EBook Gratis

APRENDIZAJE mapreduce

Free unaffiliated eBook created from
Stack Overflow contributors.

#mapreduc

e

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: mapreduce

It is an unofficial and free mapreduce ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official mapreduce.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con mapreduce

Observaciones

Esta sección proporciona una descripción general de qué es mapreduce y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de mapreduce y vincular a los temas relacionados. Dado que la Documentación para mapreduce es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Instalación o configuración

Mapreduce es una parte de Hadoop. Por lo tanto, cuando se instala Apache Hadoop (o cualquier distribución de Hadoop) MR se instala automáticamente.

MapReduce es el marco de procesamiento de datos a través de HDFS (sistema de archivos distribuido de Hadoop). Los trabajos de MR pueden escribirse usando Java, python, Scala, R, etc.

¿Qué hace mapreduce y cómo?

Mapreduce es un modelo de programación para procesar en (muy) grandes cantidades de datos.

El 'HPC' tradicional (computación de alto rendimiento) acelera los cálculos grandes en cantidades relativamente grandes de datos mediante la creación de un conjunto de computadoras altamente conectadas (que utilizan elementos como redes extremadamente rápidas y acceso rápido al almacenamiento compartido, memoria compartida) para manejar problemas informáticos Por lo general, requieren cálculos para tener acceso a los datos de los demás. Un ejemplo clásico es el pronóstico del tiempo.

Mapreduce, por otro lado, es excelente en el manejo de cálculos relativamente pequeños e independientes sobre enormes cantidades de datos. Para hacer esto posible, los datos se distribuyen en muchas computadoras (debido a la cantidad de datos), y el cálculo deseado se divide en una fase que se puede realizar en cada bit de datos de forma independiente (la fase de "mapa"). Luego se recopilan los resultados de estos cálculos independientes y se realiza una segunda parte de los cálculos para combinar todos estos resultados individuales en el resultado final (la fase de "reducción").

Ejemplo: contando votos

Imagine que tiene una gran cantidad de votos para contar, y hay un poco de trabajo para contar

cada voto (p. Ej., Averiguando en la imagen escaneada qué casilla estaba marcada).

En este caso, una implementación mapreduce sería:

Paso 1: 'Spread'

Difunde las imágenes para procesar sobre las computadoras disponibles.

Paso 2: 'Mapa'

En cada computadora, para cada imagen:

- tomar en una de las imágenes copiadas a esta computadora como entrada
- averiguar qué casilla estaba marcada
- mostrar el número (o código o nombre) del artículo votado por

Tenga en cuenta que el trabajo puede comenzar tan pronto como una computadora obtiene 1 imagen para trabajar. No es necesario que todas estas computadoras interactúen para hacer su trabajo, por lo que no es necesario que se interconecten rápidamente, tengan memoria compartida o espacio en el disco compartido.

Paso 3: 'Reunir'

Reúne todas estas salidas en 1 computadora.

Paso 4: 'Reducir'

Cuente cuántos votos hay para cada número (o código o nombre).

Este ejemplo muy básico también destaca cómo a menudo son posibles otras optimizaciones. En este caso, el paso de reducción en sí mismo se puede hacer parcialmente en cada computadora, y luego se puede hacer una reducción final en una computadora central. Esto reducirá la cantidad de trabajo en la computadora que ejecuta el paso de reducción y limitará la cantidad de datos que se deben transportar a través de la red.

Ejemplo: conteo de votos - optimizado (mediante el uso del combinador)

Paso 1: 'Spread'

Igual que antes: distribuya las imágenes para procesarlas en las computadoras disponibles.

Paso 2: 'Mapa'

Igual que antes: en cada computadora, para cada imagen:

- tomar en una de las imágenes copiadas a esta computadora como entrada
- averiguar qué casilla estaba marcada
- mostrar el número (o código o nombre) del artículo votado por

Paso 3: 'Reunir' localmente

Reúna todas las salidas de 1 computadora en la computadora misma.

Paso 4: 'Reducir' localmente

Cuente cuántos votos de cada número (o código o nombre) hay en los resultados locales y produzca estos conteos.

Paso 5: 'Reunir' globalmente

Reúne todas las salidas de las reducciones locales en 1 computadora.

Paso 6: 'Reducir' globalmente

Resuma los conteos de votos hechos localmente de cada número (o código o nombre).

Tenga en cuenta que en el paso 3 no es *necesario* esperar **todos los** resultados en ninguno de los casos siguientes:

- Si esto se vuelve demasiado para los recursos locales de las computadoras como el almacenamiento / la memoria.
- si el costo del trabajo que se debe rehacer cuando una computadora se rompe se considera demasiado grande para esperar todos los resultados locales
- Si la red ahora es libre para transportar resultados intermedios.

La recopilación local y la reducción local se pueden realizar en los resultados producidos hasta ahora en la computadora local, y esto se puede hacer en cualquier momento.

El paso de reducción local se llama el paso combinador. Este es un paso opcional que se utiliza para mejorar el rendimiento.

Lea Empezando con mapreduce en línea:
https://riptutorial.com/es/mapreduce/topic/2192/empezando-con-mapreduce

Creditos

S. No	Capítulos	Contributors
1	Empezando con mapreduce	Ani Menon, Community, Legolas