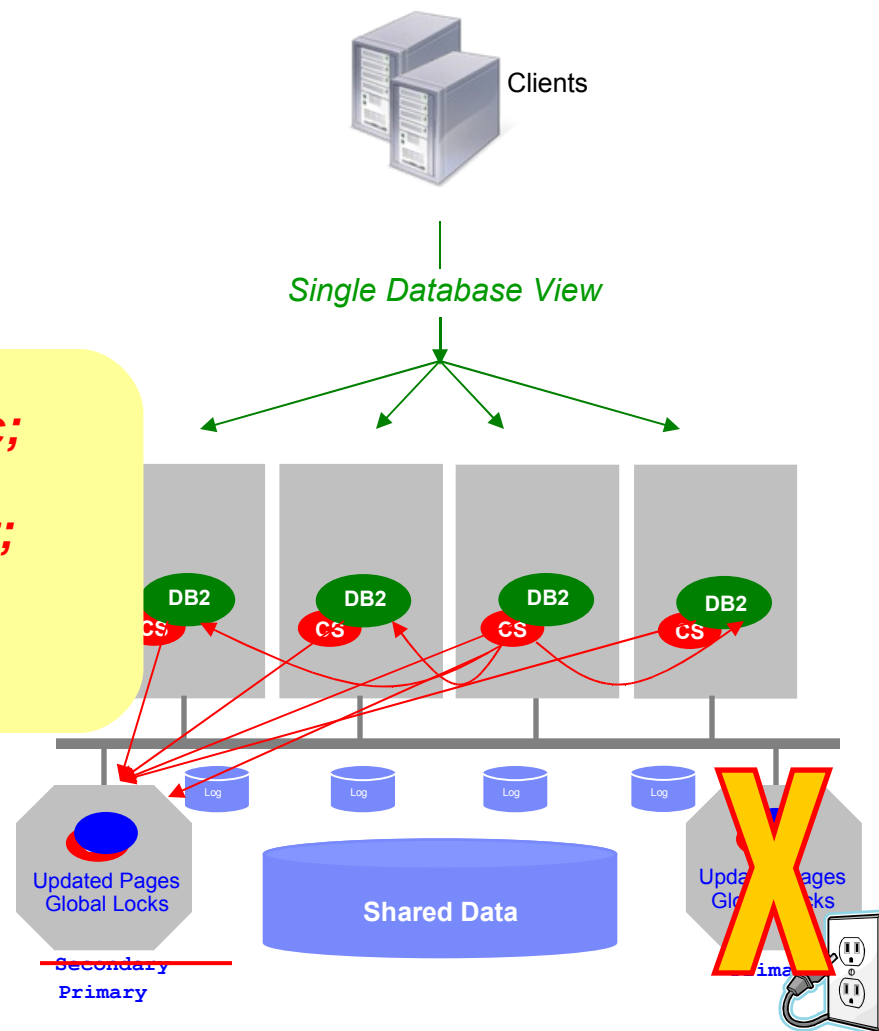
- Power restored and system re-booted
- DB2 Cluster Services automatically detects system availability
  - ▶ Informs other members and *PowerHA pureScale* servers
  - ▶ Removes fence
  - ▶ Brings up member on home host
- Client connections automatically re-routed back to member



# Primary *PowerHA pureScale* Failure

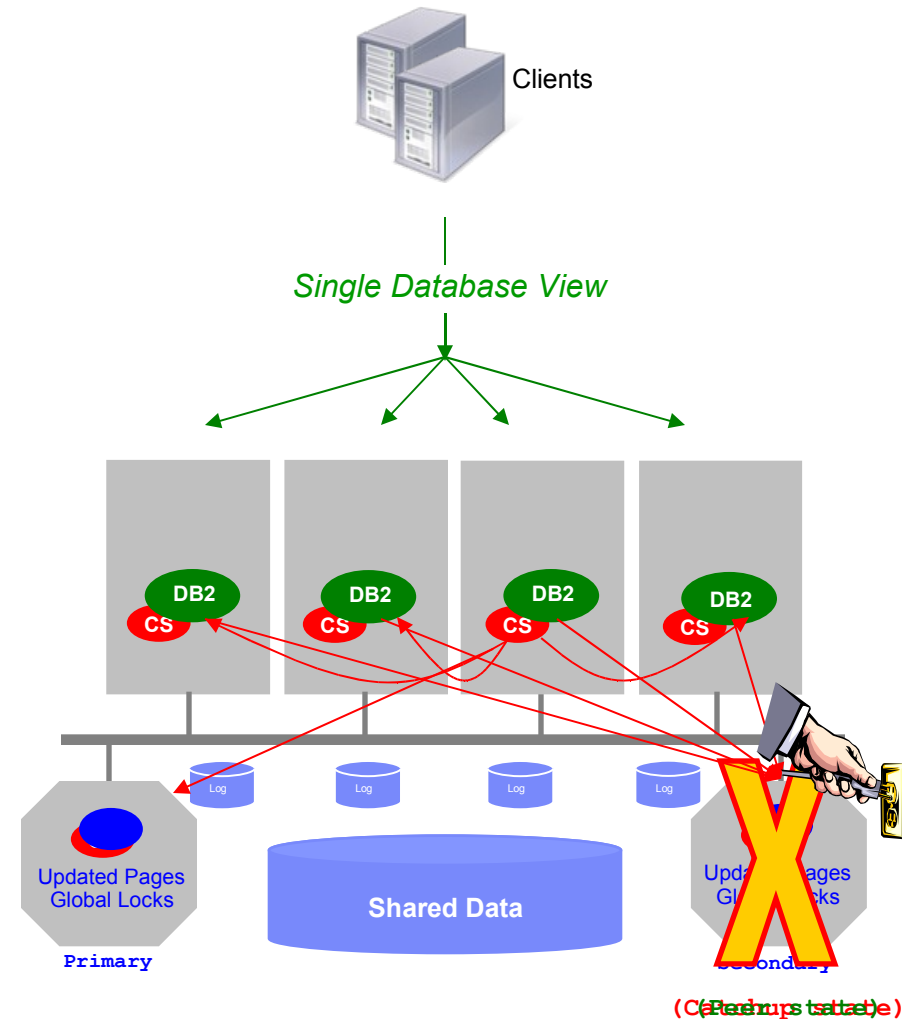
- Power cord tripped over accidentally
- DB2 Cluster Services loses heartbeat and declares primary down
  - ▶ Informs members and secondary
  - ▶ *PowerHA pureScale* service mode blocked
  - ▶ All other database activity proceeds
    - Eg. accessing pages in bufferpools, locks, sorting, aggregation, etc
- Members send missing data to secondary
  - ▶ Eg. read locks
- Secondary becomes primary
  - ▶ *PowerHA pureScale* service continues where it left off
  - ▶ No errors are returned to DB2 members

**Automatic;  
Ultra Fast;  
Online**



# PowerHA pureScale Re-integration

- Power restored and system re-booted
- DB2 Cluster Services automatically detects system availability
  - ▶ Informs members and primary
- New system assumes secondary role in 'catchup' state
  - ▶ Members resume duplexing
  - ▶ Members asynchronously send lock and other state information to secondary
  - ▶ Members asynchronously castout pages from primary to disk
- Catchup complete
  - ▶ Secondary in peer state (contains same lock and page state as primary)

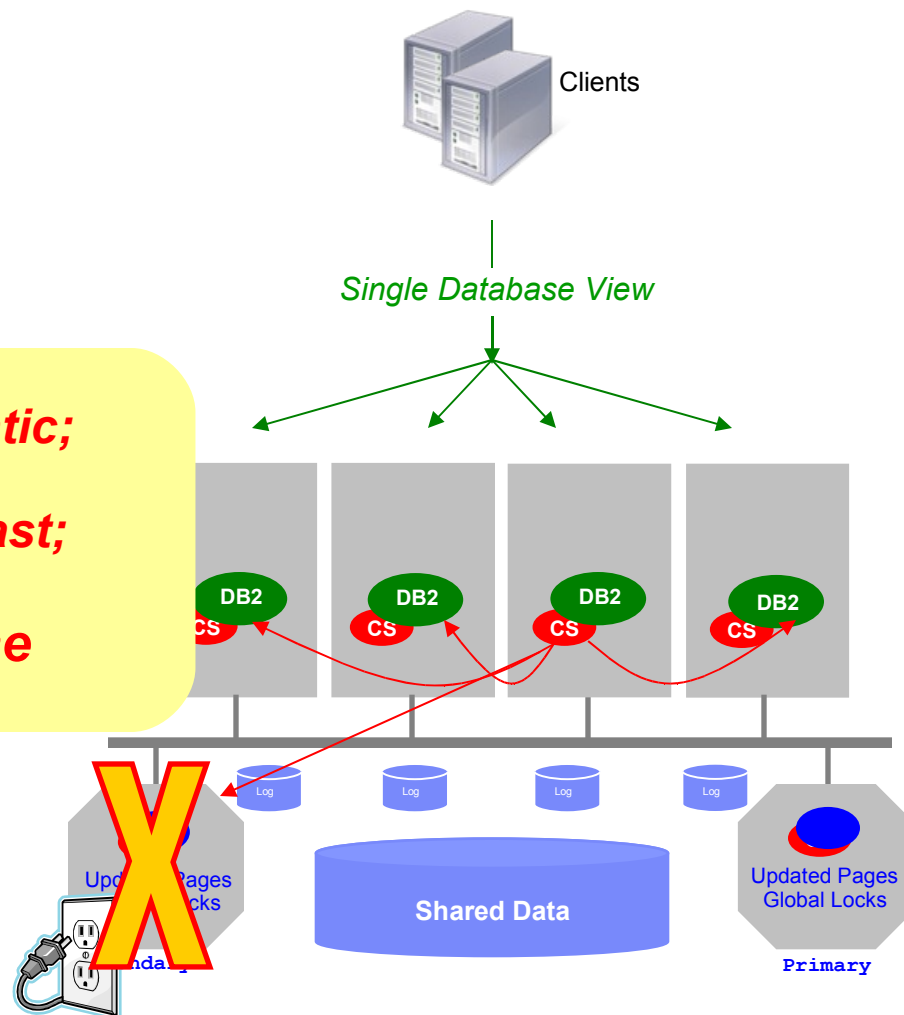


# Secondary *PowerHA pureScale* Failure

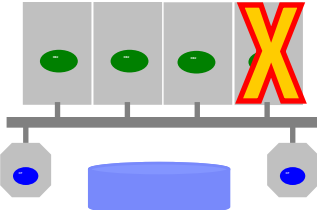
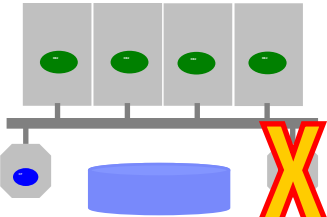
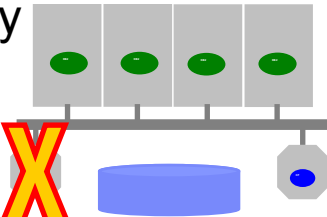
- Power cord tripped over accidentally
- DB2 Cluster Services loses heartbeat and declares secondary down
  - ▶ Informs members and primary
  - ▶ Members stop duplexing

Automatic;  
 Ultra Fast;  
 Online

- (Re-integration similar to previous chart)



# Summary (Single Failures)

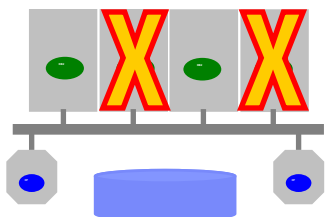
Failure Mode	Other Members Remain Online ?	Automatic & Transparent ?
<p>Member</p> 		<p>Connections to failed member transparently move to another member</p>
<p>Primary PowerHA pureScale</p> 		
<p>Secondary PowerHA pureScale</p> 		

# Simultaneous Failures

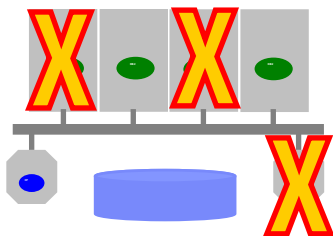
**Other Members Remain Online ?**

**Automatic & Transparent ?**

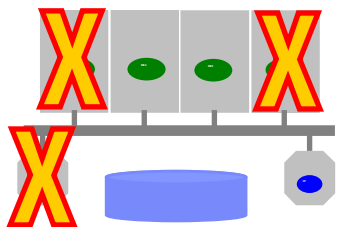
**Failure Mode**



Connections to failed member transparently move to another member



Connections to failed member transparently move to another member



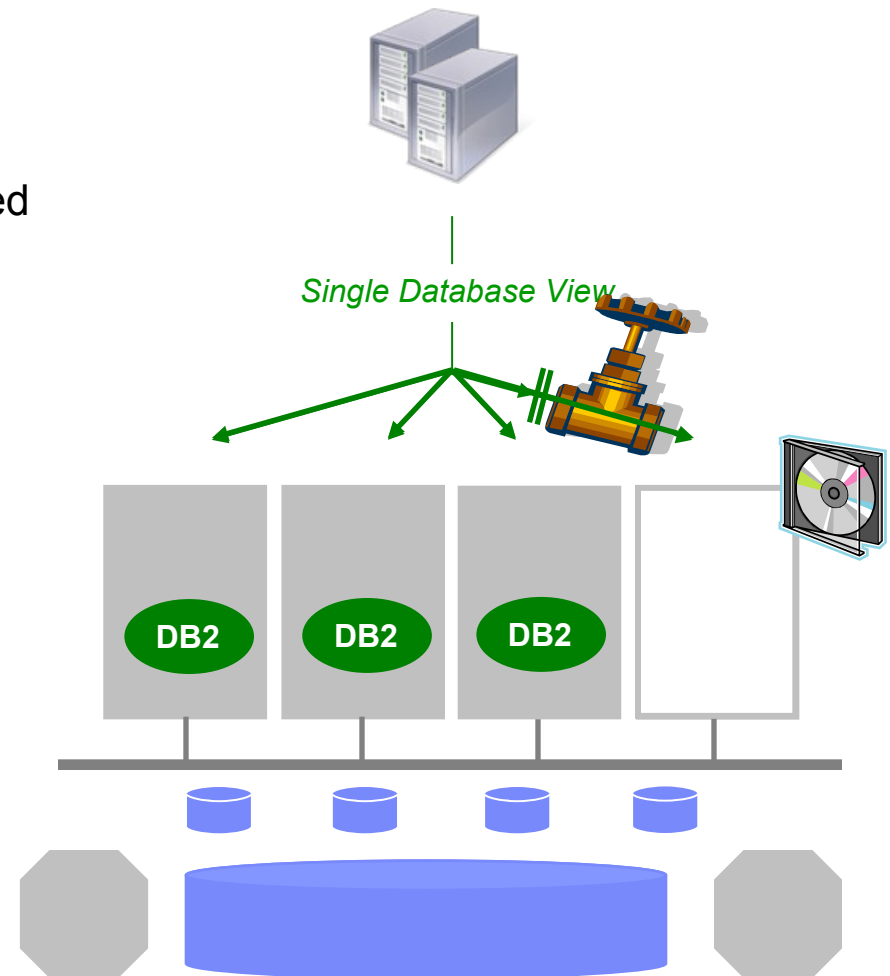
Connections to failed member transparently move to another member






# “Stealth” Maintenance : Example

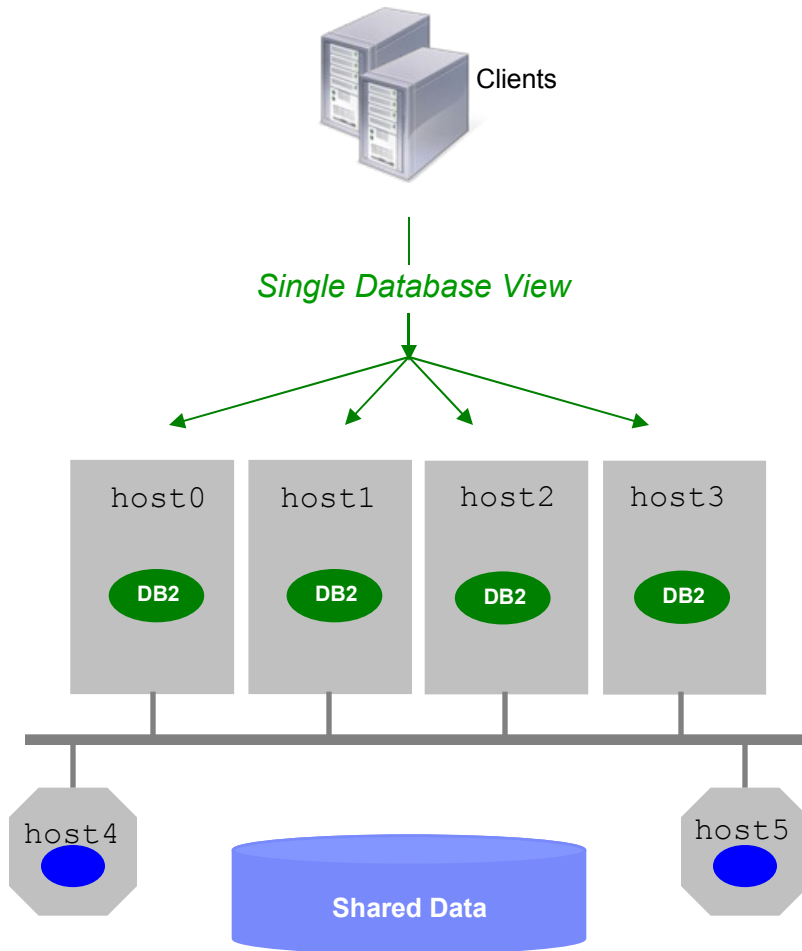
- ▶ Ensure automatic load balancing is enabled (it is by default)
- ▶ db2stop member 3 quiesce
- ▶ db2stop instance on host <hostname>
- ▶ Perform desired maintenance  
eg. install AIX PTF
- ▶ db2start instance on host <hostname>
- ▶ db2start member 3



# Agenda

- Introduction
  - ▶ Goals & Value Propositions
  - ▶ Technology Overview
  
- Technology In-Depth
  - ▶ Key Concepts & Internals
  - ▶ Efficient scaling
  - ▶ Failure modes & recovery automation
  - ▶ Stealth Maintenance
  
-  Configuration, Monitoring, Tooling
  - ▶ Cluster configuration and operational status
  - ▶ Monitoring data
  - ▶ Client configuration and load balancing
  - ▶ Installation

## db2nodes.cfg



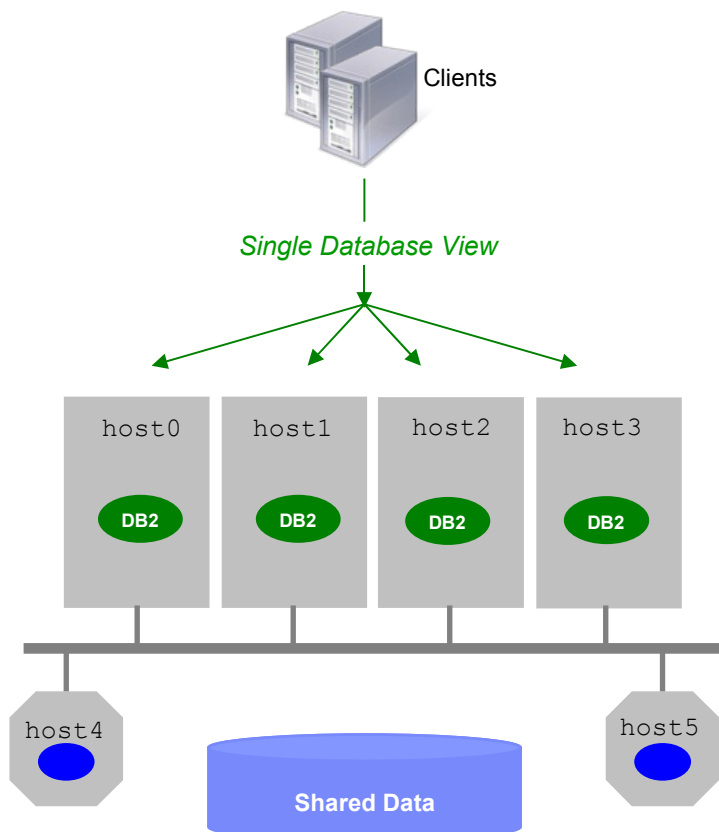
## db2nodes.cfg

```

0 host0 0 host0ib MEMBER
1 host1 0 host1ib MEMBER
2 host2 0 host2ib MEMBER
3 host3 0 host3ib MEMBER
4 host4 0 host4ib CF
5 host5 0 host5ib CF

```

# Instance and Host Status



## > db2start

```
08/24/2008 00:52:59 0 0 SQL1063N DB2START processing was successful.
08/24/2008 00:53:00 1 0 SQL1063N DB2START processing was successful.
08/24/2008 00:53:01 2 0 SQL1063N DB2START processing was successful.
08/24/2008 00:53:01 3 0 SQL1063N DB2START processing was successful.
SQL1063N DB2START processing was successful.
```

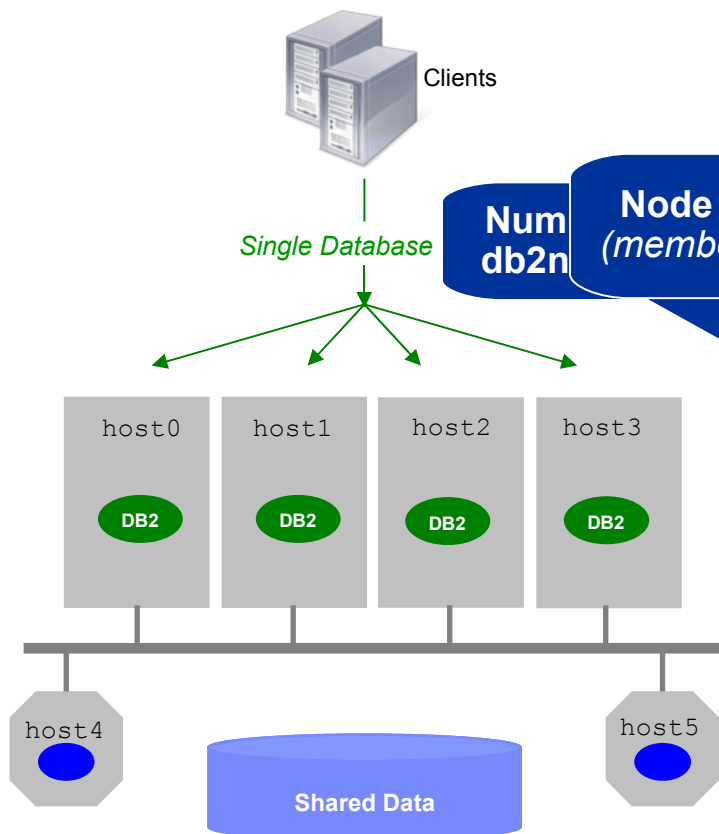
## > db2instance -list

ID	TYPE	STATE	HOME_HOST	CURRENT_HOST	ALERT
0	MEMBER	STARTED	host0	host0	NO
1	MEMBER	STARTED	host1	host1	NO
2	MEMBER	STARTED	host2	host2	NO
3	MEMBER	STARTED	host3	host3	NO
4	CF	PRIMARY	host4	host4	NO
5	CF	PEER	host5	host5	NO

HOST_NAME	STATE	INSTANCE_STOPPED	ALERT
host0	ACTIVE	NO	NO
host1	ACTIVE	NO	NO
host2	ACTIVE	NO	NO
host3	ACTIVE	NO	NO
host4	ACTIVE	NO	NO
host5	ACTIVE	NO	NO



# Instance Status



**Num db2n**

**Node type (member, CF)**

**Node state**  
 For members typically (started, stopped, waiting\_for\_failback)  
 For CFs typically (primary, peer, stopped, catchup(##%), res)

**Target host member**  
 (Member tries to run on this host when it is available.)

**Where member or CF is currently running**  
 (Normally same as home host. If differs, usually indicates home host failed and member restarting)

**Does the member or CF require attention?**  
 (Example: member restart failed)

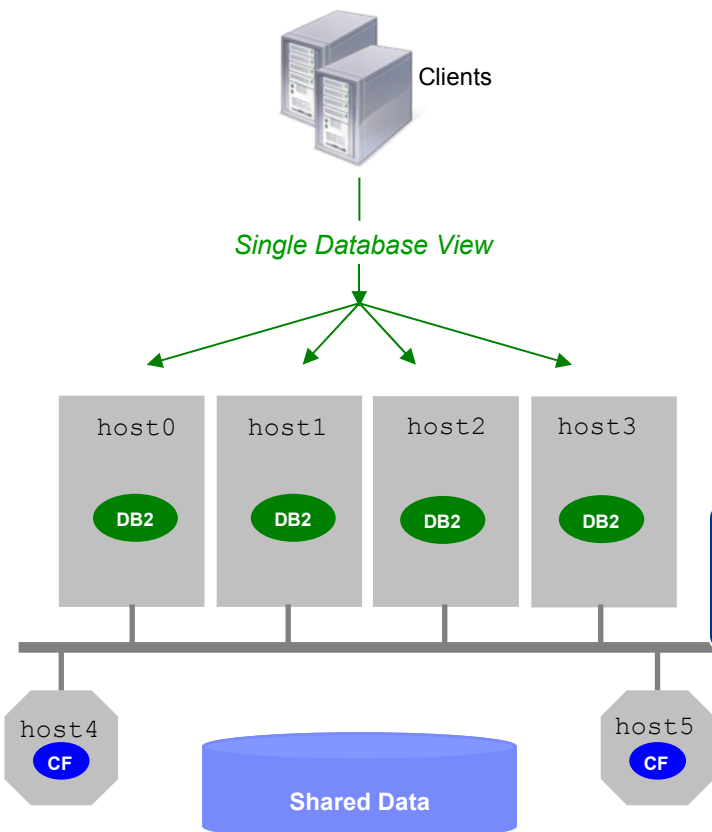
> db2instance list

ID	TYPE	STATE	HOME_HOST	CURRENT_HOST	ALERT
0	MEMBER	STARTED	host0	host0	NO
1	MEMBER	STARTED	host1	host1	NO
2	MEMBER	STARTED	host2	host2	NO
3	MEMBER	STARTED	host3	host3	NO
4	CF	PRIMARY	host4	host4	NO
5	CF	PEER	host5	host5	NO

HOST_NAME	STATE	INSTANCE_STOPPED	ALERT
host0	ACTIVE	NO	NO
host1	ACTIVE	NO	NO
host2	ACTIVE	NO	NO
host3	ACTIVE	NO	NO
host4	ACTIVE	NO	NO
host5	ACTIVE	NO	NO



# Host Status



## > db2start

```
08/24/2008 00:52:59 0 0 SQL1063N DB2START processing was successful.
08/24/2008 00:53:00 1 0 SQL1063N DB2START processing was successful.
08/24/2008 00:53:01 2 0 SQL1063N DB2START processing was successful.
08/24/2008 00:53:01 3 0 SQL1063N DB2START processing was successful.
SQL1063N DB2START processing was successful.
```

## > db2instance -li

**Has the instance been disabled on this host?**

DBAs can stop (aka disable) the instance on the host for the purposes of maintenance (eg. Upgrades). While disabled, member replication and other DB2 activity is prevented on the host.

**Does the host require attention?**

(Examples: power failure, can't communicate with host)

**Host state**  
(active) indicates the host is up and available.

(inactive) indicates the host is down and not available.

```
CF PEER host3
CF PEER host4
CF PEER host5
```

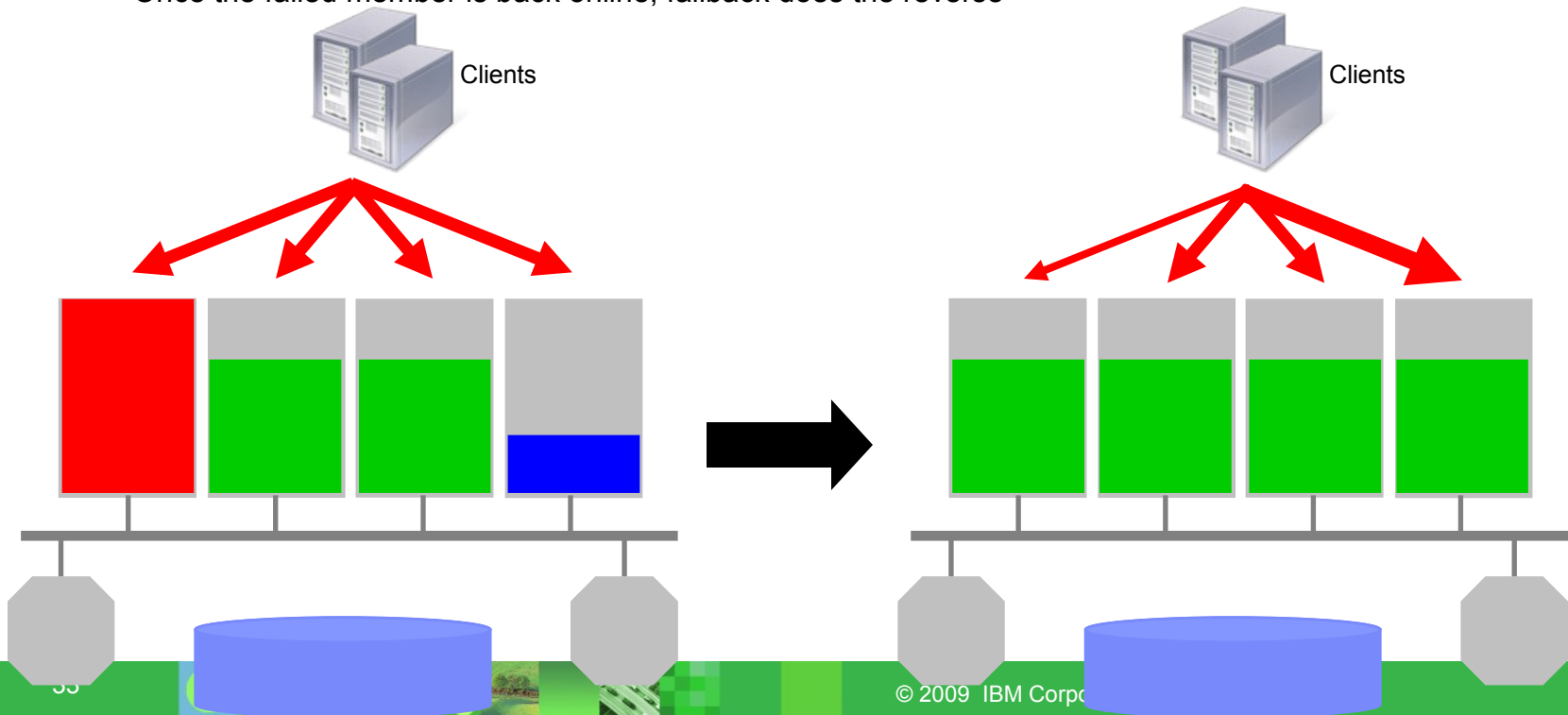
HOST_NAME	STATE	INSTANCE_STOPPED	ALERT
host0	ACTIVE	NO	NO
host1	ACTIVE	NO	NO
host2	ACTIVE	NO	NO
host3	ACTIVE	NO	NO
host4	ACTIVE	NO	NO
host5	ACTIVE	NO	NO





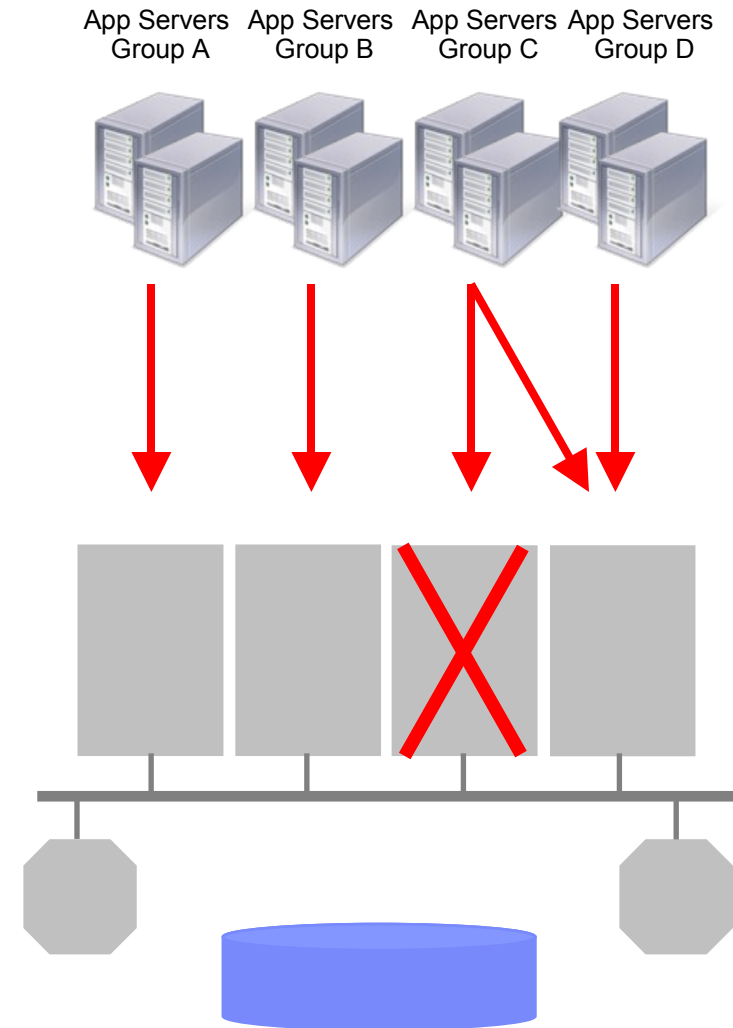
# Client Connectivity and Workload Balancing

- Run-time load information used to automatically balance load across the cluster (as in System z sysplex)
  - Load information of all members kept on each member
  - Piggy-backed to clients regularly
  - Used to route next connection (or optionally next transaction) to least loaded member
  - Routing occurs automatically (transparent to application)
- Failover
  - Load of failed member evenly distributed to surviving members automatically
- Fallback
  - Once the failed member is back online, fallback does the reverse



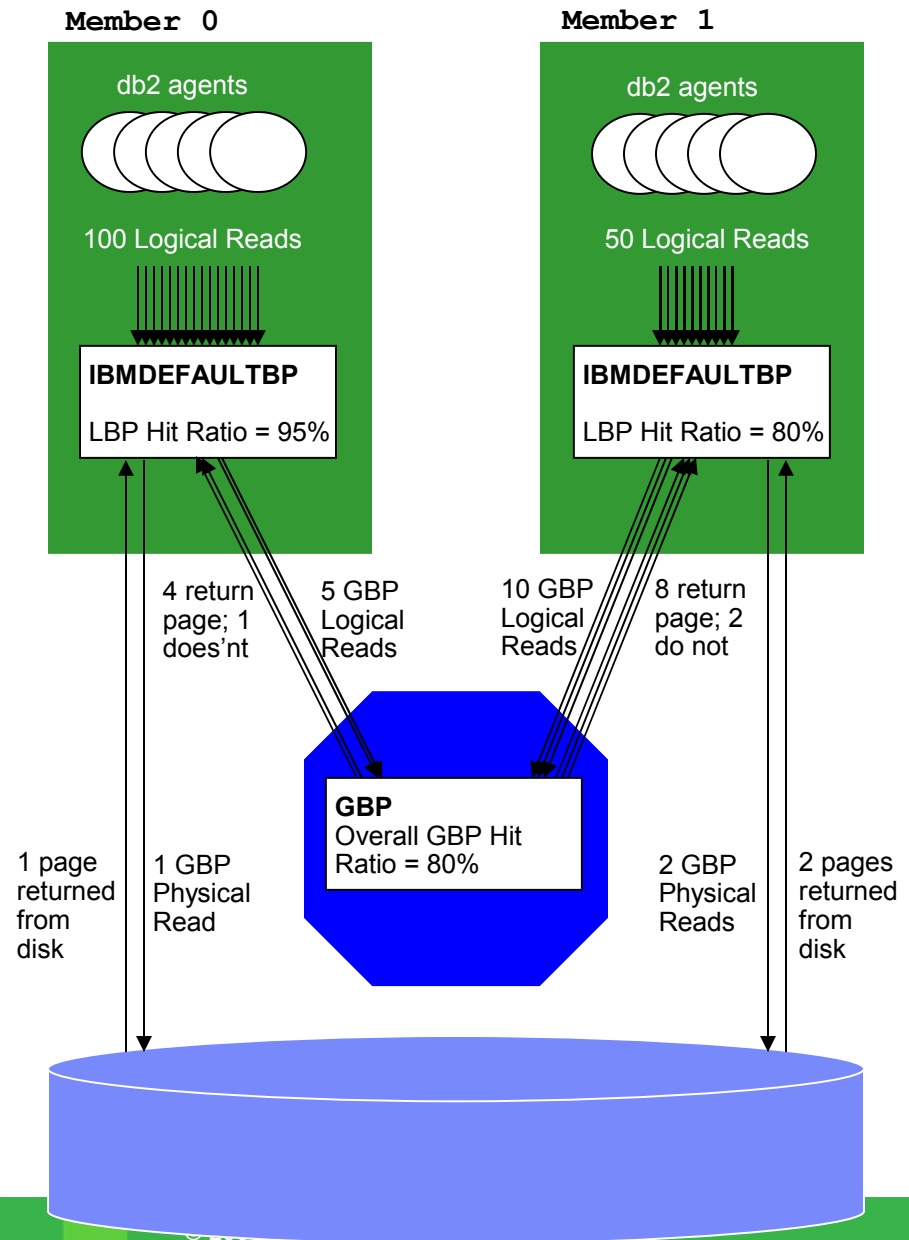
# Optional Affinity-based Routing

- Allows you to target different groups of clients or workloads to different members in the cluster
  - ▶ Maintained after failover ...
  - ▶ ... and fallback
- Example use cases
  - ▶ Consolidate separate workloads/applications on same database infrastructure
  - ▶ Minimize total resource requirements for disjoint workloads
- Easily configured through client configuration
  - ▶ db2dsdriver.cfg file

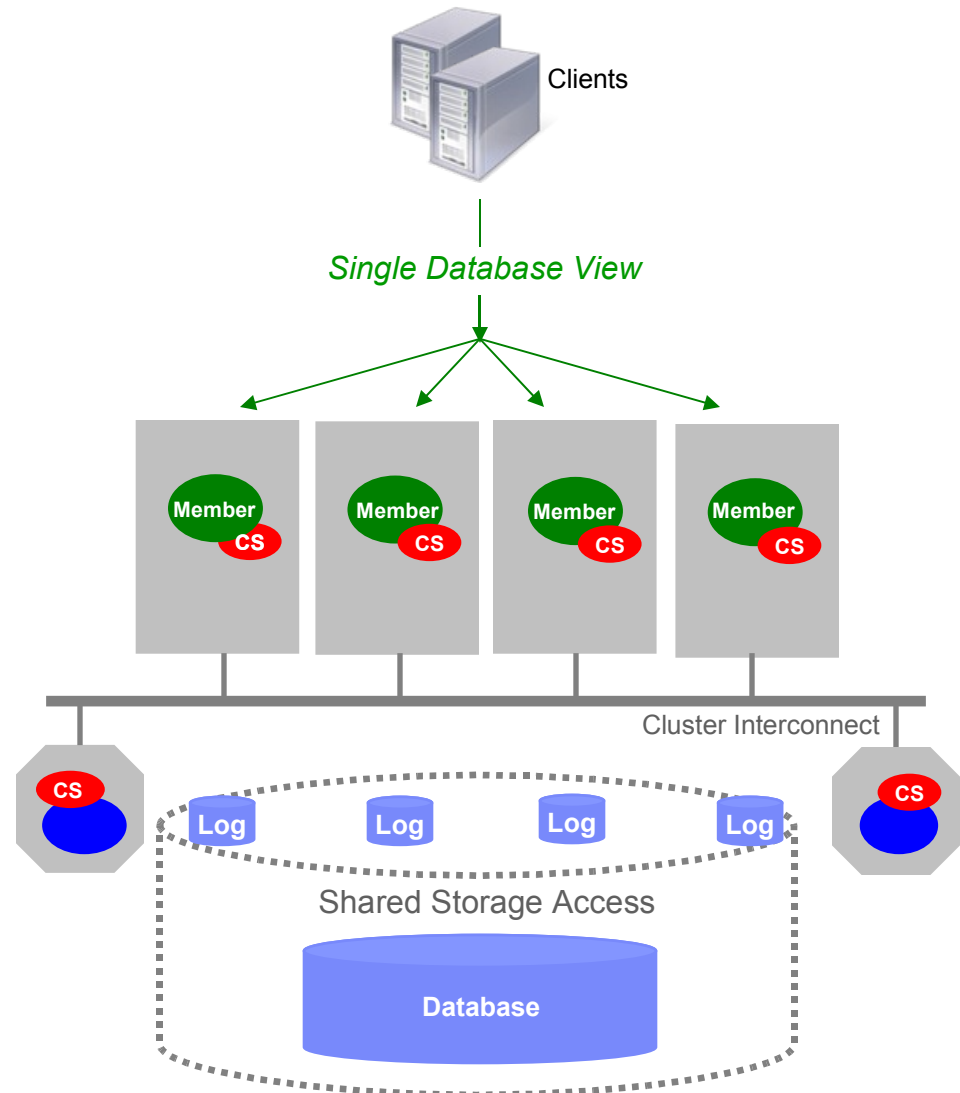


# Operational Monitoring

- New monitoring views and SQL functions
  - ▶ Global locking and global bufferpool statistics
  - ▶ Drill down into other *PowerHA pureScale* internal statistics
  - ▶ Cluster communications time
  - ▶ Cross-member page access statistics
- Drill down per member...  
... or get global view
  - ▶ Available from any member
- Event monitors “always available” mode
  - ▶ DB2 *pureScale* chooses initial member automatically
  - ▶ Fails over automatically if member fails
- Various new monitoring elements
  - ▶ Example, GBP tuning related elements (partial list):
    - DATA\_GBP\_L\_READS
    - DATA\_GBP\_P\_READS
    - INDEX\_GBP\_L\_READS
    - INDEX\_GBP\_P\_READS



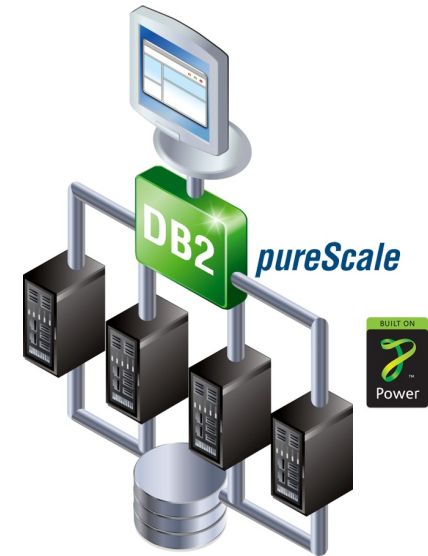
# DB2 pureScale : A Complete Solution



- **DB2 pureScale is a complete software solution**
  - ▶ Comprised of tightly integrated subcomponents
- **Single install invocation**
  - ▶ Installs all components across desired hosts
  - ▶ Automatically configures best practices
- **No cluster manager scripting or configuration required**
  - ▶ This is set up automatically, upon installation

# DB2 pureScale

- Unlimited Capacity
  - ▶ Start small
  - ▶ Grow easily, with your business
- Application Transparency
  - ▶ Avoid the risk and cost of tuning your applications to the database topology
- Continuous Availability
  - ▶ Maintain service across planned and unplanned events



# > Questions



Thank You!

# [ibm.com/db2/labchats](http://ibm.com/db2/labchats)



*Thank you for attending!*

