Novel Dynamic and Scalable Storage Management Architecture

Kunal Kumar
Dept of Information Technology
LNCT
Bhopal, India

Sushil Kumar
Dept of Information Technology
LNCT
Bhopal, India

Manish Shrivastava,PhD
Dept of Information Technology
LNCT
Bhopal, India

ABSTRACT

In the current scenario, storage management of Big Data is imposing concern for Grid Computing environments, as a large scale distributed computation System which can resolve the problem of resource sharing. In traditional approach there is high-performance computing machine consisting of dedicated servers that are used to store data storage and resource discovery. In this paper, It is proposed to be an architecture Novel Dynamic and Scalable Storage Management Architecture for Big Data Management. This allows the grid oriented storage machine to share the resource and storage space. It has a retention period to resource discovery whenever data is communicating with the virtual storage.

Keywords

GOS, SET, Data Locality, MAPE.

1. INTRODUCTION

In the previous two decades, the endless development of computational power has produced a devastating flow of data. Data is spreading at the huge amount of speed making it so difficult to handle such huge amount of data (exabytes or petabytes) [1]. Data is growing in large volume, because the organization continuously captures the collective data for decision making. Social networking also stores the huge amount of information of their respective user.

Big Data can be defined with the following properties:

Volume

Big Data is producing in huge volumes, it includes the traditional data, structured data and semi structured data [2]. Social networking site such as Facebook, serves 570 billon page views per month. Google stores 3 billon new photos every month and handles 25 billons pieces of content as Google's search and ad business [3, 4].

Velocity

Big Data's velocity deals with the speed of data come from the various resources. This characteristic of Big Data is not being limited to the incoming data and also the flow of data. Sensors devices had constantly moving to database and this is not so limited to handle enough [5, 6, 7].

Complexity

It is quite complex to maintain link, match and transfer data from the system to the respective sources. It is also necessary to connect and correlate relationship properly, whenever multiple data link is there or data can be out of control [8].

Grid is flexile, protected, integrated resource sharing among progressive collection of individuals, institution and resources. The target is to virtualized resources and allows users and application to access resource in transparent manner [9]. Storage Grid is a model for expanding and organizing storage resource geographically distributed across multiple system and network and making prominent use of available storage capacity.

Storage Resource Broker (SRB) facilitates shared collection that can be distributed across the numerous organizations [10]. The SRB provides some Grid functionality including a Virtual Organization (VO) structure for data and information, handling a multiplicity of platform, resources and data type, seamless authorization and authentication to data and information which are stored in distributed sites. DataCutter is middleware infrastructure that enables processing of wide range scientific datasets stored in archival storage systems across the environment [11]. DataCutter provides support for subsetting of datasets through multi-dimensional range queries, and application specific aggregation on scientific datasets stored in an archival storage system [12].

A typical Grid environment which is a VO consisting of a Monitoring and Discovering System, a Certification Authentication centre and Grid Service Providers. All grid services are registered in Monitoring and Discovering System. Network based systems such as Network Attached Storage (NAS) and Storage Area Network (SAN) offer a robust and easy method to control and access large amounts of storage. With the steady growth of client access demands and the required data sizes, it is a challenge to design an autonomous, dynamic, large-scale and scalable storage system which can consolidate distributed storage resources to satisfy both the bandwidth and storage capacity requirements. Grid Oriented Storage (GOS) is build to build a distributed and huge scale storage system to preserves growing demand of bandwidth and capacity [13]. The Novel Dynamic and Scalable Storage Management (NDSSM) architecture to organize Grid Oriented Service (GOS) devices into a large –scale and geographically distributed data storage system to meet the requirements imposed by all kinds of Grid applications.

2. DATA STORAGE MANAGEMENT OF BIG DATA

In this section it states that the evolution of the BIG data management within the Grid and investigation the various implementations of the data Grids that are in place today as well as those currently emerging. Current changing Electronic data management has a vast and huge history dating back to the 1950's, many data management systems have tried to make their way into the mainstream of information technology. Some of them were successfully than others; hierarchical, network, object, and in-memory are only a few examples. The most successful data management system has been the relational data management technology.

3. PROPOSED ARCHITECTURE

The proposed architecture named "Novel Dynamic and Scalable Storage Management (NDSSM)" to organize Grid storage devices into a large-scale and geographically distributed storage system to meet the requirements imposed by all kinds of Grid applications.

This architecture divides Grid storage devices into multiple geographically distributed domains to facilitate the data access locality. The architecture consists of two levels. The bottom level adopts multicast to achieve dynamic, scalable, and self-organized physical domains. The method significantly simplifies the intra-domain storage resource management. The upper level is a virtual domain that consists of geographically distributed and dynamic agents selected from each physical domain. In virtual domain we proposed a three tier architecture; process, store and archive.

Data Locality: Data locality is a measurement of data that can be selected, retrieved, compactly stored, and reused for subsequent accesses [13]. Basically, there are two types of data locality: temporal and spatial. Temporal locality denotes that the data accessed at one point in time will be accessed in the near future. Temporal locality relies on the access pattern of different applications and can therefore change dynamically. Spatial locality defines that the probability of accessing data is higher if the data near it was just accessed. Unlike the temporal locality, spatial locality is essential to managing the data by a storage system, and is relatively more stable and does not depend on applications, but rather on data organizations which is closely related to it. Reshaping access patterns can be employed to improve temporal locality [14]. Data reorganization is normally adopted to improve spatial locality. Many research efforts have been invested in exploiting the impact of access pattern and data organization of applications on the data locality to achieve performance gains. Fig -3.1 illustrates the novel DSSM architecture.

Domain Division: Choosing a suitable criterion to form GOS nodes into domains is an important factor of the NDSSM architecture. NDSSM employs the distance criterion in a geographical way to divide GOS nodes into multiple domains to facilitate the data access locality and limit the amount of communication traffic seen by each node with performance guarantee. GOS devices belonging to the same geographical area (e.g. the same enterprise or the same LAN) are formed into a domain, because the GOS devices in the area are normally placed geographically close.

GOS nodes divide into multiple domains based on:

Distance

Quantitative (bandwidth)

Qualitative (performance analysis)

With a domain based GOS Network, the scalability is achieved from a two-tiered architecture, namely, intra-domain and inter-domain. Within a domain, each GOS device is equal in functions and capabilities, and can leave or join the domain dynamically. The novelty is achieved by the dividing the virtual area storage and having a mechanism to gain the appropriate data from the resource by the concurrent user. Any node can communicate with anyone else directly. Each domain selects a domain agent for the domain members to cooperate with other domains according to domain formation algorithm. When a domain agents wants communicates with virtual domain, it first checks the data is available in the storage area, if data is available, it will access otherwise it will search in the archive.

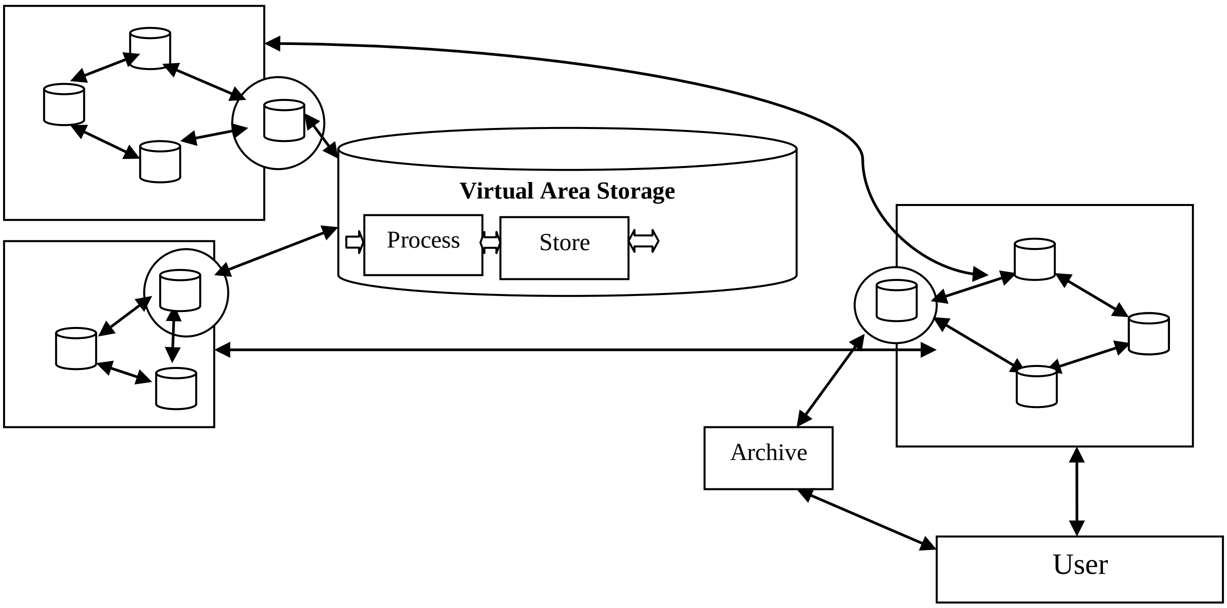

Fig 3.1 Modified DSSM

4. PROPOSED ALGORITHM

It is to be proposed that this algorithm achieves scalability in intra domain as well as inter domain. It has feature of self discovery and self organizing capable to multicast message leave and join message between GOS.

4.1 Algorithm Formulation at Bottom Level

NDSSM architecture organized as a scalable storage system. Assign with a multicast IP address to provide a single entry point the storage space. Each GOS device maintains an Adjacent Information Table (AIT) which keeps a list of all active GOS devices with their related resource information. Information consists of an IP address, reminder storage capacity, processing power. A data structure is adopted to describe the entry of AIT.

Algorithm#1: When a new GOS device wants to join the domain:

Step1: The new GOS sends "JOIN" signal to online GOS (the device sends multicast "JOIN" request of its coming and the respective local resource information such as the processing power, storage capacity, and waits until the response of other online GOS devices).
Step2: The online GOS receives "JOIN" request (the online GOS devices which grasp the "JOIN" request adding the oncoming device's information to their AIT).
Step3: The online GOS sends "ACK" to new GOS (online GOS devices send an "ACK" an acknowledgement signal back to the probing GOS device using unicast).
Step4: Create AIT (the oncoming device constructs its own AIT in terms of the ACK messages).
Algorithm#2: Leaving of a GOS device
DSSM is based on stable and trusted GOS devices; the leasing of a GOS device is normally caused by maintenance, upgrade and other reasons.
Step1: Send "LEAVE" (multicast leave message in the domain)
Step2: Receive "LEAVE" (once the online GOS devices receive the message, the devices delete the leaving device's information).
Step3: Update AIT
Step4: Repeat Step2 (if message is not received then the device is assumed to have failed and then the data is access from the Virtual Area Storage and then update their AIT.)
4.2. Algorithm Formulation at Upper Level
Algorithm#3: Select agent from particular physical domain
Step1: Select MAX [PP] (select highest processing power agent within a domain)
Step2: Check Data (send a "CHECK" to storage, if data is present it will accessed from store and otherwise it will search in the archive.)
Step2: Repeat Step1 (if fail to select highest processing power agent)
Step3: Compare agent to another GOS
Step4: Repeat Step1 (highest processing power will always reselect as an agent)
Step5: In case of same processing power unchanged (the online GOS device receive message from virtual store). .

5. RETENTION PERIOD FOR BIG DATA

The ability to maximize capacity while minimizing cost is critical for a storage platform operating at big data scale. A big data retention period will support data is to be readily and cost-effectively scaled even as the enterprise storage demands grow dramatically. The ability to maximize capacity while minimizing cost is critical for a storage platform operating at big data scale. The fixed capacity of traditional storage systems and the need to balance future storage needs with current capital expenditures, many businesses are forced to migrate their data to new systems every few years. Unfortunately for these enterprises, this migration is expensive and time-consuming [15].

The discrete systems are fundamentally disconnected from one another; they immediately become data silos that inhibit an enterprise's ability to see the big picture. To be able to fully exploit the opportunities of big data, companies must be able to access and use all of their data without ad-hoc interventions. Unlike conventional storage devices, a big data storage platform bridges legacy storage silo, rather than simply adding yet another system to the mix. Before big data, figuring out how to improve data access for globally-distributed users presented a significant challenge to IT organizations. Now, the data itself has become globally distributed, transforming the IT challenge into one of making it readily available to users and applications across the global enterprise.

A traditional centralized approach to data management is no longer practical in the age of big data, because data volumes are too large, WAN bandwidth is too limited, and the consequences of a single point of failure are too costly. A big data storage platform must be able to manage data that is distributed across the global enterprise as a single, unified pool. The application of Big Data for network planning and optimization was also seen as a key initiative, given a high ranking by 49 per cent of respondents overall and 46.9 per cent of operators. This echoes discussions that we had with industry over 2013, in which some players suggested that network complexity rather than bandwidth might fast be becoming the barrier to network growth [11, 12].

Managing data is a key priority for businesses today. The proliferation of data, however, often creates a problem of over-retention; this is not always addressed by business records policies. This creates the need for better data visibility for the user, who works with clients to construct records management policies. It is paramount to store only relevant data for a relevant period; otherwise, researching information can become ineffective. Also, one of the biggest issues that organizations are facing with the voluminous amounts of unstructured and semi-structured data it creates is to mine all the data [16].

It is advisable is to develop a data retention policy as well as date retention period as part of a paradigm shift in data management for large-scale dataset collections; such a document should address and account for vast volumes of data that a business produces, stores and processes. Establishing a data retention policy for retaining information is important for operational or regulatory compliance as well as for meeting legal and business data archival requirements.

Here there is an example SET Index for big data, because in stock exchange there is various types of message transfer is going to placed. The Thai stock market formerly known as the Security Exchange of Thailand. In 1991, its names were changed to Thai Stock Exchange of Thailand (SET) [17]. The index of the stock exchange of Thailand is known as SET index. SET index is composite market capitalization-weighted price index, which compares the current market value of all listed common stock with its value on a single base date when the stock exchange was established. In this experiment, the data were grasping from the various reliable sources, the Bank of Thailand, which contain the historical data of SET index [18], the Dow Jones index (New York), the Nikkei index (Japan), the Hang Seng (Hong Kong), and Minimum Loan Rate (the exchange rate of the Thai Baht and the US dollar, daily effective over-night federal fund rates in the USA). Since the raw data is from different stock market in the various countries, there trading days and holidays was different. So, those gaps were filled with the data from the previous day with no significant differences. The assumption underlying this study was that the missing data on non trading day will be fulfilled with the previous data. The MAPE (Mean Average Percentage Error) of two year is more convenient and smoother with respect to MAPE of one year [18, 19]. The longer period of training data can yield lower error, except the lowest error level. The error of your data is little bit lower than two years training data, but it is not significant.

In terms of trying to produce the right policy, it may be best to focus on the ability of large data sets to yield results that outweigh the legal and privacy concerns, in addition to determining the retention time, archival rules, data formats, and the permissible means of storage, access, and encryption, for instance. Overlapping policy areas may cover matters on growth and innovation, to add insight to the retention and management capabilities. For visualizing and interacting with data we required a retention period. We propose the retention period of Big Data Management is 2 years. Because of, before this time period, data is not relevant and it's not worthless to the management of the data management.

So there has to be a retention period for Big Data. Data before the period should be in the compressed form. So that searching should be faster. In case, someone needs the previous data, there is a mechanism to decompress them.

6. CONCLUSION

In this research paper, it is proposed that novel dynamic and scalable storage management architecture, based on an existing Grid environment, we propose and design a novel DSSM architecture which organizes GOS devices into different domains to enhance the data access locality in terms of geographical and distributed areas. Storage expansion is achieved by simply adding services. The novel DSSM architecture avoids the hierarchical or centralized approach of traditional Grid architecture, eliminates the flat flooding of unstructured P2P system, and provides a dynamic storage pool in Grid environments. Retention periods of Big Data, provides a mechanism for searching the data faster and provides a concept of regaining the data by the decompressing the data. The proof-of-concept prototype gives useful insights into the architecture behavior of DSSM architecture

7. REFERENCES

[1] Matthew Smith, Christian Szongott, Benjamin Henne, Gabriele von Voigt, "Big Data Privacy Issues in Public Social Media", IEEE, 6th International Conference on Digital Ecosystems Technologies (DEST), 18-20 June 2012.

[2] Sam Madden, " From Databases to Big Data", IEEE, Internet Computing, May-June 2012.

[3] "Big Data: science in the petabyte era," Nature 455 (7209):1, 2008.

[4] Stephen Kaisler, Frank Armour, J. Alberto Espinosa, William Money, "Big Data: Issues and Challenges Moving Forward", IEEE, 46th Hawaii International Conference on System Sciences, 2013.

[5] Douglas and Laney, "The importance of 'Big Data': A definition" ,2008

[6] BIG DATA: http://en.wikipedia.org/wiki/Big-data

[7] Sachchidanand Singh,Nirmala Singh," Big Data Analytics", International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, India, Oct. 2012.

[8] Michael Feldman," The Big Data Challenge: Intelligent Tiered Storage at Scale" in Actionable Market Intelligence for High Performance Computing.

[9] Ian Foster ,Carl Kesselman, Steven Tuecke, " The Anatomy of the Grid: Enabling Scalable Virtual Organizations" , Intl J. Supercomputer Applications, 2001.

[10] Storage Resource Broker: <http://it.toolbox.com/wiki/index.php/Storage_Resource_Broker>.

[11] DataCutter,<http://www.cs.umd.edu/projects/hpsl/ResearchAreas/DataCutter.htm>.

[12] M. Beynon, R. Ferreira, T. Kurc, A. Sussman, J. Saltz, DataCutter: Middleware for filtering very large scientific datasets on archival storage systems, in: Proceedings of the 17th IEEE/8th Goddard Conference on Mass Storage Systems and Technologies, USA, 2000, pp. 119–133.

[13] Yuhui Deng a,*, Frank Wang a, Na Helian b, Sining Wu c, Chenhan Liao," Dynamic and scalable storage management architecture for Grid Oriented Storage devices", Parallel Computing 34 (2008) 17–31.

[14] Ajay Kumar and Seema Bawa, "Distributed And Big Data Storage Management In Grid Computing", International Journal of Grid Computing & Applications (IJGCA) Vol.3, No.2, June 2012.

[15] Changqing Ji, Yu Li, Wenming Qiu, Uchechukwu Awada, Keqiu Li," Big Data Processing in Cloud Computing Environments", International Symposium on Pervasive Systems, Algorithms and Networks,2012.

[16] S. Rimcharoen, D. Sutivong and P. Chongstitvatana, "Prediction of the Stock Exchange of Thailand Using Adaptive Evolution Strategies," Tools with Artificial Intelligence, 2005. ICTAI 05. 17th 2005.

[17] S. Chaigusin,, C. Chirathamjaree, and J. Clayden, "Soft computing in the forecasting of the stock exchange of Thailand (SET)," Management of Innovation and Technology, 2008. ICMIT 2008. 4th, 2008.

[18] Phaisarn Sutheebanjard, Wichian Premchaiswadi "Determining the Time Period and Amount of Training Data for Stock Exchange of Thailand Index Prediction" IEEE 2010 359-363

[19] C. Worasucheep, "A New Self Adaptive Differential Evolution: Its Application in Forecasting the Index of Stock Exchange of Thailand," Evolutionary Computation, 2007. CEC 2007, 2007.