Volume-03
Issue-09
September-2018

ISSN: 2455-3085 (Online)
RESEARCH REVIEW International Journal of Multidisciplinary
www.rrjournals.com [UGC Listed Journal]

A Study on Ad-Hoc-Data Processing in Cloud Computing Process

[1]Subhendu Chatterjee & [2]Dr. Suresh Dara

[1]Research Scholar, Department of Computer Science & Engineering, Sri Satya Sai University of Technology and Medical Science, Sehore, MP (India)
[2]Department of Computer Science & Engineering, B.V RAJU Institute of technology, Narsapur, Medak, Telangana (India)

ARTICLE DETAILS

Article History
Published Online: 07 September 2018

Keywords
Ad-hoc, Cloud computing, Cloud services

ABSTRACT

The most significant point of cloud computing is that the resources and data are accumulated into data centers on the internet. These days, the cloud services like IaaS, PaaS & SaaS, have been improved in execution as application execution environments are aggregated at several levels for sharing.

1. Introduction

The ad-hoc data is stored in cash registers. Then, this stored data is analyzed with the help of the time-series. Hence, the behavior like purchasing behavior of individuals is analyzed from this ad-hocdata. According to a report, about 7 million pieces per second are accumulated at cloud centers.

This ad-hoc data is not equivalent to that is obtained in reality because of the fact that much of the data is lost while moving to the cloud centers. Many research are going on in order to reduce this data leakage.

In today's world, several kinds of data are accumulated in a cloud environment as the cost of devices of information and communication technology is decreasing day by day. There is an urgent need to analyze this massive data so that it can be helpful for the business and society.

A new technology needs to be adapted as the quantity of data is so massive which is far more than tens of terabytes or tens of petabytes. Also, these days, social infra structure services run for 24 hours and 7 days a week. Hence, there is an urgent need to change the configuration of system dynamically.

Many laboratories are developing fundamental technologies for processing ad-hoc data in a cloud environment. A new methodology has been introduced to create cloud by aggregating data. So now there is a need to change the role of cloud from application aggregation to ad-hoc data aggregation and utilization. A new technology other than information and communication technology is needed to use this kind of ad-hoc data which is of more than tens of petabytes.

Now, the scenario of cloud environment has expanded from information & communication technology applications to business processes to innovation. The aim is to increase sales by identifying valuable information via data analysis aggregated into clouds.

The most significant point of cloud computing is that the resources and data are accumulated into data centers on the internet. These days, the cloud services like IaaS, PaaS & SaaS, have been improved in execution as application execution environments are aggregated at several levels for sharing.

Ad-hoc data processing is a powerful abstraction for mining terabytes of data. Systems for massive parallel data processing, such as MapReduce and Dryad allow Internet companies, e.g., Google, Yahoo, and Microsoft, to mine large web crawls, click streams, and system logs across shared-nothing clusters of unreliable servers.

The biggest feature of innovation is that the users don't know what to do which differentiates it from traditional ICT application. There are many methods to analyze ad-hoc data. The process of data analysis must be repeated a number of times from several prospective. Also, a processing having high speed and low cost is needed in all stages of development and operation.

2. Review of related literature

Jain et al. 2012 proposed a method,Pregel, which is used to implement a programming model. In this model, each node has its own input and transfers only some messages which are needed for the next iteration to other nodes.

R. Vernica et al. 2014 proposed a 3-stage approach for end-to-end set-similarity joins. They efficiently partition the data across nodes in order to balance the workload and minimize the need for replication. Wei Lu et al. investigate how to perform kNN join using MapReduce. Mappers cluster objects into groups, then Reducers perform the kNN join on each group of objects separately. To reduce shuffling and computational costs, they design an effective mapping mechanism that exploits pruning rules for distance filtering. In addition, two approximate algorithms minimize the number of replicas to reduce the shuffling cost.

J. Ekanatake et al. 2012 proposed a method,Twister, which is an incremented MapReduce runtime which supports

Repetitive MapReduce calculations efficiently. It is used to add an extra Combine stage after Reduce stage. Thus, the output of data moves from Combine stage to next iteration's map stage.

Robert et al. 2013 proposed another method called, HaLooP, which is quite similar to Twister. HaLoop is in fact, a modified version of the MapReduce framework which supports the iterative applications by adding a 'Loop Control'. It permits to save more input and outputs during iterations. There exists a lot of iterations during the processing of ad-hoc data.

D. Kossmann et al. 2012, presented four different architectures which were based on classic multi-tier database application architecture. These four architectures are: Partitioning, Replication, Distributed Control and Caching Architecture.

It is observed that different providers have different business models and different kinds of applications are targeted by them. For example, Google, mostly, launches small applications having light work load whereas Azure launches the applications which are efficient for medium to large services. These days, most of the cloud service providers utilize hybrid architecture. This hybrid architecture has the potential to satisfy the actual service requirements.

F. Cordeiro et al. 2013, proposed BoW method. In this method, MapReduce is used to cluster very large and multi-dimensional datasets. This method permits the automatic and dynamic communication between Disk Delay and Network Delay. MapDup Reducer is a MapReduce based system which has the capability to detect near duplicates over massive datasets effectively.

C. Ranger et al 2012, implement the MapReduce framework on a number of processors in a single device. Recently, B. He at al. 2011, develop Mars which is a MapReduce framework and is based on GPS. It enhances the efficiency of the system.

T. Nylael et al. 2012 proposed a sharing framework which is known as MRShare. MRShare is used to convert a new group that can be executed more effectively by aggregating tasks into groups and evaluating each group as a single query.

John et al. 2012, proposed a method to reduce the data transfer cost. This method divides a MapReduce task into two sub-tasks : Sampling MapReduce Task and Expected MapReduce Task. In first task, input data is obtained, keys are distributed and a good partition scheme is prepared. In second task, expected MapReduce task is used to perform the partition scheme to group the intermediate keys quickly.

3. Methodology

Ad-hoc data processing architecture was used for the current research work. Since much of the ad-hoc processing tasks programmers' uses are agreeable to incremental calculation, there is a hefty prospect to reprocess preceding computations and shun intermediary outcome. However, current ad-hoc data processing architectures require the programmer to overtly split modules into sub-modules to make pipelines.

For MapReduce, even these can abscond bigevents for use again on the table. In this research, we deal with these challenges by implementing a trendy ad-hoc data processing abstraction, MapReduce, over a distributed stream processor.

MapReduce compels programmers to mention two separable, parallel phases: map and reduce. In many cases, the whole MapReduce job can be done as in-network aggregate computation.

Thus,in spite of allowing data to a central point, the distributed stream processor orchestrates incremental map and minimize computations to work on unstructured data distributed. We demonstrate the benefits of continuous, incremental MapReduce by building and querying a distributed web corpus across multiple sites.

Further than minimizing the expenditure of downloading the web to a particular point, distributed swarming specifiesenhancedevents for evaluating web indexing. For example, giving web publishers control over intranet crawling lets them to deal with the presence of dynamic content, access restrictions, emerging content types (e.g., video), and privacy concerns when creating indices.

MapReduce implementations clearly control the corresponding implementation of the map phase, the alliance of all values with a given key (called the sort), and the parallel execution of the reduce phase.

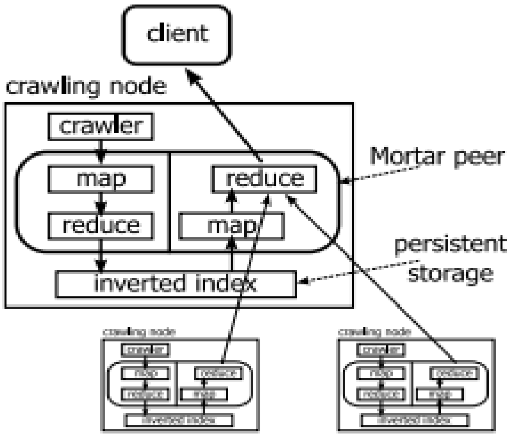

Figure 1: Incremental MapReduce producing a searchable inverted index across a set of distributed web crawlers. A client's continuous MapReduce job queries the index for a given set of keywords.

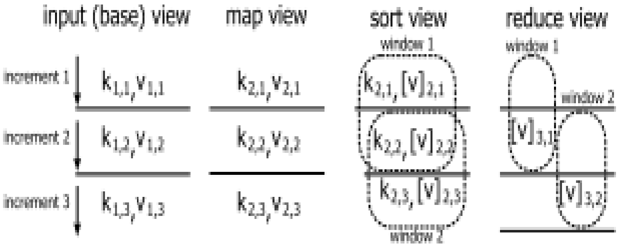

Figure 2: Incremental updates to views in MapReduce. Here the window range is two increments.

The system restarts failed tasks, gracefully dealing with machine failures and ensuring reliable operation even when operating across thousands of cheap PC's. However, current MapReduce implementations process data in isolated snap-shots.

4. Significance of the Study

For this current research, multiple crawlers discover displace areas of the web while an incessant MapReduce job builds each local index. The system converts user queries into continuous, in-network, MapReduce tasks to query the global index.

A hallucination module describes the outcome of user queries, the rate of growth of the index across the system, and the impact of node failure on outcomes. Experience with our prototype signifies that wide-area incremental MapReduce is an influential method for handling data in the cloud.

A MapReduce job breaks data processing modules into two phases: map and reduce. The map function works on individual key-value pairs, {k1, v1}, and outputs a new pair, {k2, v2}. The system creates a list of values, [v]2, for each key k2. The reduce function then creates a final value v3 from each key-value list pair.

5. Conclusion

Cloud environment is the better option to analyze ad-hoc data as it offers benefits like temporary availability of a large number of computational resources and cost reduction by allowing resources to share data.

In today's world, technology is growing at a very faster speed. A variety of data needs to be processed as the applications like social network analysis, semantic web analysis and bio-informatics network analysis are growing rapidly.

It is like a big challenge to analyze ad-hoc data. Several Governments and industries have shown their interest in ad-hoc data. This research work introduces several ad-hoc data processing techniques from system as well as application aspects. There are many big social environments like online shopping sites, social sites etc.

Companies need to track the activities of users. There are many issues like computing platform, cloud architecture, cloud database and data storage scheme. These issues need to be solved by analyzing ad-hoc data. In this research work, we discuss the data processing in cloud computing environments and issues & challenges related to this.

References

1. B. He, "The importance of 'ad-hoc data': A definition," 2011.
2. C. Ranger, "Ad-hoc data: science in the petabyte era," Nature 455 (7209): 1, 2012.
3. D. Kossmann, T. Kraska, and S. Loesing, "An evaluation of alternative architectures for transaction processing in the cloud," in Proceedings of the 2012 international conference on Management of data. ACM, 2012, pp. 579–590.
4. F. Cordeiro, J. Dean, S. Ghemawat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber, "Bigtable: A distributed structured data storage system," in 7th OSDI, 2013, pp. 305–314.
5. J. Ekanatake, "The hadoop distributed file system: Architecture and design," Hadoop Project Website, vol. 11, 2012.
6. Jain, "A survey of large scale data management approaches in cloud environments," Communications Surveys & Tutorials, IEEE, vol. 13, no. 3, pp. 311–336, 2012.
7. John Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2012.
8. R. Vernica, "Es2: A cloud data storage system for supporting both oltp and olap," in Data Engineering (ICDE), 2014 IEEE 27th International Conference on. IEEE, 2014, pp. 291–302.
9. Robert and R. Katz, "Chukwa: A system for reliable large-scale log collection," in USENIX Conference on Large Installation System Administration, 2013, pp. 1–15.
10. T. Nylael, "The Google file system," in ACM SIGOPS Operating Systems Review, vol. 37, no. 5. ACM, 2012, pp. 29–43.