Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 10, October 2015, pg.311 – 317

SURVEY ARTICLE

A SURVEY ON HIERARCHICAL DIFFUSION ALGORITHM FOR BIG DATA ENVIRONMENT

[1]V.K.SHANMUGA PRIYA, [2]G.MAHALAKSHMI
PG STUDENT, ASSISTANT PROFESSOR
[1,2]NPR COLLEGE OF ENGINEERING AND TECHNOLOGY, TAMILNADU, INDIA
[1] krishshanmuga@gmail.com; [2] mahame86@gmail.com

Abstract— Accurate solution has to be provided in big data environment using hierarchical diffusion algorithm for range amalgamate queries. Datum which can be distributed into many form and then create an advanced assessment can be construct if any inquiry appear it achieve solution can be access exactly by encapsulate narrow estimate from all distribution. Conclusion can be originate detailed form. It has been going to run in heterogeneous environment. It goes to run on heterogeneous environment. It can be the preprocess data and then upload it into the distributed file server(DFS) .Range amalgamate queries produce result directly and reduce the time efficiency compared to the subsisting process. Result can be displayed in graphical representation compared to the time duration from subsisting process with proposed concept.

Keywords— hierarchical diffusion algorithm; range amalgamate query; datum

1. INTRODUCTION

Big data is used in many applications. Nowadays so many applications run in big data concept. It can be used to save megabytes of data in data warehousing. Big data is extracted from the data warehousing concept. It can used the map reduce concept. These users may access a lot of data related to Geographical areas, political issues, neural network, health information and many more. Big data is related for social sites and media. Social sites like Google for Gmail and most preferably for the search engine, Face book, WhatsApp are hit every day by billions of people around the world. These sites improve knowledge of human

social networking, mathematicians, physicians and many more science fields by exchange of information using. All these people search valuable information in just one click for access. Big data can be used the hadoop server for improving performance. Pre-processing and then map reduce concepts are involved in big data environment. Big data processing is the main task. In this processing some frameworks are Hive. On the 6th kart announces an offer which is very cheap. Resulting in high sever processing is a very low small amount of time.

According to Flip-kart there are billions of request hit within 30 min. For processing large amount of data and analyse that data various technologies are in use as mentioned above. The more fundamental challenge for Big Data applications is to explore the large volumes of data and extract useful information or knowledge for future actions. In many situations, the knowing extraction process which has to be very efficient and close to real time as storing all observed data is nearly unachievable. The unique data quantities require an effective data analysis and prediction platform to achieve fast response and real-time classification of such Big Data. Big data sensors and growing files are used. Hierarchical diffusion algorithm can be used for partitioning a huge data into small amount of data and then create a new estimation .It is always efficient algorithm for time consumption and error reduction.

Tree like format data structure can be used on this hierarchical diffusion algorithm. One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java byte code, instead of directly to platform-specific machine code. Java byte code instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware.

End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets. Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking.

A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executables would. Just-in-Time compilers were introduced from an early stage that compiles byte codes to machine code during runtime.

Just as application servers such as Glass Fish provide lifecycle services to web applications, the Net Beans runtime container provides them to Swing applications. All new

shortcuts should be registered in "Key maps/Net Beans" folder. Shortcuts installed INS Shortcuts folder will be added to all key maps, if there is no conflict. It means that if the same shortcut is mapped to different actions in Shortcut folder and current key map folder (like Key map/Net Beans), the Shortcuts folder mapping will be ignored.

* Database Explorer Layer API in Database Explorer
* Loaders-text-db schema-Actions in Database Explorer
* Loaders-text-sql-Actions in Database Explorer
* Plug-in Registration in Java EE Server Registry

The keyword public denotes that a method can be called from code in other classes, or that a class may be used by classes outside the class hierarchy. The class hierarchy is related to the name of the directory in which the .java file is located.

The keyword static in front of a method indicates a static method, which is associated only with the class and not with any specific instance of that class. Only static methods can be invoked without a reference to an object. Static methods cannot access any class members that are not also static. The keyword void indicates that the main method does not return any value to the caller. If a Java program is to exit with an error code, it must call System.

The method name "main" is not a keyword in the Java language. It is simply the name of the method the Java launcher calls to pass control to the program. Java classes that run in managed environments such as applets and Enterprise JavaBeans do not use or need a main () method. A Java program may contain multiple classes that have main methods, which means that the VM needs to be explicitly told which class to launch from.

The Java launcher launches Java by loading a given class (specified on the command line or as an attribute in a JAR) and starting its public static void main(String[]) method. Stand-alone programs must declare this method explicitly. The String [] args parameter is an array of String objects containing any arguments passed to the class. The parameters to main are often passed by means of a command line.

2. LITERATURE REVIEW

Title 2.1: Predicting the Present with Google Trends

In this paper, Search engine can grab near-term values of economic indicators. Examples are automobile sales, travel destination planning, customer confidence. Government releases indicator of level of economic activity in many sectors." Search engine can be used to forecast near term values". These activities are available only on several weeks or few days and revised it later. It would clearly be helpful to have more timely forecasts of

these economic indicators. It would clearly be helpful to have more timely forecasts of these economic indicators. It is possible to build more sophisticated forecasting models. It may not provide dramatic increases in predictability.

Title 2.2: Online Aggregation for Large MapReduce Jobs

This system describes how to implement the aggregate query in online fashion. System takes processing takes huge time for estimate final result. For considering how much time can taken for mapreduce process in large-scale data large scale cloud computing can be based on payment. Reduce cost by using this big data. It is typically used to perform "pre-aggregation," which can reduce the amount of network traffic when the map and reduce steps are executed in a distributed environment. The user need not write the reducer and the combiner class.

Title 2.3: Fast data in the era of big data: twitter's Real-Time Related Query Suggestion Architecture

This paper provides a case study illustrating the challenges of real-time data processing in the era of "big data". We tell the story of how our system was built twice: our first implementation was built on a typical Hadoop-based analytics stack, but was later re-placed because it did not meet the latency requirements necessary to generate meaningful real-time results. Upon a cold restart, the frontend caches can serve the most-recently written results immediately without waiting for the backend. It is not sufficient

Title 2.4: Massive Scale-out of Expensive Continuous Queries

In this paper, it provides a stream splitting operator that enables such customized stream splitting. However, it is critical that the" stream splitting" itself keeps up with input streams of high volume. This is a problem when the stream splitting predicates have some costs. Therefore, to enable customized splitting of high-volume streams, we introduce a parallelized stream splitting operator, called parasplit. Parasplit was used to split the input stream in scsq-plr. The problem of scalable stream splitting is not handled by streaming MapReduce.

Title 2.5: Online Aggregation and Continuous Query support in MapReduce

In this demonstration, it described a modified MapReduce architecture that allows data to be pipelined between operators. This extends the MapReduce programming model beyond batch processing, and can reduce completion times and improve system utilization for batch jobs as well. A similar mechanism is also used to control how aggressively the

combiner function is applied. This map task does not immediately register the new spill file with the Task Tracker.

TITLE	YEAR	AUTHOR	METHODS	ADVANTAGES	DISADVANTAGES
Predicting the Present with Google Trends	2011	Hyunyoung Choi, Hal Varian	In this paper we show how to use search engine data to forecast near-term values of economic indicators. Examples include automobile sales, unemployment claims, travel destination planning, and consumer confidence.	It have used Google search data to examine how job search responded to extensions of unemployment payments.	It may not provide dramatic increases in predictability.
Fast Data in the Era of Big Data: Twitter's Real-Time Related Query Suggestion Architecture	2012	Gilad Mishne, Jeff Dalton, Zhenghua Li, Aneesh Sharma, Jimmy Lin	This paper provides a case study illustrating the challenges of real-time data processing in the era of big data". To tell about how our system was built twice: our first implementation was built on a typical Hadoop-based analytics stack, but was later re-placed because it did not meet the latency requirements necessary to generate meaningful real-time results.	upon a cold restart, the frontend caches can serve the most-recently writ-ten results immediately without waiting for the backend.	It is not sufficient
Massive Scale-out of Expensive Continuous Queries	2011	Erik Zeitler Tore Risch	This paper provide a stream splitting Operator, that enables such customized stream splitting. However, it is critical that the stream splitting itself keeps up with input streams of high volume. This is a problem when the	Parasplit was used to split the input stream in scsq-plr	The problem of scalable stream splitting is not handled by streaming MapReduce.

			stream splitting predicates have some costs. Therefore, to enable customized splitting of high-volume streams, to introduce a parallelized stream splitting operator, called parasplit.		
Online Aggregation for Large MapReduce Jobs	2011	Niketan Pansare, Vinayak Borkar, Chris Jermaine, TysonCondie	In this paper, it consider how online aggregation can be built into a MapReduce system for large-scale data processing. Given the MapReduce paradigm's close relationship with cloud computing(in that one Might expect a large fraction of MapReduce jobs to be run in the cloud),online aggregation is a very attractive technology.	This snapshot consists of all of the statistics that will be used by the Software to compute its estimate.	the user who killed the query early may not benefit directly.
Online Aggregation and Continuous Query support in MapReduce	2010	Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein	In this demonstration, to describe a modified MapReduce architecture that allows data to be pipelined between operators. This extends the MapReduce programming model beyond batch processing, and can reduce completion times and improve system utilization for batch jobs as well.	A similar mechanism is also used to control how aggressively combine function is applied.	the map task does not immediately register the new spill file with the task tracker.

3. CONCLUSION

Approximate answering approach for range-aggregate queries in big data environments. Data sampling is also well used in the field of distributed and streaming environments. Big data is nothing but unstructured, uncertain, real-time data that is present in a massive amount. Querying on such data is a bit difficult task, though there are varying technologies present in today's world. In this paper, in propose Balance partition technique and exact pattern matching technique that is useful for handling queries. Balance partition technique first divides big data into partition and store in respective partition. This partition contains indexing, which are used with an exact pattern matching technique for effective handling of queries.

In this paper, pattern matching technique is used for the retrieval of data. Partition algorithm is playing an important role for scattering of data according to data arriving on this sever. These partitions also contain an indexing system which is useful for analyzing the data. Query arrives at sever is partitioned into word. Pattern matching algorithms are used to process the relevant queries as quickly as possible. This idea willing to cover the data cube analysis and m: n problem of Fast Range Amalgamate Query technique. The map-reduce framework with pattern matching technique gives better access than any other system for query analysis.

References

[1] "Massive Scale-out of Expensive Continuous Queries", Erik Zeitler Tore Risch, 2011.
[2] "Online Aggregation for Large MapReduce Jobs", Niketan Pansare, Vinayak Borkar, Chris Jermaine, Tyson Condie, 2011.
[3] "Online Aggregation and Continuous Query support in MapReduce", Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein, 2010.
[4] "Predicting the Present with Go ogle Trends", Hyunyoung Choi, Hal Varian, 2011.
[5] "Fast Data in the Era of Big Data: Twitter's Real-Time Related Query Suggestion Architecture", Gilad Mishne, Jeff Dalton, Zhenghua Li, Aneesh Sharma, Jimmy Lin, 2012.

