Università degli Studi di Salerno
Dipartimento di Informatica

Dottorato di Ricerca in Informatica
(Ciclo XIV - Nuova Serie)

Tesi di Dottorato in Informatica

The Role of Distributed Computing in Big Data Science: Case Studies in Forensics and Bioinformatics

Abstract

Candidato:
Gianluca Roscigno
Matr.: 8880900108

Tutor:
Prof. Giuseppe Cattaneo

Coordinatore: Prof. Gennaro Costagliola

A.A. 2014/2015

Sommario

L'era dei "Big Data" sta dando vita alla generazione di grandi quantità di dati, che richiedono capacità di memorizzazione e di analisi le quali possono essere indirizzate solo dai sistemi di computazione distribuita. Per facilitare la computazione distribuita su larga scala, sono stati proposti molti paradigmi di programmazione e soluzioni, come MapReduce e Apache Hadoop, che in modo trasparente risolvono alcuni problemi relativi ai sistemi distribuiti e nascondono molti dei loro dettagli tecnici.

Hadoop è attualmente il più popolare e maturo framework che supporta il paradigma MapReduce, ed è ampiamente usato per memorizzare e processare grosse quantità di dati adoperando un insieme di computer. Le soluzioni come Hadoop sono attraenti poiché semplificano la "trasformazione" di un'applicazione non parallela a quella distribuita, adoperando strumenti generali e senza richiedere molte competenze. Tuttavia, senza qualsiasi attività di ingegnerizzazione degli algoritmi, alcune applicazioni realizzate non sono del tutto veloci ed efficienti, e possono soffrire di diversi problemi ed inconvenienti quando sono eseguite su un sistema distribuito. Infatti, un'implementazione distribuita è una condizione necessaria ma non sufficiente per ottenere prestazioni notevoli rispetto ad una controparte non parallela. Quindi, è necessario valutare come le soluzioni distribuite vengono eseguite su un cluster Hadoop, e/o come le loro prestazioni possono essere migliorate per ridurre il consumo delle risorse e i tempi di completamento.

In questa tesi mostreremo come le implementazioni basate su Hadoop possono essere migliorate utilizzando attentamente le attività di ingegnerizzazione degli algoritmi, di messa a punto, di profilazione e i

miglioramenti al codice. È anche analizzato come è possibile raggiungere questi obiettivi lavorando su alcuni *punti cruciali*, tali come: la computazione locale ai dati, la dimensione della partizione di input, il numero e la granularità dei sotto-problemi, la configurazione del cluster, la rappresentazione dell'input/output, etc.

In particolare, per indirizzare queste questioni, noi scegliamo alcuni casi di studio provenienti da due aree di ricerca dove il problema del grande ammontare dei dati sta crescendo, ossia *Digital Image Forensics* e *Bioinformatica*. Noi descriviamo principalmente vere e proprie implementazioni per mostrare come progettare, ingegnerizzare, migliorare e valutare soluzioni basate su Hadoop per il problema della *Source Camera Identification*, cioè il riconoscimento della fotocamera usata per scattare una data immagine digitale, utilizzando l'algoritmo di *Fridrich et al.*, e per due dei principali problemi in Bioinformatica, ovverosia il *confronto senza allineamento delle sequenze* e l'estrazione delle *statistiche globali o locali dei k-meri*.

I risultati ottenuti dalle nostre implementazioni migliorate mostrano che esse sono sostanzialmente più veloci delle corrispondenti applicazioni non parallele, e notevolmente più veloci rispetto alle corrispondenti semplici implementazioni basate su Hadoop. In alcuni casi, per esempio, la nostra soluzione per le statistiche dei k-meri è all'incirca 30 volte più veloce rispetto alla nostra semplice implementazione basata su Hadoop, e circa 40 volte più veloce rispetto ad un analogo strumento costruito su Hadoop. Inoltre, le nostre applicazioni sono anche *scalabili*, ossia i tempi d'esecuzione sono (approssimativamente) dimezzati quando si raddoppiano le unità di computazione. Infatti, le attività di ingegnerizzazione degli algoritmi basate sull'implementazione di miglioramenti astuti, e coadiuvate da accurate attività di profilazione e messa a punto, possono portare a migliori risultati sperimentali, evitando possibili problemi.

Noi anche evidenziamo come i miglioramenti, i consigli, gli stratagemmi e gli approfondimenti proposti possono essere adoperatati in altre

aree di ricerca e problemi.

Sebbene Hadoop semplifica alcune attività degli ambienti distribuiti, dobbiamo accuratamente conoscerlo per raggiungere prestazioni degne di nota. Non è sufficiente essere un esperto del dominio applicativo per costruire implementazioni basate su Hadoop, infatti, al fine di ottenere buone prestazioni, un esperto di sistemi distribuiti, d'ingegneria degli algoritmi, di messa a punto, di profilazione, etc. è anche richiesto. Quindi, le migliori prestazioni dipendono pesantemente dal grado di cooperazione tra l'esperto di dominio e l'ingegnere degli algoritmi distribuiti.