Introduzione

Oggi giorno uno dei mercati più floridi è quello della vendita di dati in quanto molte compagnie li utilizzano per effettuare studi e previsioni di mercato, analisi dei gusti degli utenti, ecc. I maggiori acquisitori di dati sono senza dubbio i social network, come Twitter, Facebook, Instagram, visto il loro utilizzo massivo da parte degli utenti di tutto il mondo. Tuttavia, la compagnia che riesce ad acquisire il maggior numero di dati è Google, in quanto negli anni è diventato il motore di ricerca per eccellenza. L’analisi di questi dati ha portato alla realizzazione di nuovi approcci per poterli gestire in modo veloce, cosa che non si sarebbe riuscita a fare con le normali tecniche. Il modo più efficiente ad oggi è quella di analizzare i dati attraverso l’elaborazione parallela che permette di gestire problemi di grandi dimensioni in tempi relativamente brevi. Le elaborazioni parallele sono implementate distribuendo la computazione su un grande numero di computer.

Il nostro elaborato mira ad evidenziare la potenza del calcolo parallelo distribuito su reti di calcolatori confrontandolo con un’elaborazione locale sia in maniera sequenziale, sia sfruttando i thread. Le varie elaborazioni andranno ad analizzare un dataset formato da un insieme di tweet estratti dal social network Twitter.

Nei successivi capitoli introduttivi andremo a spiegare i big data e le tecnologie hadoop e spark. Successivamente verranno presentati i vari passi che hanno portato alla realizzazione di questo progetto.

Big Data

I Big Data sono una raccolta di dati grande e complessa così estesa in termini di volume, velocità e varietà da richiedere tecnologie e metodi analitici specifici per l'estrazione di valore o conoscenza. Con i big data la mole dei dati è dell'ordine degli zettabyte, ovvero miliardi di terabyte, quindi si richiede una potenza di calcolo parallelo e massivo con strumenti dedicati eseguiti su decine, centinaia o anche migliaia di server.

Le caratteristiche fondamentali dei Big Data possono essere descritte nelle tre “V”:

* Volume: capacità di acquisire, memorizzare ed accedere a dati di grandi dimensioni;
* Velocità: capacità di analizzare i dati in tempo reale o quasi;
* Varietà: riferito alle varie tipologie di dati che possono essere strutturate o non strutturate.

Hadoop

Apache Hadoop è un framework sotto una licenza libera che supporta applicazioni distribuite con elevato accesso ai dati, permettendo alle applicazioni di lavorare con migliaia di nodi e petabyte di dati. Ispirato dalla MapReduce di Google e dal Google File System, è un progetto Apache di alto livello costruito e usato da una comunità globale di contributori, che usano i linguaggi di programmazione Java e Scala.

Hadoop è composto da HDFS, MapReduce e YARN. Col passare del tempo si è costruito un ecosistema aggiungendo nuovi strumenti progettati appositamente per integrarsi con questi tre.

- HDFS

L'Hadoop Distributed File System (in sigla HDFS) è un file system distribuito, portabile e scalabile scritto in Java. Un cluster in Hadoop tipicamente possiede uno o più name node (su cui risiedono i metadati dei file) e un insieme di data node (su cui risiedono, in blocchi di dimensione fissa, i file dell'HDFS).

I file in HDFS sono suddivisi in blocchi e distribuiti tra i vari nodi del cluster mantenendo anche

delle repliche per garantire maggiore sicurezza.

Hadoop cerca di far eseguire a ciascun nodo i calcoli sui blocchi di file presenti nel proprio disco, in

questo modo si suddivide il carico di lavoro aumentando le prestazioni di calcolo e diminuendo

traffico di rete.

L’HDFS non permette operazioni di modifica su file, ma solo di creazione e cancellazione.

-MapReduce

Il MapReduce è stato ispirato dalle funzioni Map e Reduce utilizzate nella programmazione funzionale. Nella fase di Map i dati di input sono una serie di record che vengono elaborati singolarmente (da processi processi chiamati Mapper) in modo da restituire risultati che verranno successivamente aggregati (da processi chiamati Reducer).

Ogni singolo nodo possiede un blocco del file su cui esegue la sua parte di elaborazione.

-YARN

Introdotto in hadoop 2.0, YARN si impegna ad allocare le risorse a varie applicazioni in modo efficace. Esegue due demoni, che si occupano di due diverse attività:

* il gestore delle risorse, che esegue il monitoraggio dei lavori e l'allocazione delle risorse alle applicazioni
* il master dell'applicazione, che monitora lo stato di avanzamento dell'esecuzione.

Spark

Apache Spark è una piattaforma open source per l’elaborazione di analisi dei dati su larga scala, progettata per essere veloce e generica, tuttavia non include un sistema di gestione dei dati e pertanto viene solitamente distribuito su Hadoop o su altre piattaforme di archiviazione.

La sua caratteristica più importante è il suo cluster computing in memoria che è responsabile di aumentare la velocità di elaborazione dei dati.

Spark sfrutta il concetto di RDD, Resilient Distributed Dataset, che rappresenta una collezione immutabile e partizionata di dati su cui è possibile operare in parallelo.

Le proprietà chiave di un RDD sono:

* Ogni RDD è immutabile, cioè una volta creato non lo si può cambiare, se non creandone un altro mediante una trasformazione;
* Ogni RDD può solo essere creato inizialmente a partire dai dati su disco (presi da HDFS) oppure a partire da altri RDD;
* Le trasformazioni possibili per creare nuovi RDD sono poche, deterministiche e ripetibili: si può mapparli (cioè trasformarli da un array di chiave-valore a un altro array di chiave-valore), filtrarli (partire da un array e crearne un altro filtrando i dati), unire due RDD;
* Questo approccio serve perchè un singolo pezzo (partizione) del RDD possa essere ricostruito a partire dalla sequenza di trasformazioni che lo hanno generato;
* Ogni RDD può restare in memoria oppure essere materializzato su disco, a scelta del programmatore (RDD in memoria ma inutilizzati da tempo vengono comunque automaticamente scaricati su disco dal processo di gestione che gira sul worker node);
* Ogni RDD è descritto da un set completo di metadati che consentono la ricostruzione di una delle sue partizioni in caso di fault: dove si trovano le partizioni, quali sono gli RDD padre, quale è la sequenza di trasformazioni, detta LINEAGE, che lo hanno generato.

Spark nasce come un sistema per creare e gestire job di analisi basati su trasformazioni di RDD. Dato che gli RDD nascono e vivono in memoria, l’esecuzione di lavori iterativi o che trasformano più volte un set di dati, sono immensamente più rapide di una sequanza di map-reduce. 10, anche 100 volte più veloci, perchè il disco non viene mai (o quasi mai) impiegato nell’elaborazione.

Un’altra caratteristica è che supporta più linguaggi fornendo API scritte in Scala, Java, Python o R rendendolo così molto flessibile.