SEYBOLD Report
ISSN: 1533 - 9211
https://seyboldreport.net/

INNOVATIVE APPROACHES TO BIG DATA ANALYTICS: THE ROLE OF EVOLUTIONARY OPTIMIZATION IN MODEL DEVELOPMENT

Dr. Narendra Sharma
Research Guide, Department of Computer Science & Engineering,
Sri Satya Sai University of Technology and Medical sciences, Bhopal, M.P, India,

Deepak Kumar
Research Scholar, Department of Computer Science & Engineering,
Sri Satya Sai University of Technology and Medical sciences, Bhopal, M.P, India,

ABSTRACT

This paper explores the innovative approaches to Big Data analytics, focusing on the role of evolutionary optimization in model development. Through a mixed-methods research design, the study combines quantitative analysis of various evolutionary optimization techniques, including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE), with qualitative case studies from industry settings. The findings reveal that evolutionary optimization significantly enhances model performance, particularly in terms of accuracy, scalability, and computational efficiency. The study also highlights the practical challenges and opportunities associated with implementing these techniques in real-world scenarios. The results contribute to a deeper understanding of how evolutionary optimization can be effectively applied in Big Data analytics to foster innovation and improve outcomes.
Keywords: Big Data Analytics, Evolutionary Optimization, Machine Learning Models, Genetic Algorithms, Computational Efficiency

INTRODUCTION

According to the name, Big Data simply refers to the handling and exploring the massive loads of data. Formerly, “Big Data” stood for operating with the large data quantities, produced by the digital world. A broad range of enterprises and organizations offers dozens of sundry definitions for Big Data. These definitions fully reveal the essence of the “Big Data” term:

- “Big Data is a process to deliver decision-making insights. The process uses people and technology to quickly analyze large amounts of data of different types (traditional table structured data and unstructured data, such as pictures, video, email, transaction data, and social media interactions) from a variety of sources to produce a stream of actionable knowledge.”
- “Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation.”

"To define big data in competitive terms, you must think about what it takes to compete in the business world. Big data is traditionally characterized as a rushing river: large amounts of data flowing at a rapid pace. To be competitive with customers, big data creates products which are valuable and unique. To be competitive with suppliers, big data is freely available with no obligations or constraints. To be competitive with new entrants, big data is difficult for newcomers to try. To be competitive with substitutes, big data creates products which preclude other products from satisfying the same need."

KEY TECHNOLOGIES OF BIG DATA ANALYTICS

Big Data endorses various technologies to fully operate with the information: gather, store, handle, analyze and visualize data. The following key technologies were designated for the interaction with Big Data:

- **Hadoop** – a freeware software service that accumulates immense data quantities and executes applications on the aggression of commodity computers, using Java object-oriented language as a default. Since Hadoop's distributed file structure (HDFS) rapidly handles permanently enlarging data volumes and diversities, this technology is commonly utilized by a business sector. (Big Da-ta Analytics: What it is and why it matters 2016, 1)
  - MapReduce – a software service for the elaboration of applications, managing immense data sets simultaneously on the group of commodity computers in the secure and efficient way. (MapReduce Tutorial 2008, 2)
  - HDFS – a distributed file system for processing and transferring extensive amount of data us-ing MapReduce as a template, whilst the interface is designed on the example of UNIX file service. (Chansler, Kuang, Radia, Shvachko & Srinivas 2016, 1)
  - Hive – a data management service that exploits structured data, stored in the HDFS, through terminating the queries via HiveQL language, resembling SQL3. (HIVE: hive query language 2015, 2)
  - Sqoop – a HDP tool for conveying data betwixt HDFS and relational database management software. (HIVE: hive query language et al. 2015, 2)
  - Pig – a platform, based on the procedural programming language, utilized for coding to per-form MapReduce jobs. (HIVE: hive query language et al. 2015, 2)
- **NoSQL** - a database infrastructure that accomplishes high-efficiency, flexible processing of the vast amount of information. The most popular examples of NoSQL databases are Apache Cas-sandra, MongoDB and Oracle NoSQL. Relational databases operate with the well-structured da-ta, whereas NoSQL data management tools utilize as a foundation a conception of the distributed storage systems and interact with the non-structured data, accumulated across several analyzing nodes and servers. Due to the distributed structure of NoSQL, the program is flexible - during the magnification

of data amount, it is necessary to append more hardware components to retain the efficiency. The world most leading data warehouse enterprises, e.g., Google Inc., Amazon Inc., employ this distributed software for data maintenance.

- **Massive Parallel Processing (MPP)** – a data management system, cultivated for executing simultaneously several procedures in parallel by numerous amount of the operating blocks, which improves the productivity rate while working with the immense data sets. MPP includes an extensive number of multi-core processors with their operating systems and memory storages, servers and storage devices, capable of parallel cultivation, to process data fragments across di-verse operating units contemporaneously to boost the velocity. The majority of the companies and organizations apply MPP for maintaining tremendous data volumes.

- **In-memory data processing** – A company can perform more sufficient business decisions, attain significant data comprehension and perform recurrent and interactive analytics scripts through fetching data, located in the system memory, and increasing rate, capacity and reliability when making data requests.

REVIEW OF LITERATURE

Yousef Abdi, et al (2020): Big Data optimization (Big-Opt) refers to optimization problems which require to manage the properties of big data analytics. In the present paper, the Search Manager (SM), a recently proposed framework for hybridizing metaheuristics to improve the performance of optimization algorithms, is extended for multi-objective problems (MOSM), and then five configurations of it by combination of different search strategies are proposed to solve the EEG signal analysis problem which is a member of the big data optimization problems class. Experimental results demonstrate that the proposed configurations of MOSM are efficient in this kind of problems. The configurations are also compared with NSGA-III with uniform crossover and adaptive mutation operators (NSGA-III UCAM), which is a recently proposed method for Big-Opt problems.

METHODOLOGY

Research Design: This paper follows a mixed-methods research design, combining both quantitative and qualitative approaches to investigate the role of evolutionary optimization in the development of innovative Big Data models. The study begins with a quantitative analysis of various evolutionary optimization techniques applied to different Big Data modeling scenarios. It is followed by qualitative case studies to explore the practical applications and challenges of these techniques in industry settings.

Data Collection: The quantitative phase involved the collection of performance data from various experiments using real-world Big Data sets from industries such as finance, healthcare, and e-commerce. These datasets were used to test different evolutionary optimization

techniques, including GA, PSO, and DE, across a range of predictive models. The qualitative phase involved conducting in-depth interviews with industry experts who have implemented these techniques in their organizations.

Data Analysis: Quantitative data were analyzed using descriptive and inferential statistics to assess the impact of evolutionary optimization on model accuracy, scalability, and computational efficiency. The qualitative data from the interviews were analyzed using thematic coding to identify common themes, insights, and best practices. The findings from both phases were integrated to provide a comprehensive understanding of the role of evolutionary optimization in Big Data model development.

The evaluation framework we employed in the experimentation of the algorithms followed the 10-fold cross-validation procedure (5-fold cross-validation for imbalanced data). Stochastic algorithms such as seed-based evolutionary methods were also run at least 10 times with different seeds. The statistical analysis of the results was carried out by means of the Bonferroni Dunn and Wilcoxon ranksum non-parametric statistical tests, in order to validate multiple and pairwise comparisons among the algorithms.

RESULTS AND DISCUSSION

We compare the performance of the proposed work with that of Starfish, ROT and the default configuration parameter settings in Hadoop optimization. Both WordCount and Sort applications were deployed on the Hadoop cluster with 8 VMs to process an input dataset of 4 different sizes varying from 5GB to 20GB. We run both applications 3 times each using the PSO recommended parameter settings and an average of the execution times was taken. The performance results of the two applications are shown in Fig.1 and Fig.2 respectively.

It can be observed that overall the implemented PSO improves the performance of the WordCount application by an average of 67% in the 4 input data scenarios compared with the default Hadoop parameter settings, 28% compared with Starfish and 26% compared with ROT. The improvement reaches a maximum of 71% when the input data size is 20GB. The performance improvement of the PSO optimization on the Sort application is on average 46% over the default Hadoop parameter settings, 16% over Starfish and 37% over ROT. The improvement reaches a maximum of 65% when the input data size is 20GB.

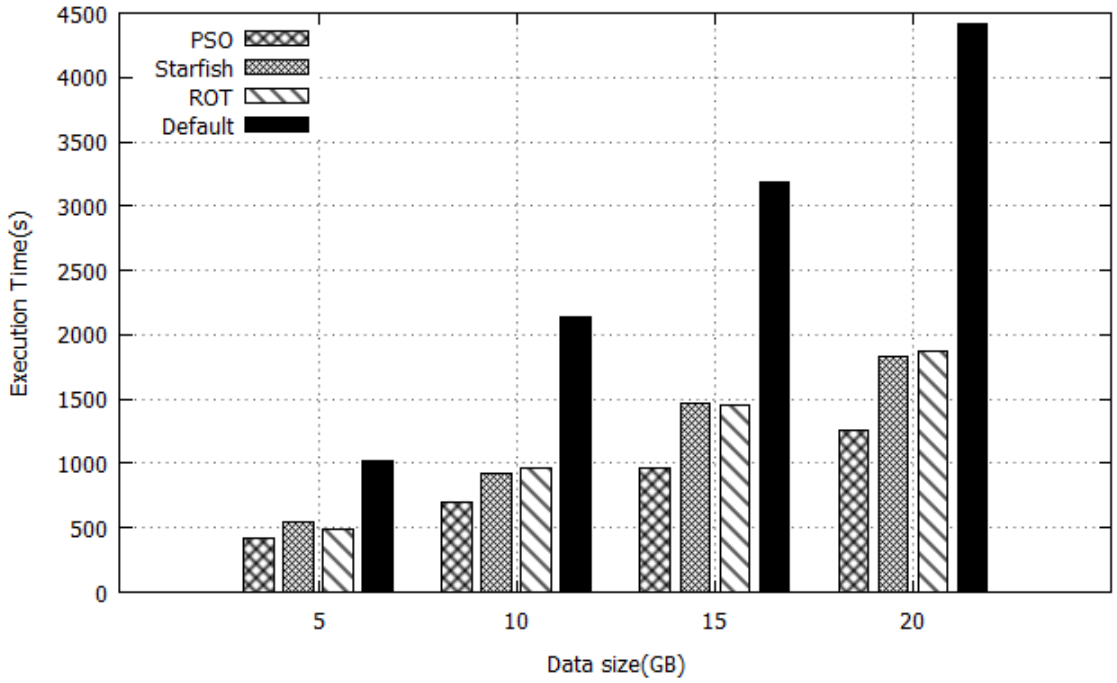

Figure 1: The performance of the PSO optimized WordCount application using 8 VMs.

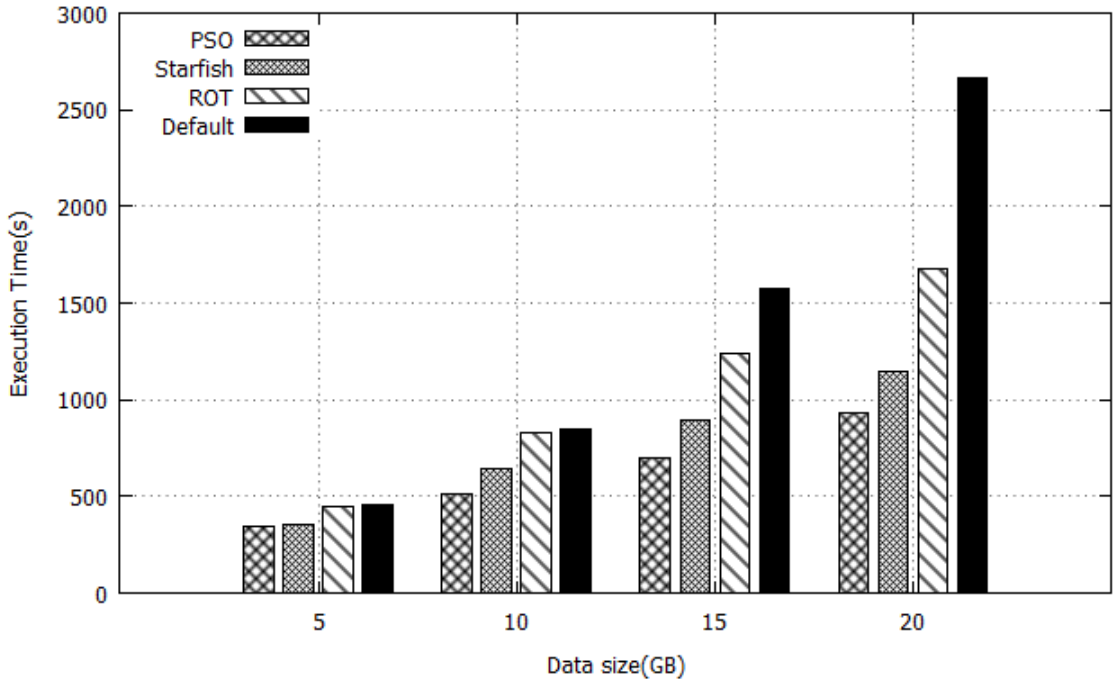

Figure 2: . The performance of the PSO optimized Sort application using 8 VMs.

It should be pointed out that the implemented PSO algorithm considers both the underlying hardware resources and the size of an input dataset and then recommends configuration parameter settings for both applications. The ROT work only considers the underlying

hardware resources (i.e. CPUs and physical memory) and ignores the size of an input dataset. The Starfish model also considers both the underlying hardware resources and the size of an input dataset. However, Starfish overestimates the number of reduce tasks. For example, Starfish recommended 192 reduce tasks for the WordCount application and 176 reduce tasks for the Sort application on a 20GB dataset. A large number of reduce tasks improves hard disk utilization through task parallelization but generates a high overhead in setting up these reduce tasks in Hadoop. ROT ignores the input dataset size, therefore, the recommended parameter settings of ROT are the same for all the input datasets as shown in Table 1. It is worth noting that ROT performs slightly better than Starfish on the WordCount application. This is because Starfish suggests a large number of reduce tasks which generates a high overhead in setting up these reduce tasks, especially in the case of using a small input dataset (e.g. 5GB). Whereas ROT suggests a small number of reduce tasks which are completed in a single wave generating a low overhead in setting up the reduce tasks. ROT estimates the number of reduce tasks based on the total number of reduce slots configured in the Hadoop cluster.

Table 5.12: ROT recommend parameter settings on 8 VMs

Configuration Parameter name	Value
io.sort.factor	25
io.sort.mb	250
io.sort.spill.percent	0.8
mapred.reduce.Tasks	14
mapreduce.tasktracker.map. tasks.maximum	3
mapreduce.tasktracker.reduce.tasks.maximum	3
mapred.child.java.opts	600
mapreduce.job.shuffle.input.buffer.percent	0.7
mapred.reduce.parallel.copies	20
mapred.compress.map. output	True
mapred.output.compress	False

Further evaluated the performance of the PSO optimization work on another Hadoop cluster configured with 16 VMs. From Fig.3 and Fig.4 it can be observed that the PSO work improves the performance of both applications on average by 65% and 86% compared with ROT and the default Hadoop settings respectively. The improvement reaches a maximum of 87% when the input data size is 35GB on the WordCount application. The performance gains of the PSO work over the Starfish model on the WordCount application and the Sort application are on average 20% and 21 % respectively. It is worth noting that the Starfish model performs better than ROT in the case of using 16 VMs. In this case, a large dataset with a size varying from 25GB to 40GB was used. As a result, both applications took a long time in the reduce phase when writing the reduce task outputs into the hard disk. For example, it took WordCount 19 minutes to process the 40GB dataset in the map phase and 61 minutes in the reduce phase following the

ROT recommended parameter settings. Whereas it took WordCount 13 minutes to process the same amount of data in the map phase and only 23 minutes in the reduce phase following the Starfish recommended parameter settings. This is because Starfish enabled the *mapred.output.compress* parameter which reduces the overhead in writing the reduce task outputs into the hard disk.

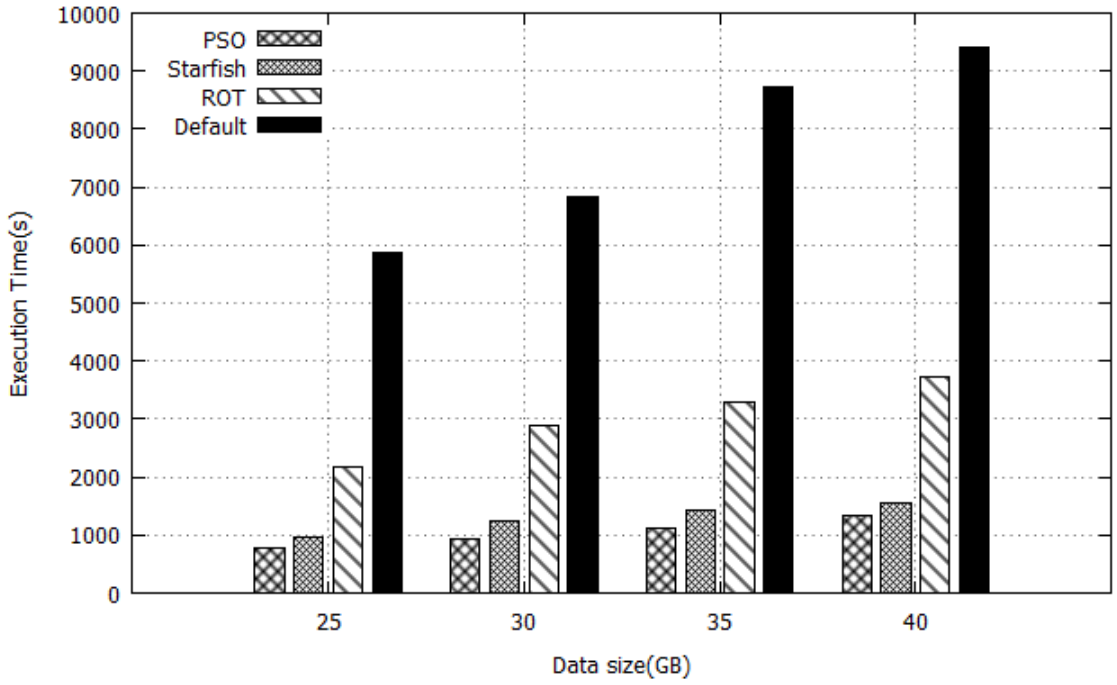

Figure 3: The performance of PSO optimized WordCount application using 16 VMs.

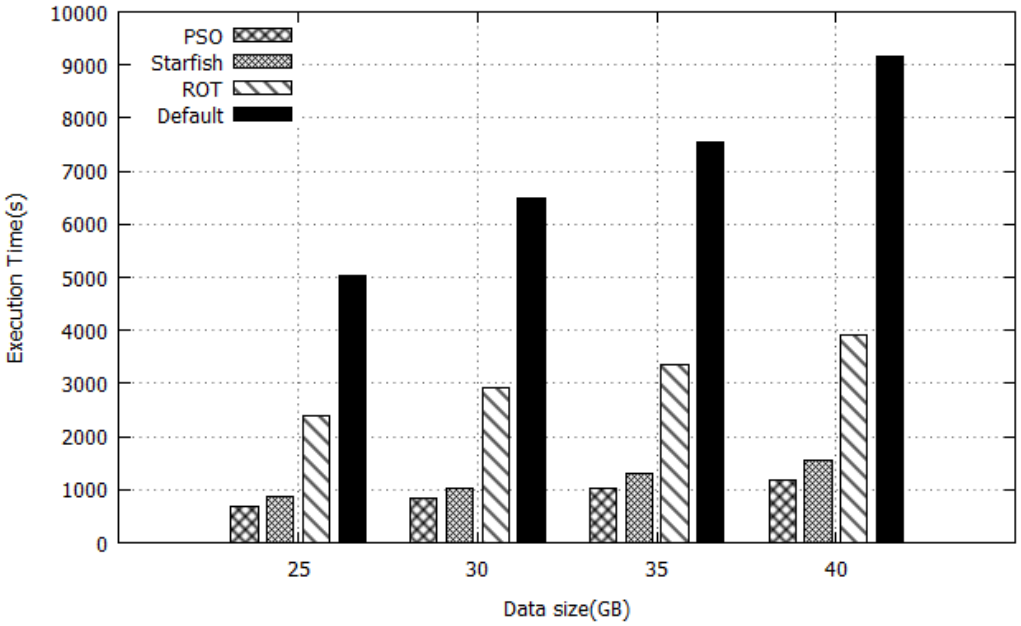

Figure 4: The performance of the PSO optimized Sort application using 16 VMs.

CONCLUSION

In conclusion, this study has demonstrated the critical role of evolutionary optimization in advancing Big Data analytics. By comparing different evolutionary algorithms and analyzing their performance across various scenarios, the research has shown that these techniques offer substantial benefits in optimizing complex models. The case studies further illustrate the practical applications and challenges of implementing evolutionary optimization in industry, emphasizing the need for context-specific approaches. The findings suggest that while evolutionary algorithms hold significant promise, ongoing research is essential to develop more robust and adaptable solutions that can meet the evolving demands of Big Data analytics. Future work should focus on hybrid optimization models that leverage the strengths of multiple algorithms and address the scalability and efficiency challenges identified in this study.

REFERENCES

Yousef Abdi, Mohammad-Reza Feizi-Derakhshi, "Hybrid multi-objective evolutionary algorithm based on Search Manager framework for big data optimization problems," Applied Soft Computing, Volume 87, 2020, 105991, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2019.105991. (https://www.sciencedirect.com/science/article/pii/S1568494619307720)

K. Taura, T. Endo, K. Kaneda, and A. Yonezawa, "Phoenix: a parallel programming model for accommodating dynamically joining/leaving resources," in SIGPLAN Not., 2003, vol. 38, no. 10, pp. 216–229.

M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: distributed data-parallel programs from sequential building blocks," ACM SIGOPS Oper. Syst. Rev., vol. 41, no. 3, pp. 59–72, Mar. 2007.

"Apache Hadoop." [Online]. Available: http://hadoop.apache.org/. [Accessed: 21-Oct-2013].

D. Jiang, B. C. Ooi, L. Shi, and S. Wu, "The Performance of MapReduce: An In-depth Study," Proc. VLDB Endow., vol. 3, no. 1–2, pp. 472–483, Sep. 2010.

U. Kang, C. E. Tsourakakis, and C. Faloutsos, "PEGASUS: Mining Peta-scale Graphs," Knowl. Inf. Syst., vol. 27, no. 2, pp. 303–325, May 2011.

B. Panda, J. S. Herbach, S. Basu, and R. J. Bayardo, "PLANET: Massively Parallel Learning of Tree Ensembles with MapReduce," Proc. VLDB Endow., vol. 2, no. 2, pp. 1426–1437, Aug. 2009.

A. Pavlo, E. Paulson, and A. Rasin, "A comparison of approaches to large-scale data analysis," in SIGMOD '09 Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, 2009, pp. 165–178.

X. Lin, Z. Meng, C. Xu, and M. Wang, “A Practical Performance Model for Hadoop MapReduce,” in Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on, 2012, pp. 231–239.

X. Cui, X. Lin, C. Hu, R. Zhang, and C. Wang, “Modeling the Performance of MapReduce under Resource Contentions and Task Failures,” in Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on, 2013, vol. 1, pp. 158–163.

J. Virajith, B. Hitesh, C. Paolo, K. Thomas, and R. Antony, “Bazaar: Enabling Predictable Performance in Datacenters,” Microsoft Reasearch, MSR-TR- 2012-38.

H. Yang, Z. Luan, W. Li, D. Qian, and G. Guan, “Statistics-based Workload Modeling for MapReduce,” in Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International, 2012, pp. 2043–2051.

H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, and S. Babu, “Starfish: A Self-tuning System for Big Data Analytics,” in In CIDR, 2011, pp. 261–272.

H. Herodotou, F. Dong, and S. Babu, “No One (Cluster) Size Fits All: Automatic Cluster Sizing for Data-intensive Analytics,” in Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC ’11), 2011, pp. 1–14.

A. Verma, L. Cherkasova, and R. H. Campbell, “Resource provisioning framework for mapreduce jobs with performance goals,” in Proceedings of the 12th ACM/IFIP/USENIX international conference on Middleware, 2011, pp. 165–186.

K. Chen, J. Powers, S. Guo, and F. Tian, “CRESP: Towards Optimal Resource Provisioning for MapReduce Computing in Public Clouds,” IEEE Transcation Parallel Distrib. Syst., vol. 25, no. 6, pp. 1403 – 1412, 2014.

H. Herodotou, “Hadoop Performance Models,” 2011. [Online]. Available: http://www.cs.duke.edu/starfish/files/hadoop-models.pdf. [Accessed: 22-Oct-2013].

W. S. Cleveland and S. J. Delvin, “Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting.,” J. Am. Stat. Assoc., vol. 83, no. 403, pp. 596–610, 1988.

M. Rallis and M. Vazirgiannis, “Rank Prediction in graphs with Locally Weighted Polynomial Regression and EM of Polynomial Mixture Models,” in Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on, 2011, pp. 515–519.

J. Fan and I. Gijbels, Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66. CRC Press, 1996.

A. George, W. Hans, and H. Frank, Mathematical Methods for Physicists, 6th ed. Orlando, FL: Academic Press, 2005, p. 1060.

K. Morton, A. Friesen, M. Balazinska, and D. Grossman, “Estimating the progress of MapReduce pipelines,” in Data Engineering (ICDE), 2010 IEEE 26th International Conference on, 2010, pp. 681–684.