HADOOP - Método Emergente

MapReduce

- Introducción:

MapReduce se usa en aquellos problemas en los que se encuentran involucrados grandes conjuntos de datos que deben ser procesados por una gran cantidad de nodos, realizando el procesamiento en paralelo con datos obtenidos de un sistema de archivos (no estructurado) o de una base de datos (estructurado).

- Requerimiento:

Mejorar el rendimiento de lectura de los datos.

- Atributo de calidad:

Rendimiento/Performance.

- Táctica:

A través de las funciones Map, se transforma un conjunto de datos en pares de clave/valor. Cada uno de estos elementos está ordenado por su clave.
Y mediante Reduce, se combinan los valores con la misma clave en un mismo resultado.

ID Clúster:

- **Introducción**:

Cuando un DataNode se inicia, realiza un Handshake con el NameNode con el propósito de verificar que el ID del DataNode coincida con el del NameNode.

- Requerimiento:

Proteger la integridad de los datos.

- Atributo de calidad:

Confiabilidad.

- Táctica:

Los DataNodes con ID diferente al del NameNode, no podrán unirse al clúster.
Cuando se crea un nuevo DataNode, no posee ID. Al ingresar a algún cluster, se le asigna el ID de ese cluster.

Heartbeats:

- Introducción:

Durante el funcionamiento normal, el DataNode envía señales al NameNode para confirmar que esta operativo y que sus datos están disponibles.

- Requerimiento:

Verificar el correcto funcionamiento de los nodos y la disponibilidad de sus datos.

- Atributo de calidad:

Confiabilidad.

- Táctica:

El DataNode envía heartbeats al NameNode con un intervalo de tiempo de, por defecto, tres segundos.
Si el NameNode no recibe esas señales durante un tiempo "largo", asume que el DataNode esta fuera de servicio y crea nuevas copias de los archivos que contenía.

Replicación:

- Introducción:

Una característica importante de Hadoop es el particionamiento de los datos en miles de servidores.

- Requerimiento:

Garantizar la durabilidad de los datos y la disponibilidad a los mismos.

- Atributo de calidad:

Confiabilidad.
Escalabilidad.
Disponibilidad de los datos.

- Táctica:

Copiar el contenido de los archivos en varios DataNodes.

Backup:

- Introducción:

El NameNode puede llevar a cabo el rol de BackupNode, el cual es capaz de crear checkpoints. Si el NameNode falla, el BackupNode utiliza su checkpoints más reciente (que esta en memoria) y de esta manera recompone el último estado del NameNode.

- Requerimiento:

Respaldar el fallo del NameNode.

- Atributo de calidad:

Confiabilidad.

- Táctica:

Restaurar el ultimo estado operativo del NameNode a traves del checkpoint del BackupNode.

Block Scanner:

- Introducción:

El escaner detecta si un bloque es corrupto. Si es así, lo notifica al NameNode.
Este lo marca como corrupto, pero no lo elimina inmediatamente, primero realiza una copia buena del bloque, y recién cuando la copia esta creada elimina el bloque corrupto.

- Requerimiento:

Verificar si un bloque es corrupto o esta dañado.

- Atributo de calidad:

Confiabilidad.

- Táctica:

Cada DataNode ejecuta un escaner periódicamente que escanea sus copias y verifica los datos almacenados.