Project Proposal

Accelerating chameleon on Hadoop with Spark

CS550 – Advanced Operating System
September 15th 2017

Xuzhu Chen A20385468
Peng Wang A20386791
Junyi Wang A20388013

1. Introduction

As people need to process data with high-volume, high velocity and high variety, they developed a scalable, reliable distributed computing software, called Hadoop. Spark is a fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation.

k-NN (k-nearest neighbors algorithm) is a non-parametric method used for classification and regression.

In this project, we will focus on achieve Chameleon algorithm (a improvement algorithm of k-NN) on hadoop with spark, compare it with the result of k-NN on hadoop with spark, and then optimize it.

2. Background Information

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The project includes these modules:

- Hadoop Common: The common utilities that support the other Hadoop modules.
- Hadoop Distributed File System (HDFS™): A distributed file system that provides high-throughput access to application data.
- Hadoop YARN: A framework for job scheduling and cluster resource management.
- Hadoop MapReduce: A YARN-based system for parallel processing of large data sets.

Spark is a Hadoop-related project which increase the reading/writing speed from both memory and disk, and helps increasing efficiency.

Chameleon algorithm is a clustering algorithm using dynamic modeling, its advantage is it can discover natural clusters of different shapes and sizes. It is the aggregate of k-NN and two-phase clustering.

3. Problem Statement

The Chameleon algorithm's key feature is that it accounts for both interconnectivity and closeness in identifying the most similar pair of clusters. It thus avoids the limitations discussed earlier. Furthermore, Chameleon uses a novel approach to model the degree of interconnectivity and closeness between each pair of clusters. This approach considers the internal characteristics of the clusters themselves. Thus, it does not depend on a static, user-supplied model and can automatically adapt to the internal characteristics of the merged clusters.

Compared with CURE and DBScan, chameleon made less mistakes than CURE. Though DBScan can finds the correct clusters, it cannot effectively find clusters with variable internal densities.

4. Related Work

Chameleon algorithm is a clustering algorithm raised in 1999. It Combines initial partition of data with hierarchical clustering techniques it modifies clusters dynamically. As for Spark, it is originally developed by University of California, Berkeley, and maintained by Apache Software Foundation as a part of Hadoop project till now. Spark is a memory based execution engine as a

replacement of MapReduce.

5. proposed solution

There is some algorithm improvements about the Chameleon algorithm, but most of them focus on improving the accuracy of algorithm. For our project, we aim at both accuracy improvement and efficiency improvement. In this way, we need to modify the Chameleon algorithm, parallelize it and implement it on Spark. We provide some evaluation on performance of original Chameleon algorithm and modified Chameleon algorithm execute on Hadoop cluster with different number of nodes.

The proposed solution consists of:

- Designing a new parallel algorithm based on Chameleon with better performance.
- Implementing the new algorithm with Spark on Hadoop cluster.
- Evaluating the performance of the new algorithm.

6. Evaluation

The evaluation of this project contains the following;

- Performance comparison of the new algorithm and the original Chameleon algorithms running on the same cluster.
- Performance comparison of the new algorithm running on Hadoop cluster (with Spark), VS. on single machine.

For the sake of these comparisons we will use "MixSim" to creature different size of testing data. All tests and comparisons will be run on Azure cloud which is more stable and has a much better scalability compare to local machine.

comparative aspects includes:

- number of nodes in cluster
- number of processors for each node
- Transfer time (transfer data between nodes)
- Centroid Updating Time
- Labeling time

7. Conclusion

A good promotion of algorithm can typically improve the efficiency of the computation. Improving an algorithm not only can improve the idea of the algorithm but also can promote the way to process the algorithm. Using multiple threads to process the knn algorithm requires a set of processor with suitable performance.

Our promotion of knn algorithm considers both algorithm and the parallel model, because the parallel algorithm can directly improve the efficiency of algorithm. As for promoting the idea of the algorithm, we aim at Chameleon algorithm. Then we will parallelize it on the spark. We also will performance the original knn algorithm on the spark. Finally, we focus on the analysis the efficiency of chameleon algorithm. We can find the apparently improvement of speed and little decrease of the accuracy. Then the acceleration of Chameleon on the spark is successful.

8. Addition Resources

8.1 Time line

week	task
1-3	Gather information, study related papers
4-5	Build the spark structure
6	Run based knn algorithm on spark
7	Run chameleon algorithm and parallelize it on spark
8	Data training and gather the data information
9	Analysis the efficiency of algorithm
10	Write the final report and presentation

8.2 Deliverables

- One final report in PDF form.
- One final Powerpoint presentation.
- Source code

8.3 Work Arrangement

- Xuzhu Chen: achieve the chameleon algorithm on the spark & data analysis
- Junyi Wang: achieve the basic knn algorithm on the spark & data analysis
- Peng Wang: parallelize chameleon on the spark & data analysis

References

[1] A. Aji F. Wang H. Vo R. Lee Q. Liu X. Zhang J. Saltz "Hadoop GIS: A high performance spatial data warehousing system over MapReduce" PVLDB vol. 6 no. 11 pp. 1009-1020 2013.

[2] M. Armbrust R. S. Xin C. Lian Y. Huai D. Liu J. K. Bradley X. Meng T. Kaftan M. J. Franklin A. Ghodsi et al. "Spark sql: Relational data processing in spark" ACM pp. 1383-1394 2015.

[3] X. Meng J. Bradley B. Yuvaz E. Sparks S. Venkataraman D. Liu J. Freeman D. Tsai M. Amde S. Owen et al. "Mllib: Machine learning in apache spark" vol. 17 no. 34 pp. 1-7 2016.

[4] D. Vohra Apache hbase primer 2016.

[5] Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm. J. R. Stat. Soc. Ser. B 1977, 39, 1–38.

[6] B. Samanthula, Y. Elmehdwi, W. Jiang, K-nearest neighbor classification over semantically secure encrypted relational data, TKDE, vol. 27, no. 5, 2015, pp. 1261-1273.

[7] H. Kim, H. Kim, J. Chang, A kNN query processing algorithm using a tree index structure on the encrypted database, in Big Data and Smart Computing (BigComp), 2016, pp. 93-100.

[8] S. Bhatkar, R. Sekar, and D. C. DuVarney, "Efficient techniques for comprehensive protection from memory error exploits," in USENIX Security Symposium (Security '05), 2005.

[9] Shweta Kharya, "Using Data Mining Techniques for Diagnosis and Prognosis of Cancer Disease", International Journal of Computer Science and Information Technology, vol. 2, no. 2, pp. 55-66, April 2012.

[10] M. Marta-Almeida M. Ruiz-Villarreal J. Pereira P. Otero M. Cirano X. Zhang and R. Hetland "Efficient tools for marine operational forecast and oil spill tracking" Marine Pollution Bulletin vol. 71 no. 1–2 pp. 139–151 2013.