International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 2, 2014. ISSN: 2348-6600
Reference ID: IJCS-039 PAGE NO: 215-219

Transpose-Minify Model for data processing in Cloud

S.Asha, Final MCA, T.Menaka, Final MCA, Dr.N.Nagadeepa, Ph.D,HOD
Department of Computer Applications, V.S.B.Engineerign College,Karur-636 111. India
ashasekar7@gmail.com,menagamcavsb@gmail.com,nagadeepa1012@gmail.com

Abstract— *The user can use the data and perform the operation in the cloud at any time. Thus the cloud gives highly security and efficient to use the internet based on their needs. In cloud computing has given many factors. The interesting things are hardware cost is very low, the storage area and power capacity are very high, and the growth of data generated by digital media, authoring web, scientific instrument, physical simulation, etc. The important in cloud computing is how it will effective to use and store the data and query and analyze the dataset these are most important challenges in cloud computing. In order to provide solution for this problem software framework. We use Transpose-Minify Framework. It is used for managing the data in effective way.*

Keywords-map reduces, data processing, transpose, minify.

1.Introduction

Umbrella term is named as cloud computing. To describe list of sophisticated on-demand computing services. It was originally provided by commercial providers, they are Microsoft, Google, and Amazon. The computing infrastructure denotes a model.

It can be viewed as cloud. The important principles in this model it provides Storage, Computing, and Software as a service. Raw computing and storage provide addition called cloud. Cloud computing provide wide range of software services. The development tools and APIs will also include the developer to allow them to build seamlessly scalable application on their services.

Indeed, cloud computing provide three services they are Software as a service (Saas), Platform as a service (Paas), Infrastructure as a service(Iaas) and also cloud provide some features they are customization, self-service, pay for usage (metering and billing), elasticity. Hybrid cloud, Public cloud, Private cloud these are deployment models which was provides by further cloud. The cloud offered important feature is user data anywhere in the world and which can operate the unknown machine in remotely by the user.

Now a day's large amount of data is produced by organization in day to day life. but among these the interest thing in cloud computing is that the cloud computing has motivated by many factors they are system hardware, low of cost, increase the computing power, storage capacity and massive growth in data size generated by digital media (images, videos, audio) web authoring, physical simulation, scientific instrument, etc.

Till the end how effectively store, query, analyze, and utilize these immense dataset is the still in main challenges in the cloud. To manage the data produced by the organization in effective way for that this paper a novel highly decentralized software framework implemented is named as Map Reducing Technology. To combine the cloud computing technologies to effectively store, query, analyze, and utilize the organizations data is the main aim of this paper.

This paper is structure as follows: Section II: discuss about Transpose-Minify Model. The data processing in the cloud is the frame work of software in minify model. In Section III: presents the existing and proposed system analysis. Section IV: contains the main features of transpose-minify framework. Section V: implements the sculptor-Serf architecture for the Transpose-Minify Framework. Section VI: discuss the Transpose Minify Framework. It is implemented using the Transpose and minify functions. Section VII: presents the some of the related implementations for the Transpose-Minify Framework done in the cloud and also it discuss about conclusions and future work.

2. Transpose-Minify Model

Many large-scale computing problems have been solved by a software framework called Transpose-Minify. Large set of data set can be processed by using this programming model.

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 2, 2014. ISSN: 2348-6600
Reference ID: IJCS-039 PAGE NO: 215-219

The Transpose-Minify contain two main functions. They are Transpose and Minify.

2.1 Transpose Function
The similar data items can be storing and searching called this Transpose Function.

2.2 Minify Function
The summary operation is performed by this procedure.
The Transpose-Minify provides many useful features such as simplicity, fault tolerance, and scalability. The cloud computing programming is the most powerful realization of data-intensive.
The traditional data intensive programming model for cloud computing has been easier-to-use, efficient, reliable replacement in this computing.
The data-center software track is proposed to form basis of this center. Many field the Transpose-Minify can be applied they are data and compute-intensive applications, machine language, graphic programming, multi-core programming.

3. System Analysis

3.1 Existing System

The traditional data intensive system was used for managing the large amount of data produced in organization in the past. While transferring the large amount of data to distant CPU it is not suitable for cloud computing due to the bottleneck of the Internet. The lack in scalability and there is no enough space to store large amount of data these are the drawbacks of traditional method. And also the query processing will not be supported.

3.2 Proposed System
In proposed system Transpose-Minify software framework is called as a novel approach. It is to use effectively and large data sets can be managed. The simplicity, fault tolerance, and scalability are the main features. In this model both computing and data resources are co-located, thus benefiting the service providers and minimizing the communication cost.

4.Main Features of Transpose-Minify Framework

4.1 Simplicity
The parallelization and concurrency control both are responsible for the Transpose-Minify runtime, and also it allow the programmers to easily distributed applications and design parallel.

4.2 Manageability
It provides two level of management
1. The input data have been managed and prepare the data to execute.
2. The output data also been managed and reduced data also been received.

4.3 Scalability
The performance of the Transpose-Minify potentially increases then the nodes also increase.

4.4 Fault Tolerance and Reliability
The data in the GFS are distributed on clusters with thousands of nodes. Thus if any nodes or hardware failures means it can be removed and simply installing the new node in their place. Moreover, Transpose-Minify, taking the advantage of the replication in GFS, It can be achieved by high reliability they are(1) When a host node is going off-line it will rerunning all task (completed or in progress), (2) rerunning the failed tasks on another node, and (3) when the task are slowing down launching backup task and causing a bottleneck to the entire job.

5. Transpose-Minify Implementation

The Transpose-Minify framework uses the sculptor- Serf architecture (Fig.1).

5.1 Transpose step

The input has been taken by sculptor node and it can be divided sub problems into smaller, and distribute them to serf nodes. In turn this is done by serf node again, leading to a multi-level tree structure. The smaller problems are process by serf node and the answer passes back to its sculptor node.

5.2 Minify step
The all sub problems answers has been collected by sculptor node and it combine in one way to form the output-thus the problem answer was originally trying to solve it.

5.3 The Function of Sculptor
Master is also called as sculptor and it is responsible for Querying the Name Node for the block locations, The slave can scheduling the task on which hosting the task's block and Monitoring the successes and tasks of the failure.

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 2, 2014. ISSN: 2348-6600
Reference ID: IJCS-039 PAGE NO: 215-219

5.4 The Function of Serf

Slave is also called serf, it can execute the tasks as directed by the master.

6. Implementation of Transpose-Minify

6.1 Steps

- The input can be prepared by user.
- Then the prepared input is send to sculptor of the architecture.
- Then the sculptor segments input and assigns to serf nodes.
- Then the serf nodes will process the input and produce the correct output.
- Then the serf node sends the out file to sculptor

For following the above listed steps the pivotal-appraisal pair is generated for Transpose-Minify functions:

- For Transpose (p1,a1)->list (p2,a2)
- For reducing (p2,list (a2))->list (a3)

The user program in the Transpose Minify library first split the input files into M pieces of 16 to 64 megabytes (MB) per piece. After it start many copies of program on a cluster. Among that one of them is "Sculptor" and rests of them are "Serf". Then the Sculptor is responsible for scheduling (assigns the Transpose and Minify tasks to the worker) and monitoring (monitors the task progress and the serf health). The sculptor assigns the task to an idle serf, when the Transpose task arise means, taking into the account the data locality. The corresponding input split content has been read by the Serf and emits a pivotal/appraisal pair to the user-defined Transpose function. So the first buffered in the memory are intermediate key/value pairs produced by Transpose function and then periodically written to a local disk, partitioned into R sets by the partitioning function. The location of these stored pairs to the Serf passes by the location of the sculptor which read the buffered data from the Transpose using remote procedure calls (RPC). So that all occurrences of the same key are grouped together it then sorts the intermediate keys. For each key, to minify the function the worker passes the corresponding intermediate value for its entire occurrence. Finally the output is available in R output files (one per Minify task).

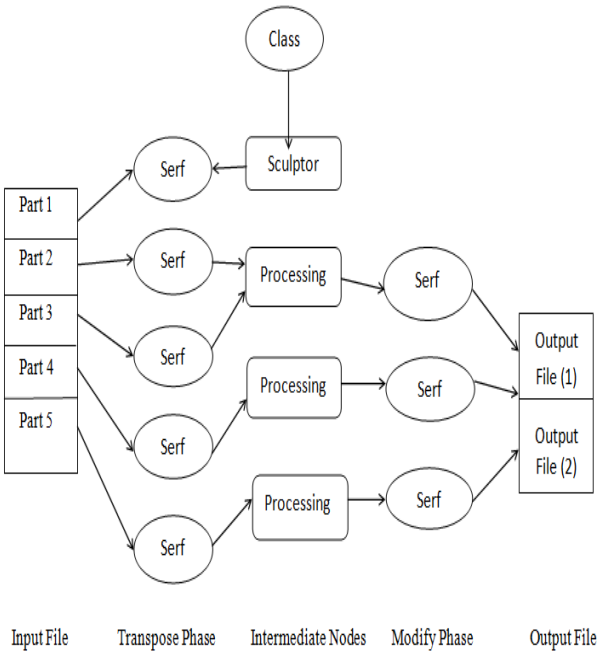

Figure.1 Sculptor- Serf architecture

7. Transpose-Minify Implementations for the Cloud

The Transpose-Minify framework can be implemented in various fields using the cloud computing technology.

7.1 Hadoop

The Hadoop common (7), file system ,RPC, and serialization libraries includes Hadoop core formerly and provides the basic services for building a cloud computing environment with commodity hardware.

The Transpose-Minify framework and the Hadoop Distributed File System (HDFS) are the two fundamental subprojects. To run on cluster of commodity machine Hadoop Distributed File System has been designed. It is highly fault tolerance and is appropriate for data-intensive; the application data provides the high speed access. The highly reliant on its shared file system is Transpose-Minify framework (i.e., it comes with plug-ins for HDFS, Cloud Store (15), and Amazon Simple Storage Service S3 (16)).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 2, 2014. ISSN: 2348-6600
Reference ID: IJCS-039 PAGE NO: 215-219

7.2 Disco

The Nokia (21) has been developed the Disco open-source Transpose-Minify implementation. The Disco core is written in Erlang, jobs in Python typically write by the users of Disco. Nokia research center has been started Disco as a lightweight framework for rapid scripting of distributed data processing tasks. In parsing and reformatting data, data clustering, probabilistic modeling data mining, full-text indexing, and log analysis with hundreds of gigabytes of real-world data Disco has been used successfully. Disco is based on the master-slave architecture. When the client send jobs to Disco master and it adds to job queue, and run them in cluster when CPUs become available. There is a Worker supervisor for each node. They are responsible for spawning and monitoring all the running Python worker processes within that node. The assigned task run by the Python and then it send addresses of resulting files to master through their supervisor.

7.3 Mapreduce.NET

The .NET platform realization of Transpose-Minify is Mapreduce.NET (22). To provide the support for a wide variety of data-intensive and compute intensive applications is aim (e.g., MRPGA is an extension of Transpose-Minify for GA applications based on Transpose-Minify.NET (23)). Transpose-Minify .NET is designed for the Windows platform, with emphasis on reusing as many existing Windows component as possible. The Transpose-Minify .NET runtime library is assisted by several components service from Aneka (24, 25) and run on WinDFS. Aneka is a .NET-based platform for enterprise and public cloud computing. In a public cloud environment it supports the development and deployment of .NET-based cloud applications, such as Amazon EC2. Besides Aneka, Reduce .NET is using WinDFS, a distributed storage service over the .NET platform. WinDFS manages the stored data by providing an object-based interface with a flat name space. Moreover, Map Reduce .NET can also works the Common Interface File System (CIFS) or NTFS.

7.4 Skynet

Ruby implementation of Transpose-Minify is a skynet (17, 26), created by Geni. Skynet is "an adaptive, self-upgrading, fault tolerant and fully distributed system with no single point of failure" (17). The plug-in based message queue architecture is heart of Skynet, this message queuing allowing workers to watch out for each other. If the work fails, another worker will notice and pickup that task. Currently there are two message queue implementations available: one is Rinda that uses Tuple space and another one is MY SQL. Skynet works by putting task on message queue that are picked up by skynet workers. After loading the code at startup the tasks execute skynet workers.

7.5 Grid Gain

Open cloud platform is a grid, developed in java. Grid Gain enables user to develop and run applications on private or public clouds. The Transpose-Minify paradigm is at a core of what Grid Grain does. It discuss about the process of splitting an initial task into multiple subtask, executing these subtask in parallel and aggregation (reducing) results back to one final result. In Transpose-Minify implementations new features have been added in Grid Grain they are: distributed task session, checkpoints for long running tasks, early and late load balancing, and affinity co-location with data grids.

8. CONCULSION AND FUTUREWORK

In this paper introduces the Transpose-Minify Framework Which is important programming model for next-generation distributed systems, namely cloud computing

In this paper the Transpose-Minify model presented different impacts in computer science discipline, along with different efforts around the world. The implementation of Transpose-Minify has been a lot of effort in development, still there is more to be achieved in terms of Transpose-Minify optimizations and implementing this simple model in different areas. The future work of this paper is using the Transpose-Minify Framework it performing the optimization.

REFERENCES

[1]. I. Foster, Yong Zhao, I. Raicu and S. Lu, Cloud computing and grid computing 360-degree compared, in Proceedings of the Grid Computing Environments Workshop (GCE '08), 2008, pp.

[2].A. Szalay, A. Bunn, J. Gray, I. Foster, I. Raicu. The Importance of Data Locality in Distributed Computing Applications, in Proceedings of NSF Workflow, 2006.

[3]. A. S. Szalay, P. Z. Kunszt, A. Thakar, J. Gray, D. Slutz, and R. J. Brunner, Designing and mining multi-terabyte astronomy archives: The Sloan Digital Sky Survey, in Proceedings of the SIGMOD International Conference on Management of Data, 2000, pp. 451_462.

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 2, 2014. ISSN: 2348-6600
Reference ID: IJCS-039 PAGE NO: 215-219

[4]. J. Dean and S. Ghemawat, MapReduce: Simplified data processing on large clusters, Communications of the ACM, 51(1):107_113, 2008.
[5]. S. Ghemawat, H. Gobioff, and S. T. Leung, The Google File System, in Proceedings of the 19th ACM Symposium on Operating Systems Principles, LakeGeorge, NY, October, 2003, pp.

Published by SK Research Group of Companies (SKRGC).