POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

A Design-time Optimization Framework for Private Cloud Big Data Systems

Relatore:
Prof. Danilo ARDAGNA — Politecnico di Milano
Correlatori:
Dott. Michele CIAVOTTA — Politecnico di Milano
Dott. Eugenio GIANNITI — Politecnico di Milano

Tesi Magistrale di:
Jacopo RIGOLI
Matricola 817279

Anno Accademico 2015-2016

Contents

List of Figures

List of Tables

Abstract

Nowadays, more and more enterprises exploit Big Data technologies in order to extract knowledge from huge datasets. Applications can encompass business analytics, can support decision-making, log-analysis, machine learning. Hence, a reliable and scalable computing solution is highly required for dealing with the quantity and the complexity of the information to process. Hadoop is an already widespread technology, implementing the MapReduce computational model and featuring a resource negotiator, as well as a distributed file system, thus providing a highly scalable parallel computation, ensuring also distribution, fault-tolerance, reliability, and monitoring. Tez is a framework built on top of Hadoop that allows to build applications by setting their workflows as direct acyclic graphs. Even if it is an extremely important task, still there are no tools that fully support developers in activities related to the capacity planning of MapReduce or Tez applications. In order to minimize costs and fulfill the required number of concurrent users, it is important to properly set computational resources to avoid wastes and to guarantee a certain level of performance. For this reason, this thesis proposes a tool to optimize the allocation of resources and to guarantee that execution times of both MapReduce and Tez tasks respect estabilished deadlines. Having such tool available at design-time enables operators to make more informed decisions about the technology to use and to fully exploit the potential offered by the Cloud infrastructure. The core of this work is the development of a greedy algorithm to optimize the allocation of resources and the overall cluster cost in private cloud scenarios. It exploits analytical models to evaluate execution time of MapReduce and Tez jobs and is implemented in three steps: an approximated solution obtained by a machine learning technique is initially obtained. Then, a greedy search optimizes the capacity allocation for individual applications by performing the minimum number of analytical simulations. Finally, a mixed integer model identifies the cluster of minimum costs and the VMs to physical machine assignment.

Sommario

Oggigiorno, sempre più aziende sfruttano tecnologie Big Data per estrarre conoscenza da enormi dataset. Le applicazioni includono business-analysis in grado di supportare il processo decisionale, log-analysis, il machine learning. Una soluzione di elaborazione altamente affidabile e scalabile è necessaria per elaborare grandi quantità di informazioni complesse. Hadoop è una tecnologia già diffusa che implementa il modello di computazione MapReduce, un negoziatore di risorse e un file system distribuito, fornendo così computazioni parallelizzabili, scalabili, assicurando anche distribuzione, fault-tolerance, affidabilità e monitoraggio. Tez è un framework costruito su Hadoop che permette di creare applicazioni basate su direct acyclic graph. Nonostante sia un aspetto estremamente importante, ancora non ci sono strumenti che supportano pienamente gli sviluppatori in attività legate alla pianificazione delle risorse da assegnare alle applicazioni MapReduce o Tez. Per minimizzare i costi e supportare un numero di utenti concorrenti, è importante impostare correttamente le risorse, evitando sprechi e garantendo un certo livello di prestazioni. Per questo motivo, questa tesi propone uno strumento per ottimizzare l'allocazione delle risorse e per garantire che i tempi di esecuzione di task MapReduce e Tez rispettino le scadenze impostate. Tale strumento, disponibile in fase di progettazione, consente agli operatori di prendere decisioni più consapevoli sulla tecnologia da utilizzare e per sfruttare appieno le potenzialità offerte dalle infrastrutture Cloud. Questo lavoro si basa sullo sviluppo di un algoritmo greedy per ottimizzare l'allocazione delle risorse e il costo globale dei cluster in scenari cloud privato, sfruttando modelli analitici per valutare il tempo di esecuzione dei task MapReduce e Tez. Inizialmente, si ottiene una soluzione approssimata con una tecnica di machine learning. Poi, una ricerca greedy ottimizza l'assegnazione di capacità alle varie applicazioni, eseguendo il numero minimo di simulazioni analitiche. Infine, un mixed integer model identifica il cluster di costo minimo e l'assegnazione delle macchine virtuali su quelle fisiche.

CHAPTER 1

Introduction

In recent years, data has grown in an explosive way. Every day, 2.5 quintillion bytes of data are created. This is so much that the 90 percent of the data in the world today were produced within the past two years. That is why the embracement of Big Data is steadily increasing, it has moved from an experimental projects to mission-critical, enterprise-wide deployments providing new insights, competitive advantage, and business innovation [30]. This ever-increasing number of applications range from fraud detection to one-to-one marketing, encompassing business analytics and support to decision making both in private and public sectors. In order to cope with this unprecedented amount of data that many companies need to process in a timely way, new technologies are increasingly adopted by the industry, following the *Big Data* paradigm.

From the technological perspective, MapReduce programming model has been for many years the de-facto standard in this field. Indeed, it is capable of analyzing in a very efficient way large amounts of unstructured data, thus resulting a viable solution to support both the variety and volume requirements of Big Data operations [34]. MapReduce has been adopted in multiple application domains, e.g., machine learning, log analysis, and data mining [58]. The most common open source implementation of this computational model, Apache Hadoop, is able to manage large datasets allowing other applications, built on the top of it, to efficiently elaborate even Petabytes of data. Further improvements have been introduced with the Apache Tez framework, in which job tasks are not forced to follow a strict two-phase order, but tasks can form an arbitrary complex Directed Acyclic Graph (DAG) and the are no longer I/O barriers in intermediate computations.

Performance prediction of Big Data applications is extremely important, e.g., for correctly planning the required size a cluster (either physical or in the public cloud) must have to handle a certain workload. However, the adoption of Hadoop and other Big Data technologies is complex. The deployment and setup of an implementation is time-consuming, expensive, and resource-intensive. Nowadays, this problem is getting even worse due to recent Hadoop and Tez enhancements that increases the situation complexity. Thus, modeling performance of computational paradigms newer than MapReduce, e.g., Apache Tez, become extremely challenging.

The focus in this thesis is to provide a software tool able to support system administrators and operators in the capacity planning process of shared Hadoop 2.x in private clusters, supporting applications with deadline guarantees and a number of concurrent users. Having such tool available at design-time enables operators to make more informed decisions about the technology to use and to fully exploit the potential offered by the Cloud infrastructure.

This thesis is organized as follows. Chapter 2 will discuss the state of the art, giving an overview of the technologies under study, in particular Cloud computing, Hadoop, Tez, MapReduce and its job performance models. Next, Chapter 3 is going to show how we developed mathematical programming models to solve the capacity planning problem in a Private Cloud cluster. After that, in Chapter 4, we will show the architecture of the implented tool. In Chapter 5, an exploration of the characteristics of the proposed models is given. In the end, Chapter 6 will wrap up this work and draws conclusions on the outcomes. Furthermore, it points out relevant issues that remain open and will be the focus of future work.

CHAPTER 6

Conclusions and Future Work

Throughout this thesis we investigated the capacity planning problem for MapReduce and Tez applications running on Hadoop private cloud clusters. After exploring the technical aspects related to Cloud computing, MapReduce, Apache Hadoop, Tez as well as the application of the SVR Machine Learning, we focused on the development of a tool that could be exploited for solving the capacity allocation problem in a scalable way. Then, the solution methods were validated through empirical analyses.

In particular, this work proposed a tool able to optimize the allocation of resources and to guarantee the execution times of both MapReduce and Tez jobs with respect to established deadlines. The core of this tool is a greedy algorithm able to optimize the allocation of resources and the overall cluster cost exploiting accurate analytical models. Having such tool available at design-time enables operators to make more informed decisions about the technology to use and to fully exploit the potential offered by private cloud infrastructures.

Building upon the outcomes of this work, it is possible to investigate further open issues and relevant research questions.

The greedy search technique allows to obtain very accurate predictions by simulating the whole Hadoop system, at the price of long execution times. The tool shown a promising direction involving the combination of analytical models (QN,SWN) and machine learning techniques (SVR). Through a further use of the existing expensive models it is possible to generate lots of data that can be used in order to further improve the training set of the SVR, drastically reducing execution times.

Since the provided tool supports the capacity allocation problem for the execution of complex DAGs, other big data frameworks that take advantage from them can be investigated. Examples of such frameworks that are attracting increasing attention in the industry are Spark and Storm.

Bibliography

[1] *Amazon EC2 Pricing*. URL: http://aws.amazon.com/ec2/pricing/ (visited on 03/17/2015).

[2] *Amazon EC2 Pricing*. URL: http://aws.amazon.com/ec2/pricing/ (visited on 07/16/2016).

[3] *Amazon Elastic Compute Cloud (EC2)*. URL: http://aws.amazon.com/ec2.

[4] *Amazon Elastic MapReduce*. URL: https://aws.amazon.com/elasticmapreduce/ (visited on 08/30/2016).

[5] *Amazon Simple Storage Service*. URL: https://aws.amazon.com/s3/ (visited on 09/17/2016).

[6] *Amazon Web Services*. URL: http://aws.amazon.com/ (visited on 03/17/2015).

[7] *Apache Hadoop*. URL: http://hadoop.apache.org (visited on 10/22/2016).

[8] *Apache Hive*. URL: https://hive.apache.org (visited on 10/22/2016).

[9] *Apache Tez*. URL: http://tez.apache.org/ (visited on 08/29/2016).

[10] D. Ardagna, S. Bernardi, E. Gianniti, S. Karimian Aliabadi, D. Perez-Palacin, and J. I. Requeno. "Modeling Performance of Hadoop Applications: A Journey from Queueing Networks to Stochastic Well Formed Nets". In: *ICA3PP*. (Granada, Spain). Dec. 2016. In press.

[11] S. Baarir, M. Beccuti, D. Cerotti, M. De Pierro, S. Donatelli, and G. Franceschinis. "The GreatSPN Tool: Recent Enhancements". In: *ACM SIGMETRICS Performance Evaluation Review* 36.4 (2009), pp. 4–9.

[12] S. Bardhan and D. A. Menascé. "Queuing Network Models to Predict the Completion Time of the Map Phase of MapReduce Jobs". In: *Proc. International Computer Measurement Group Conf.* (Las Vegas, NV). Dec. 2012.

[13] J. Barr. *Host Your Web Site in the Cloud: Amazon Web Services Made Easy*. 1st ed. Sitepoint, 2010. isbn: 978-0980576832.

[14] M. Bertoli, G. Casale, and G. Serazzi. "JMT: Performance Engineering Tools for System Modeling". In: *SIGMETRICS Perform. Eval. Rev.* 36.4 (2009), pp. 10–15. issn: 0163-5999. doi: 10.1145/1530873.1530877.

[15] R. H. Byrd, J. Nocedal, and R. A. Waltz. "Knitro: An Integrated Package for Nonlinear Optimization". In: *Energy*. Vol. 83. 2006, pp. 35–59. isbn: 978-0-387-30065-8. doi: 10.1007/0-387-30065-1_4. arXiv: arXiv:1011.1669v3. url: http://www.ziena.com/papers/integratedpackage.pdf$%5Cbackslash$nhttp://link.springer.com/10.1007/0-387-30065-1%7B%5C_%7D4.

[16] I. Carrera, F. Scariot, C. Geyer, and P. Turin. *An Example for Performance Prediction for Map Reduce Applications in Cloud Environments*.

[17] J. Dean and S. Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters". In: *6th Symposium on Operating Systems Design and Implementation*. 2004, pp. 137–149.

[18] *Flexyscale*. url: http://www.flexiscale.com/ (visited on 03/17/2016).

[19] R. Fourer, D. M. Gay, M. Hill, B. W. Kernighan, and T. B. Laboratories. "AMPL: A Mathematical Programming Language". In: *Management Science* 36 (1990), pp. 519–554. issn: 01716468. doi: 10.1007/BF01783416.

[20] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990. isbn: 0716710455.

[21] E. Gianniti. "Game theory models for MapReduce : joint admission control and capacity allocation". MA thesis. Politecnico di Milano, 2015. url: http://hdl.handle.net/10589/107281 (visited on 11/15/2016).

[22] *GoGrid*. url: http://www.gogrid.com/ (visited on 03/17/2016).

[23] *Google App Engine*. url: https://developers.google.com/appengine/ (visited on 03/17/2016).

[24] *Google Apps for Work*. url: http://www.google.com/enterprise/apps/business/ (visited on 03/17/2016).

[25] *Google Compute Engine*. URL: https://cloud.google.com/products/compute-engine (visited on 03/17/2016).

[26] *Google Inc.* URL: http://www.google.com/about/company/ (visited on 03/17/2015).

[27] *Hadoop MapReduce Next Generation - Capacity Scheduler*. URL: http://hadoop.apache.org/docs/r2.3.0/hadoop-yarn/hadoop-yarn-site/CapacityScheduler.html (visited on 10/22/2016).

[28] *Hadoop MapReduce Next Generation — Capacity Scheduler*. URL: http://hortonworks.com/blog/benchmarking-apache-hive-13-enterprise-hadoop/.

[29] H. Herodotou. *Hadoop Performance Models*. Tech. rep. Duke University, June 6, 2011. arXiv: 1106.0940 [cs.DC].

[30] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi. "Big Data and Its Technical Challenges". In: *Commun. ACM* 57.7 (July 2014), pp. 86–94. ISSN: 0001-0782.

[31] K. Kambatla, G. Kollias, V. Kumar, and A. Grama. "Trends in Big Data Analytics". In: *Journal of Parallel and Distributed Computing* 74.7 (2014), pp. 2561–2573. ISSN: 0743-7315. DOI: 10.1016/j.jpdc.2014.01.003. URL: http://www.sciencedirect.com/science/article/pii/S0743731514000057.

[32] M. Khan, Y. Jin, M. Li, Y. Xiang, and C. Jiang. "Hadoop Performance Modeling for Job Estimation and Resource Provisioning". In: *IEEE Trans. Parallel Distrib. Syst.* 27.2 (Feb. 2016), pp. 441–454. ISSN: 1045-9219. DOI: 10.1109/TPDS.2015.2405552. URL: http://dx.doi.org/10.1109/TPDS.2015.2405552.

[33] *KVM*. URL: http://www.linux-kvm.org/ (visited on 03/17/2016).

[34] D. Laney. *3D Data Management: Controlling Data Volume, Velocity, and Variety*. Tech. rep. META Group, 2012.

[35] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik. *Quantitative System Performance. Computer System Analysis Using Queueing Network Models*. Prentice-Hall, 1984. URL: http://homes.cs.washington.edu/~lazowska/qsp/ (visited on 04/07/2016).

[36] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon. "Parallel Data Processing with MapReduce: A Survey". In: *SIGMOD Rec.* 40.4 (Jan. 2012), pp. 11–20. ISSN: 0163-5808. DOI: 10.1145/2094114.2094118.

[37] D.-R. Liang and S. K. Tripathi. “On Performance Prediction of Parallel Computations with Precedent Constraints”. In: *IEEE Transactions on Parallel and Distributed Systems* 11.5 (May 2000), pp. 491–508.

[38] M. Lin, L. Zhang, A. Wierman, and J. Tan. “Joint Optimization of Overlapping Phases in MapReduce”. In: *SIGMETRICS Performance Evaluation Review* 41.3 (2013), pp. 16–18. DOI: 10.1145/2567529.2567534.

[39] X. Lin, Z. Meng, C. Xu, and M. Wang. “A Practical Performance Model for Hadoop MapReduce”. In: *International Conference on Cluster Computing Workshops*. IEEE. 2012. DOI: 10.1109/ClusterW.2012.24.

[40] M. Malekimajd, D. Ardagna, M. Ciavotta, A. M. Rizzi, and M. Passacantando. “Optimal Map Reduce Job Capacity Allocation in Cloud Systems”. In: *SIGMETRICS Perform. Eval. Rev.* 42.4 (June 2015), pp. 51–61. ISSN: 0163-5999. DOI: 10.1145/2788402.2788410.

[41] M. Malekimajd, A. M. Rizzi, D. Ardagna, M. Ciavotta, M. Passacantando, and A. Movaghar. “Optimal Capacity Allocation for Executing MapReduce Jobs in Cloud Systems”. In: *Proceedings of the 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*. (Timisoara, Romania). 2014, pp. 385–392.

[42] P. Mell and T. Grance. *The NIST Definition of Cloud Computing*. Tech. rep. July 10, 2009. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (visited on 03/17/2016).

[43] *Microsoft Azure Storage*. URL: http://azure.microsoft.com/en-us/services/storage/ (visited on 09/17/2016).

[44] *Microsoft Corporation*. URL: http://www.microsoft.com/ (visited on 03/17/2016).

[45] *Microsoft HDInsight*. URL: http://azure.microsoft.com/en-us/services/hdinsight/ (visited on 08/30/2016).

[46] *Microsoft Office 365*. URL: http://office365.microsoft.com/ (visited on 03/17/2016).

[47] *Microsoft Windows Azure*. URL: http://www.windowsazure.com/ (visited on 03/17/2016).

[48] *Microsoft Windows Azure Virtual Machines*. URL: http://www.windowsazure.com/en-us/home/features/virtual-machines/ (visited on 03/17/2016).

[49] *Onlive*. URL: http://www.onlive.com/ (visited on 03/17/2016).

[50] *Rackspace*. URL: http://www.rackspace.com/ (visited on 03/17/2016).

[51] *Salesforce*. URL: http://www.force.com/ (visited on 03/17/2016).

[52] *SAP*. URL: http://www.sap.com/ (visited on 03/17/2016).

[53] C. Shanklin. *Benchmarking Apache Hive 13 for Enterprise Hadoop*. URL: https://hadoop.apache.org/docs/r2.4.1/hadoop-yarn/hadoop-yarn-site/CapacityScheduler.html.

[54] *State of the Cloud Report*. URL: http://www.mcit.gov.eg/Upcont/Documents/Reports%20and%20Documents_1252016000_RightScale-2016-State-of-the-Cloud-Report.pdf (visited on 10/22/2016).

[55] *TCP-DS Benchmark*. URL: http://www.tpc.org/tpcds/ (visited on 03/17/2016).

[56] *The Digital Universe in 2020*. URL: http://idcdocserv.com/1414.

[57] A. Verma, L. Cherkasova, and R. H. Campbell. "ARIA: Automatic Resource Inference and Allocation for MapReduce Environments". In: *Proceedings of the Eighth International Conference on Autonomic Computing*. June 2011.

[58] E. Vianna, G. Comarela, T. Pontes, J. M. Almeida, V. A. F. Almeida, K. Wilkinson, H. A. Kuno, and U. Dayal. "Analytical Performance Models for MapReduce Workloads". In: *International Journal of Parallel Programming* 41.4 (2013), pp. 495–525.

[59] E. Vianna, G. Comarela, T. Pontes, J. Almeida, V. Almeida, K. Wilkinson, H. Kumo, and U. Dayal. "Analytical Performance Models for MapReduce Workloads". In: *Int J Parallel Prog* 41 (2013), pp. 495–525. DOI: 10.1007/s10766-012-0227-4.

[60] E. Vianna, G. Comarela, T. Pontes, J. Almeida, V. Almeida, K. Wilkinson, H. Kumo, and U. Dayal. "Modeling the Performance of the Hadoop Online Prototype". In: *23rd International Symposium on Computer Architecture and High Performance Computing*. IEEE. 2011, pp. 152–159. DOI: 10.1109/SBAC-PAD.2011.24.

[61] *VMware Inc.* URL: http://www.vmware.com/ (visited on 03/17/2016).

[62] X. Wu, Y. Liu, and I. Gorton. "Exploring Performance Models of Hadoop Applications on Cloud Architecture". In: *Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures*. QoSA '15. Montréal, QC, Canada: ACM, 2015, pp. 93–101. ISBN: 978-1-4503-3470-9. DOI: 10.1145/2737182.2737197. URL: http://doi.acm.org/10.1145/2737182.2737197.

[63] *Xen Hypervisor*. URL: http : / / www . xen . org/ (visited on 03/17/2016).

[64] F. Yan, L. Cherkasova, Z. Zhang, and E. Smirni. "Optimizing Power and Performance Trade-offs of MapReduce Job Processing with Heterogeneous Multi-core Processors". In: *CLOUD*. 2014. DOI: 10.1109/CLOUD.2014.41.

[65] X. Yang and J. Sun. "An Analytical Performance Model of MapReduce". In: *CCIS*. IEEE. 2011.

[66] Q. Zhang, L. Cheng, and R. Boutaba. "Cloud Computing: State-of-the-Art and Research Challenges". In: *J. Internet Services and Applications* 1.1 (2010), pp. 7–18. DOI: 10.1007/s13174-010-0007-6.

[67] Z. Zhang, L. Cherkasova, and B. T. Loo. "Parameterizable Benchmarking Framework for Designing a MapReduce Performance Model". In: *Concurrency Computat.: Pract. Exper.* (2014). DOI: 10.1002/cpe.3229.