Performance tuning Apache Drill on Hadoop Clusters with Genetic Algorithms

Improving industry standards for advanced analytics and business intelligence

Roger Bløtekjær

Thesis submitted for the degree of
Master of science in Informatikk: språkteknologi
30 credits

Institutt for informatikk
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2018

Performance tuning Apache Drill on Hadoop Clusters with Genetic Algorithms

Improving industry standards for advanced analytics and business intelligence

Roger Bløtekjær

0.1 Summary

0.2 Foreword

I got nothing. Put this on a different page though, maybe with a dramatic font or something.

Contents

Chapter 1

Introduction/Background ?

1.1   Target group

This thesis covers the deep technical aspects of big data analysis and genetic algorithms. All techniques used will be explained in detail, but it is advised to have a certain degree of technical insight before reading.

1.2   Area of research

Hadoop and MapReduce are already well established technologies employed in countless applications around the world, Apache Drill however is a fairly new concept with little to no research from the community. We propose a new method of implementing Hadoop clusters with Apache Drill, automatically optimizing the performance tuning in a lightweight and low effort way.

1.3   Personal motivation

The subject for this master thesis is a natural continuation of our previous work *Hadoop MapReduce Scheduling Paradigms*, published in 2017, in the 2nd IEEE International Conference on Cloud Computing and Big Data Analysis (ISSS-BDA 2017). Back then the topic was haphazardly picked from a list of eligible ones, but the more we read into it - the more we understood the incredible use cases for Hadoop within the massive industries that are driving forces for our technological advancements. As is common in IT, levels of abstraction get added on top of proven technologies both to make implementation better, and often to increase performance. Since then Apache Drill has seen plenty of stable builds and proven itself to be potentially industry-changing in the way we handle our data - entering a schema-on-the-fly paradigm.

1.4   Research method in brief

Throughout this thesis we will develop an entire suite of tools centered around a genetic algorithm for automatically optimizing Apache Drill on top of a Hadoop

cluster, tested on big and diverse sample sets. Metrics will NEED MORE RE-SEARCH TO DECLARE

1.5 Most relevant previous findings

There arent really any research done on the impact of tuning Apache Drill.

1.6 Why is this worthwhile

During my previous research the motivation for all the papers surveyed was mostly the same. **First and foremost, it was well acknowledged that Hadoop clusters show very suboptimal performance with out-of-the-box settings. Secondly, it was shown that finding optimal parameters for a Hadoop cluster is very time consuming and hard. As organizations often run with a lack of resources, time and domain-specific engineers, this is a field ripe for improvement.**

1.7 How far will this advance the field?

Hopefully this will provide a fully functional, open source light weight framework allowing companies to easily deploy Hadoop Clusters without worrying about tailoring the solution or suboptimal performance. If the task proves to be too big, this will lay the foundation for further work to make a de facto solution. Or something.

1.8 Structure of the report

Add comments about every chapter here.

Chapter 2

Related literature and theoretical focus

2.1 Performance tuning MapReduce as a Research Field

There is a ton of research to be read about MapReduce optimization, trying to tune map- and reduce-slots, and improve the inherent job tracker. Looking at a few essential papers within this field shows a general consensus that tuning default performance parameters in a variety of environments will lead to 10-30% performance increase, which naturally results in considerable cost savings. For instance Min Li et. al. could report that "*Our results using a real implementation on a representative 19-node cluster show that dynamic performance tuning can effectively improve MapReduce application performance by up to 30% compared to the default configuration used in YARN.*"[1] Furthermore, manually tuning these clusters are too demanding for most businesses to even consider, requiring both deep technical and domain-specific insight. These findings further maintain our vision of bringing auto-tuning to the Apache Drill framework, which we believe to be the next natural step forwards, even paradigm-defining, for big data handling.

2.2 Gunther

Gunther evaluated methods for optimizing Hadoop configurations using machine learning and cost-based models, but found them inadequate for automatic tuning. Thus they introduced a search-based approach with genetic algorithms, designed to identify crucial parameters to reach near-optimal perfomance of the cluster.[2] This paper tells us that genetic algorithms as an approach to performance tune big data clusters already is a proven method. However, the scope is set to Hadoop as an out-of-the-box enterprise solution, whereas we take the next step within the schema-on-the-fly paradigm of Apache Drill.

2.3   mrOnline

"MapReduce job parameter configuration significantly impacts application performance, yet extant implementations place the bur- den of tuning the parameters on application programmers. This is not ideal, especially because the application developers may not have the system-level expertise and information needed to select the best configuration. Consequently, the system is utilized inef- ficiently which leads to degraded application performance."[1]

Chapter 3

Presentation of domain where technology is used

BMC Software, Inc states the following about Apache Hadoop:

Financial services companies use analytics to assess risk, build investment models, and create trading algorithms; Hadoop has been used to help build and run those applications. Retailers use it to help analyze structured and unstructured data to better understand and serve their customers. In the asset-intensive energy industry Hadoop-powered analytics are used for predictive maintenance, with input from Internet of Things (IoT) devices feeding data into big data programs. Telecommunications companies can adapt all the aforementioned use cases. For example, they can use Hadoop-powered analytics to execute predictive maintenance on their infrastructure. Big data analytics can also plan efficient network paths and recommend optimal locations for new cell towers or other network expansion. To support customer-facing operations telcos can analyze customer behavior and billing statements to inform new service offerings. Examples of Hadoop There are numerous public sector programs, ranging from anticipating and preventing disease outbreaks to crunching numbers to catch tax cheats. Hadoop is used in these and other big data programs because it is effective, scalable, and is well supported by large vendor and user communities. Hadoop is a de facto standard in big data.[3]

Chapter 4

Method

4.1 Technology

4.1.1 Hadoop features

Hadoop common

Hadoop common consists of a few select core libraries, that drives the services and basic processes of Hadoop, such as abstraction of the underlying operating system and file system. It also contains documentation and Java implementation specifications.

HDFS

HDFS (Hadoop Distributed File System) is a highly fault tolerant, distributed file system designed to run efficiently, even on low-cost hardware. HDFS is tailor made to express large files and huge amounts of data, with high throughput access to application data and high scalability across an arbitrary amount of nodes.

YARN

YARN (Yet Another Resource Negotiator) is actually a set of daemons that run in the cluster to handle how the jobs get distributed.

- There is a NM (NodeManager) that represents each node in the cluster, monitoring and reporting to the RM whether or not they are idle, and resource usage (CPU, memory, disk, network). A node in a Hadoop cluster divides its resources into abstract Containers, and reports on how many containers there are available for the RM to assign jobs to.

- There is an AM (ApplicationMaster) per job, or per chain of jobs, representing the task at hand. This AM gets inserted into containers on nodes, when a job is running.

- Finally there is a global RM (ResourceManager), which is the ultimate authority on how to distribute loads and arbitrate resources. The RM consists of two entities, the Scheduler and the ApplicationsManager.

- The ApplicationsManager is responsible for accepting job submissions (at this point represented as an ApplicationMaster), i.e by doing code verification on submitted jobs, changing their status from "submitted" to "accepted". Once a job is accepted, the ApplicationsManager sends the ApplicationMaster to the scheduler to negotiate and supply containers for the job on the nodes in the cluster. The ApplicationsManager also has services to restart failed ApplicationMasters in containers, and retry entire jobs.

- The Scheduler is a pure scheduler in the sense that it only cares about delivering tasks to idle slots, based on free resources on the node. It calculates distributions across multiple nodes / containers, and can prioritize - i.e compute smaller jobs in front of large ones to more effectively complete job queue, even though the large job was submitted first. The Scheduler does not care for monitoring task completions or failures, simply distributing loads on the cluster.

The ResourceManager and the NodeManager form the data-computation framework. The per-application ApplicationMaster is, in effect, a framework specific library and is tasked with negotiating resources from the ResourceManager and working with the NodeManager(s) to execute and monitor the tasks.[4]

MapReduce

4.1.2 Zookeeper

4.1.3 Apache Drill

9

Chapter 5

Results

Chapter 6

Discussion

Chapter 7

Conclusion

Bibliography

[1] Li, M., Zeng, L., Meng, S., Tan, J., Zhang, L., Butt, A. R., & Fuller, N. (2014, June). *Mronline: Mapreduce online performance tuning. In Proceedings of the 23rd international symposium on High-performance parallel and distributed computing (pp. 165-176).* ACM.

[2] Liao, G., Datta, K., & Willke, T. L. (2013, August). *Gunther: Search-based auto-tuning of mapreduce.* In European Conference on Parallel Processing (pp. 406-419). Springer Berlin Heidelberg.

[3] BMC Software, Inc (2018 January) *http://www.bmc.com/guides/hadoop-examples.html*

[4] Apache Software Foundation (January 2018)*https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html*