ILLINOIS INSTITUTE OF TECHNOLOGY

Project Proposal

CS554: Data-Intensive Computing

Fall 2018

Title:

XSearch : Disk Data Pool

People involved:

a). Krishna Bharadwaj - A20398222
b). Pooja Patel - A20396099
c). Animesh Patni - A20403240

Abstract:

We live in a world now where the advances in computing and computer systems have led to a tremendous increase in the amount of data generated. Studies have shown that a major portion of the generated data is unstructured, and it is growing at 60% annually. With this upsurge of data, it is of paramount importance that we can retrieve the data and retrieve it as quickly as possible. This is especially true in the world of high performance computing(HPCs) where searches are performed over terabytes of files. In scenarios such as the one stated above, searching, which usually is a trivial task on a personal computer could get very tricky and end up consuming a lot of time. In this project, we look at some of the commonly used open source libraries such as Lucene, Lucene PlusPlus and Xapian. The idea is to benchmark these libraries on a single node instance over text and metadata search. We observe the Indexing throughput, Query Throughput and the Index size and finally, identify potential bottlenecks in the system and perhaps come up with solutions to overcome them.

Background information:

Information Retrieval has come a long way, from searching in small datasets to today dealing with huge datasets. There have been researches done on how to optimize and increase the performance even more. New libraries and techniques have come up, but still there is some room for improvement. Scaling up the system has shown some improvements and by using different indexing techniques and developing dynamic indexing techniques has also shown some improvements. Indexing data properly is still a challenge. We will be benchmarking different techniques of implementing the search and emphasizing more on the throughput for the information retrieval from the disk.

Problem statement:

The Problem we are addressing is from the information retrieval side of Computer Science. Information Retrieval is a process of obtaining relevant information based on the search criteria from the resources of information The search can be based on content as well as the metadata. In computer system the materials are represented as files while the collection of information resources is comprised of filesystem. To improve the performance and efficiency of search on text or metadata in the supercomputers is what we are looking at. Also this area is not explored or focused so much previously.

Related work:

As we delved deeper into the topics of information retrieval, we stumbled upon many studies focusing on performance evaluation of various information retrieval libraries. However, we notice that in many such studies, the emphasis was primarily on the accuracy of the search result. EVALUATION OF INFORMATION RETRIEVAL SYSTEMS by Keneilwe Zuva and Tranos Zuva talks about evaluating information retrieval systems based on precision and recall. However, our emphasis is firmly on query throughput, Index throughput and index size. To put this in simple terms, we look at how fast a search can be performed rather than how accurately the search is performed

Proposed solution:

There can be multiple ways to reach the end goal and benchmark different libraries, as we right now have a overview of the problem and the research we have put in we came up with one solution path we think we can achieve the goal of benchmarking different libraries. It begins with creating an Linux instance on the chameleon cluster (one node) and deciding the specification of the system we are going to work on. The specification that includes Read-only memory, number of cores, disk size etc. We will be using textual data to perform the search task and Wikipedia has one of the richest textual collection of all. So, we will be taking Wikipedia dumps and splitting them into small chunks using scripts (Python or R). One more aspect of this information retrieval is performing search on metadata, we will be taking metadata dumps and would perform textual search as well as metadata search separately and benchmarking them of different libraries. We will be looking at various libraries but the first one we would be looking at is Apache Lucene as this is one of the most extensively used libraries, we will be working on different parameters so that we can achieve the best outcome possible. Benchmarking phase for Apache Lucene will be very interesting as we will be tuning the parameters even more to achieve the best result we possibly can. Analyzing the outcome is one of the most important phases and we will be looking into the previous researches and implementation of Apache Lucene to more analyze and improve our findings. The second library we will be working on is LucenePlusPlus. LucenePlusPlus uses whole different structure and parameters and first understanding those and implementing basic information retrieval will be our first step. Then we will be benchmarking the same and would be performing the Parameter tuning for better results. Analyzing our findings with the previous implementations and researches of LucenePlusPlus would help us increase the performance. The above-mentioned libraries would be implemented using C Language. The third library we are going to use is Xapian and this is different from the above too. We will be implementing this using C++ and performing parameter tuning while benchmarking the same. We will be looking precious implementation of the same to get a better analysis of the outcome. There is one more aspect we might look at is implementing a data structure (C-Trie) to improv the indexing mechanism of our model as indexing data dynamically is one of the most efficient way of performing information retrieval. Finally, we will be comparing each libraries and implementation and would be looking at which one is performing the best and under what setup. What is the bottle-neck for the model and which component or parameter is acting as one? How can we overcome that bottleneck? This solution path may change as we proceed further and we would like to continue working on this problem in the future as well, cause information retrieval on large data and who can we achieve it even faster be it metadata or textual information.

Evaluation:

We will be focused on calculating the indexing throughput, query throughput and index size for each of the libraries in a single threaded and a multi-threaded environment. We plan to import the Wikipedia dump build an index on the Wikipedia dump for each of the IR libraries mentioned above. Hence, the indexing throughput will be calculated programmatically using the formula **<size of data to be indexed>/<time taken>. The unit for measuring indexing throughput will be MB/s.**

The index size would be the size of the index created after indexing. **The unit for measuring index size will be MB.**

The query throughput is calculated as **<size of index>/<time taken to search>. The unit for measuring query throughput will be MB/s.**

Upon measuring the above quantities, we will perform an empirical analysis on the data acquired, and thus, we will be able to identify for each of the libraries the potential bottlenecks in the system and how to overcome them.

Timeline with weekly goals:

Weeks	Goals
August – Week 4	Research on three different projects
September – Week 1	Brainstorming on xSearch
September – Week 2	Creating Instance on chameleon and getting hands on using the instance on chameleon
September – Week 3	Setting up and Benchmarking Lucene
September – Week 4	Parameter tuning for Lucene in order to improve the performance
October – Week 1	Analysis on the results obtained and Finding the bottleneck component
October – Week 2	Setting up and Benchmarking LucenePlusPlus and working on Project Midterm progress report / presentation
October – Week 3	Parameter tuning for LucenePlusPlus in order to improve the performance
October – Week 4	Analysis on the results obtained and Finding the bottleneck component
November – Week 1	Setting up and Benchmarking xapian
November – Week 2	Parameter tuning of xapian in order to improve the performance
November – Week 3	Analysis on the results obtained and Finding the bottleneck component

November – Week 4	Working on the results obtained from all the three experiments and analysis on it along with making the final report and preparing project final presentation

Deliverables:

The deliverables of the project will be:

- Final Project report
- Final Project Presentation
- Code
- Experiment results
- Comparisons in the form of charts or tables
- Analysis of the result

Conclusion:

The following will be the learning from this project:

- Information retrieval from disk on one node from large amount of data
- We will be working on different open source information retrieval libraries and will learn and evaluate how each one of them is performing
- We will learn how to scale a system so that we can get better efficiency and performance
- Also will learn more about various disk and information retrieval from those disk
- We will evaluate what component of the system is acting as a bottleneck for the search and what can be done   to overcome the same
- The project will be a success if we are able to successfully identify bottlenecks in the system and explore possible solutions

References:

[1] Alexandru Iulian Orhean, Kyle Chard, Ioan Raicu.“XSearch: Distributed Information Retrieval in Large-Scale Storage Systems”

[2] Alexandru Iulian Orhean, Itua Ijagbone, Dongfang Zhao, Kyle Chard, Ioan Raicu. “Toward Scalable Indexing and Search on Distributed and Unstructured Data”, IEEE Big Data Congress 2017

[3] Itua Ijagbone, “Scalable indexing and searching on distributed file systems”. Master thesis. Illinois Institute of Technology, 2016

[4] keneilwe zuva and tranos zuva. “Evaluation of information retrieval Systems”

[5] Dongfang Zhao, Ning Liu, Dries Kimpe, Robert Ross, Xian-He Sun, and Ioan Raicu. “Towards Exploring Data-Intensive Scientific Applications at Extreme Scales through Systems and Simulations”

[6] T. Leibovici, “Taking back control of hpc file systems with robinhood

policy engine,”

[7] S. Chafle, J. Wu, I. Raicu, and K. Chard, “Optimizing search in unsharded largescale distributed systems.”

[8] Patrick Glauner, Jan Iwaszkiewicz , Jean-Yves Le Meur and Tibor Simko, “Use of Solr and Xapian in the Invenio document repository software”

[9] LucenePlusPlus OpenHub : https://www.openhub.net/p/LucenePlusPlus

[10] http://wwwhome.cs.utwente.nl/~hiemstra/papers/IRModelsTutorial-draft.pdf

[11] http://lucene.apache.org/

[12] https://github.com/luceneplusplus/

[13] https://xapian.org/