Information Management

IBM DB2 pureScale

Luis Reina (luis_reina@es.ibm.com)
IBM SWG Information Management

Data Management

¿Qué es DB2 pureScale?

Brings Cluster Technology to distributed platforms in partnership with Power Systems

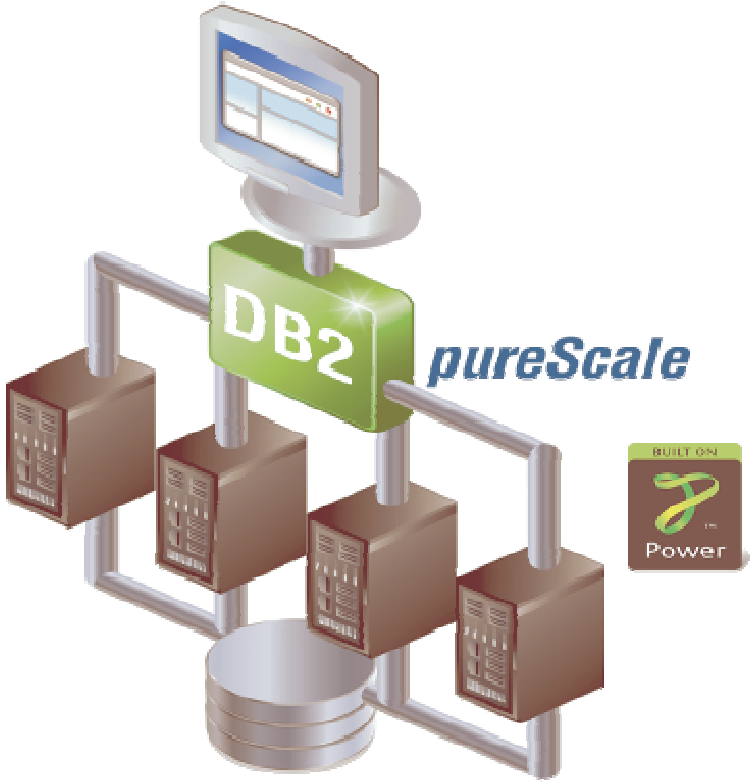
Easy for applications to run on pureScale

Multiple servers appear as one database

Seamlessly add or remove servers to meet changing business needs

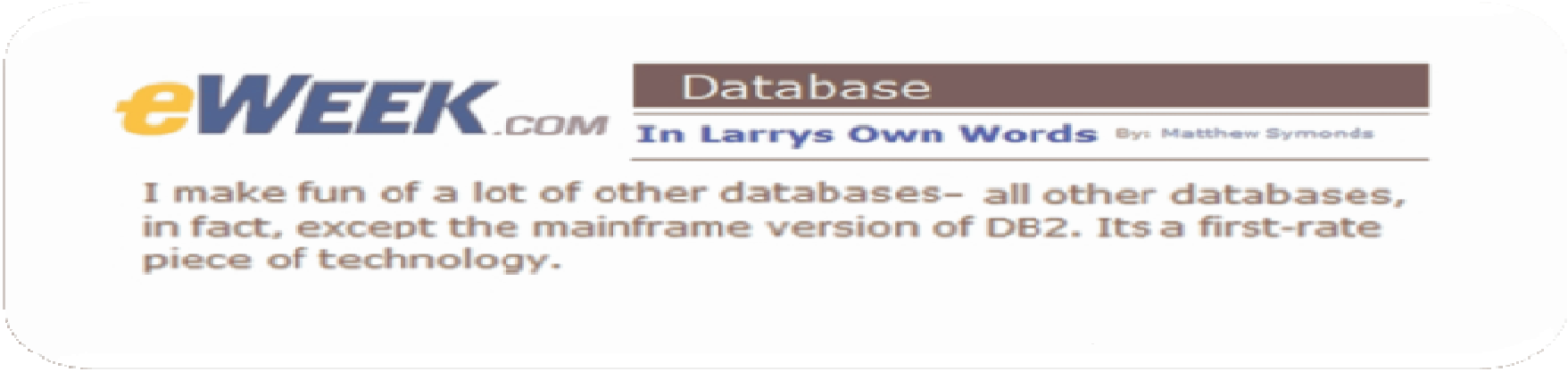
Delivers levels of Scalability and Availability not seen before on distributed platforms

Users never know when one or more members fail – DB2 Pure Scale remains running



DB2 for zOS Data Sharing

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

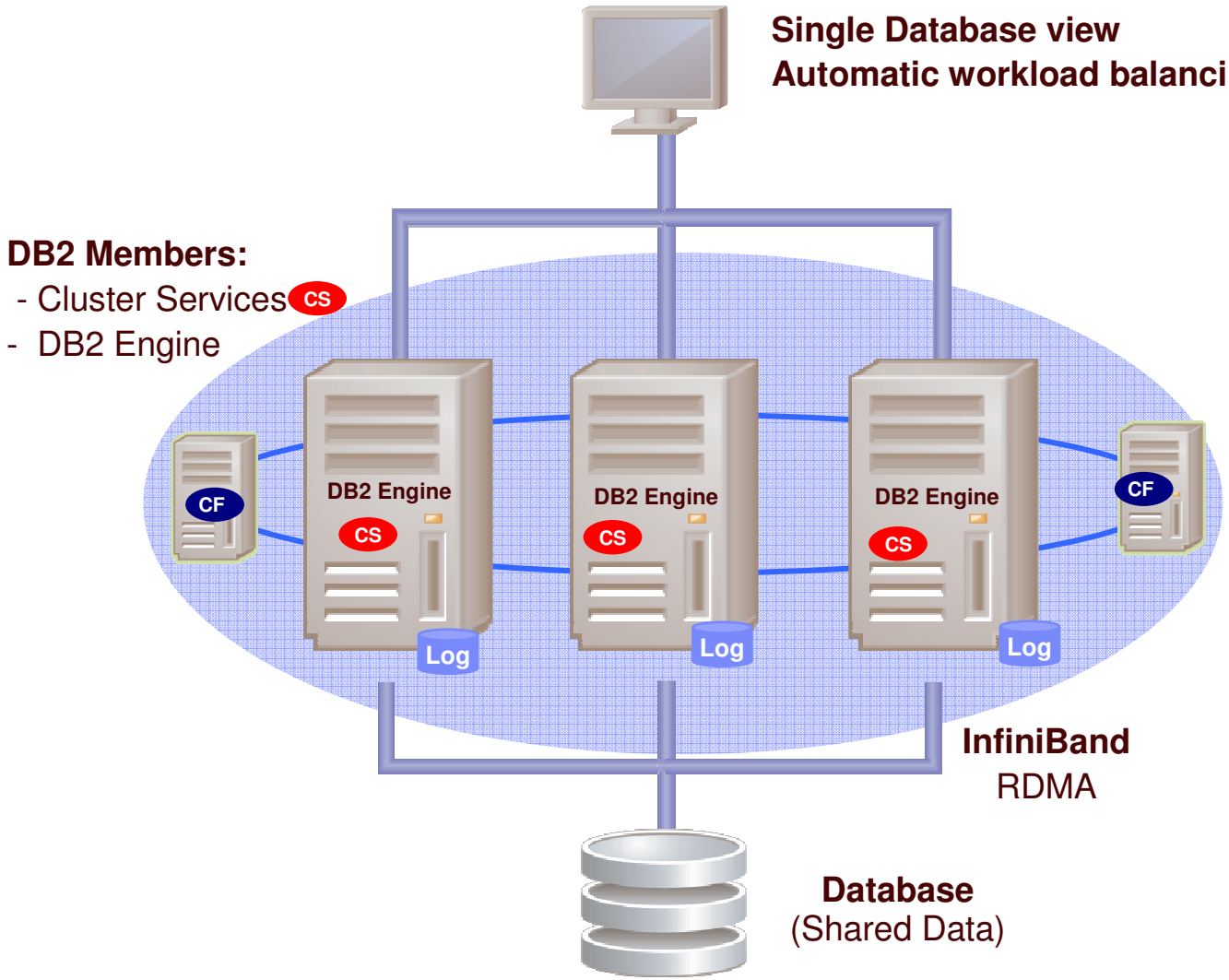


- **Why?**
  - The Coupling Facility!!
    - Centralized locking, centralized buffer pool deliver superior scalability and superior availability
  - The entire environment on z/OS uses the Coupling Facility
    - CICS, MQ, IMS, Workload Management, and more

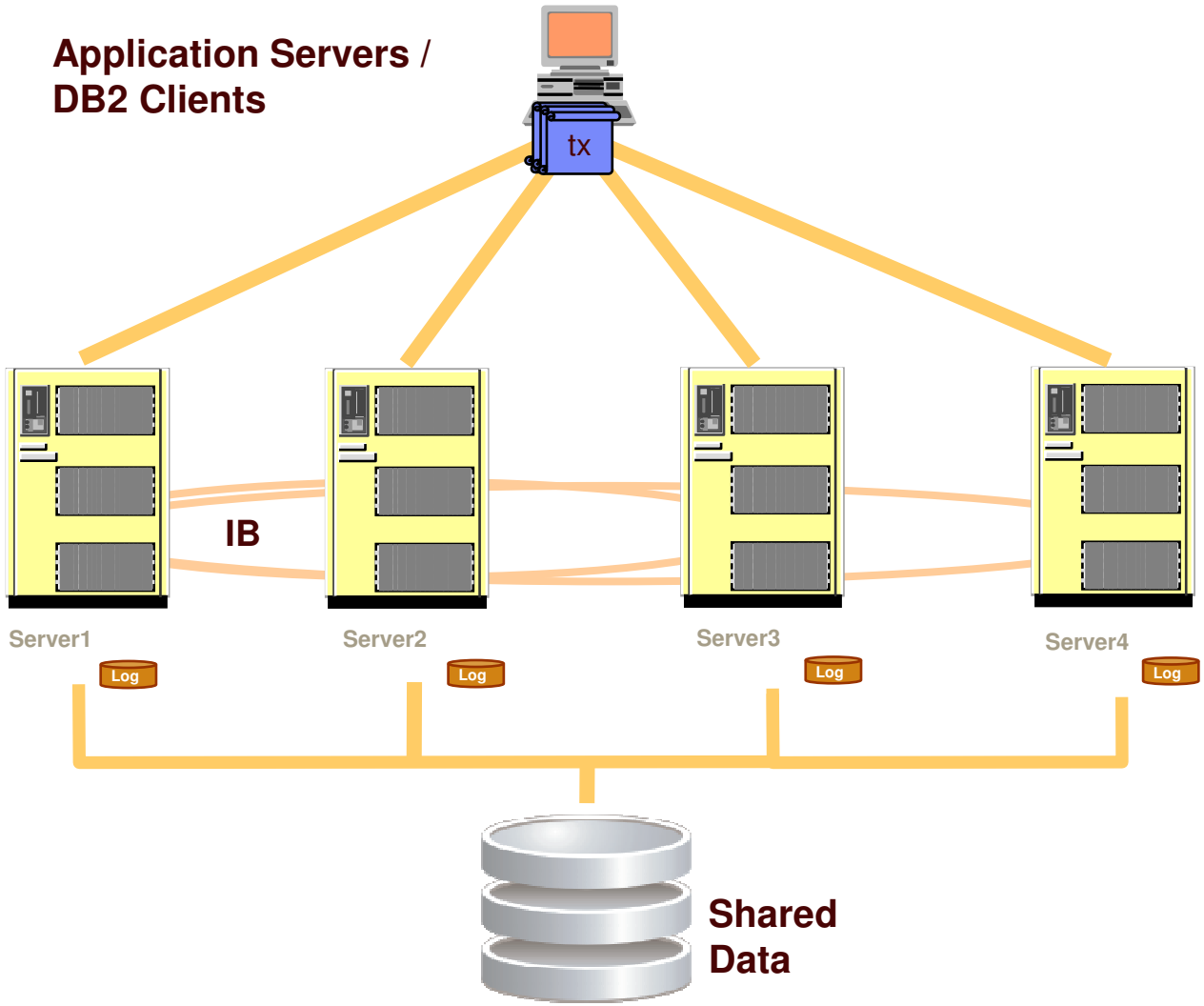
Objetivos de DB2 pureScale

- **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

Visión de la Arquitectura de DB2 pureScale

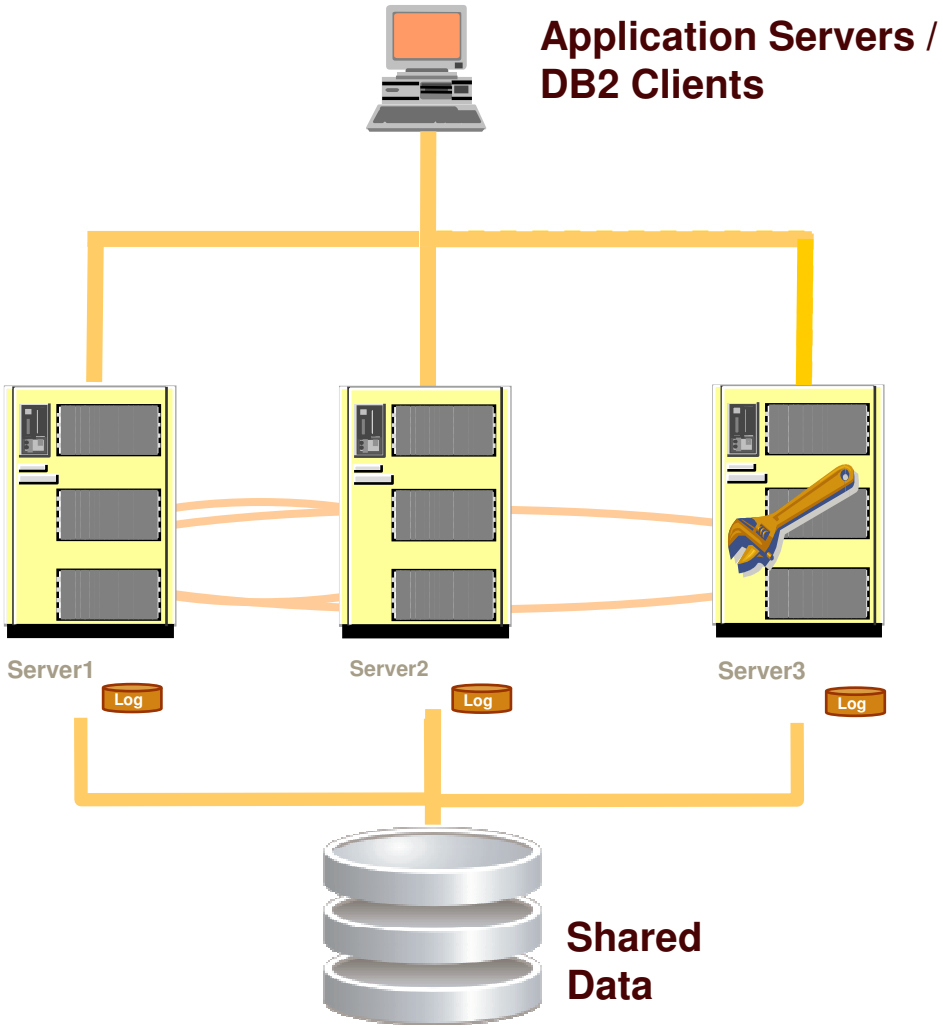


Crecimiento Sencillo



- Crecimiento Sencillo
- Uso instantáneo del nuevo nodo
- Balanceo de Carga
- Sin cambio en la aplicaciones
- Hasta 128 nodos en la versión inicial

Paradas Planificadas



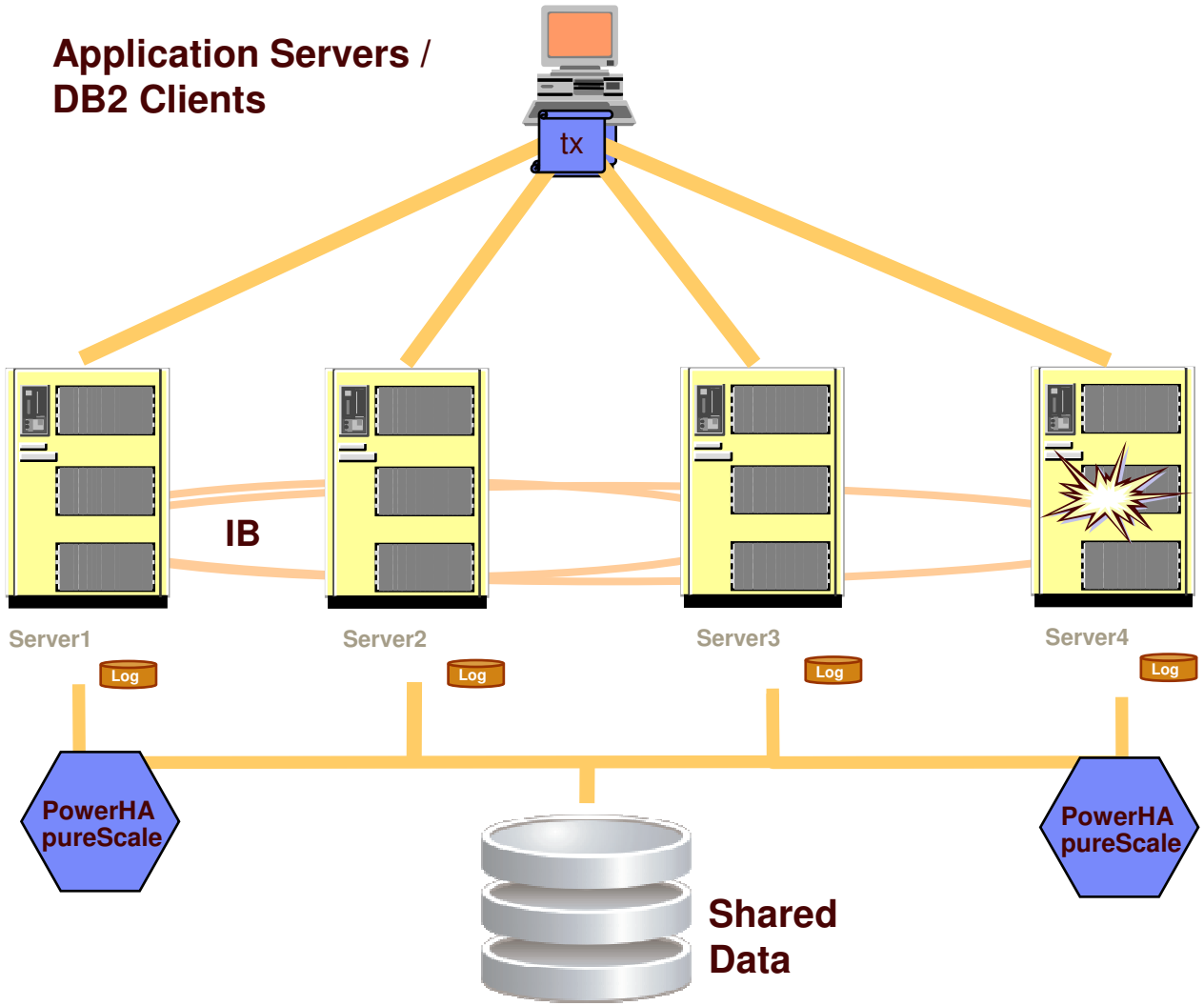
Failover **transparente a las aplicaciones**

Continuous Availability

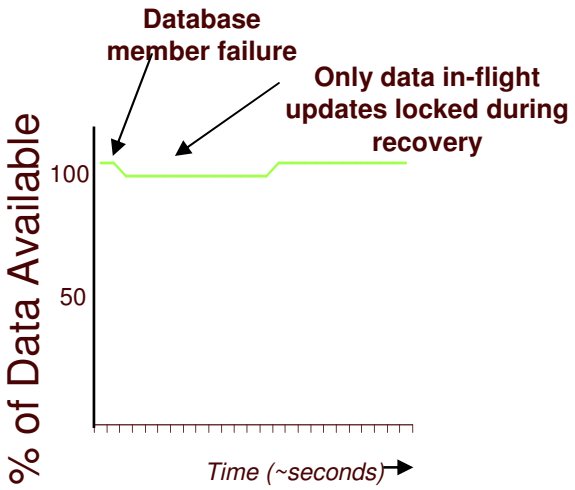
Administration

Reconexión automática de los clientes

Paradas No Planificadas



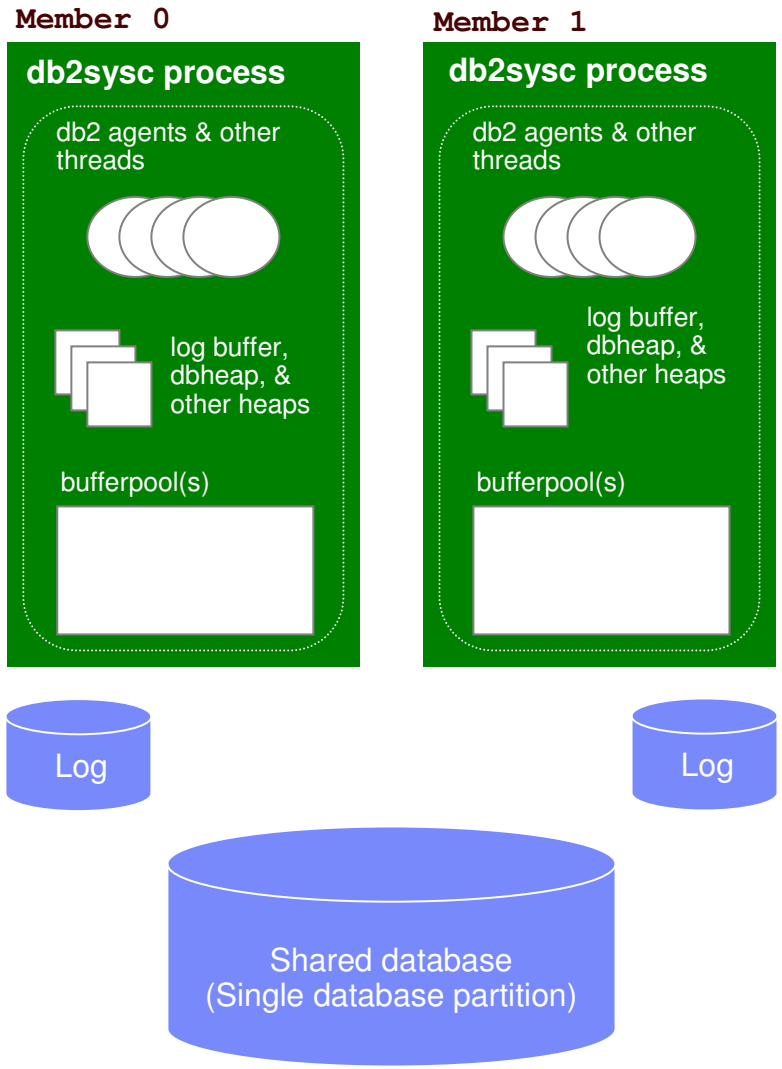
- **Disponibilidad Maximizada**
- **Datos en Lectura no se bloquean**
- **Sólo se bloquean los datos modificados en vuelo**
- **Las aplicaciones no tienen que reconectarse**



CONCEPTOS

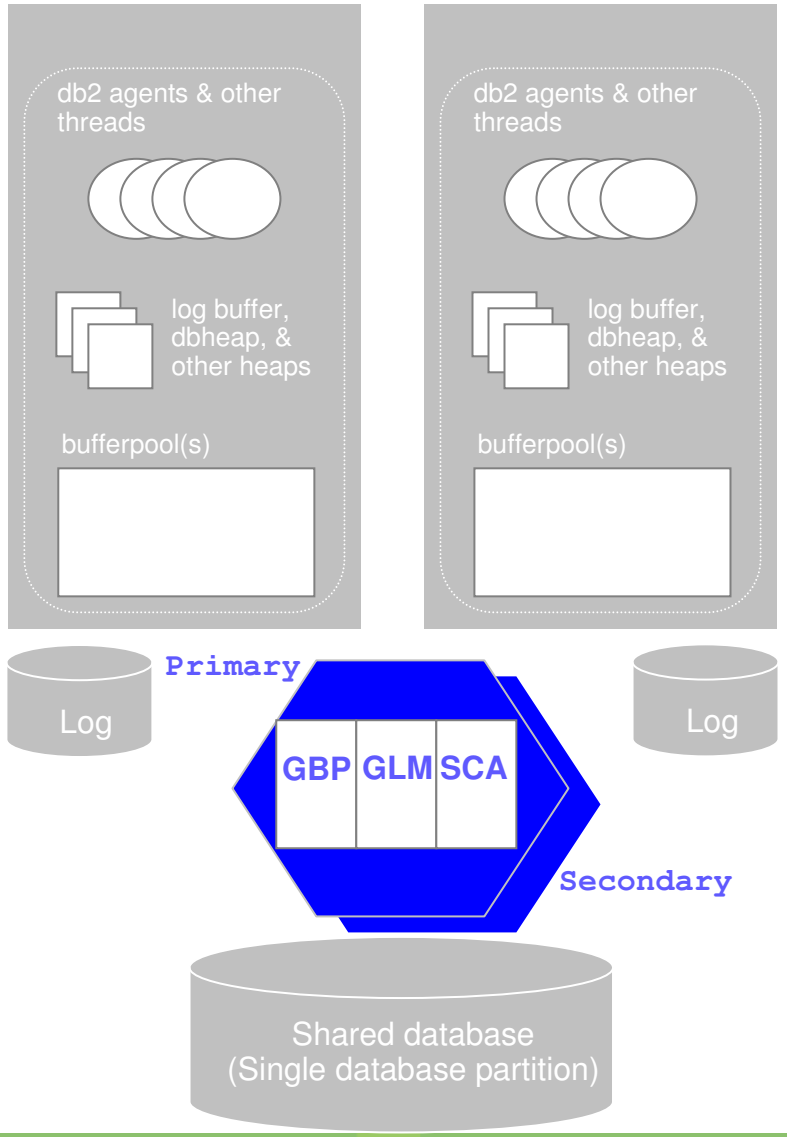
¿Qué es un Miembro de DB2 pureScale?

- **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)



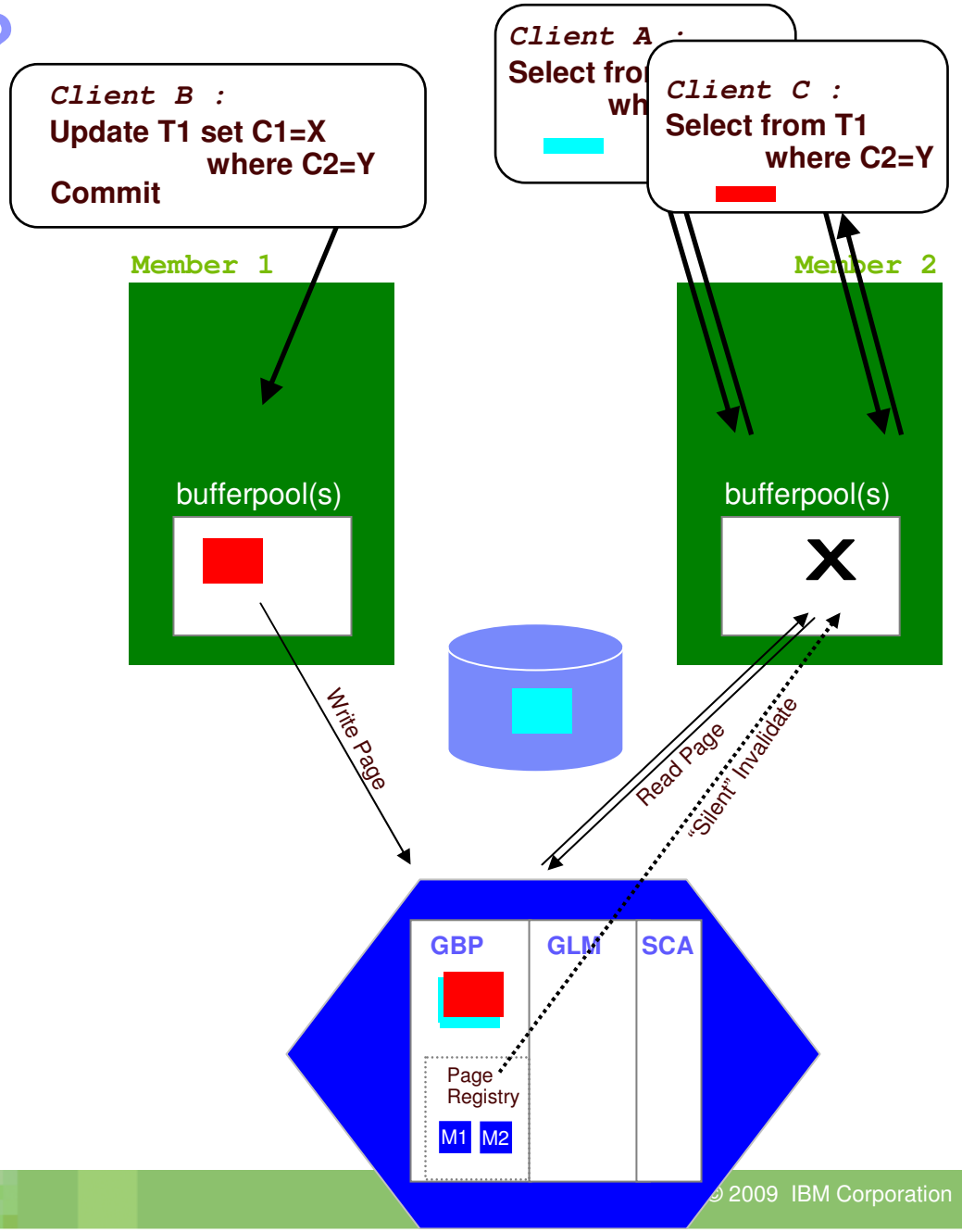
¿Qué es PowerHA pureScale?

- **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



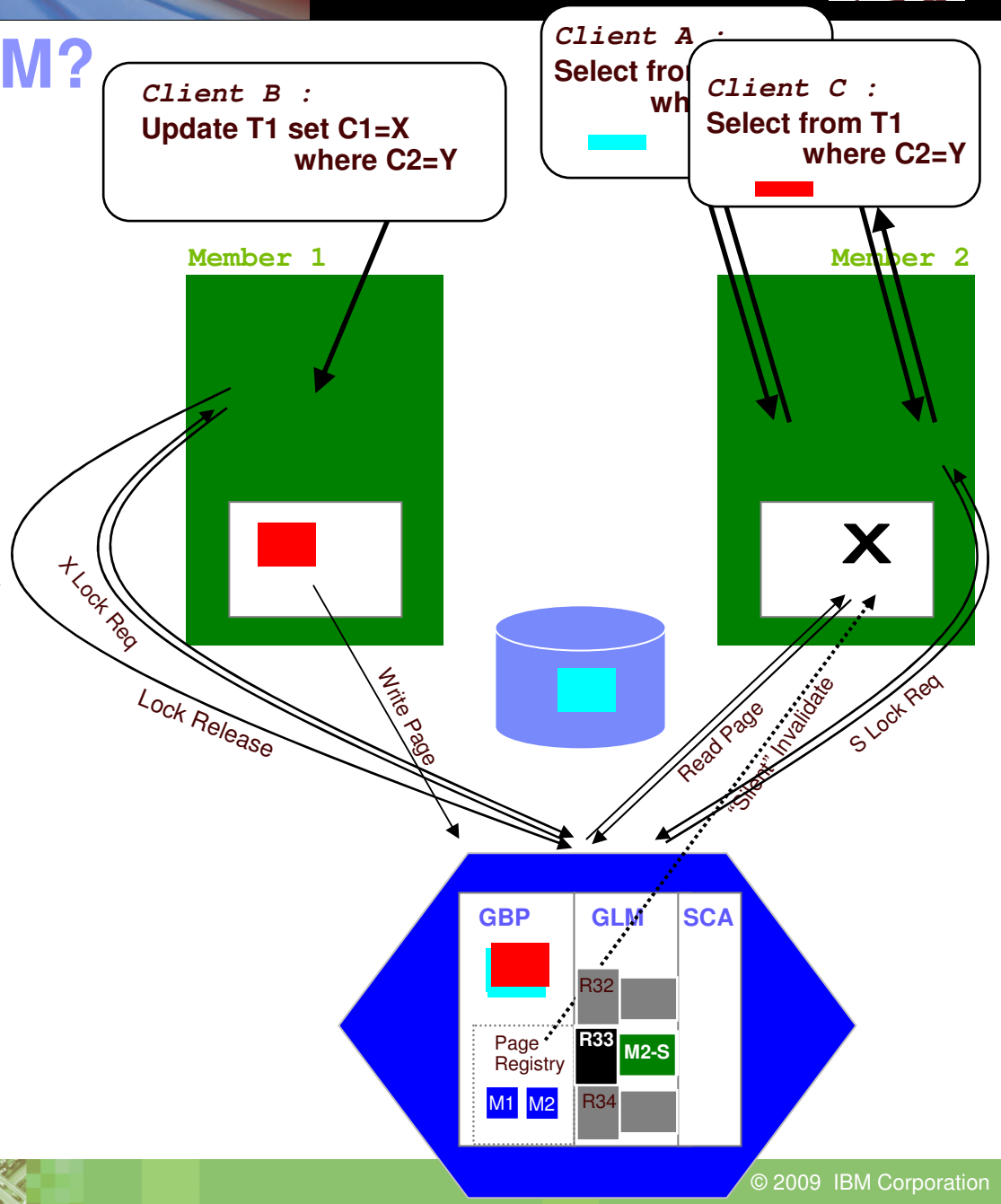
¿Para qué sirve el GBP?

- **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



¿Para qué sirve el GLM?

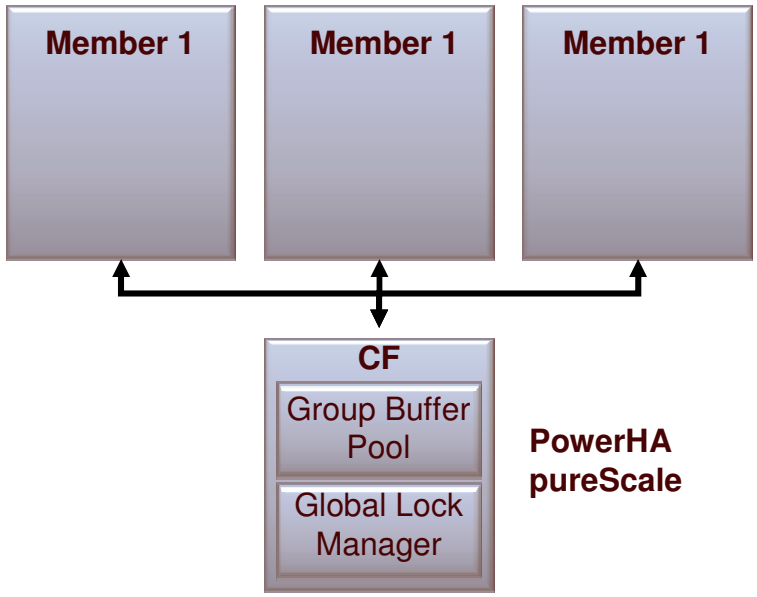
- **Grants locks to members upon request**
  - If not already held by another member, or held in a compatible mode
- **Maintains global lock state**
  - Which member has what lock, in what mode
  - Also - interest list of pending lock requests for each lock
- **Grants pending lock requests when available**
  - 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



Las Claves de la Escalabilidad y Disponibilidad

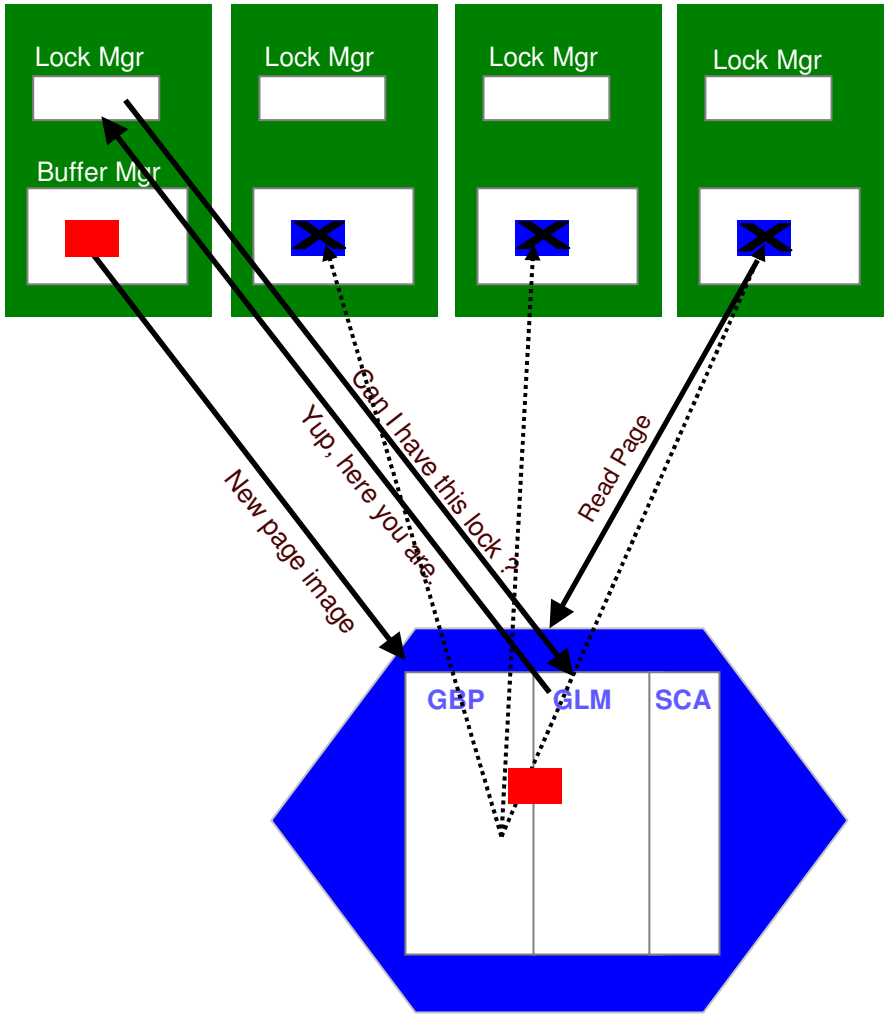
- **Efficient Centralized Locking and Caching**
  - As the cluster grows, DB2 maintains one place to go for locking information and shared pages
  - Optimized for very high speed access
    - DB2 pureScale uses Remote Direct Memory Access (RDMA) to communicate with the powerHA pureScale server
    - 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



REMOTE DIRECT MEMORY ACCESS (RDMA)

- **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**

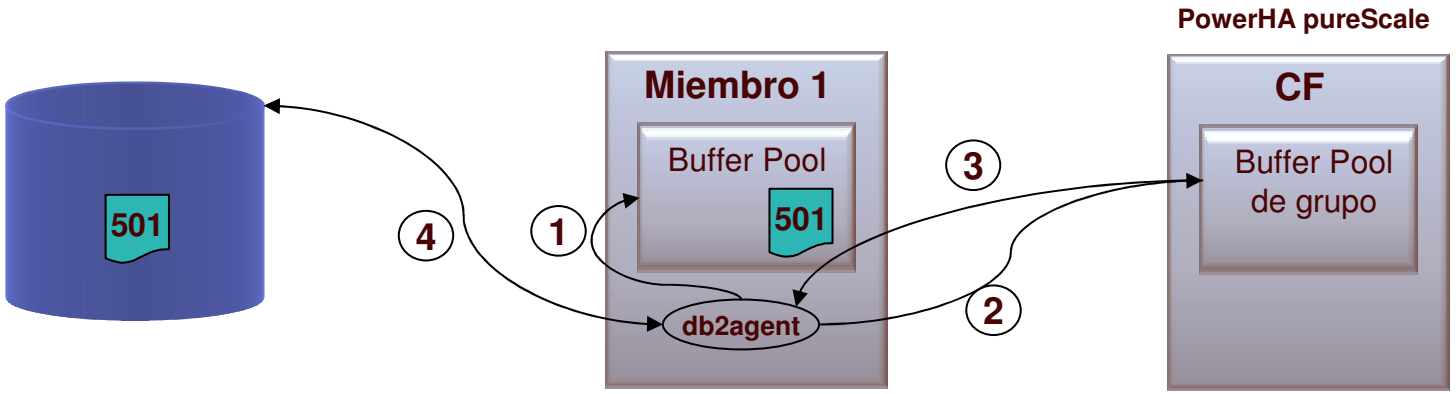


ESCALABILIDAD

¿Qué ocurre en pureScale para leer una pagína?

Agente en el Miembro 1 quiere leer página 501

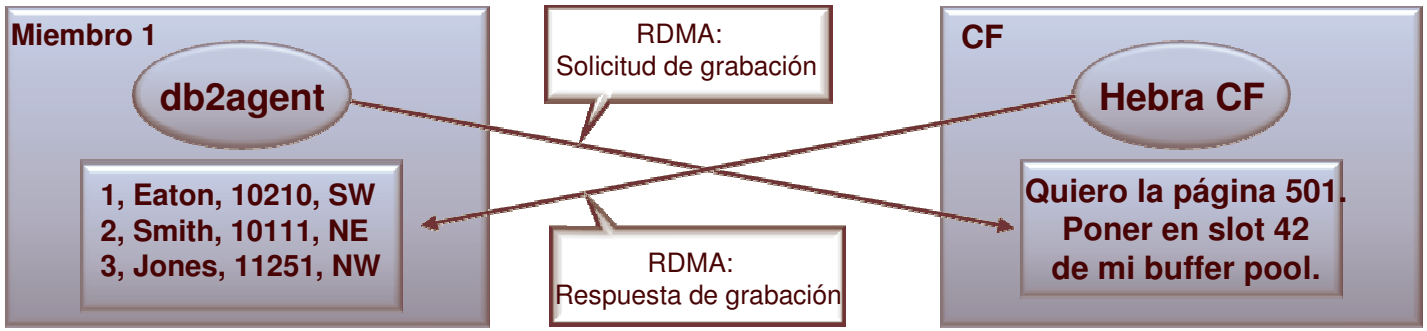
1. El db2agent busca en el buffer pool local: “página no encontrada”
2. El db2agent ejecuta una llamada de Lectura y Registro (RaR) RDMA directamente a la memoria del CF
   - Sin cambio de contexto, sin llamadas del kernel.
   - Solicitud es sincronizada al CF
3. CF responde que no tiene la página (nuevamente vía RDMA)
4. El db2agent luego lee la página en el disco



Las ventajas de Leer y Registro con RDMA

1. **El agente en el Miembro 1 escribe directamente a la memoria del CF con:**
   - El numero de página que quiere leer
   - El slot en el Buffer pool donde quiere fijar la página
2. **El CF responde escribiendo directamente a la memoria del Miembro 1:**
   - Indicando que la página no fue encontrada **o**
   - Con la página solicitada.

- **El tiempo total de principio a fin para el RAR medido en <u>microsegundos</u>**
- **Las llamadas son muy rápidas; es posible que el agente aún esté en el CPU, listo para recibir la respuesta**



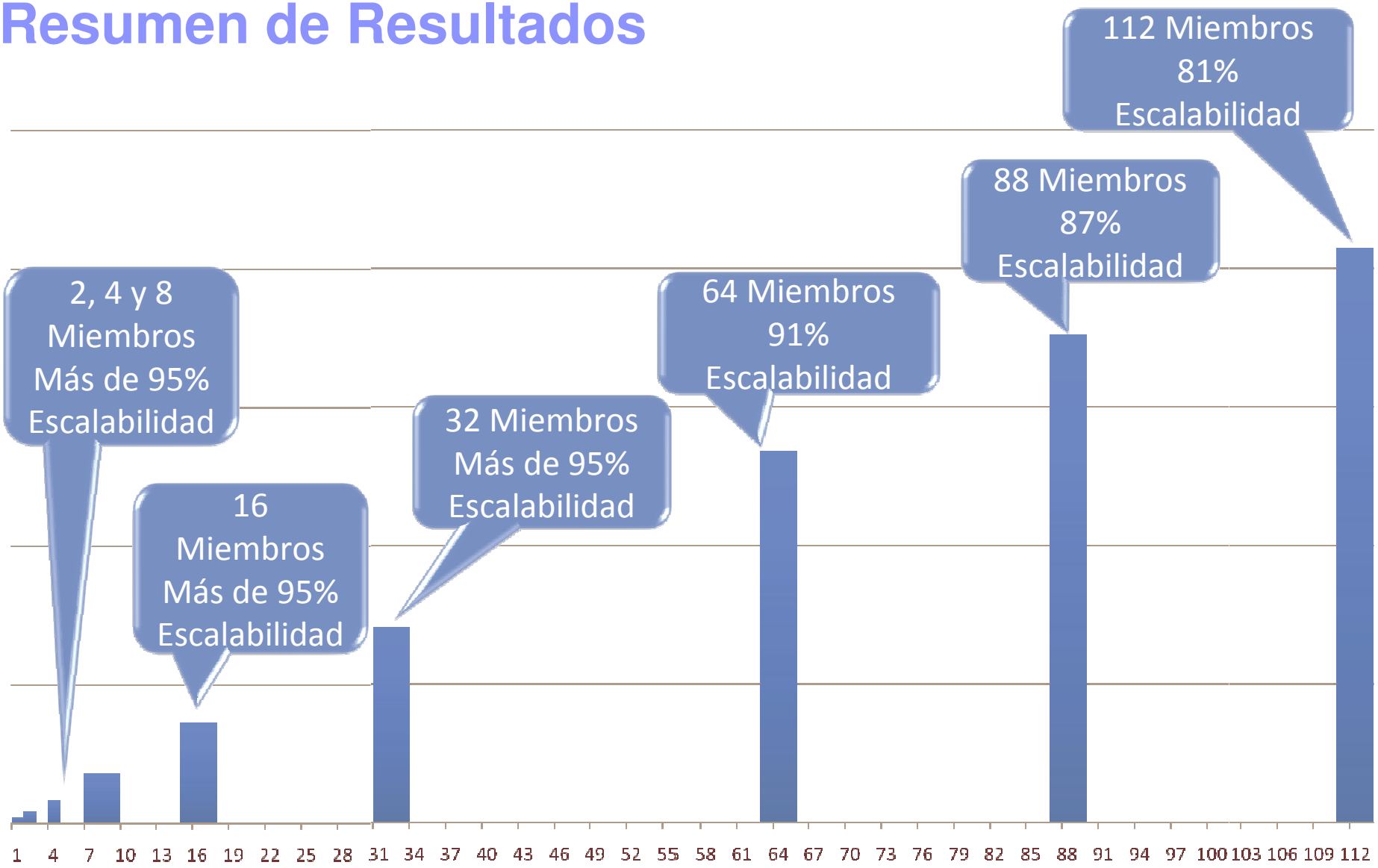
Mucho más escalable, no necesita localidad de datos

Prueba de Escalabilidad de la Arquitectura de DB2 pureScale

- **Prueba con carga de trabajo tipo *Web Commerce*.**
  - La mayoría de la carga es de lectura, pero **no únicamente de lectura.**
- **No dejar que las aplicaciones tengan conocimiento específico del clúster.**
  - **Sin direccionamiento de transacciones a los miembros.**
  - Las transacciones obtienen acceso a filas aleatoriamente .
  - Tenemos que demostrar escalabilidad transparente de aplicaciones.
- **Hacer pruebas hasta con más de 100 miembros.**

.

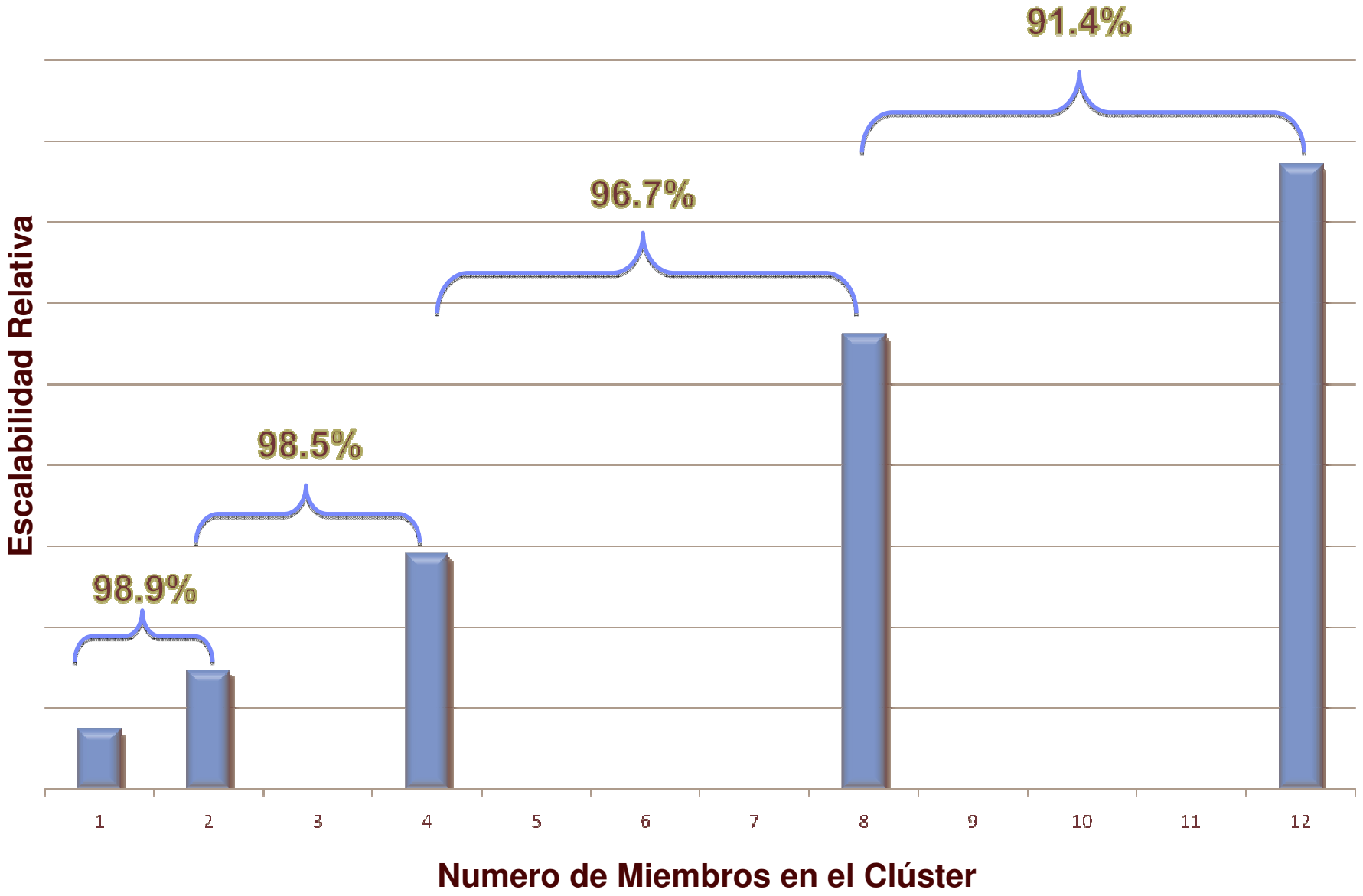
Resumen de Resultados



Ejemplo de Escalabilidad: *OLTP*

- **Utilizamos una carga de trabajo con más escritura**
  - 1 transacción de escritura para cada 4 de lectura
  - Ésta es una típica proporción de lectura/escritura de cargas de trabajo OLTP
- **La aplicación sigue sin conocimiento del clúster**
  - **Sin direccionamiento de transacciones a los miembros**
  - Demostrar escalabilidad transparente de la aplicación
- **Sistema Redundante**
  - Utilizando 14 sistemas p550 con 8-núcleos incluyendo PowerHA pureScale™ duplex
- **Escalabilidad es superior al 90%**

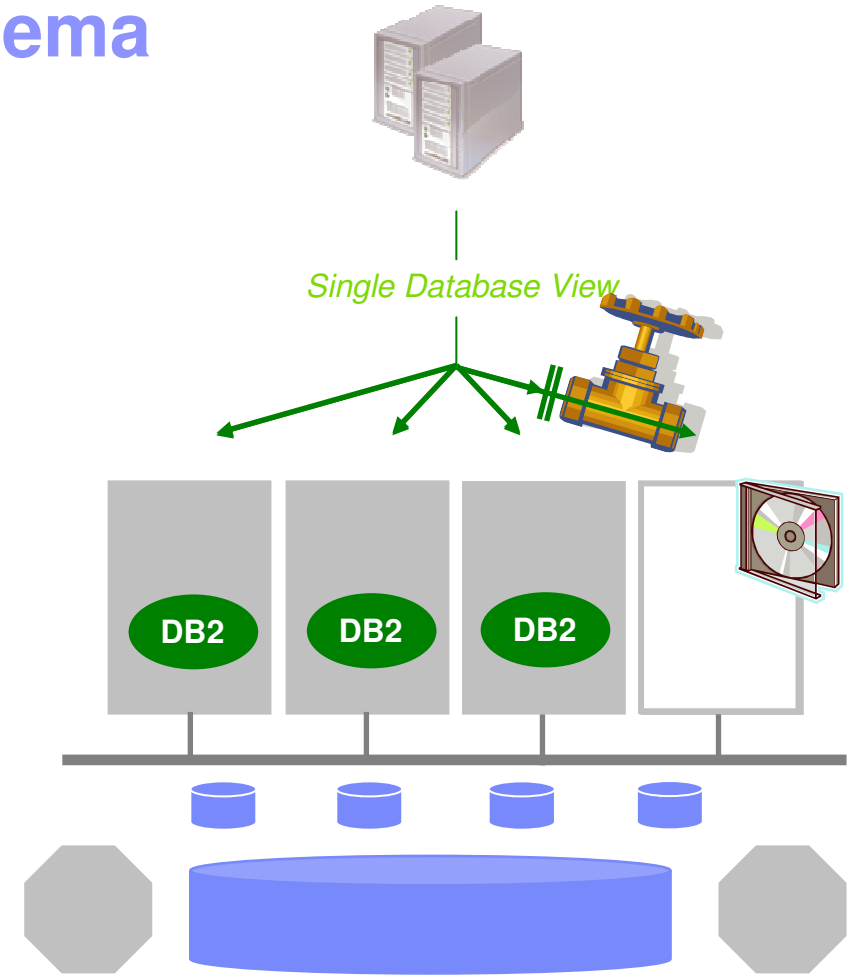
Resumen de Resultados: OLTP



DISPONIBILIDAD

Mantenimiento del Sistema

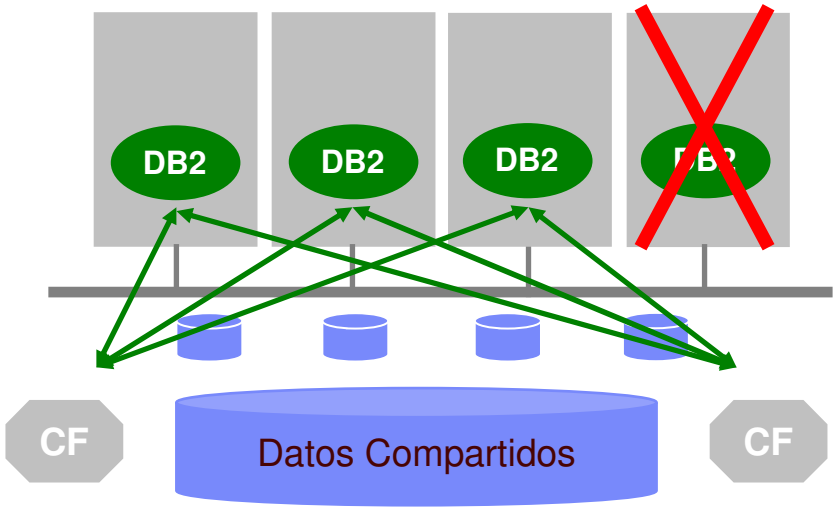
- **Goal: allow DBAs to apply system maintenance without negotiating an outage window**



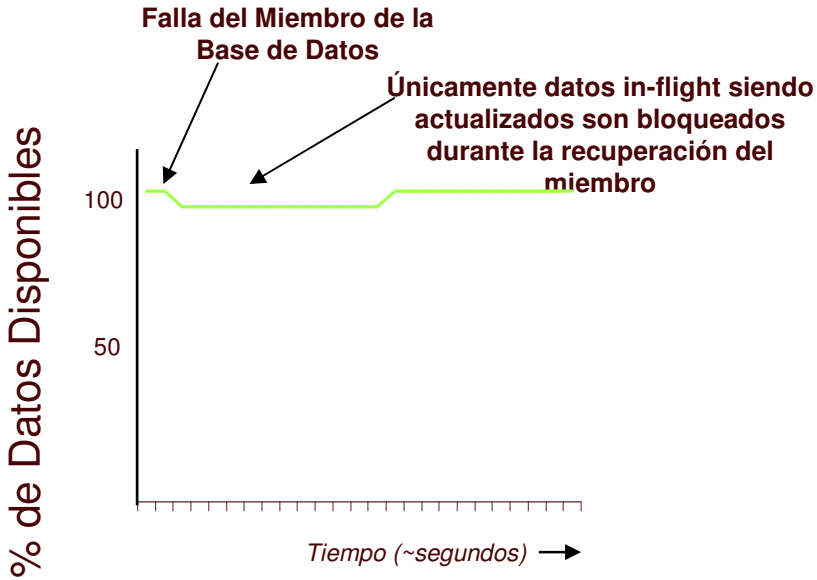
- **Procedure:**
  1. Drain (aka Quiesce)
  2. Remove & Maintain
  3. Re-integrate
  4. Repeat until done

Recuperación Online

- **El diseño de DB2 pureScale es para maximizar la disponibilidad durante un fallo y ejecutar el proceso de recuperación rápidamente**



- **Cuando el miembro de la base de datos falla, únicamente los datos *in-flight* permanecen bloqueados hasta que se halla completado la recuperación del miembro**
  - In-flight = datos siendo actualizados por el miembro que falló durante el tiempo de la falla



- **Meta de tiempo para disponibilidad de fila**
  - <20 segundos

Pasos después de que un Miembro falle

1. **Detección del fallo**
   - Detección de fallo Software por debajo del segundo.
   - Detección de fallo Hardware por debajo de 3 segundos.
     - *Heart beat* y otros chequeos.
     - No falsos *takeovers* (debidos a congestión)

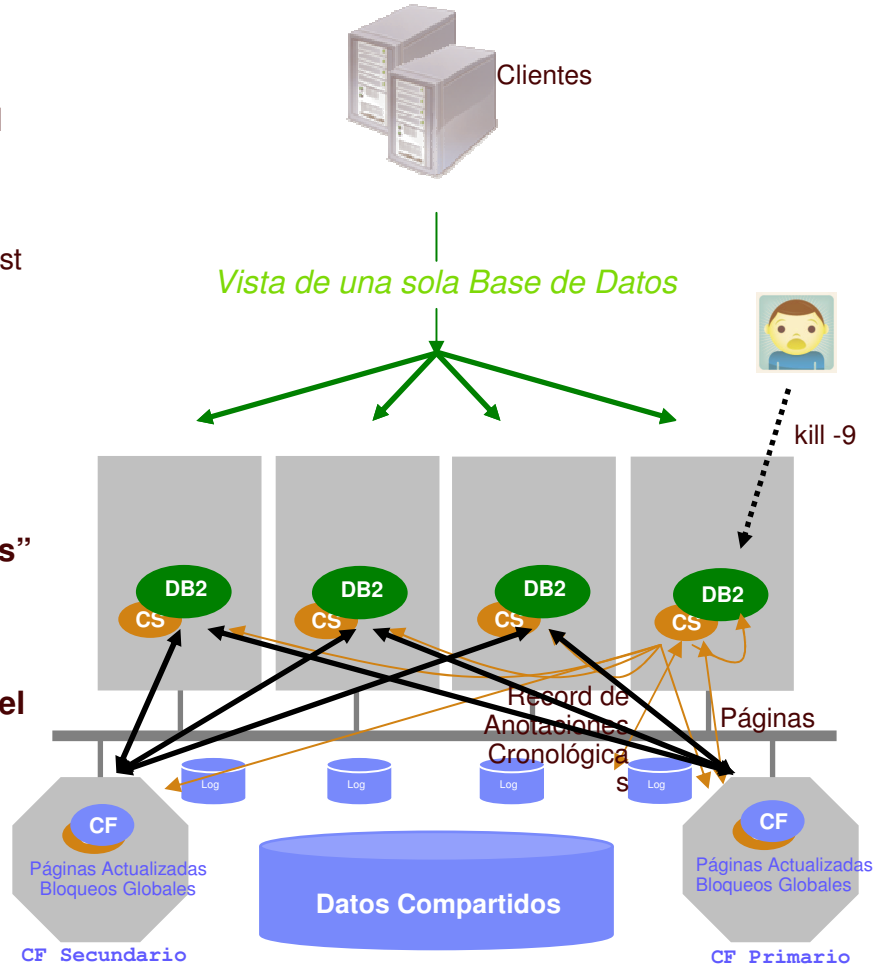
2. **El proceso de recuperación coge directamente desde la CF:**
   - Las páginas que necesitan ser reparadas
   - Ubicación de los ficheros de log para empezar el recovery

3. **Arrancar la instancia Light Instance para hacer redo y undo recovery.**

Ejecutar el Slide Show

Resumen fallo de Miembro

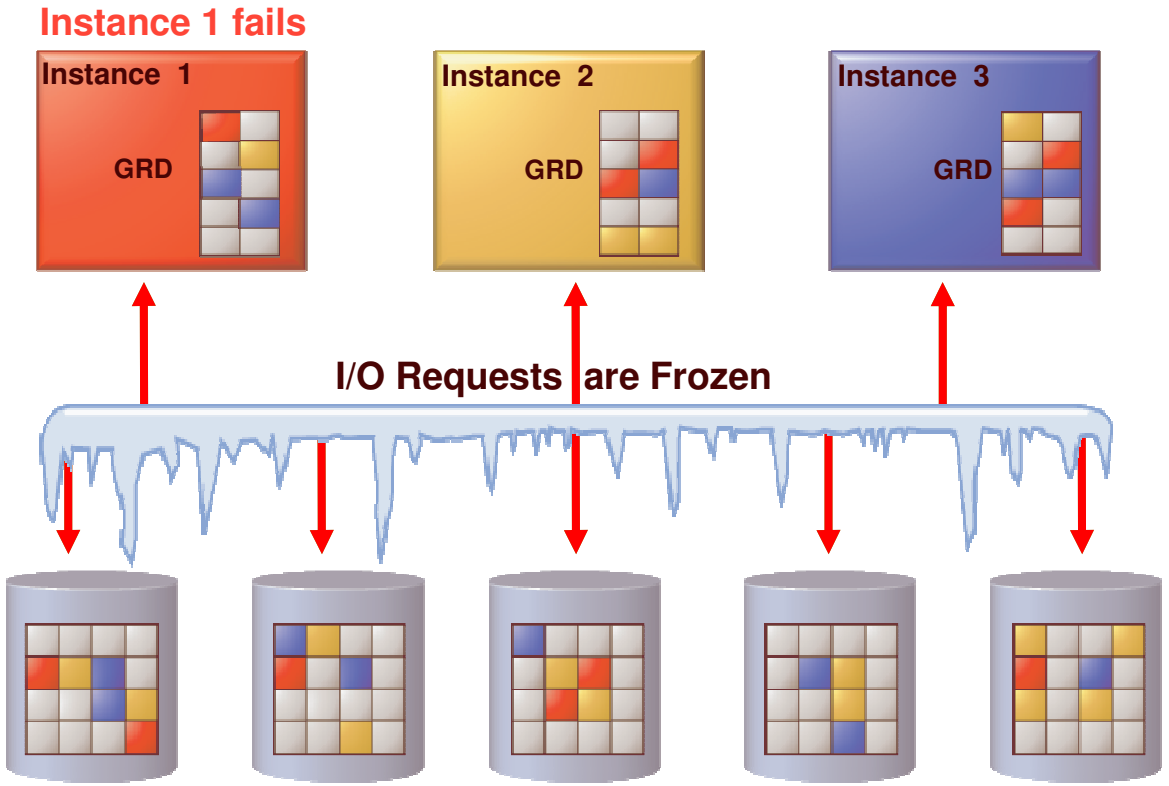
- **Fallo de un Miembro**
- **Los servicios de clúster de DB2 detectan automáticamente el fallo del miembro**
  - Informa a otros miembros y CFs
  - Comienza el rearranque automático del miembro en el mismo host (u otro remoto si es necesario).
  - El rearranque del miembro es como un *crash recovery*, pero mucho mas rápido.
    - Redo es limitado a transacciones en vuelo.
    - Beneficios del uso del caché de pagina en la CF.
- **El cliente es redirigido transparentemente a miembros "sanos" del cluster.**
- **Los otros miembros están completamente disponibles todo el tiempo *"Online Failover"***
  - CF mantiene los bloqueos de actualizaciones adquiridas por el miembro que ha caído.
  - Otros miembros pueden continuar leyendo y actualizando los datos no bloqueados por el miembro caído.
- **Rearranque del Miembro termina**
  - Los bloqueos son liberados y todos los datos están totalmente disponibles.



Comparativa con Oracle RAC

Con RAC la E/S de todo el cluster se Congela

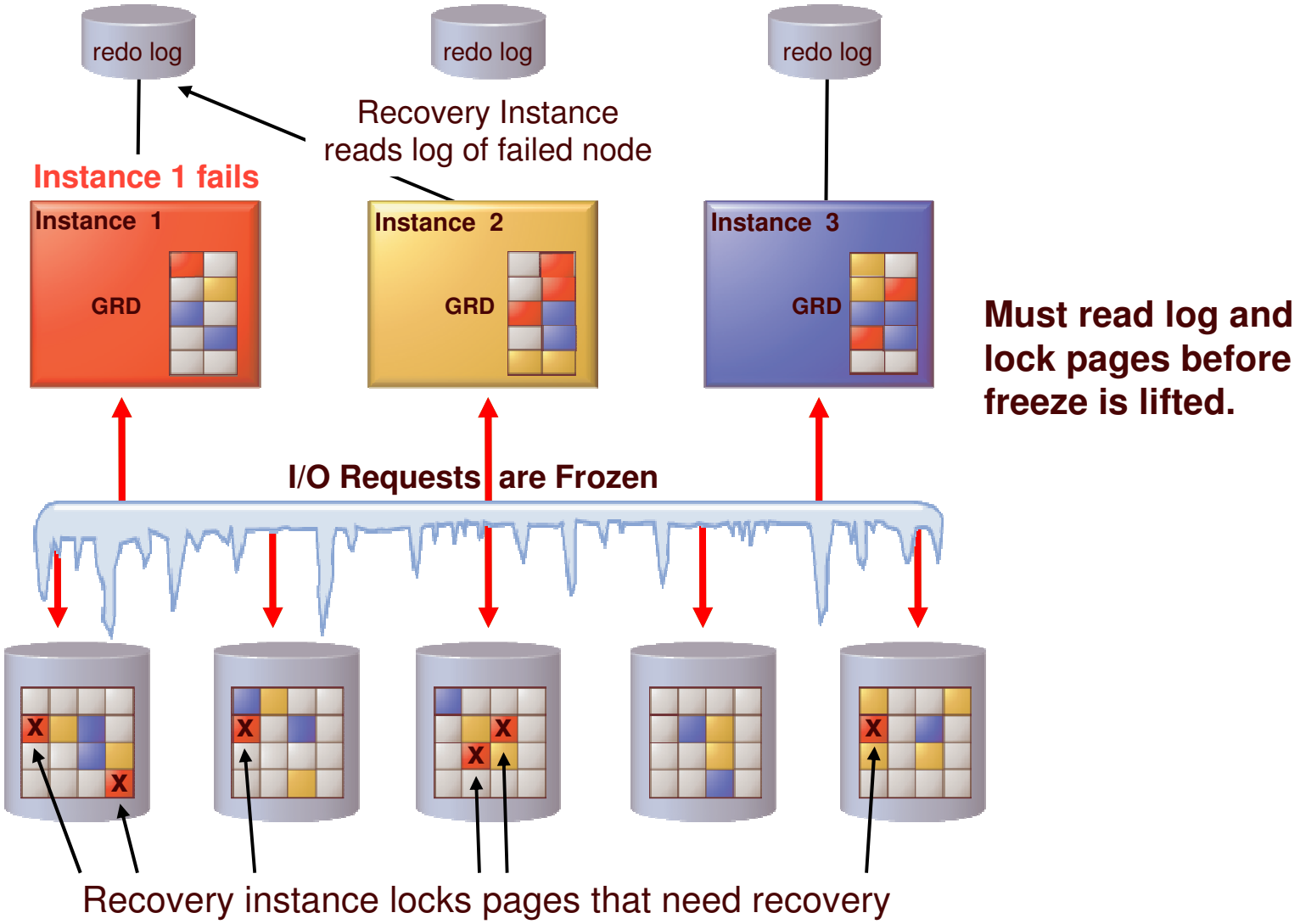
- **Global Resource Directory (GRD) Redistribution**



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

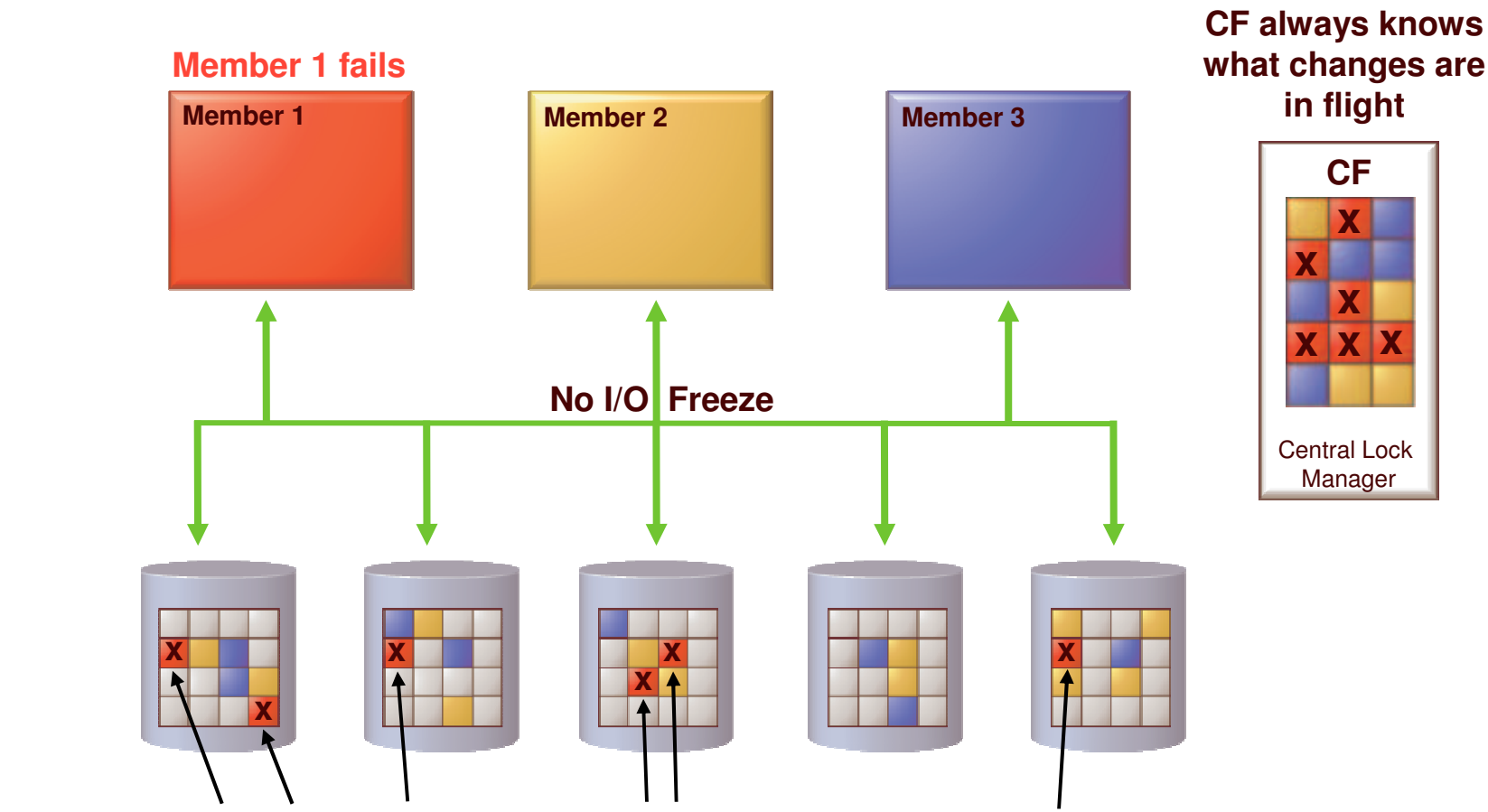
No Lock Updates

Con RAC Las paginas que necesitan *recovery* se bloquean

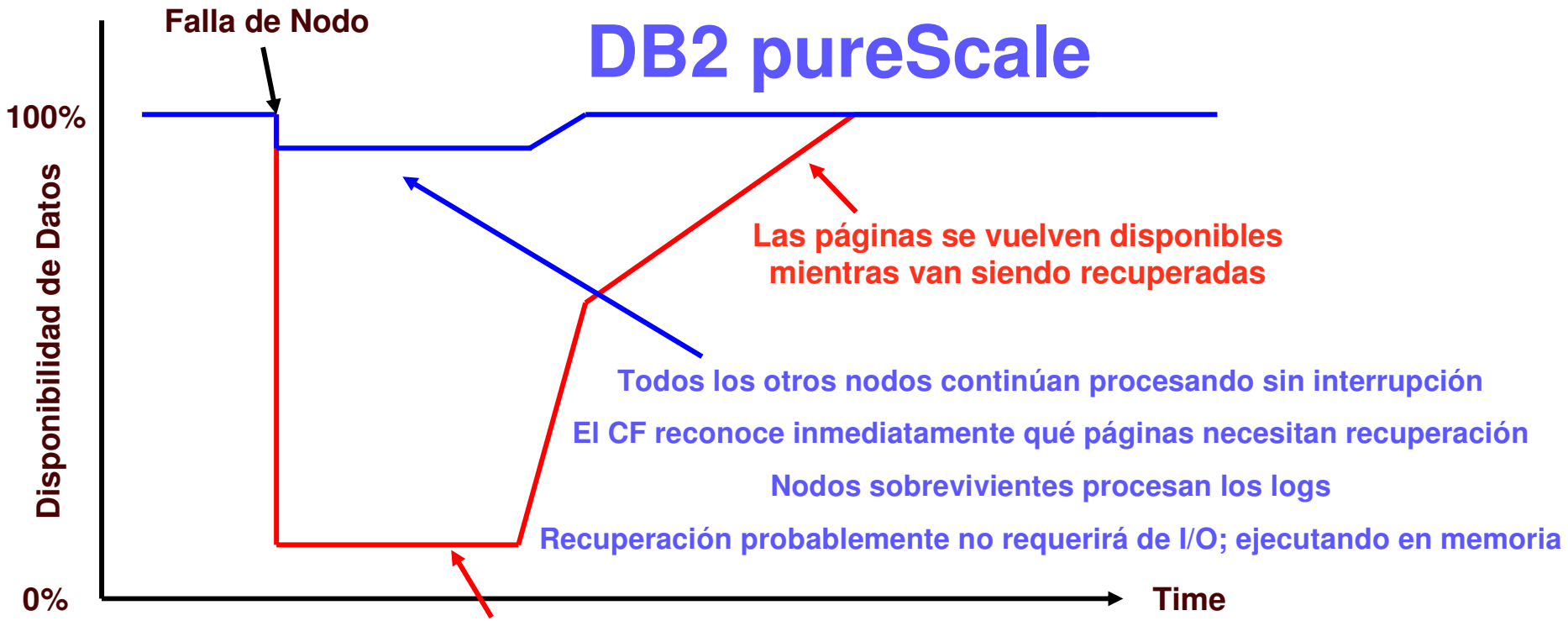


Must read log and lock pages before freeze is lifted.

Con DB2 pureScale no hay ninguna Congelación de E/S



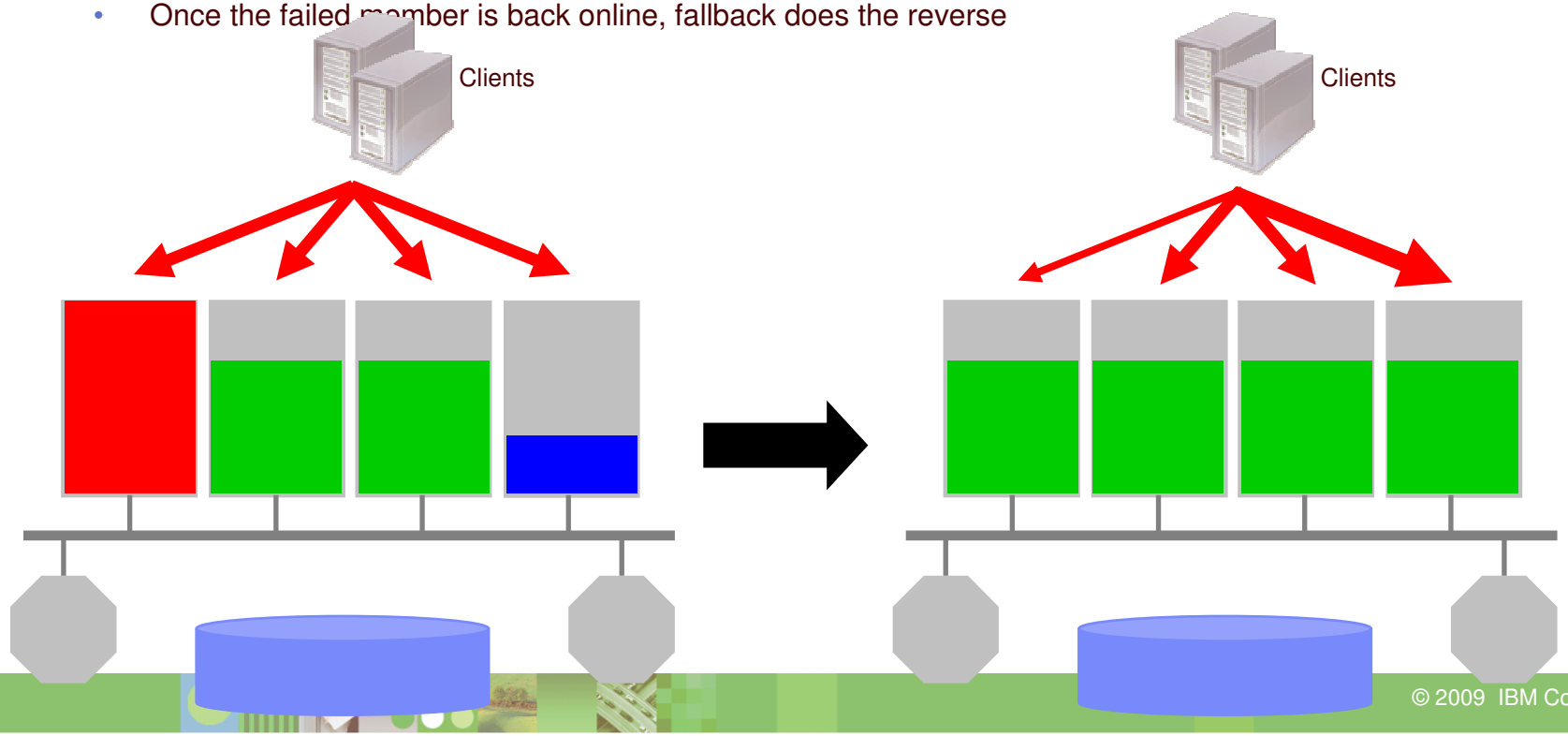
Comparativa de Disponibilidad



BALANCEO DE CARGA

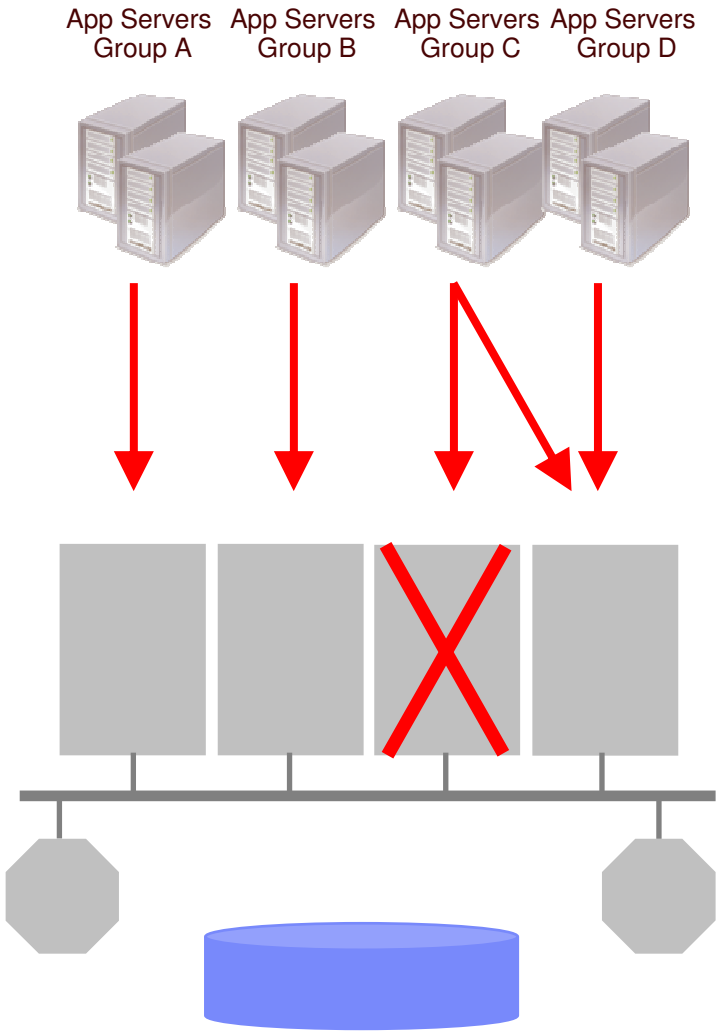
Balanceo automático de Carga

- **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



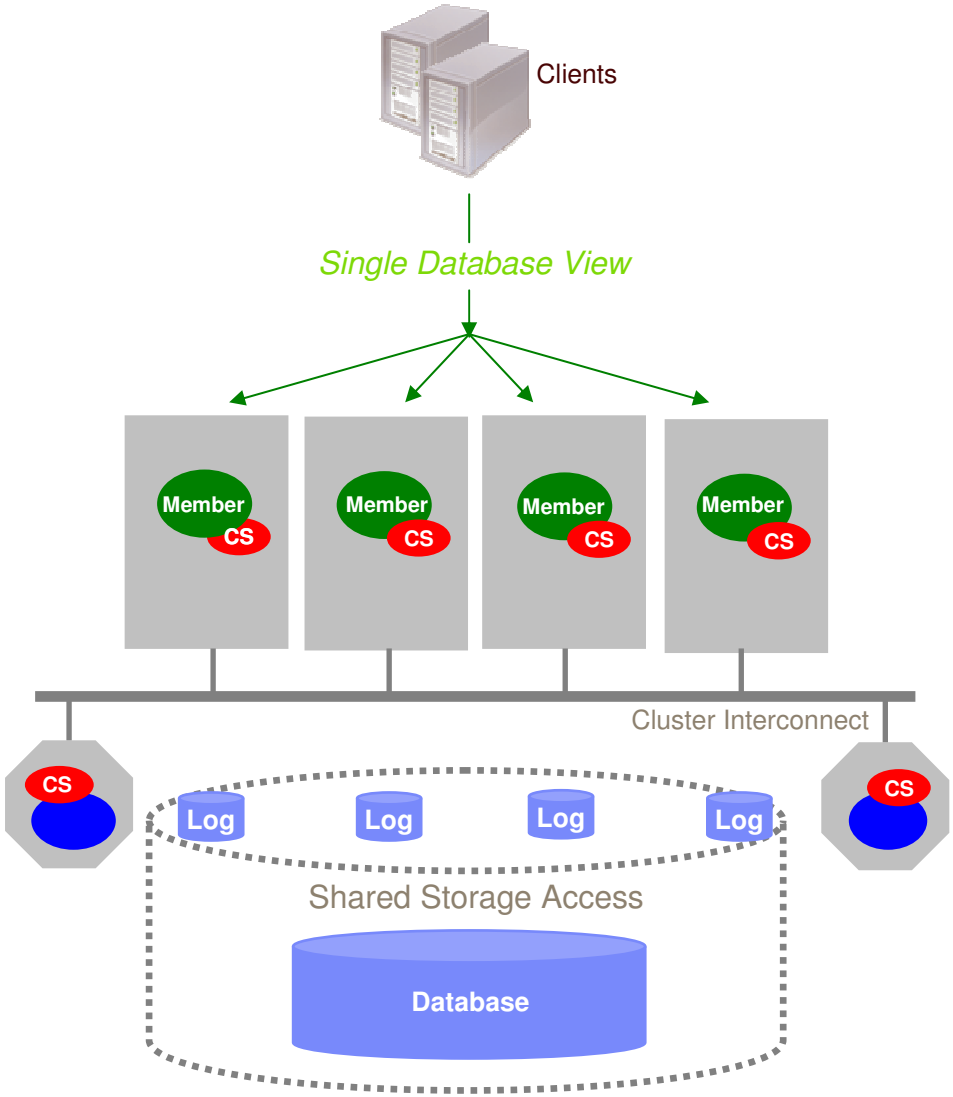
Conexión basada en Afinidad (Opcional)

- **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



FACILIDAD DE INSTALACION Y USO

Una Solución Completa



- ***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

Fácil de Usar

Un “bundle” contiene todo lo que se necesita para instalar y configurar.

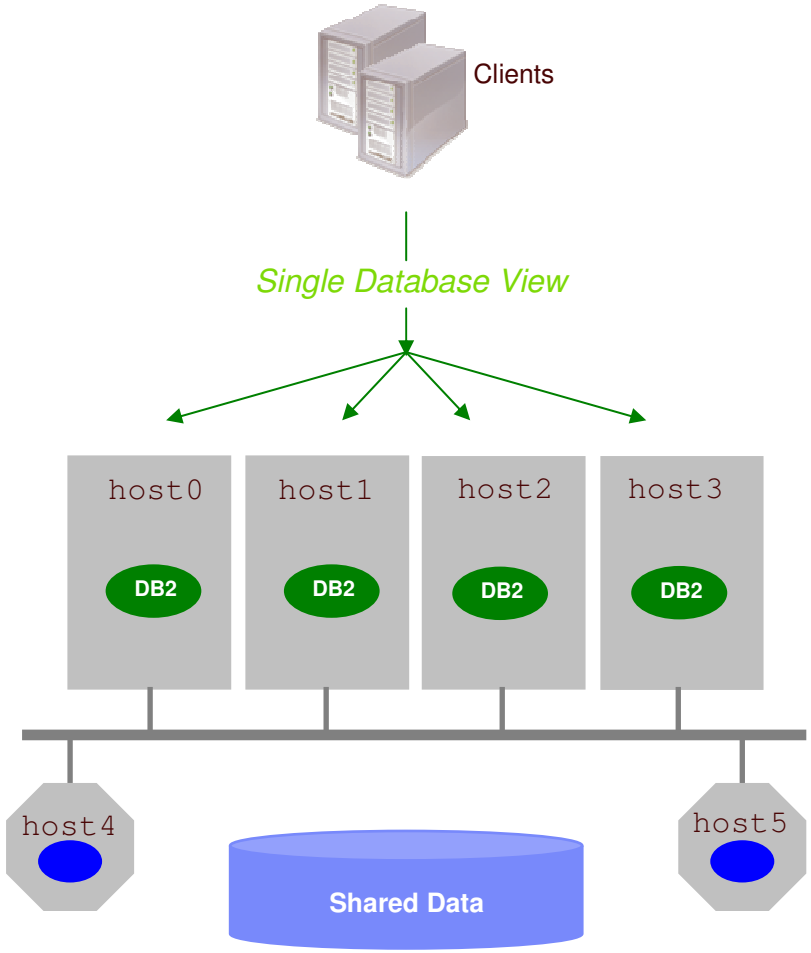
Instalación y configuración del cluster automatizado en la instalación.

Escalabilidad sin cambio en las aplicaciones.

Añadir un nuevo miembro requiere sólo 2 comandos.

Quitar un miembro requiere un sólo comando

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
```

Conclusiones

- **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