POLITECNICO
MILANO 1863

Department of Electronics, Informatics and Bioengineering
M.Sc. course of Computer Science and Engineering

A Hybrid Machine Learning Approach for Big Data Performance Evaluation

Supervisor:
Prof. Danilo Ardagna — Politecnico di Milano
Co-Supervisors:
Dr. Eugenio Gianniti — Politecnico di Milano
Dr. Marco Lattuada — Politecnico di Milano

Master of Science Thesis of:
Vahid Heidari
Matriculation ID 850973

April 2018

Contents

List of Figures

List of Tables

Abstract

Nowadays running big data applications on clouds are growing rapidly and have become critical for almost every industry. Because of that, it is often important to predict with fair confidence the execution time of submitted applications, for instance when service level agreements (SLAs) are established with end-users. In other words, users may want to determine how jobs execution time changes when the available cloud resources (in terms of, e.g., virtual machines type and their number) change. However, running experiments in real cloud environments is generally expensive and time consuming. Therefore, exploiting a reasonably accurate model for performance evaluation and prediction of cloud applications has a great importance. Performance prediction models are extremely useful to aid development and deployment of big data applications; either for design time decisions or run time system reconfiguration.

Map Reduce framework became one of the most popular platforms for data analytics. Apache Spark extends the MapReduce model and using Resilient distributed Datasets (RDDs) and Directed Acyclic Graphs (DAGs) empowers Spark to be a fast and general-purpose big data computing platform.

One approach for performance prediction is to develop white box analytical models based on DAG simulators or approximation formulas for predicting performance metrics. Another approach is machine learning based techniques which embody the black box, and infer performance models based on the relations among the input and output variables of a system that are observed during an initial training phase.

This thesis, validates and optimizes a hybrid (also known as gray box) approach for performance prediction of big data applications running on clouds, which exploits both analytical modeling and machine learning techniques and it is able to achieve a good accuracy without too many time consuming and costly experiments on a real setup.

Sommario

Attualmente, le applicazioni big data su cloud stanno avanzando rapidamente, diventando di cruciale importanza per quasi tutti i settori industriali. Per questa ragione, risulta spesso essenziale prevedere con una certa accuratezza i tempi di esecuzione delle applicazioni lanciate, per esempio quando gli SLA (Service Level Agreement) vengono stabiliti con gli end-users. In altre parole, gli utenti vorrebbero poter determinare quanto varia il tempo di esecuzione quando le risorse disponibili su cloud variano (in termini per esempio di tipologia e numero di macchine virtuali). Tuttavia, eseguire esperimenti in real cloud environment è di solito costoso e richiede un elevato consumo di tempo. Di conseguenza, l'utilizzo di modelli ragionevolmente accurati per la valutazione e la predizione delle prestazioni delle applicazioni cloud è di fondamentale importanza. I modelli di predizione delle prestazioni sono estremamente utili per incoraggiare lo sviluppo e la diffusione di applicazioni big data e anche per le decisioni sul tempo di progettazione o per la riconfigurazione del run time system. Il framework Map Reduce è diventato una delle piattaforme più utilizzate per l'analisi dei dati. Apache Spark estende il modello MapReduce, l'utilizzo di RDD (Resilient distributed Datasets) e di DAG (Directed Acyclic Graphs) e consente a Spark di essere una piattaforma di elaborazione di big data rapida e generale.
Un possibile approccio per la predizione delle prestazioni consiste nello sviluppo di modelli analitici white box basati su simulatori DAG o su formule di approssimazione per la predizione delle metrics performance. Un altro approccio è rappresentato dal machine learning che incorpora la black box e deduce modelli di prestazione basati sulle relazioni tra variabili di input ed output di un sistema che vengono osservate in una fase di training iniziale. Questa tesi convalida ed ottimizza un approccio ibrido (noto anche come gray box) per la predizione delle performance delle applicazioni big data che girano su cloud, che impiega sia modelli analitici che tecniche di machine learning ed è in grado di raggiungere una buona accuratezza senza impiegare troppo tempo o effettuare costosi esperimenti su un setup reale.

CHAPTER 1

Introduction

Big Data is revolutionizing all aspects of our lives ranging from enterprises to consumers, from science to government. [26]

In this context, the MapReduce (MR) framework became the most popular platform [11] for data analytics because of its simplicity, generality, and maturity [48]. Apache Spark, on the other hand, provides a user friendly programming interface to decrease coding efforts and provide better performance in a majority of the cases with problems related to big data.
Apache Spark started as a research project at UC Berkeley in the AMPLab, with the goal to design a programming model that supports a much wider class of applications than MapReduce, while maintaining its automatic fault tolerance.
Spark offers an abstraction called Resilient distributed Datasets (RDDs)[46] a distributed memory abstraction that lets programmers perform in memory computations on large clusters in a fault-tolerant manner. In practice, Spark can easily obtain a 10x speedup over Hadoop on specific scenarios [46] and it is the most promising framework that will probably support the execution of big data applications for the next 5–10 years [14].

Clouds are cost-effective platforms to support big data systems, as resources (e.g., nodes) can be allocated and deallocated on demand, in response to the applications requirements and QoS needs.

In this context, one of the main challenges [30, 44] is that the execution time of big data applications is generally unknown in advance. Because of this, performance analysis is usually done empirically through experimentation, requiring a costly setup [21].

In addition, big data systems are becoming a central force in society, thus requiring the development of intelligent systems, which provide QoS

guarantees to their users. Performance prediction models are extremely useful to aid development and deployment of big data applications; either for design time decisions or run time system reconfiguration. Design time models can help, e.g., to determine the appropriate size of a cluster or to predict the budget required to run Hadoop/Spark in public clouds. Such models can be used also at run time, allowing a dynamic adjustment of the system configuration [4, 34], e.g., to cope with workload fluctuations or to reduce energy costs.

One approach for performance prediction is to develop white box analytical models (AMs) based on DAG simulators, queueing networks (QNs), Petri nets (PNs) and so on for predicting performance metrics. However, analytically modeling big data applications is very challenging, since this requires a deep knowledge of the system behavior. Moreover, in a cloud context performance prediction is even more complex, since applications execution time may also vary as a consequence of resource contention and performance degradation introduced by the underlying virtualization layer [8].

To ensure AM tractability in such complex systems, AM-based performance models typically rely on simplifying assumptions that reduce their accuracy. On the other hand, black box machine learnings (MLs) deal with the study and construction of algorithms that can learn from data and make predictions on it without a priori knowledge about the internals of the target system. In recent years, a growing number of successful researches were done to explore the possibility of using ML techniques to predict performance of complex computer systems [24, 28, 45]. Therefore, ML can also be exploited for predicting the execution time of Spark jobs. To be able to predict accurately, ML models should be built during a training phase with a sufficient amount of experimental data from different workloads, using different parameters and configurations. However, running several experiments in cloud environments would be costly and time consuming. Though ML often provides good accuracy in regions for which it is well trained, it shows poor precision in regions for which none or very few samples are known.

Hybrid machine learning, which can be considered as a gray box modeling technique [6, 16, 25, 38], is a new approach for performance prediction that tries to achieve the best of the AM and ML worlds by mixing the two. Such models can be exploited to support design-time decision-making during the development and deployment phases of big data applications. These models can then be also kept alive at run-time to conduct the dynamic adjustment of the system configuration [34].

Focus of this thesis is to provide a design time combined AM/ML model to estimate big data applications execution time built on Spark, running

in cloud clusters. At first, AMs are used to initially model the response time of the big data application and initiate some synthetic samples. This analytical data is used to train an initial ML model. During an iterative and incremental process, new data from the operational system is fed into the ML model to build a more accurate performance predictor. In addition, some intuitions are exploited to provide more accurate predictions while consuming less data from the operational systems, which, due to resource contention, might be affected by noise.

The accuracy of the models is evaluated on real systems by performing experiments based on the TPC-DS industry benchmark for business intelligence data warehouse applications. The Italian supercomputing center, CINECA, and the Microsoft Azure HDInsight [1] data platform has been considered as target deployment.

The proposed hybrid approach (HML) is compared against two ML-based techniques, and a graybox technique proposed by Didona et al. [18] in terms of different metrics defined to indicate prediction accuracy.

Using different regression techniques and hyper parameter optimization, HML outperforms baseline machine learning techniques, and provides better extrapolation and interpolation capabilities.
HML also surpasses analytical models which usually have acceptable extrapolation, by achieving more accurate results.
In comparison to iterative machine learning (IML), beside surpassing in extrapolation and interpolation capabilities, HML significantly reduces the number of training samples gathered from the operational system.
Comparing to other hybrid machine learning approaches, i.e., Didona et al.'s work [17], based on "No Free Lunch" theorem we cannot declare there is a clear winner in terms of accuracy or number of data samples required to build a model, but HML demonstrates to be more robust to the choice of the underling ML model and easier to be fine-tuned.

This thesis is organized as follows:
Chapter 2 outlines the state of the art. Chapter 3 presents the hybrid machine learning approach for performance evaluation of big data applications. In the Chapter 4 implementation library of the proposed approach is described. Experimental results are presented in Chapter 5. And finally conclusions are drawn in Chapter 6.

[1] https://azure.microsoft.com/en-us/services/hdinsight/

CHAPTER 6

Conclusions and Future Work

This thesis, proposed, validated and optimized a novel hybrid machine learning algorithm (HML), which is able to use analytical model (AM) and machine learning (ML) in synergy to model and predict the execution time of jobs running on the most widely used big data frameworks such as Spark.

With respect to state of the art work, HML is particularly effective to predict the performance of big data applications when shared physical environments characterized by high resource contention are considered.

The proposed approach can use different regression techniques and performs hyper parameter optimization in all of the used ML techniques.
The results from this thesis implies that HML outperforms baseline machine learning techniques, providing better extrapolation and interpolation capabilities.
HML provides a powerful methodology for a better approximation of Big Data job execution time on cloud environment, these approximations are more accurate than analytical models which usually have acceptable extrapolation capabilities.
In comparison to iterative machine learning (IML), findings of this thesis indicate that HML reduces significantly the number of training samples to be gathered from the operational system. In most of the experiments, HML gathered only one sample while IML was reaching the maximum number of the iterations.
Comparing to other hybrid machine learning approaches, i.e., Didona et al.'s work [17], we cannot declare there is a clear winner in terms of accuracy or number of data samples required to build a model, but our approach

demonstrated to be more robust to the choice of the underling ML model and easier to be fine-tuned.

Building upon the outcomes of this work, it is possible to investigate further open issues and relevant research questions.
Different hyper parameter optimization methods such as random search can be used instead of grid search to make the procedure faster.
More research can explore the effects of using different weights for training the model in HML. Future works can also concentrate on effect of using neural network algorithms instead of the used ML methods in HML, to investigated how prediction accuracy is affected.

Bibliography

[1] P. et al. “Scikit-learn: Machine Learning in Python”. In: *JMLR* 12 (2011), pp. 2825–2830.

[2] O. Alipourfard, H. H. Liu, J. Chen, S. Venkataraman, M. Yu, and M. Zhang. “CherryPick: Adaptively Unearthing the Best Cloud Configurations for Big Data Analytics.” In: *NSDI*. 2017.

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

[4] D. Ardagna, C. Ghezzi, and R. Mirandola. “Rethinking the Use of Models in Software Architecture”. In: *Proc. of QoSA*. 2008.

[5] S. Arlot and A. Celisse. “A survey of cross-validation procedures for model selection”. In: *Statistics Surveys*. 2010, pp. 40–79.

[6] E. Ataie, E. Gianniti, D. Ardagna, and A. Movaghar. “A Combined Analytical Modeling Machine Learning Approach for Performance Prediction of MapReduce Jobs in Cloud Environment”. In: *Proc. of SYNASC*. 2016.

[7] Bishop. *Pattern Recognition And Machine Learning*. Springer, 2006.

[8] G. Casale, M. Tribastone, and P. G. Harrison. “Blending randomness in closed queueing network models”. In: *Perform. Eval.* 82 (2014), pp. 15–38.

[9] C.-C. Chang and C.-J. Lin. “LIBSVM: a library for support vector machines”. In: *ACM Trans. on Intelligent Systems and Technology* 2.27 (2011), pp. 1–27.

[10] V. Dalibard, M. Schaarschmidt, and E. Yoneki. “BOAT: Building auto-tuners with structured Bayesian optimization”. In: *WWW 2017 Proc.*

[11] J. Dean and S. Ghemawat. “MapReduce: Simplified Data Processing on Large Clusters”. In: *Comm. of ACM* 51.1 (2008), pp. 107–113.

[12] 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.

[13] C. Delimitrou and C. Kozyrakis. "Quasar: resource-efficient and QoS-aware cluster management". In: *ACM SIGPLAN Notices*. Vol. 49. 4. ACM. 2014, pp. 127–144.

[14] Derrick. "Survey shows huge popularity spike for Apache Spark". In: (2015). URL: http://fortune.com/2015/09/25/apache-spark-survey.

[15] D. Didona, P. Felber, D. Harmanci, P. Romano, and J. Schenker. "Identifying the optimal level of parallelism in transactional memory applications". In: *Computing* 97.9 (2015), pp. 939–959.

[16] D. Didona, F. Quaglia, P. Romano, E. Torre, and U. Roma. "Enhancing Performance Prediction Robustness by Combining Analytical Modeling and Machine Learning". In: *Proc. of ACM/SPEC*. 2015.

[17] D. Didona and P. Romano. "Hybrid Machine Learning/Analytical Models for Performance Prediction: a Tutorial". In: *Proc. of ACM/SPEC*. 2015.

[18] D. Didona and P. Romano. "On Bootstrapping Machine Learning Performance Predictors via Analytical Models". In: (2014). arXiv: arXiv:1410.5102v1.

[19] D. Didona, P. Romano, S. Peluso, and F. Quaglia. "Transactional Auto Scaler : Elastic Scaling of Replicated In-Memory Transactional Data Grids". In: *ACM Trans. on Autonomous and Adaptive Systems* 9.2 (2014), pp. 1–32.

[20] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. 6th ed. 2015. ISBN: 978-1-4614-7138-7.

[21] G. P. Gibilisco, M. Li, L. Zhang, and D. Ardagna. "Stage aware performance modeling of DAG based in memory analytic platforms". In: *Proc. of CLOUD*. 2016.

[22] J. Han, M. Kamber, and J. Pei. *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2012. ISBN: 978-0-12-381479-1.

[23] H. Herodotou, F. Dong, and S. Babu. "No One (Cluster) Size Fits All: Automatic Cluster Sizing for Data-intensive Analytics Categories and Subject Descriptors". In: *Proc. of ACM Symposium on Cloud Computing*. 2011.

[24] E. Ipek, S. A. McKee, B. R. de Supinski, and R. Caruana. "Efficiently exploring architectural design spaces via predictive modeling". In: *Proc. of ASPLOS*. 2006.

[25] F. Isaila, P. Balaprakash, S. M. Wild, D. Kimpe, R. Latham, R. Ross, and P. Hovland. "Collective I/O tuning using analytical and machine learning models". In: *Proc. of IEEE Cluster Computing*. 2015, pp. 128–137.

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

[27] K.J. Milmann and M. Avaizis. "Scientific Python". In: *Computing in Science & Engineering* 11 (2011).

[28] P. Lama and X. Zhou. "AROMA: automated resource allocation and configuration of MapReduce environment in the cloud". In: *Proc. of ICAC*. 2012.

[29] Leo Breiman. *Random Forests*. 2001.

[30] 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.

[31] M.Bertoli, G.Casale, and G.Serazzi. "JMT: performance engineering tools for system modeling". In: *ACM SIGMETRICS Performance Eval. Review* 36.4 (2009), pp. 10–15.

[32] M. Malekimajd, D. Ardagna, M. Ciavotta, and A. M. Rizzi. "Optimal Map Reduce Job Capacity Allocation in Cloud Systems". In: *ACM SIGMETRICS Perf. Evaluation Review* 42.4 (2015), pp. 51–61.

[33] K. Ousterhout, R. Rasti, S. Ratnasamy, S. Shenker, and B. G. Chun. "Making Sense of Performance in Data Analytics Frameworks". In: *Proc. of NSDI* (2015).

[34] J. Polo, Y. Becerra, D. Carrera, M. Steinder, I. Whalley, J. Torres, and E. Ayguad. "Deadline-Based MapReduce Workload Management". In: *IEEE Trans. on Network and Service Management* 10.2 (2013), pp. 231–244.

[35] R.E. Fan et al. "LIB LINEAR: A library for large linear classification." In: *The Journal of Machine Learning Research* (2008).

[36] A. M. Rizzi. "Support vector regression model for BigData systems". In: *ArXiv e-prints* (Dec. 2016). arXiv: 1612.01458 [cs.DC].

[37] D. Rughetti, P. D. Sanzo, B. Ciciani, F. Quaglia, and S. Universit. "Analytical/ML Mixed Approach for Concurrency Regulation in Software Transactional Memory". In: *Proc. of IEEE/ACM CCGrid*. 2014.

[38] P. D. Sanzo, F. Quaglia, B. Ciciani, A. Pellegrini, D. Didona, P. Romano, R. Palmieri, and S. Peluso. "A flexible framework for accurate simulation of cloud in-memory data stores". In: *Simulation Modelling Practice and Theory* 58 (2015), pp. 219–238.

[39] A. J. Smola and B. Schölkopf. "A tutorial on support vector regression". In: *Statistics and Computing* 14.3 (2004), pp. 199–222. ISSN: 1573-1375. DOI: 10.1023/B:STCO.0000035301.49549.88. URL: http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88.

[40] G. Tesauro, N. K. Jong, R. Das, and M. N. Bennani. "On the use of hybrid reinforcement learning for autonomic resource allocation". In: *Cluster Computing* 10 (2007), pp. 287–299.

[41] E. Thereska and G. R. Ganger. "IRONModel: Robust Performance Models in the Wild". In: *Proc. of ACM SIGMETRICS*. 2008.

[42] V. N. Vapnik. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN: 0-387-94559-8.

[43] S. Venkataraman, Z. Yang, M. J. Franklin, B. Recht, and I. Stoica. "Ernest: Efficient Performance Prediction for Large-Scale Advanced Analytics." In: *NSDI*. 2016.

[44] A. Verma, L. Cherkasova, and R. H. Campbell. "ARIA: Automatic Resource Inference and Allocation for MapReduce Environments". In: *Proc. of ICAC*. 2011.

[45] N. Yigitbasi, T. L. Willke, G. Liao, and D. Epema. "Towards Machine Learning-Based Auto-tuning of MapReduce". In: *Proc. of MASCOTS*. 2013.

[46] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. "Spark: Cluster Computing with Working Sets". In: *Proc. of HotCloud 2010*.

[47] M. Zaharia, P. Wendell, A. Konwinski, and H. Karau. *Learning Spark*. O'Reilly Media, Inc., 2015. ISBN: 978-1-4493-5862-4.

[48] Y. Zhang, S. Chen, Q. Wang, and G. Yu. "MapReduce: Incremental MapReduce for Mining Evolving Big Data". In: *IEEE Trans. on Knowledge and Data Engineering* 27.7 (2015), pp. 1906–1919.

backmatter/references.bib