

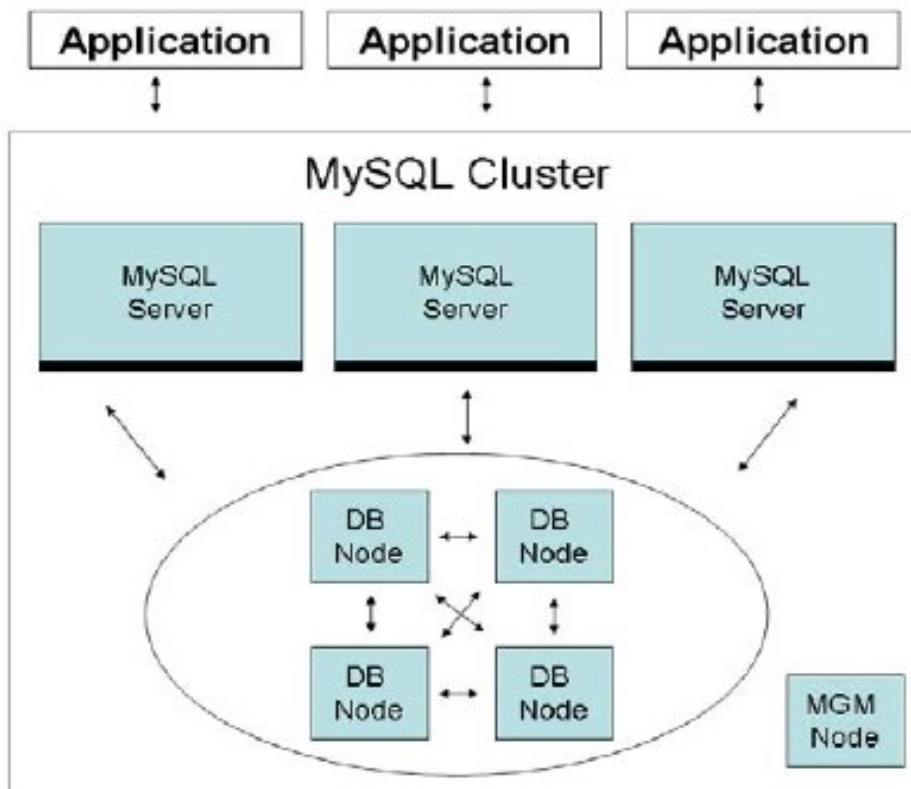
# MySQL

## Introducción

MySQL Cluster está diseñado para tener una arquitectura distribuida de nodos sin punto único de fallo.

MySQL Cluster consiste en 3 tipos de nodos:

1. Nodos de almacenamiento, son los nodos principales del sistema. Todos los datos son almacenados en los nodos de almacenamiento. Los datos son continuamente replicados para asegurarse de que los datos están disponibles aunque caigan varios nodos. Los nodos de almacenamiento mantienen todas las transacciones de la base de datos.
2. Los nodos de administración mantienen la configuración y son usados para cambiar la configuración del sistema. Normalmente un único nodo de administración es usado aunque pueden usarse varios. Estos nodos únicamente se usan en la implantación del sistema y en las reconfiguraciones, lo que significa que los nodos de almacenamiento son funcionales son los nodos de administración.
3. Los nodos servidor MySQL son servidores de MySQL accediendo a los nodos de almacenamiento. El servidor MySQL proporciona a desarrolladores una interfaz SQL estándar y MySQL que realiza las llamadas a los nodos de almacenamiento evitando la necesidad de la programación a bajo nivel.



En un servidor MySQL en un cluster MySQL está conectado a todos los nodos de almacenamiento y hay varios servidores MySQL en el mismo cluster MySQL. Todas las transacciones ejecutadas en los servidores MySQL son mantenidas por el conjunto común de nodos de almacenamiento. Esto

significa que tan pronto una transacción ha sido ejecutada en un servidor MySQL, es resultado es visible a través de todos los servidores MySQL conectados al cluster.

## Arquitectura de sistema de alta disponibilidad

- Los servidores MySQL están conectados con todos los nodos de almacenamiento. Si un nodo de almacenamiento falla, el servidor MySQL puede usar fácilmente otro
- Los datos de un nodo de almacenamiento son replicados en múltiples nodos de almacenamiento.
- Los nodos de administración pueden ser reiniciados y parados sin afectar al estado del cluster.
- Los datos se replican de forma asíncrona lo que implica tiempos muy bajos de failover en caso de caída de un nodo
- Los nodos han sido usando la arquitectura “no compartir nada”. Cada nodo tiene su propio almacenamiento de disco y memoria (También existe la opción de compartir disco y memoria para cuando varios nodos se encuentran en el mismo servidor)
- No existe un punto único de fallo. Cualquier nodo puede ser matado sin pérdida de datos ni parada de las aplicaciones que usan la base de datos.
- Las aplicaciones se conectan al Cluster de MySQL usando los servidores MySQL lo que permite:
  - Independencia de los datos, esto significa que la aplicación puede ser escrita sin tener conocimiento del almacenamiento físico de los datos.
  - Transparencia en la red y la distribución, esto significa que la aplicación depende de ningún modo de detalles operacionales de la red, no de la distribución de los datos en los nodos de almacenamiento.
  - La replicación de los datos a los nodos de almacenamiento es transparente
  - Una interfaz estándar SQL de fácil uso para desarrolladores.

## Replicación Síncrona

- Todos los datos en la base de datos se replican a los nodos de un mismo grupo
- Cada transacción es propagada a todos los nodos de almacenamiento apropiados durante la transacción.
- Cuando la transacción va a ser aplicada, se envía una petición a todos los nodos de almacenamiento envueltos en ella. Cuando todos los nodos indican que están preparados se aplica la transacción y la aplicación es informada del éxito de la acción.
- Para asegurarse de que la base de datos es consistente, si uno de los nodos de almacenamiento falla durante la transacción, la transacción es abortada y la aplicación es informada de ello para que se inicie de nuevo la transacción.

## **Detección de fallos**

Existen dos métodos de detección:

### ***Pérdida de la comunicación***

Si un nodo de almacenamiento notifica que la conexión entre dos nodos se ha perdido, el resto de nodos es inmediatamente informado y se clasifica el nodo como fallido. Los nodos fallidos automáticamente reinician y se conectan al cluster de MySQL como nodos nuevos, no afectando a la aplicación. La pérdida de comunicación es la forma más rápida de detectar que un nodo ha fallado.

### ***Fallo Heartbeat***

Puede ser que un nodo falle por motivos diferentes a una pérdida de comunicación, por ejemplo fallos de disco, estos fallos hacen que el nodo se comporte de forma anómala pero no tiene porque producirse una pérdida de comunicación.

Para evitar estos problemas cada nodo envía una señal al siguiente (los nodos de almacenamiento están estructurados de forma lógica en anillo) en caso de que este no responda, se envían 3 señales más y si sigue sin responder se marca el nodo como fallido.

## **Particionamiento de red**

Cuando varios nodos caen, el resto de nodos usa un protocolo de particionamiento de red para comprobar si están en mayoría teniendo más de un nodo en cada grupo de nodos o teniendo más de la mitad de los nodos activos por cada grupo. Esto asegura que las transacciones son completamente aplicadas eliminando la posibilidad de que un grupo de nodos actúe solo y solamente se apliquen parte de las transacciones.

## **Determinando el orden de fallo**

En raras ocasiones cuando múltiples nodos fallan a la misma vez, en este caso un protocolo de failover de aplicación de transacciones en 2 fases es utilizado para determinar en que orden los nodos han ido fallando para asegurarse que pueden ser restaurados.

El protocolo se ejecuta en dos pasos:

1. Todos los nodos envían una lista de todos los nodos de almacenamiento que se consideran que han fallado a un nodo de almacenamiento maestro.
2. El maestro determina e informa a todos los nodos del orden de fallo de los nodos.

## **Recuperación de nodos**

### ***Recuperación simple de un nodo***

Si un nodo falla, el nodo reinicia preguntando a un nodo backup (por ejemplo un nodo en el mismo

grupo de nodos) por la porción que debe almacenar. El nodo backup envía esta información porción-a-porción al nodo reiniciado.

El nodo tras sincronizarse está preparado para procesar transacciones inmediatamente.

## ***Recuperación múltiple***

Mediante el protocolo de determinación de fallo se especifica el orden de caída de los nodos y a través de un nodo maestro se van restaurando uno a uno.

## **Recuperación del Sistema**

### ***Archivado***

Todas las operaciones de la base de datos se archivan en un log que se aplicará en caso de recuperación del nodo para mantener la base de datos al día.

### ***Puntos de control Locales***

Cada cierto tiempo a través de un logaritmo de punto de control se crea una foto consistente de los datos del nodo, de forma que para la recuperación del nodo se carga y se aplica el log, así los datos del nodo estarán actualizados

### ***Puntos de control Globales***

Desde que MySQL Cluster es una base de datos de memoria principal, las transacciones primero se aplican en la memoria principal. Mientras haya nodos vivos en todos los nodos de los grupos, la replicación hace los datos seguros. Para recuperarse de fallos de sistema (cuando todos los nodos fallan). MySQL Cluster vacía el log a disco durante un punto de control global, a veces llamado aplicación de transacciones de grupo. Después de un punto de control global, todas las transacciones también se aplican a disco.

Para controlar el estado de aplicación de una transacción, MySQL Cluster asigna una identificación de punto de control global a cada transacción finalizada. El identificador global de punto de control especifica cual hasta donde ha llegado la aplicación de la transacción y puede usarse para verificar para comprobar que la aplicación ha sido aplicada a disco. Si, para una instancia el punto de control 15 ha finalizado, el resultado de todas las transacciones con identificación de punto de control global menor o igual que 15 han sido aplicadas a disco.

## ***Recuperación de Sistema***

Se realiza en 2 pasos:

1. Cada foto es cargada por cada nodo de almacenamiento. Después el log es aplicado.
2. Se recuperan todas las transacciones usando la identidad de un punto de control menor o igual que la más reciente terminada. Con esto se asegura que las transacciones más recientes han sido aplicadas.

# Replicación y Particionado transparente

Cuando una aplicación quiere ejecutar una nueva transacción, MySQL Cluster usa uno de los nodos de almacenamiento para ejecutarla. Si el nodo está caído, la transacción pasa automáticamente a otro nodo de almacenamiento. Un algoritmo de Round-Robin selecciona automáticamente el siguiente nodo de almacenamiento .

## Escenarios de Fallo

### ***Falla un nodo MySQL Server***

Puede ser reiniciado y vuelto a conectar al Cluster. El nodo reiniciado puede ser conectado a cualquier nodo de almacenamiento. La base de datos se puede servir a través de cualquier otro nodo MySQL Server.

### **Fallo de un nodo de almacenamiento**

Si un nodo de almacenamiento cae, todos los nodos son informados por la pérdida de conexión con este o por la conexión heartbeat. Desde que los datos son replicados, siempre hay otro nodo de almacenamiento para servir a las peticiones de transacciones.

### ***Nodo de administración falla***

Desde que los nodos de almacenamiento y servidores son independientes del servidor de administración , puede fallar todas las veces que quiera pues no afecta al funcionamiento del sistema.

### ***Fallos de conexión***

Si la conexión entre los nodos de almacenamiento se rompe, los nodos obtienen información de que los nodos están caídos. La pérdida de conexión se trata igual que la caída de un nodo. La primera parte del protocolo de fallo en 2 fases es usado para determinar que nodos son inalcanzables, y después el MySQL Cluster se recompone con los nodos a los que es posible conectarse.

Las conexiones poseen un sistema de superación de fallos para manejar los fallos de comunicación. Usando este sistema, las conexiones TCP/IP tienen un tiempo de superación de fallo de unos 100 milisegundos , y la interfaz Scalable Coherent tiene un tiempo de fallo de 100 microsegundos, EL sistema de superación de fallos efectivamente oculta problemas de tarjetas PCI, problemas de cable, y fallos de alternancia por mensajes de enrutado a través de otras conexiones.

### ***Fallos de disco***

Todos los nodos de almacenamiento almacenan sus propios datos. Si un nodo detecta que su sistema de archivos se está corrompiendo , termina su ejecución. Después de que el sistema haya sido arreglado, el nodo es reiniciado usando el protocolo de recuperación de nodo

## **Conclusión**

MySQL Cluster es una base de datos de alta disponibilidad usando una arquitectura basada en no compartir nada y una interfaz SQL estándar.

MySQL Cluster es tolerante a fallos de varios de los nodos de almacenamiento y se reconfigura en el aire para enmascarar los fallos. Las capacidades de auto curación , como la transparencia de la distribución de los datos y particionamiento de la aplicación, dan como resultado un simple modelo de programación que permite a los desarrolladores incluir fácilmente en sus aplicaciones sin código complejos de bajo nivel.

Mirar la atomicidad transaccional