Information Management

IBM

DB2 pureScale Overview

Chris Eaton
WW Technical DB2 Specialist
ceaton@ca.ibm.com

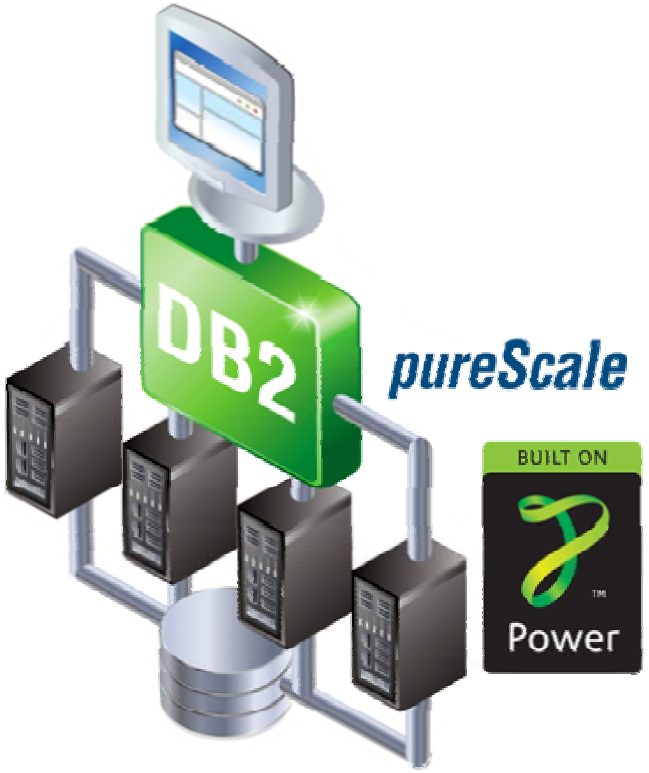
Agenda

- What is DB2 pureScale?
- Where Does DB2 pureScale Come From?
- Transparent Application Scalability
- High Availability
- Geographic clustering
- Competitive Comparison for Availability
- Competitive Comparison for Scalability

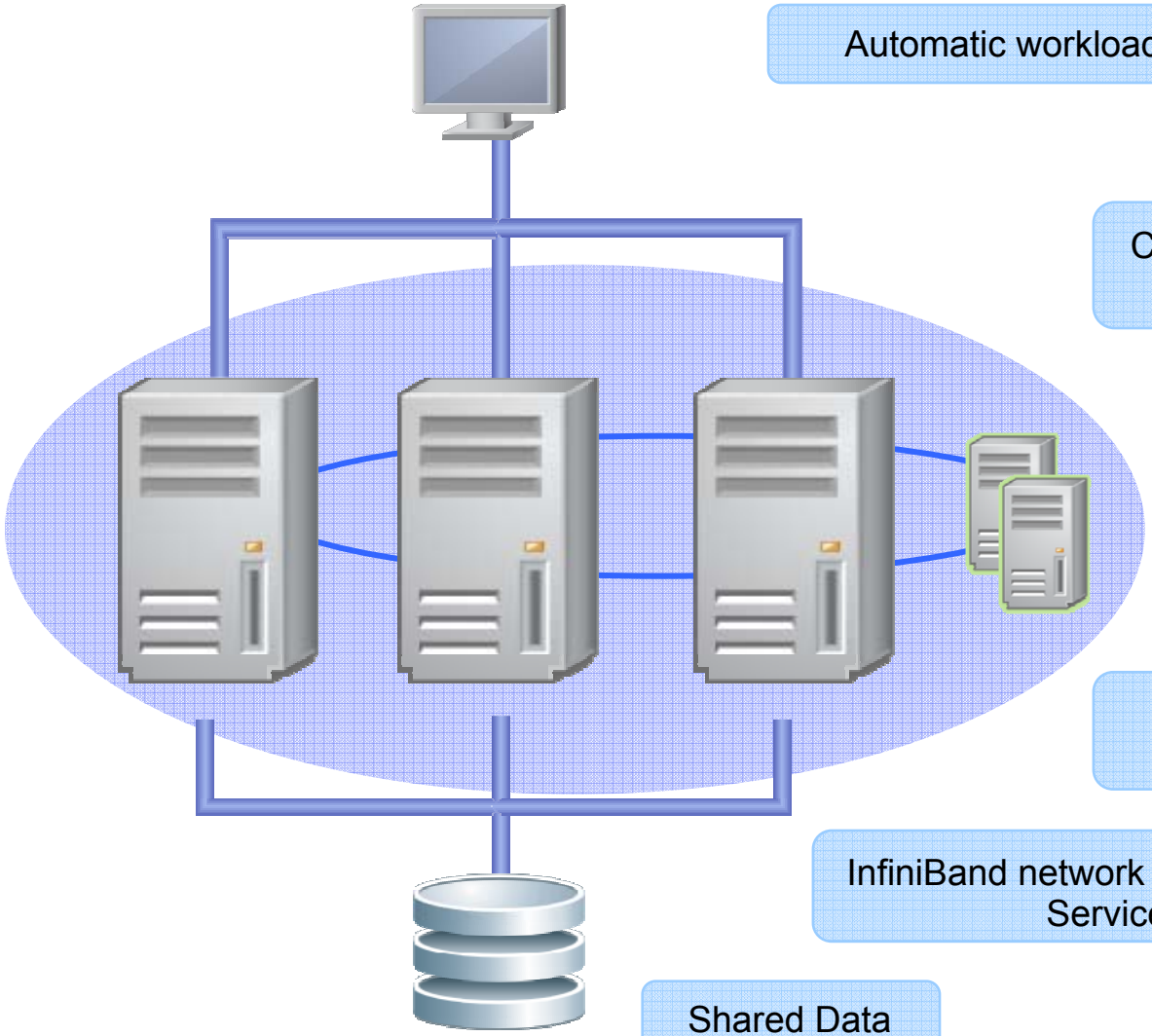
What is DB2 pureScale?

DB2 pureScale

- **Unlimited Capacity**
  - Buy only what you need, add capacity as your needs grow
- **Application Transparency**
  - Avoid the risk and cost of application changes
- **Continuous Availability**
  - Deliver uninterrupted access to your data with consistent performance



DB2 pureScale Architecture



Automatic workload balancing

Cluster of DB2 nodes running on Power servers

Leverages the global lock and global cache manager technology from z/OS

Integrated Cluster Manager

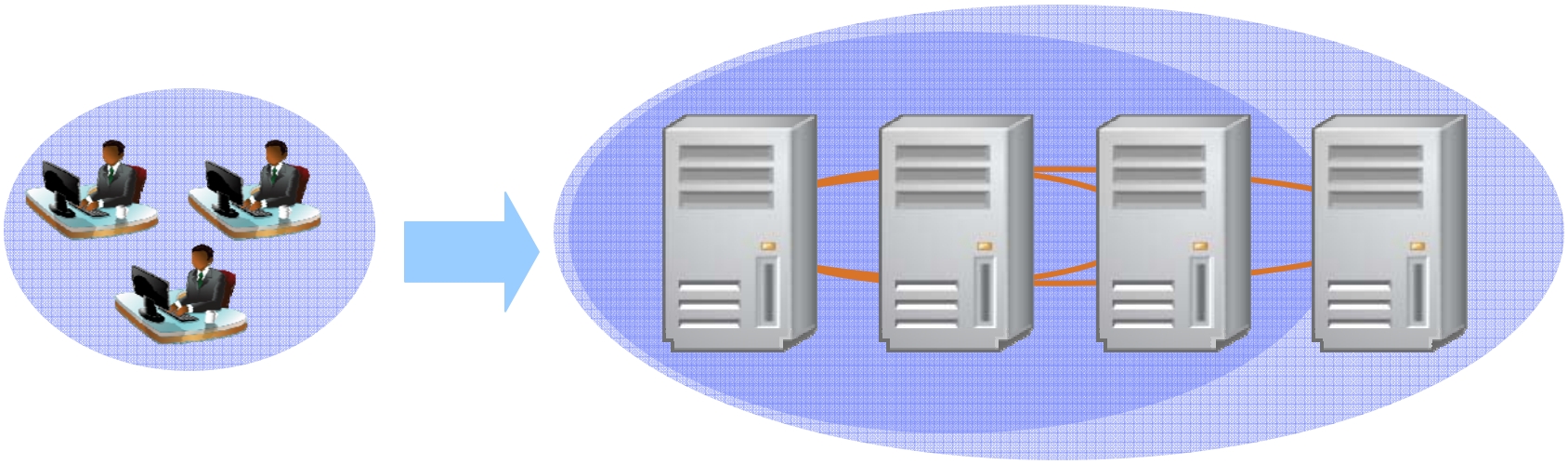
InfiniBand network & DB2 Cluster Services

Shared Data

Application Transparency

Take advantage of extra capacity instantly

- No need to modify your application code
- No need to tune your database infrastructure



Your DBAs can add capacity without re-tuning or re-testing

Your developers don't even need to know more nodes are being added

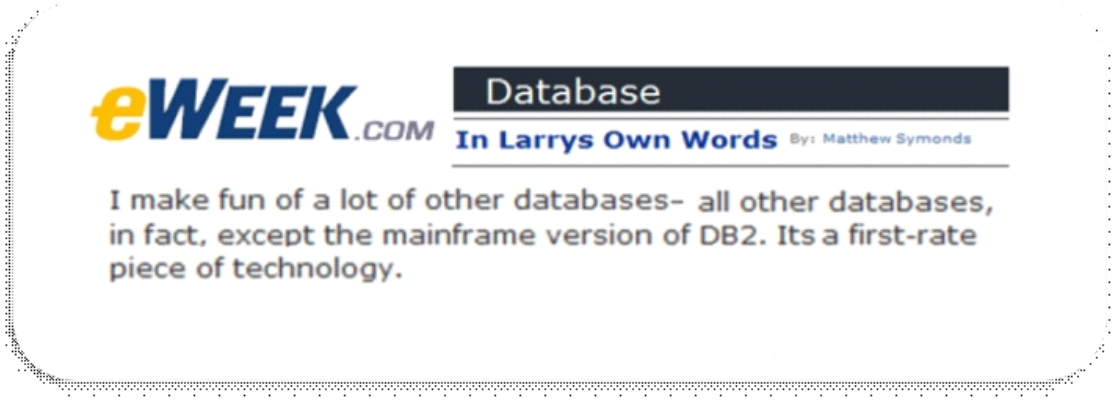
IBM

Where Does DB2 pureScale Come From?

DB2 for z/OS Data Sharing is the Gold Standard

- **Everyone recognizes DB2 for z/OS as the "Gold" standard for scalability and high availability**

- **Even Oracle agrees:**

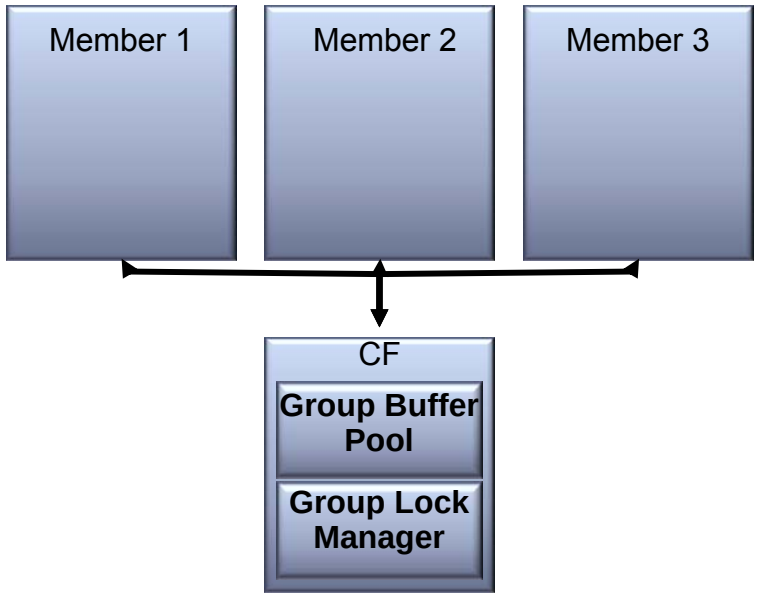


- **Why?**
  - The Coupling Facility!!
    - Centralized locking, centralized buffer pool deliver superior scalability and superior availability
  - The entire environment on z/OS uses the Coupling Facility
    - CICS, MQ, IMS, Workload Management, and more

The Key to Scalability and High Availability

- **Efficient Centralized Locking and Caching**
  - As the cluster grows, DB2 maintains one place to go for locking information and shared pages
  - Optimized for very high speed access
    - DB2 pureScale uses Remote Direct Memory Access (RDMA) to communicate with the powerHA pureScale server
    - No IP socket calls, no interrupts, no context switching

- **Results**
  - Near Linear Scalability to large numbers of servers
  - Constant awareness of what each member is doing
    - If one member fails, no need to block I/O from other members
    - Recovery runs at memory speeds



Information Management

IBM

Transparent Application Scaling

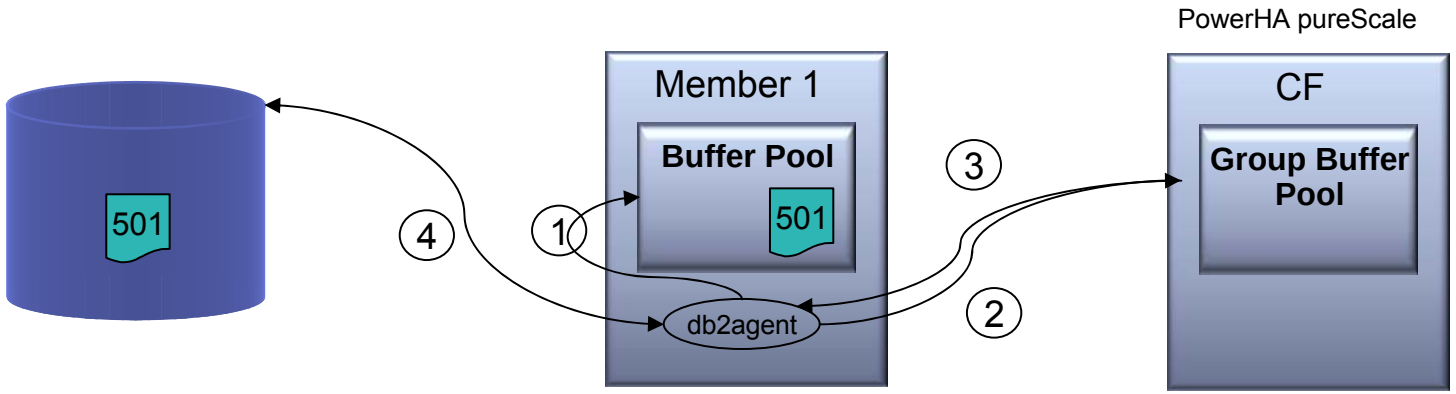
Transparent Application Scalability

- **Scalability without application or database partitioning**
  - Centralized locking and real global buffer pool with RDMA access results in real scaling without making application cluster aware
    - Sharing of data pages is via RDMA from a true shared cache
      - not synchronized access via process interrupts between servers)
    - No need to partition application or data for scalability
      - Resulting in lower administration and application development costs
  - Distributed locking in RAC results in higher overhead and lower scalability
    - Oracle RAC best practices recommends
      - Fewer rows per page (to avoid hot pages)
      - Partition database to avoid hot pages
      - Partition application to get some level of scalability
      - All of these result in higher management and development costs

What Happens in DB2 pureScale to Read a Page

Agent on Member 1 wants to read page 501

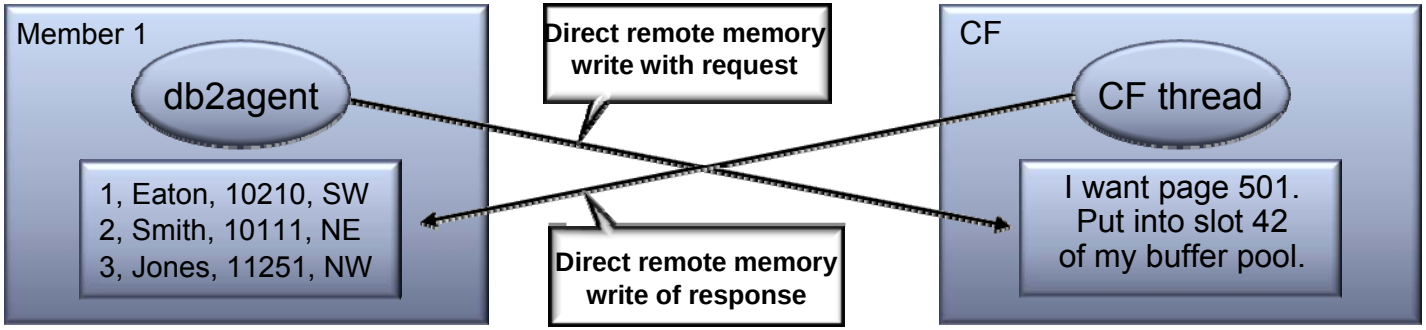
1. db2agent checks local buffer pool: page not found
2. db2agent performs Read And Register (RaR) RDMA call directly into CF memory
   - No context switching, no kernel calls.
   - Synchronous request to CF
3. CF replies that it does not have the page (again via RDMA)
4. db2agent reads the page from disk



The Advantage of DB2 Read and Register with RDMA

1. **DB2 agent on Member 1 writes directly into CF memory with:**
   - Page number it wants to read
   - Buffer pool slot that it wants the page to go into
2. **CF either responds by writing directly into memory on Member 1:**
   - That it does not have the page **or**
   - With the requested page of data

- **Total end to end time for RAR is measured in microseconds**
- **Calls are very fast, the agent may even stay on the CPU for the response**

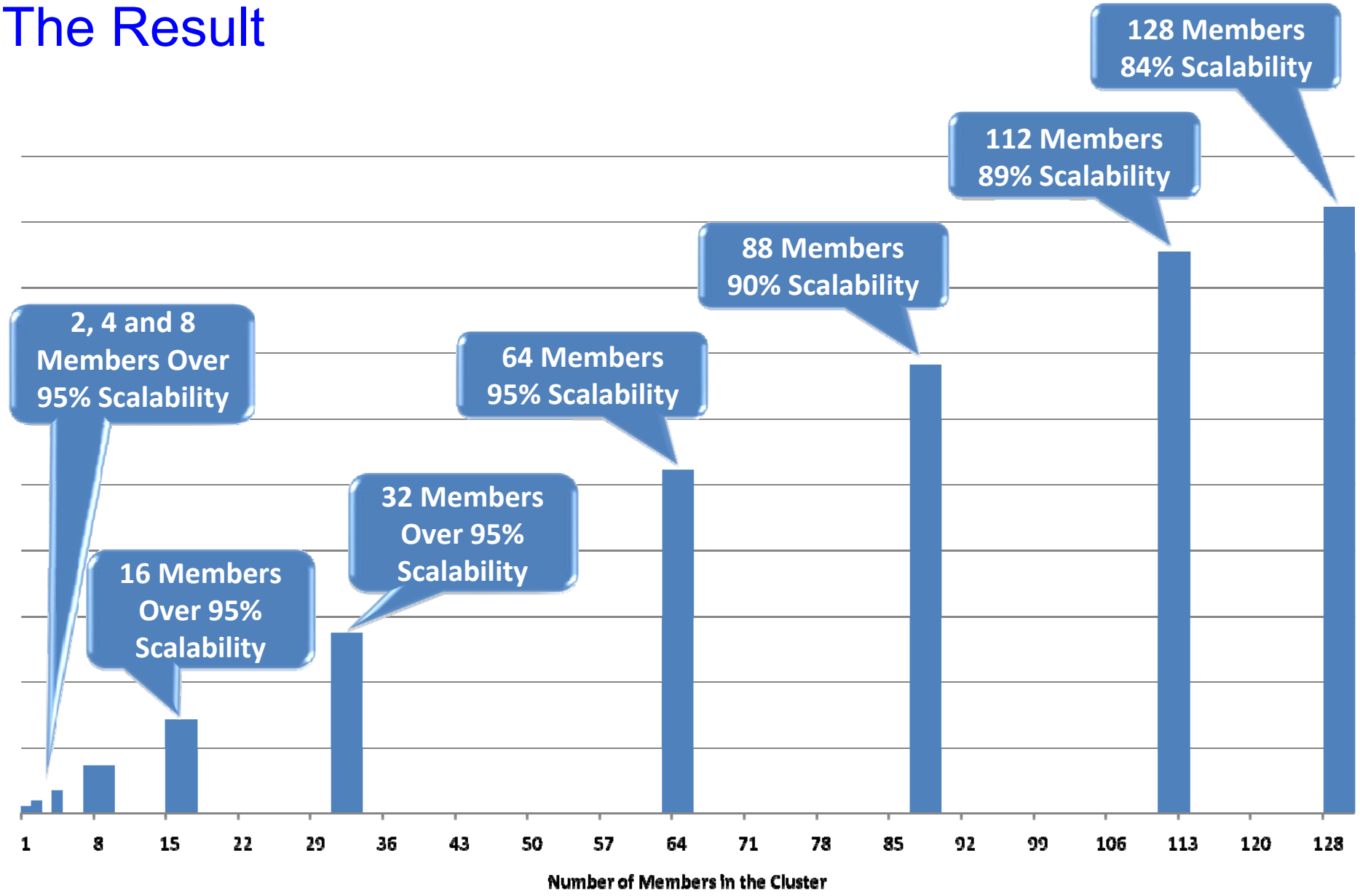


Much more scalable, does not require locality of data

Proof of DB2 pureScale Architecture Scalability

- **How far will it scale?**
- **Take a web commerce type workload**
  - Read mostly but **not read only**
- **Don’t make the application cluster aware**
  - **No routing of transactions to members**
  - Demonstrate transparent application scaling
- **Scale out to the 128 member limit and measure scalability**

The Result

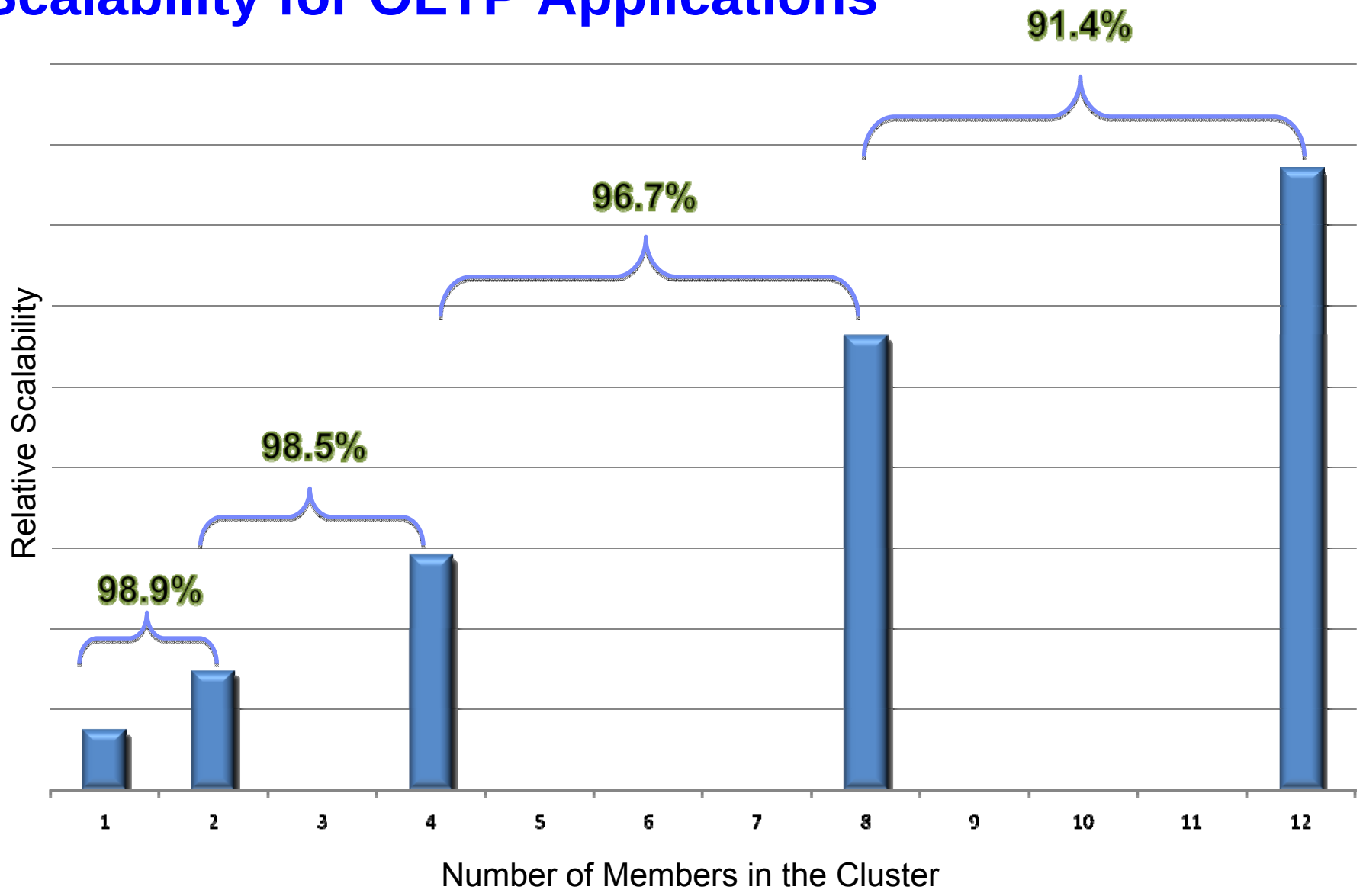


Validation testing includes capabilities to be available in future releases.

Dive Deeper into a 12 Member Cluster

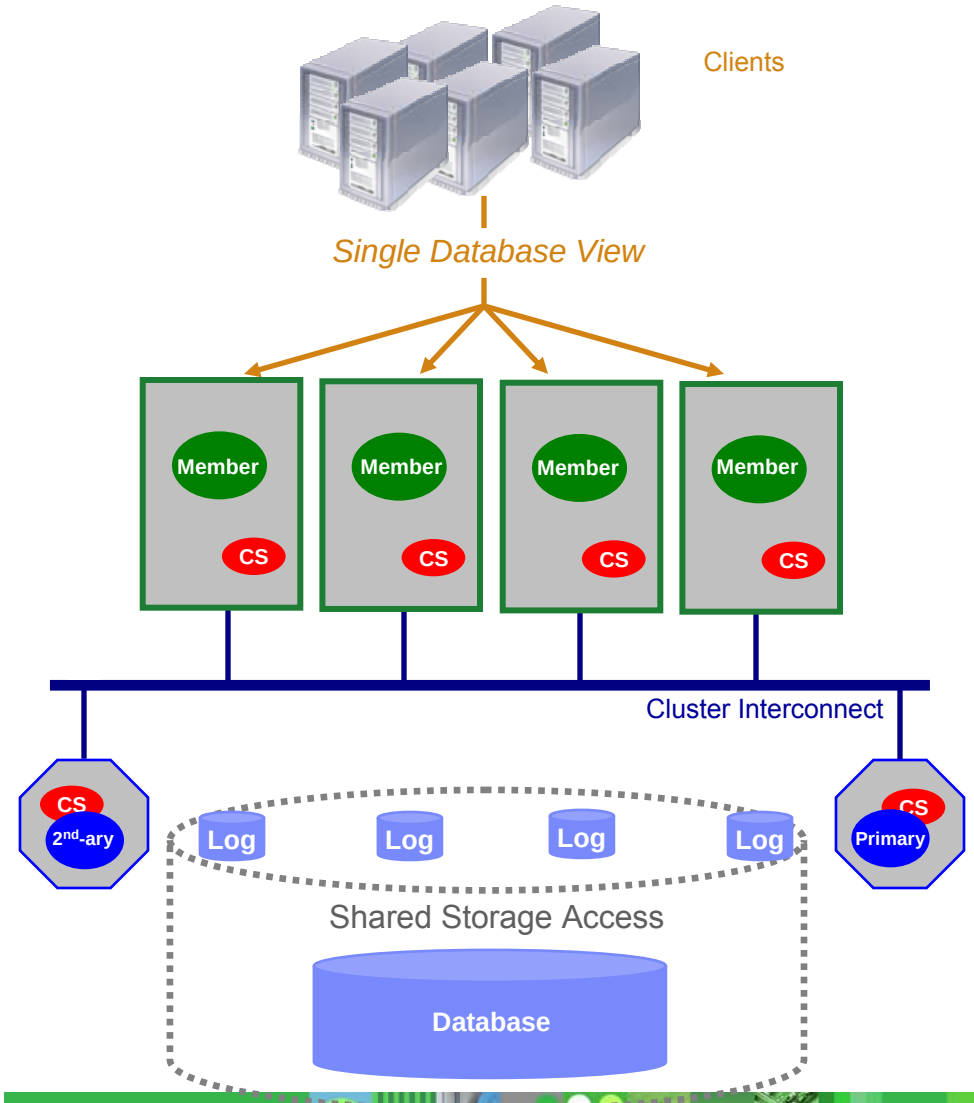
- **Looking at more challenging workload with more updates**
  - 1 update transaction for every 4 read transactions
  - Typical read/write ratio of many OLTP workloads
- **No cluster awareness in the application**
  - **No routing of transactions to members**
  - Demonstrate transparent application scaling
- **Redundant system**
  - 14 8-core p550s including duplexed CFs
- **Scalability remains above 90%**

Scalability for OLTP Applications



Technology Overview

Leverages 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)

Cluster caching facility (CF) from STG

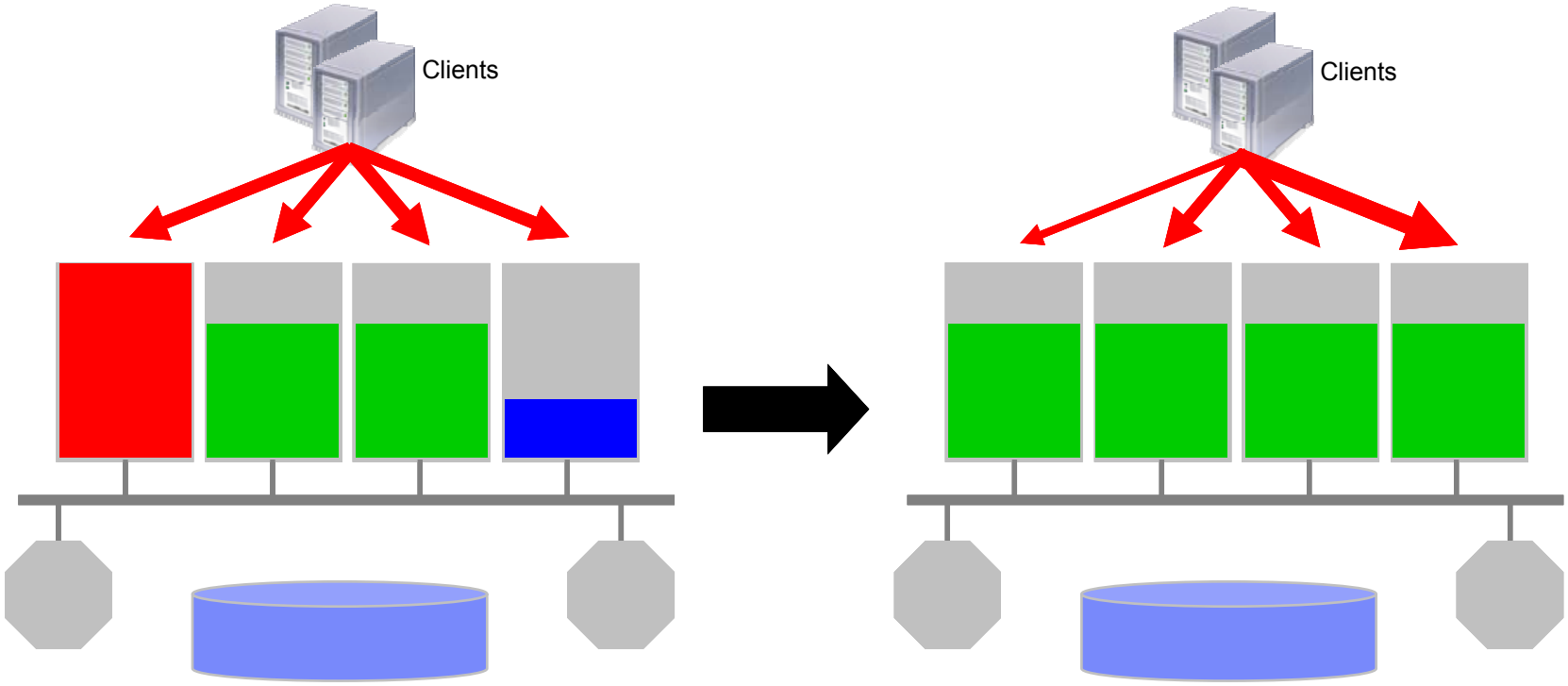
- Efficient global locking and buffer management
- Synchronous duplexing to secondary ensures availability

Data sharing architecture

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

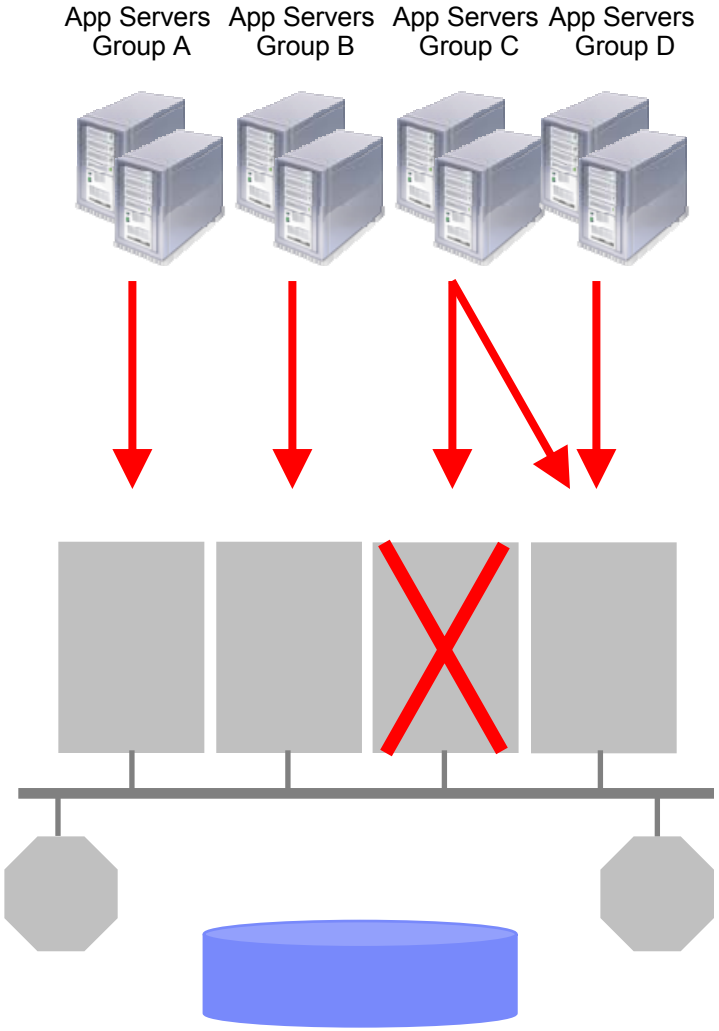
Workload Balancing

- **Run-time load information used to automatically balance load across the cluster**
  - Shares design with 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

- **Target different workloads to different members**
  - 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



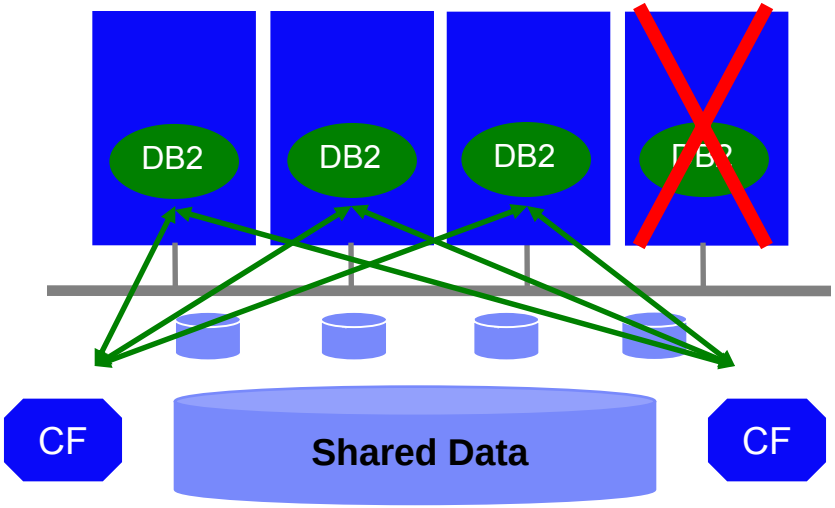
Information Management

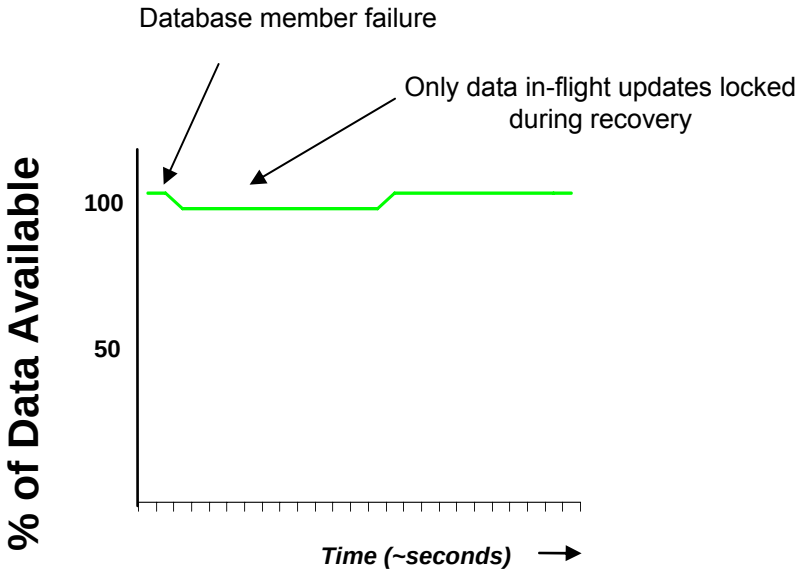
IBM

High Availability

Online Recovery

- **DB2 pureScale design point is to maximize availability during failure recovery processing**
- **When a database member fails, only *in-flight* data remains locked until member recovery completes**
  - In-flight = data being updated on the failed member at the time it failed
- **Target time to row availability**
  - <20 seconds





Steps Involved in DB2 pureScale Member Failure

1. **Failure Detection**

2. **Recovery process pulls directly from CF:**
   - Pages that need to be fixed
   - Location of Log File to start recovery from

3. **Restart Light Instance performs redo and undo recovery**

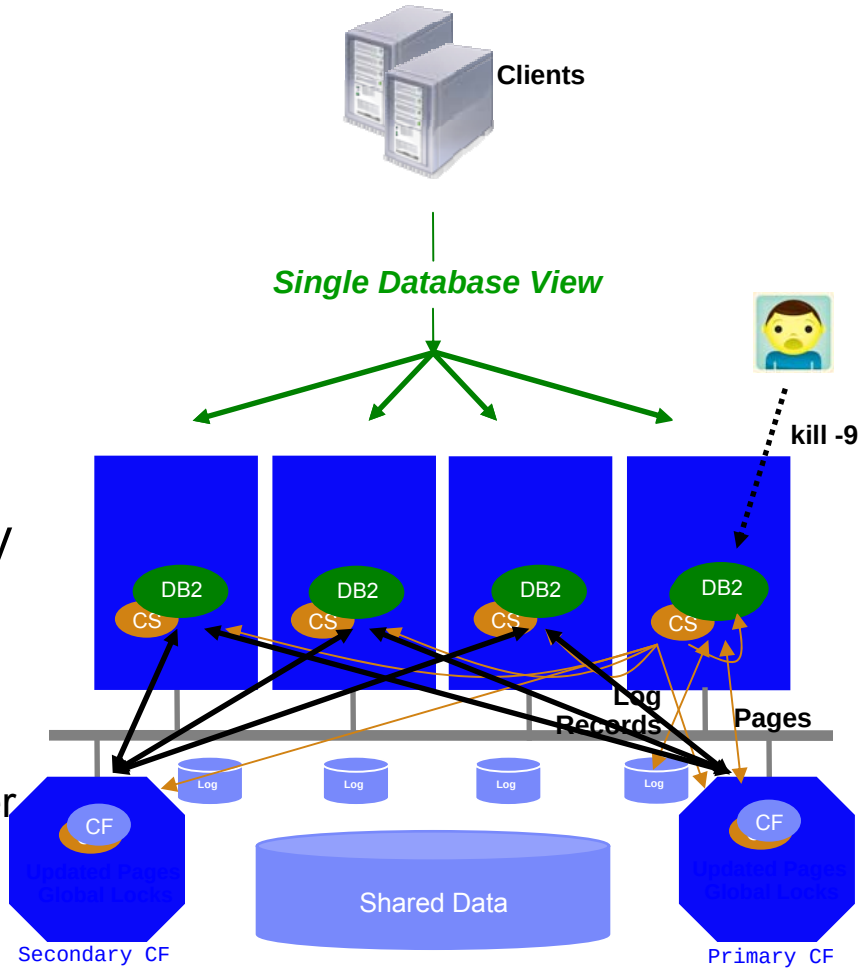
Failure Detection for Failed Member

- **DB2 has a watchdog process to monitor itself for software failure**
  - The watchdog is signaled any time the DB2 member is dying
  - This watchdog will interrupt the cluster manager to tell it to start recovery
  - Software failure detection times are **a fraction of a second**
- **The DB2 cluster manager performs very low level, sub second heart beating (with negligible impact on resource utilization)**
  - DB2 cluster manager performs other checks to determine congestion or failure
  - Result is hardware failure detection in under 3 seconds without false failovers

Member Failure Summary

Run in Slide Show

- Member Failure
- DB2 Cluster Services automatically detects member's death
  - Inform other members, and CFs
  - Initiates automated member restart on same or remote host
  - Member restart is like crash recovery in a single system, but is much faster
    - Redo limited to in-flight transactions
    - Benefits from page cache in CF
- Client transparently re-routed to healthy members
- Other members fully available at all times – *"Online Failover"*
  - CF holds update locks held by failed member
  - Other members can continue to read and update data not locked for update by failed member
- Member restart completes
  - Locks released and all data fully available



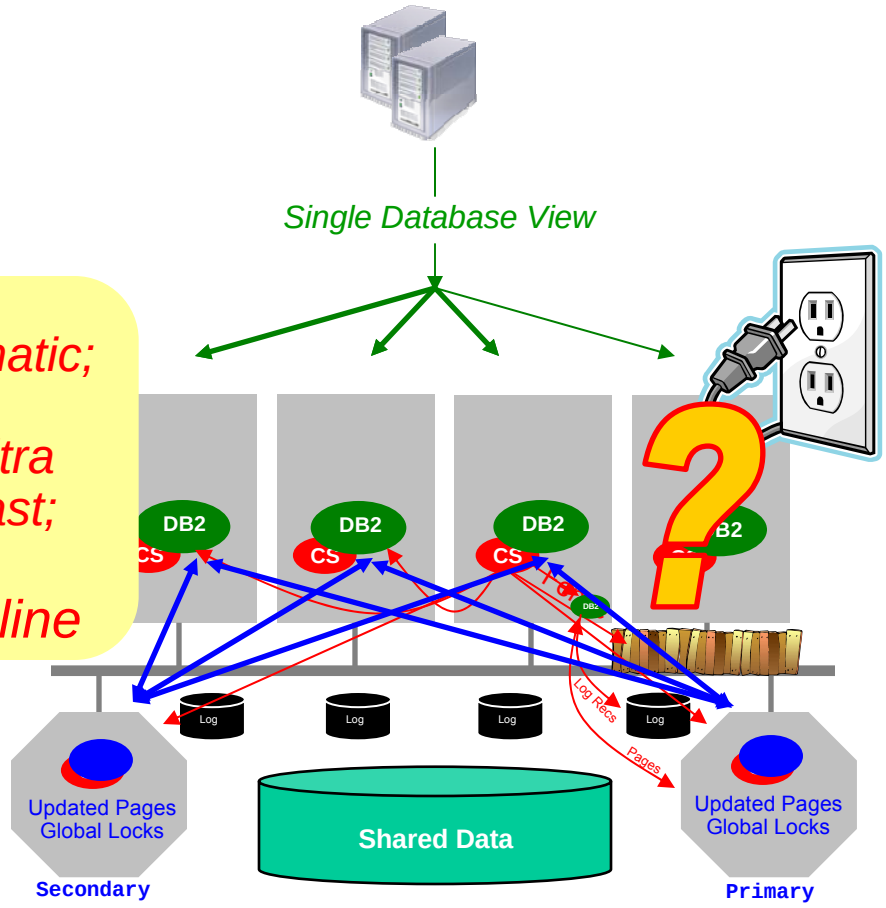
Member Hardware Failure

- **Power cord tripped over accidentally**
- **DB2 Cluster Services looses heartbeat and declares member down**
  - Informs other members & *CF* 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 da… much faster
    - Redo limited to inflight tr… (due to FAC)
    - Benefits from page cach…
- **In the mean-time, client conn… automatically re-routed to he…**
  - Based on least load (by defau…
  - Pre-designated failover meml…
- **Other members remain fully a… 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;

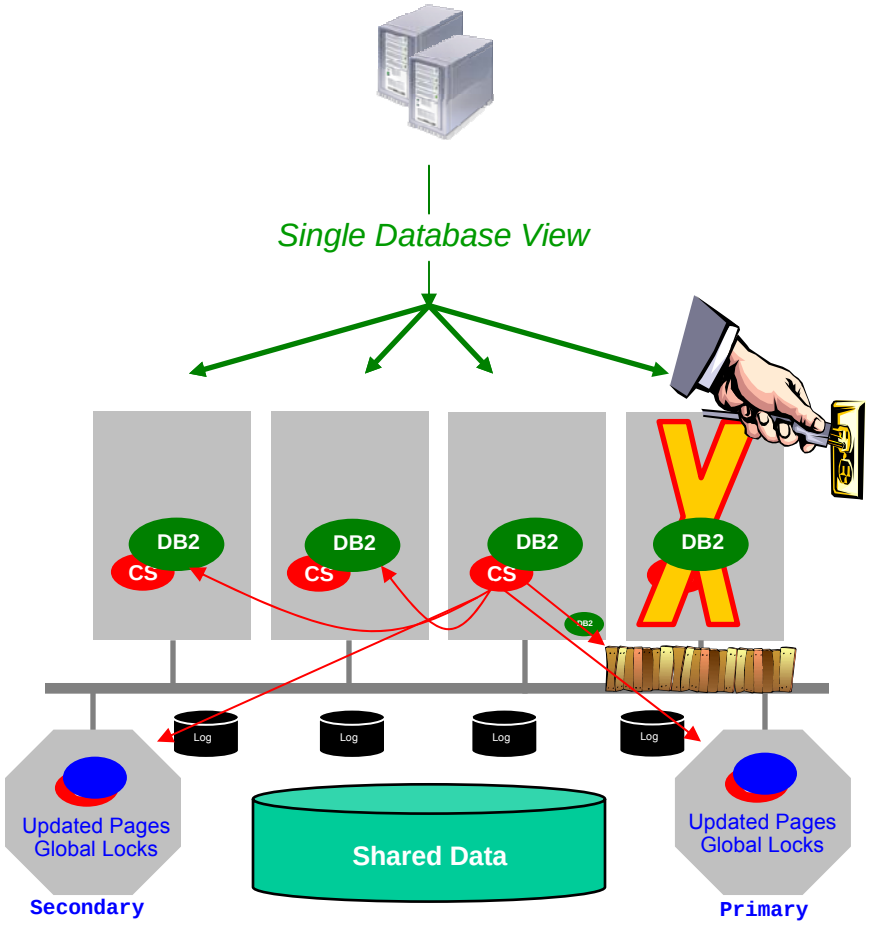
Online



Almost all data remains available. Affected connections transparently re-routed to other members.

Member Failback

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



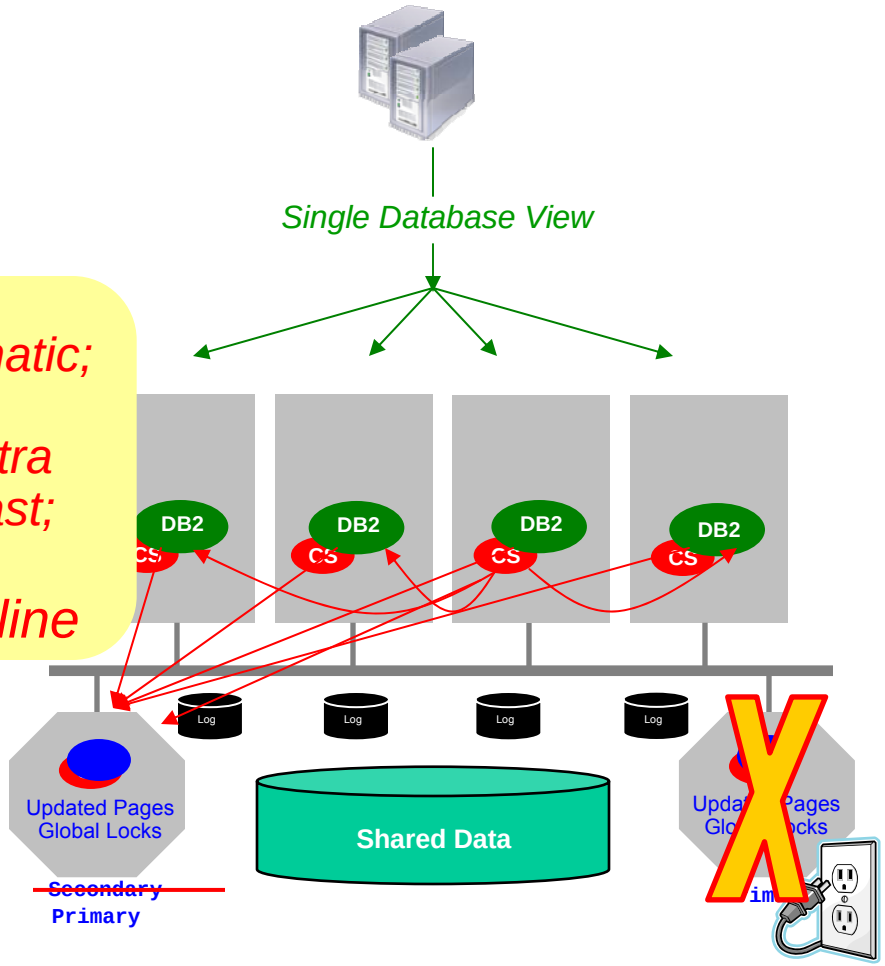
Primary CF HW Failure

- **Power cord tripped over accidentally**
- **DB2 Cluster Services looses heartbeat and declares primary down**
  - Informs members and secondary
  - *CCF* service momentarily
  - All other database activi proceeds normally
  - Eg. accessing pages in b existing locks, sorting, aggregation, etc
- **Members send missing d secondary**
  - Eg. read locks
- **Secondary becomes primary**
  - *CCF* service continues where it left off
  - No errors are returned to DB2 members

Automatic;

Ultra Fast;

Online



All data remains available. Completely transparent to members and transactions.

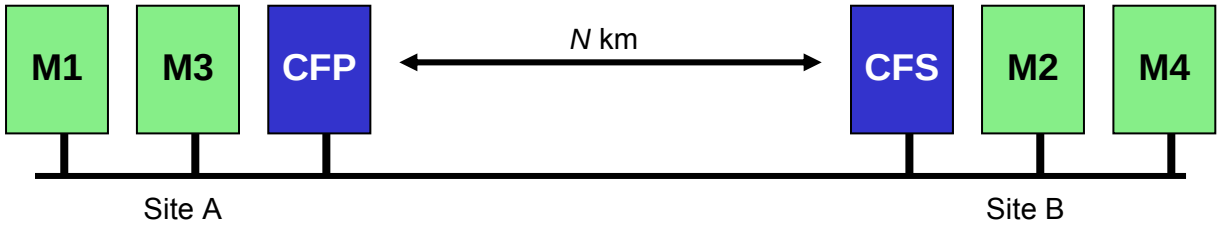
Information Management

IBM

DB2 pureScale Stretch Cluster

What is a Stretch Cluster?

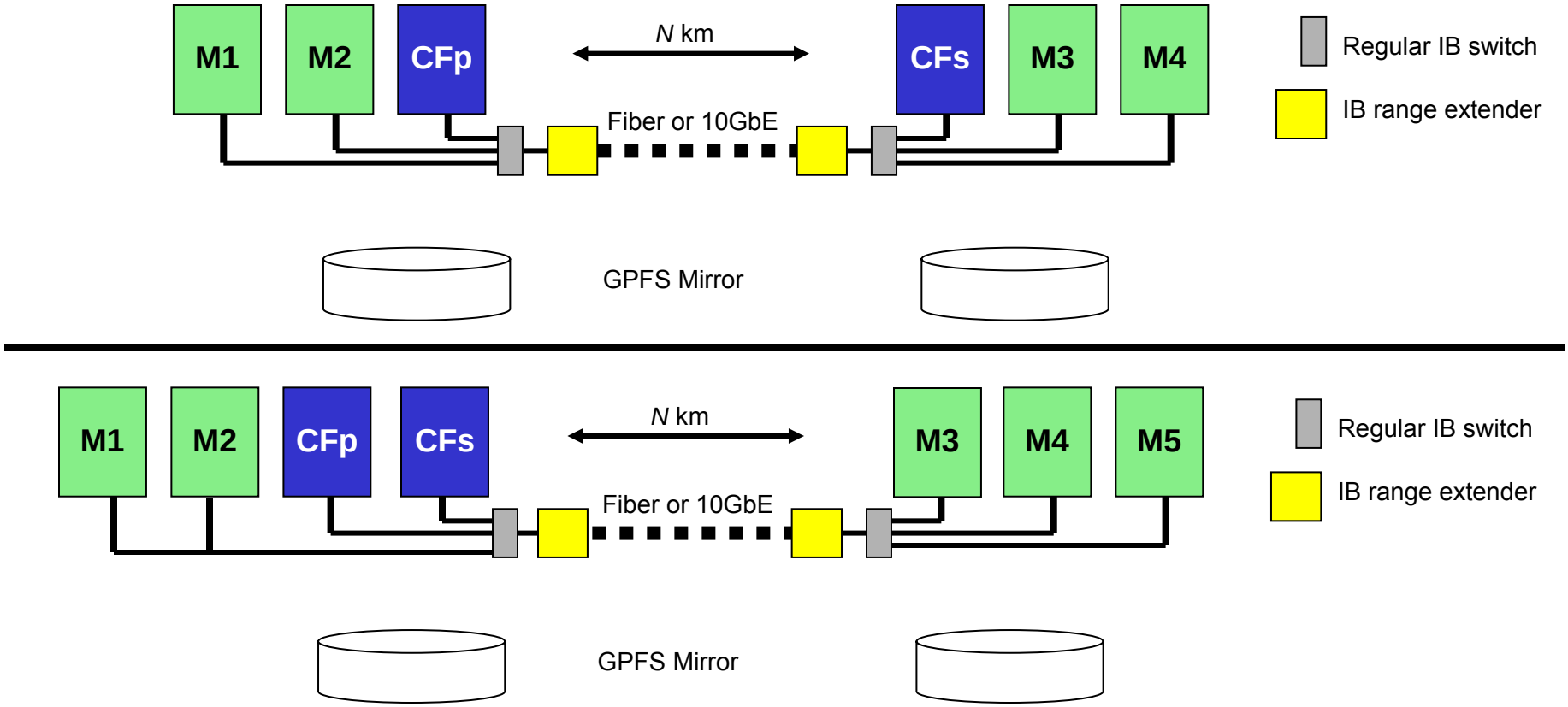
- **A 'stretch' or geographically dispersed cluster spans two sites at distances of tens of km**
  - Goal
    - Provide active / active access to a common database or databases across the cluster
  - Enables a level of DR support suitable for many types of disaster



- **For example**
  - DB2/z Geographically Dispersed Parallel Sysplex (GDPS)
  - http://www-03.ibm.com/systems/z/advantages/gdps/index.html

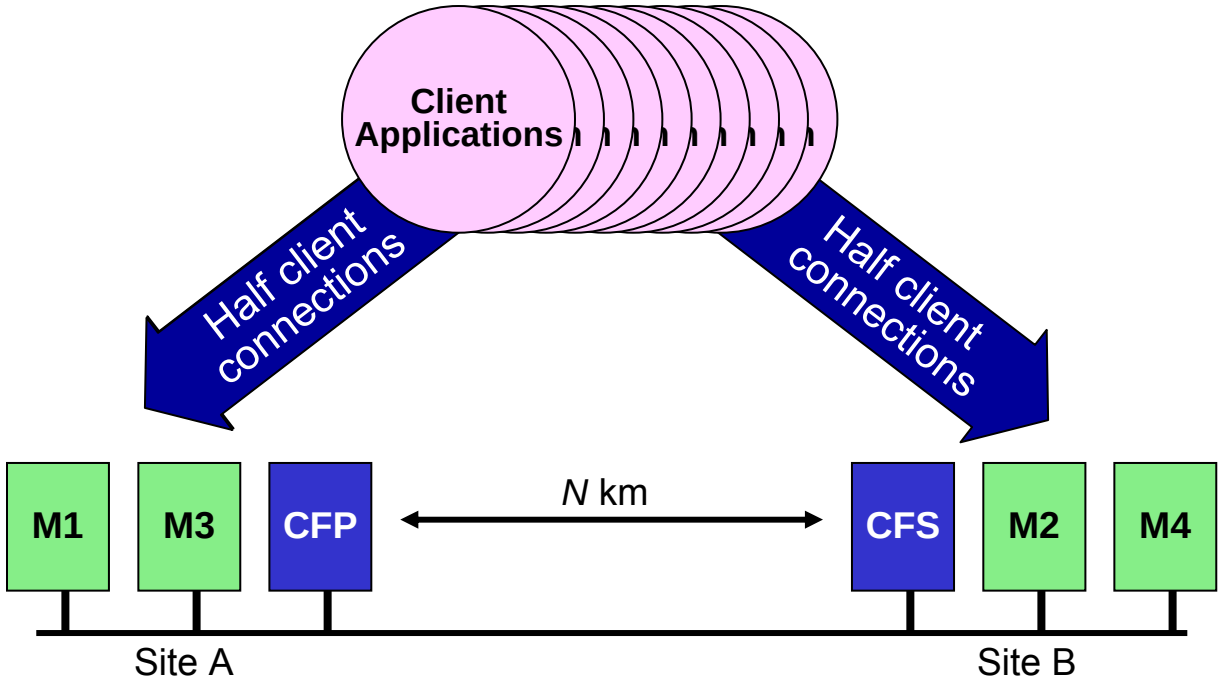
Active/Active DR via “Stretch Cluster"

- **Approach**
  - 1 or more members on both sites - all active
  - One CF on both sites – or both CFs on one side (DR site reporting)
  - GPFS mirroring for storage redundancy



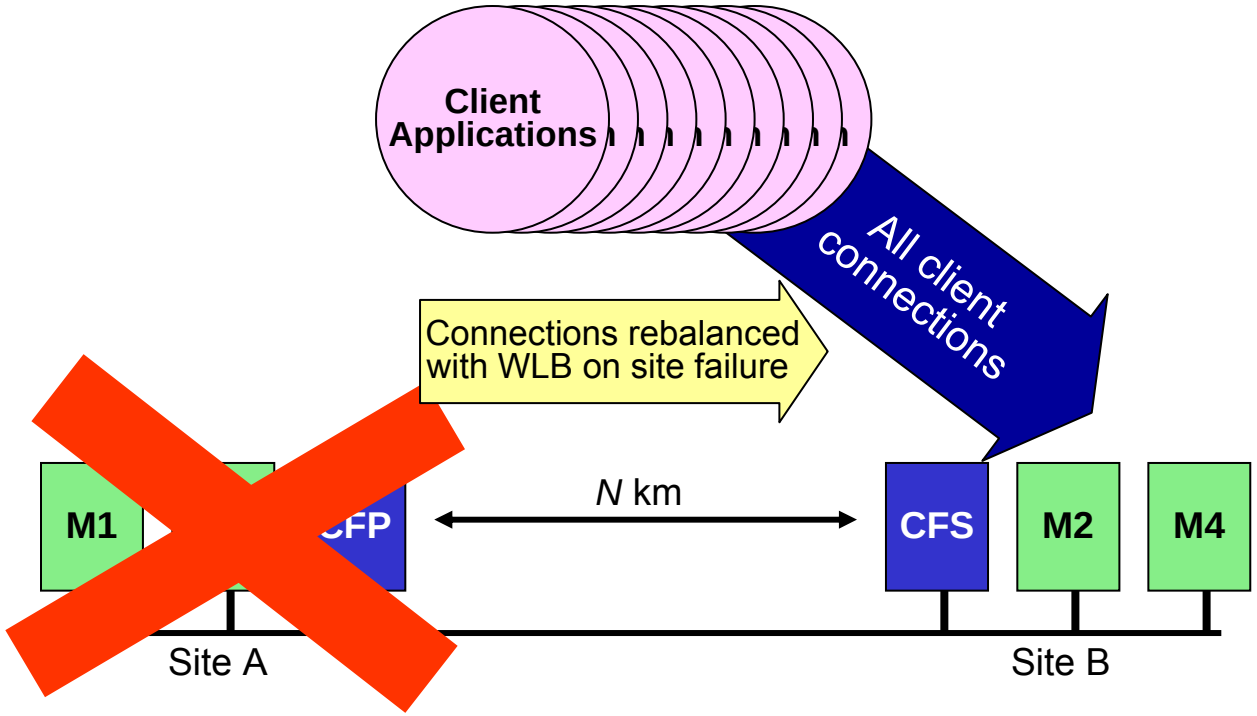
Target scenario

- **Both sites active & available for transactions during normal operation**
- **On failures, client connections are automatically redirected to surviving members**
  - Applies to both individual members within sites, and total site failure



Target scenario

- **Both sites active & available for transactions during normal operation**
- **On failures, client connections are automatically redirected to surviving members**
  - Applies to both individual members within sites, and total site failure



Information Management

IBM

Competitive Comparison for Availability

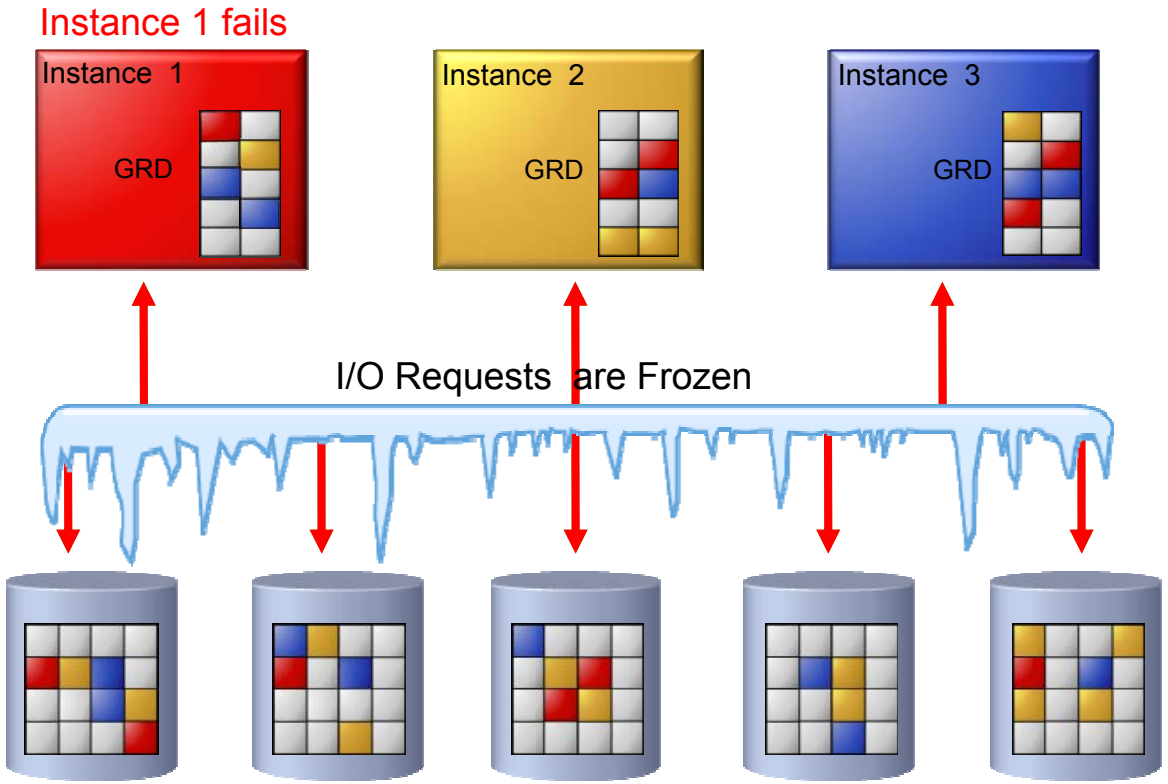
Steps involved in a RAC node failure

1. **Node failure detection**
2. **Data block remastering**
3. **Locking of pages that need recovery**
4. **Redo and undo recovery**

Unlike DB2 pureScale, Oracle RAC does not centralize lock or data cache

With RAC – Access to GRD and Disks are Frozen

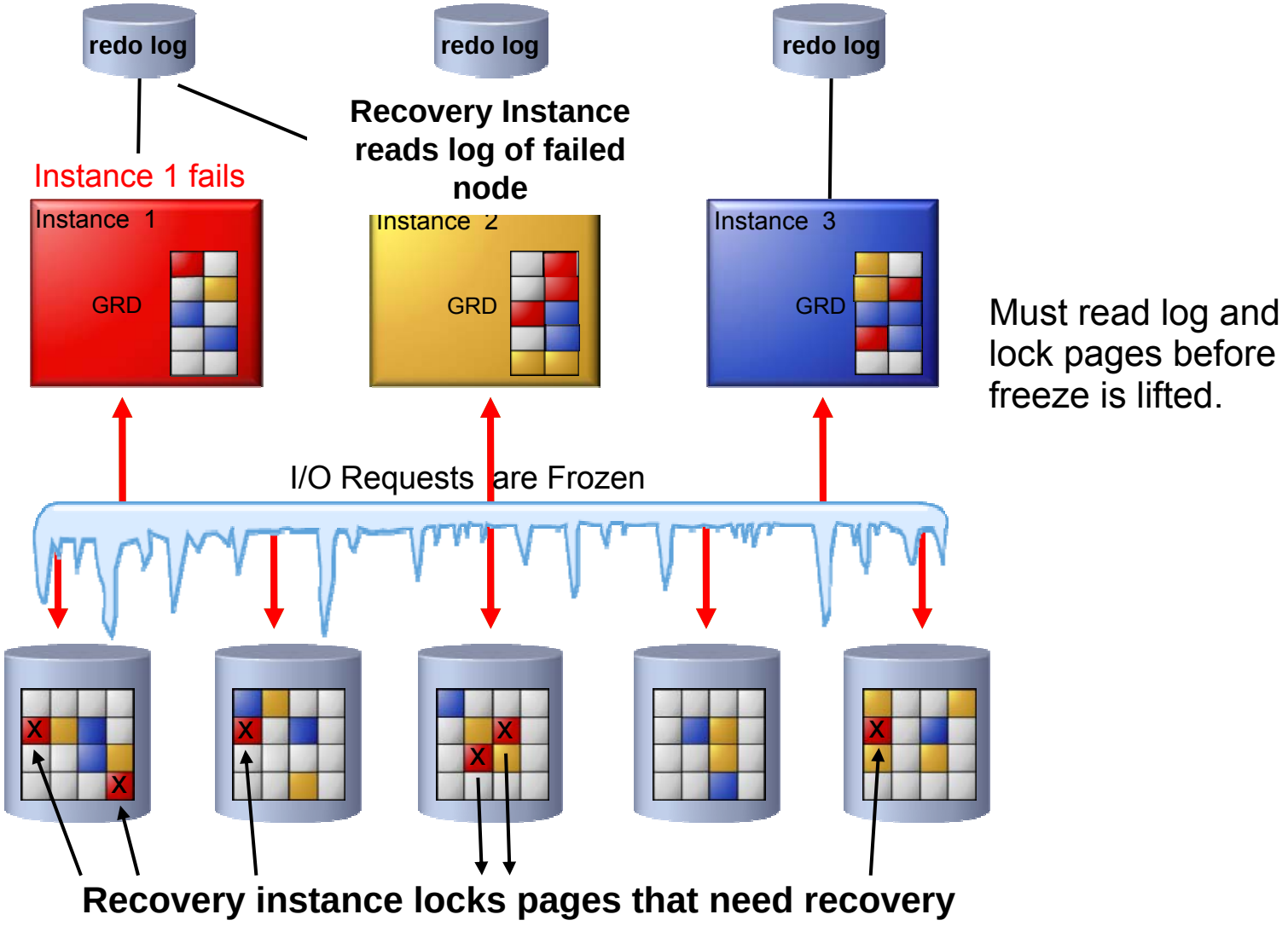
- **Global Resource Directory (GRD) Redistribution**



No more I/O until pages that need recovery are locked

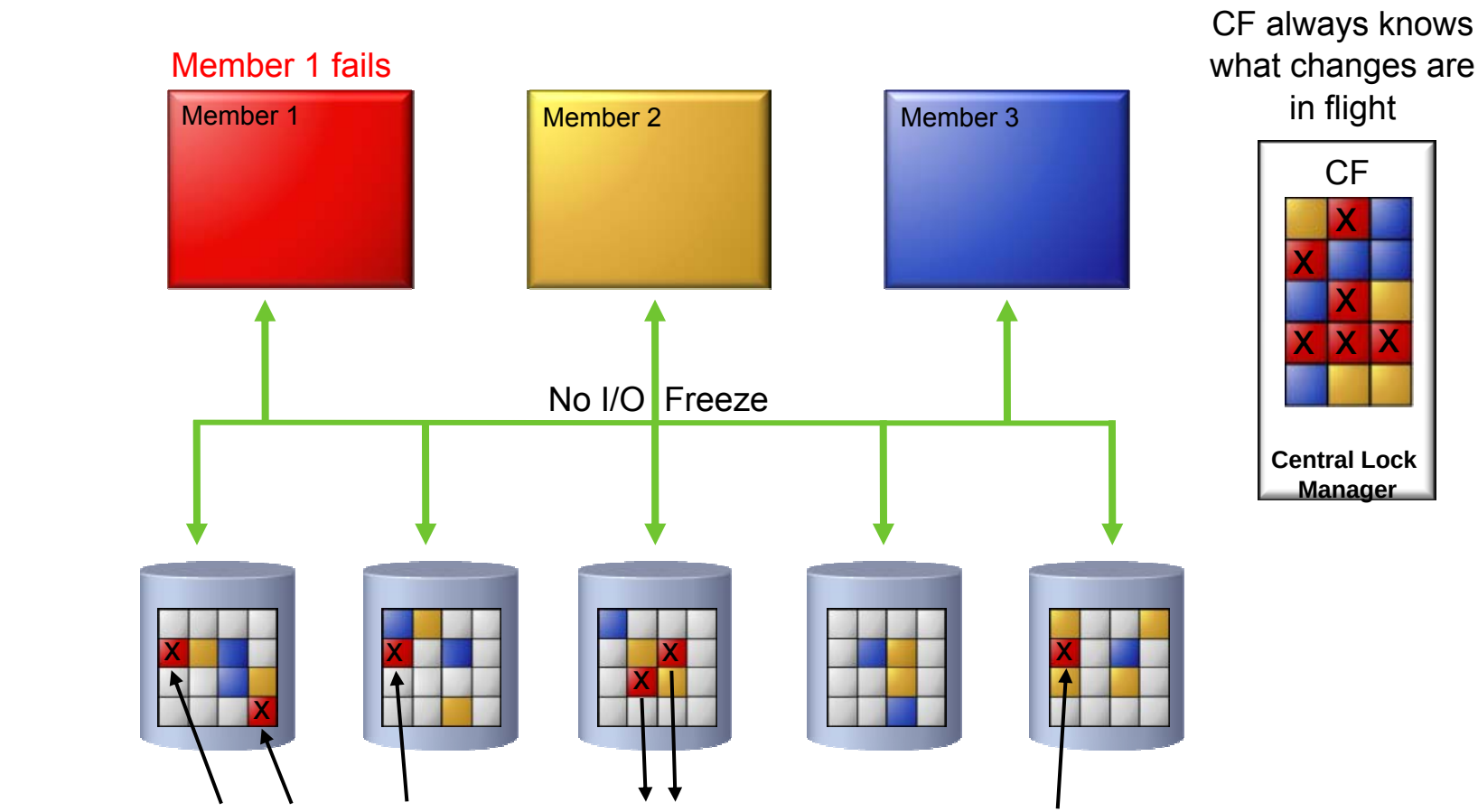
No Lock Updates

With RAC – Pages that Need Recovery are Locked



Must read log and lock pages before freeze is lifted.

DB2 pureScale – No Freeze at All



Information Management

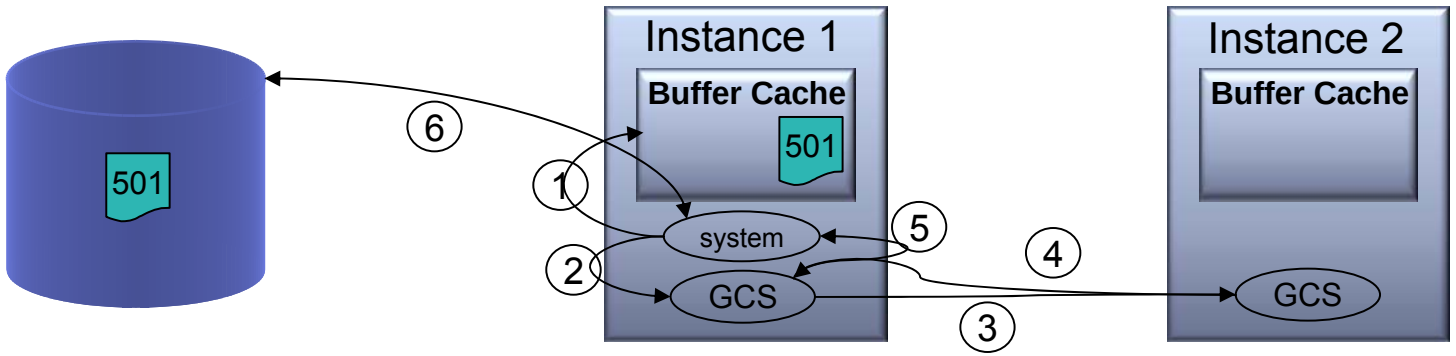
IBM

Competitive Comparison for Scalability

Oracle RAC - Single Instance Wants to Read a Page

Process on Instance 1 wants to read page 501 mastered by instance 2

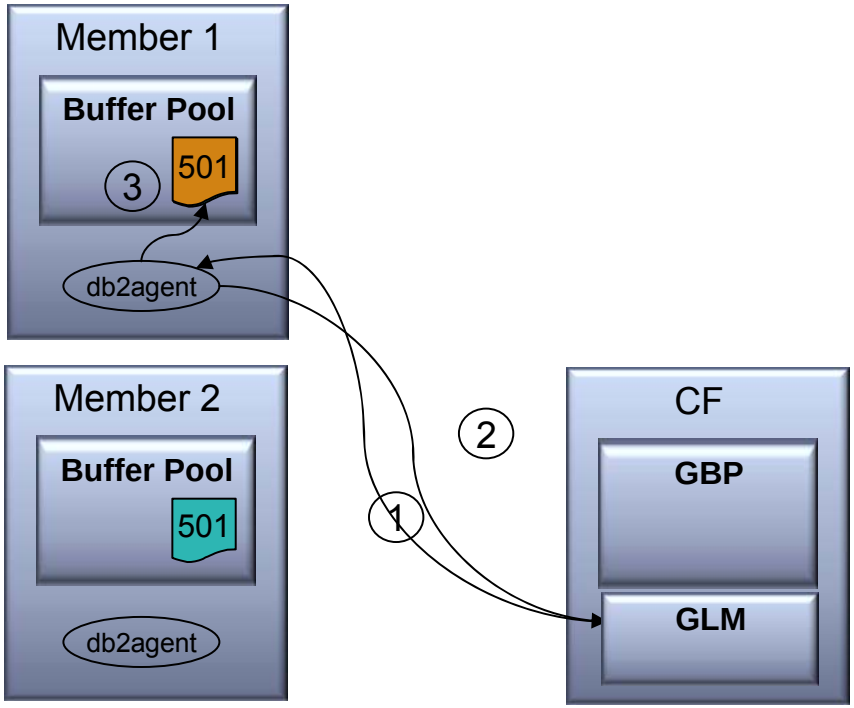
1. System process checks local buffer pool: page not found
2. System process sends an IPC to the Global Cache Service process to get page 501
   - Context Switch to schedule GCS on a CPU
   - GCS copies request to kernel memory to make TCP/IP stack call
3. GCS sends request over to Instance 2
   - IP receive call requires interrupt processing on remote node
4. Remote node responds back via IP interrupt to GCS on Instance 1
5. GCS sends IPC to System process (another context switch to process request)
6. System process performs I/O to get the page



DB2 pureScale - Member 1 Updates a Row

1. **Agent makes a Set Lock State (SLS) RDMA call to CF for X-lock on the row and P-lock to indicate the page will be updated**
   - Prevents other members from making byte changes to the page at the exact same time as this member
   - SLS call takes as little as 15 microseconds end to end
2. **CF responds via RDMA with grant of lock request**
3. **Page updated**

- **At this point Member 1 does not need to do anything else**
  - P-lock is only released in a lazy fashion
  - If another Member wants it, they can have it but otherwise Member 1 keeps it until commit

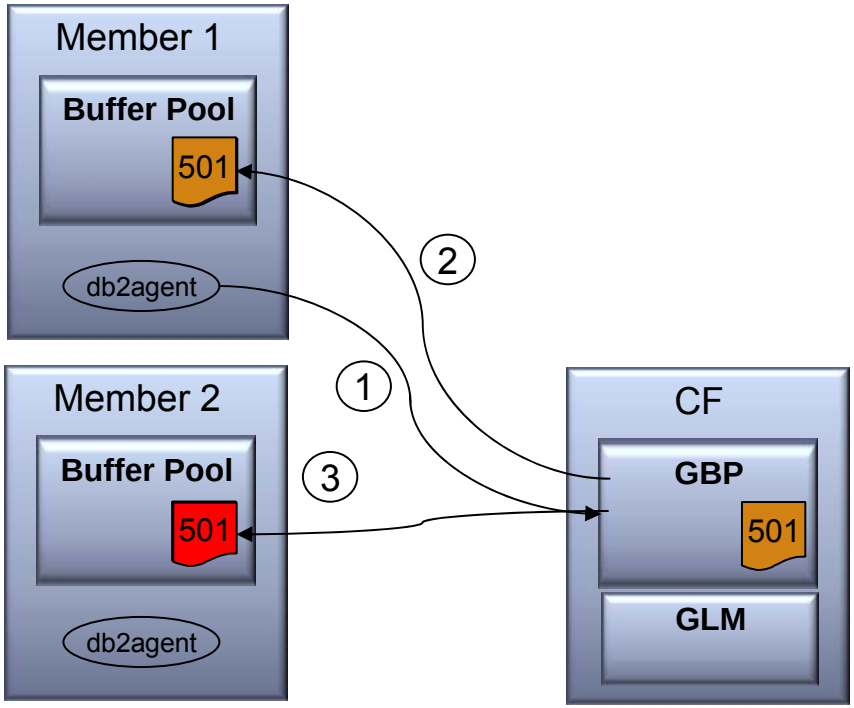


DB2 pureScale - Member 1 Commits Their Update

1. **Agent makes a Write And Register Multiple (WARM) RDMA call to the CF for the pages it has updated**

2. **CF will pull all the pages that have been updated directly from the memory address of Member 1 into its global buffer pool**
   - P-Locks released if not already released (as are X-locks for the row)

3. **CF invalidates the page in all other members that have read this page by directly updating a bit in the other members buffer pool directory**
   - Before a member accesses this changed page again it must get the current copy from the Global Buffer Pool



DB2 pureScale - Two Members Update Same Page

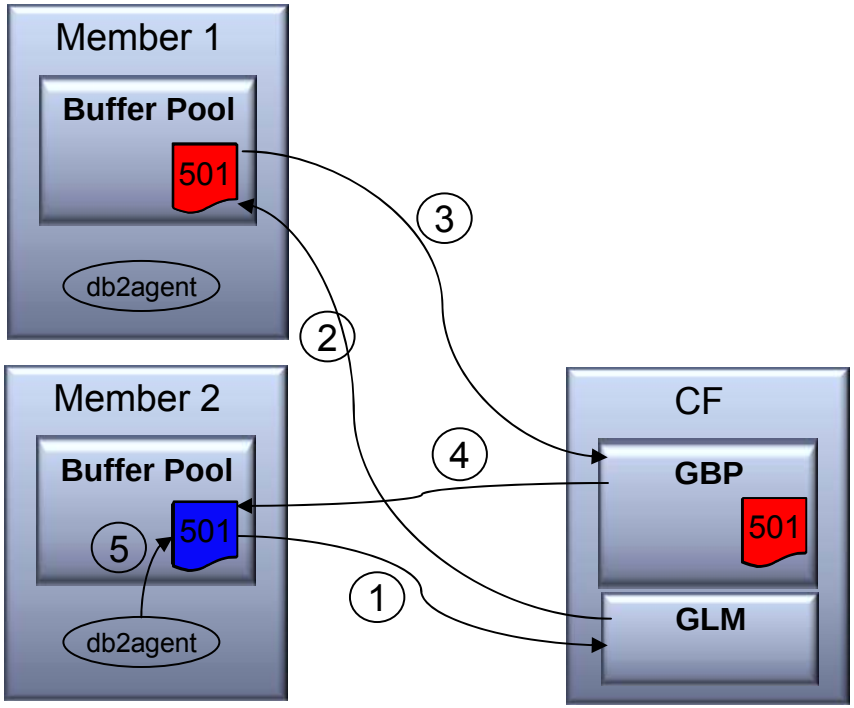
1. **Agent on Member 2 makes a Set Lock State (SLS) RDMA call to CF for X-lock on the row and P-lock to indicate the page will be updated**
   - The P-Lock contends with the lock held my Member 1

- **GLM tells Member 1 to release its P-lock**

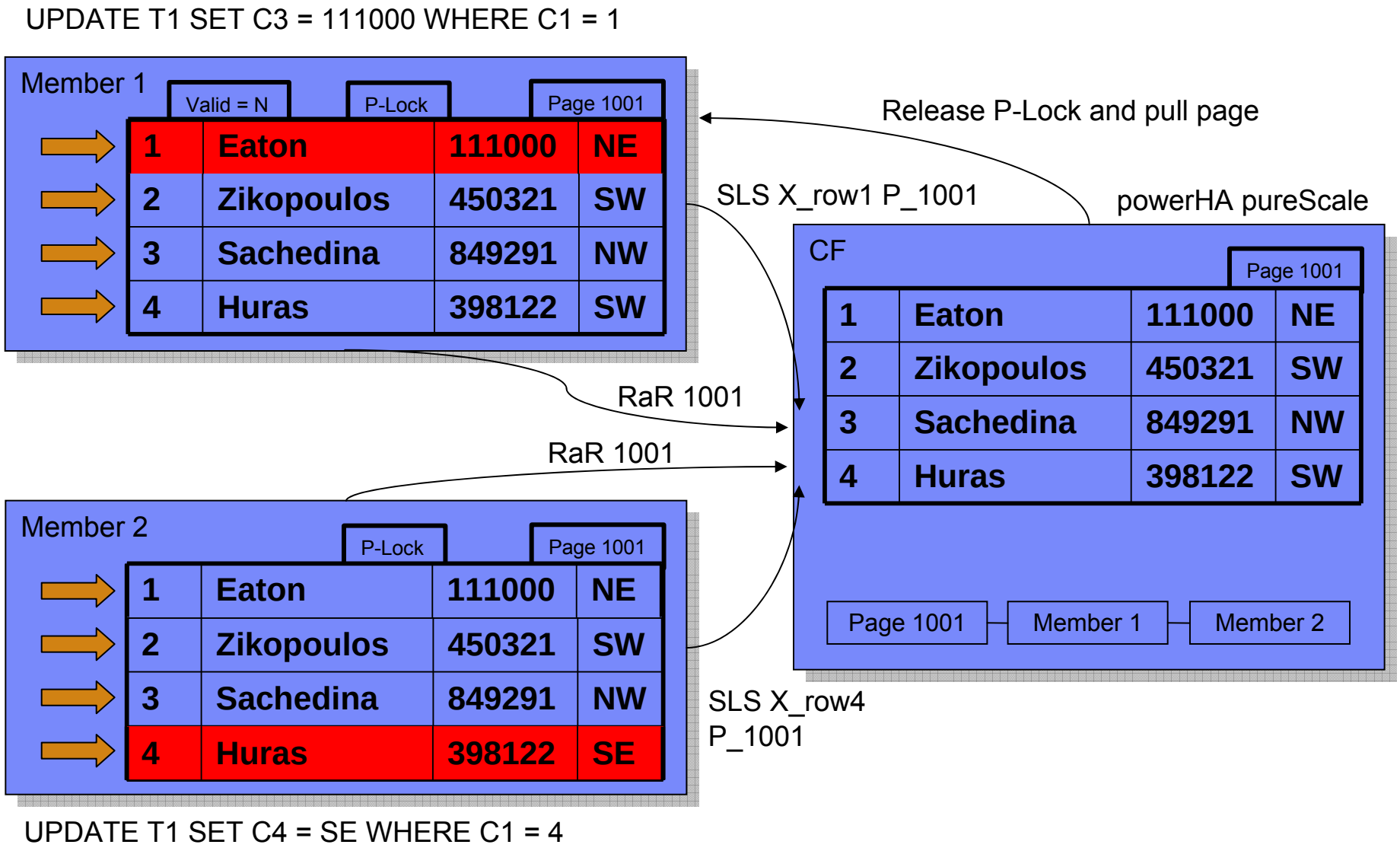
- **Member 1 completes the update and has the page pulled from its local memory into the GBP via WARM request**
  - Note that the P-lock is not required for a transaction boundary
  - Only held for the duration of the time it takes to make the byte changes to the page

1. **CF responds via RDMA with grant of lock request and pushes the updated page into Member 2's memory and invalidates other member copies**

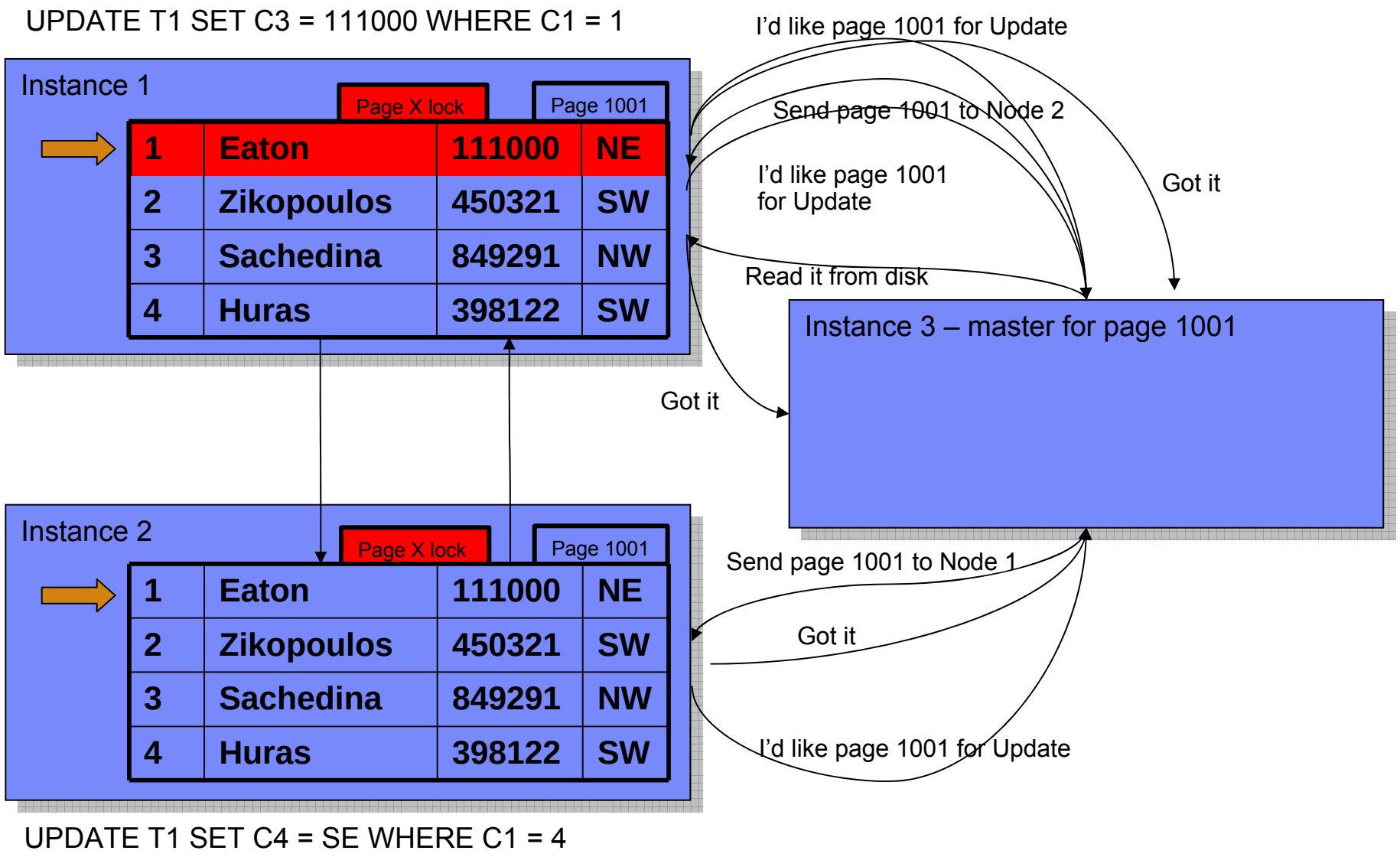
- **Page updated**



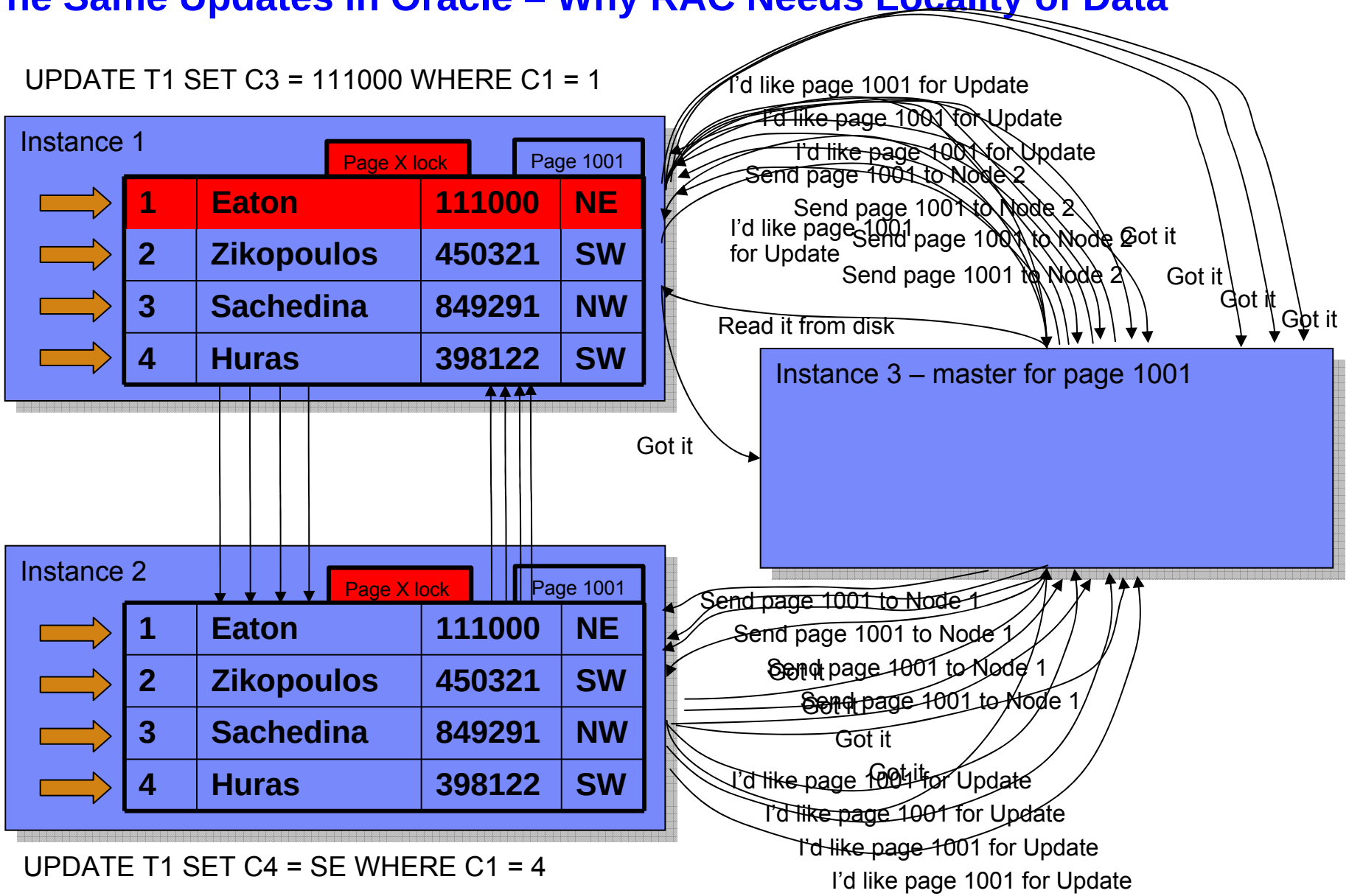
A Closer Look at 2 Members Updating the Same Page (Different Rows)



The Same Updates in Oracle – Why RAC Needs Locality of Data



The Same Updates in Oracle – Why RAC Needs Locality of Data



Scalability Differences

- **Oracle RAC must lock a page whenever there is the intent to update that page**
- **DB2 pureScale must lock a page whenever rows are actually being changed on that page**
- **DB2 pureScale improves concurrency between members in a cluster which results in better scalability and less of a need for locality of data**

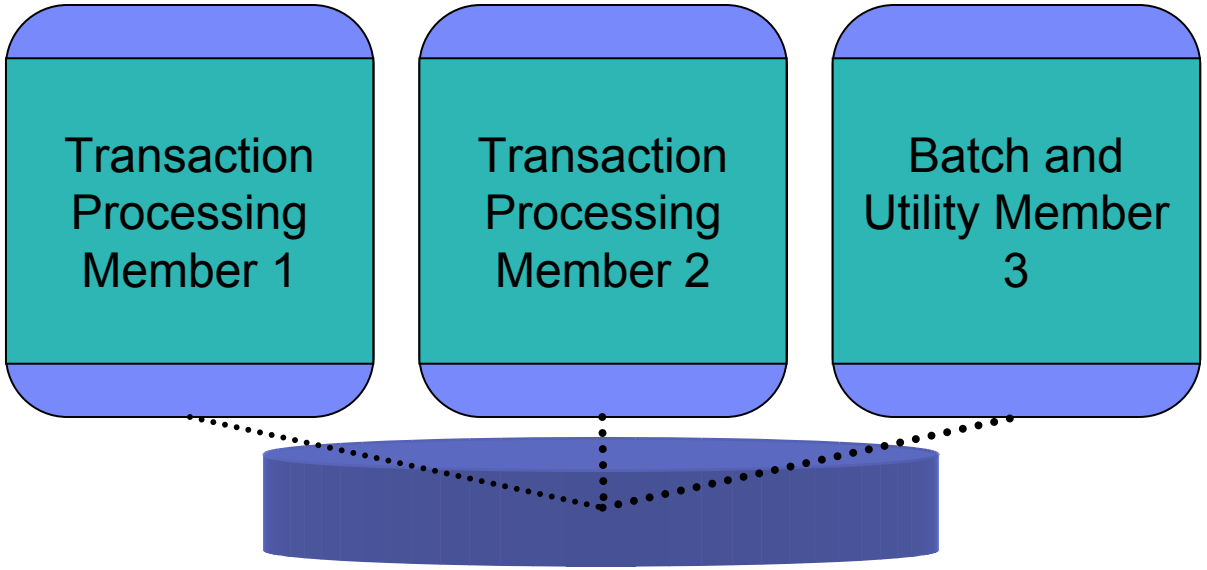
Information Management

IBM

DB2 pureScale for Mixed Workloads and Consolidation

DB2 Advantages for Mixed Workloads

- **Many OLTP workloads already have a large batch component**
- **DB2 already has strong customer proof points**
  - Large Chinese Financal with OLTP/Batch on a 40TB system
  - Large global ERP system with 24TB
- **Add in pureScale to allow Batch on one node and transactional on the other nodes in the cluster**

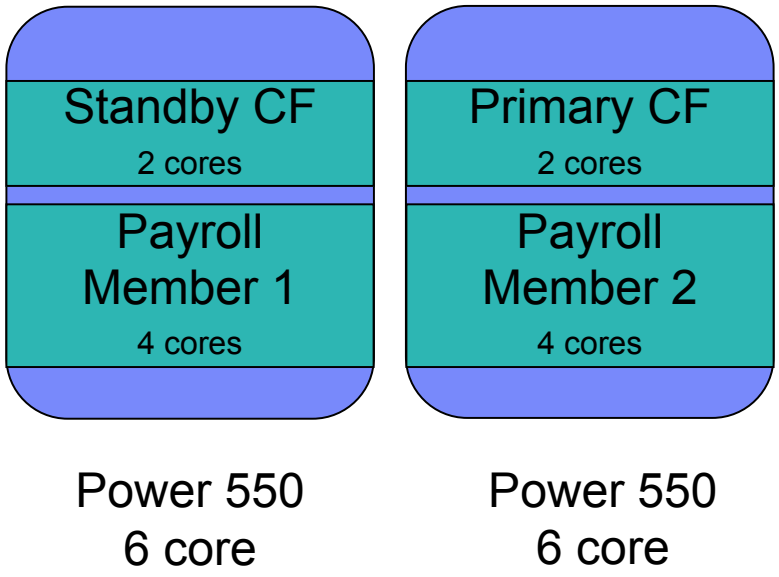


DB2 on Power Advantages for Consolidation

- **Consolidation needs:**
  - Effective workload management
    - DB2 WLM integrated with AIX WLM – delivers most effective resource utilization across the entire server
    - Oracle workload management does not share across instances and cannot be used with AIX WLM
  - Dynamic resource allocation
    - DB2 self tuning memory can immediately leverage additional resources or scale back if resources needed elsewhere
    - Oracle can only share resources within a single instance
  - Ultra high reliability
    - pureScale delivers higher availability than Oracle RAC
    - Power Systems 3x-4x more reliable than x86 servers
  - Simplified problem determination
    - Exadata has separate storage servers and database servers making problem determination even more difficult
  - Scalability (scale out and scale up) transparently to the application

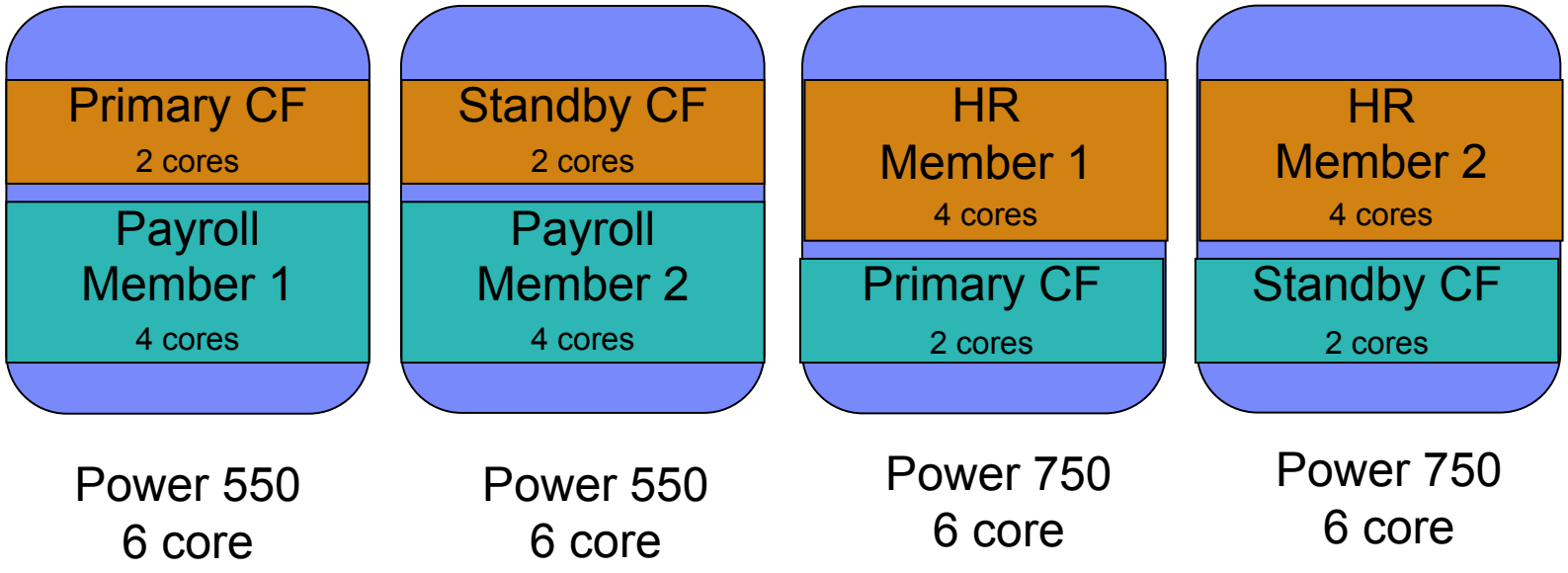
Consolidation for Cost Savings on DB2 pureScale

- **Start with the server sizes customized to fit your application**
  - Don't overbuy servers
  - Flexibility in server size and cores per server
  - Flexibility in amount of storage allocated



Consolidation for Cost Savings on DB2 pureScale

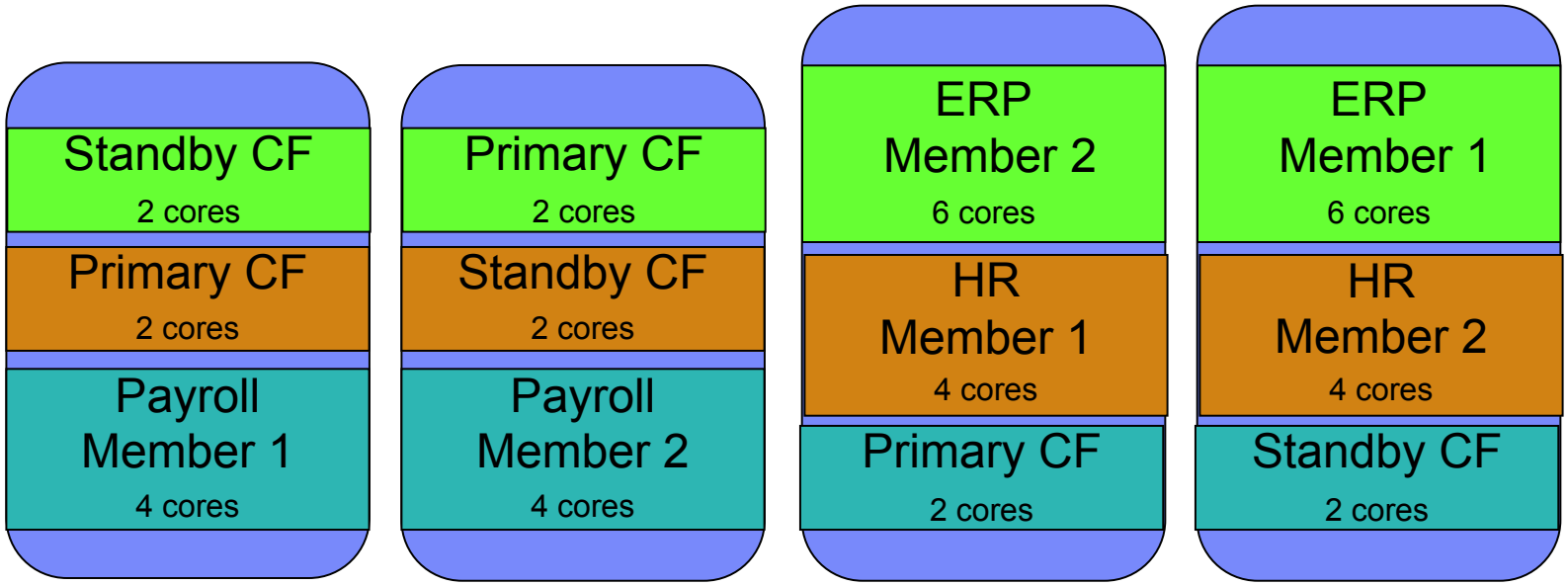
- **Grow with next generation servers**
  - Scale up on existing servers or out onto new servers
  - Mix and match servers to fit your application profile
  - Leverage resources across the cluster (utilize the strengths of PowerVM)



750 supports up to 16 cores

Consolidation for Cost Savings on DB2 pureScale

- **Consolidate onto the number of servers that make sense for your business**
  - Purchase the most cost effective servers – not one size fits all



750 supports up to 16 cores

Summary – What can DB2 pureScale Do For You?

- Deliver higher levels of scalability and superior availability
- Better concurrency during regular operations
- Better concurrency during member failure
- Result in less application design and rework for scalability
- Improved SLA attainment
- Better consolidation platform – sized to fit, not one size fits all
- Lower overall costs for applications that require high transactional performance and ultra high availability

Additional Resources

- **DB2 pureScale Information Center**
  - http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/index.jsp
- **DB2 pureScale web page**
  - http://www-01.ibm.com/software/data/db2/linux-unix-windows/editions-features-purescale.html
- **My blog entries on pureScale**
  - http://it.toolbox.com/blogs/db2luw/db2-purescale-scalability-part-1-35173