Multilevel Threshold Secret Sharing Scheme to Secure MapReduce Computations in Cloud Computing Environment

M. Eyalarasi, P. Shanthi and A. Umamakeswari
School of Computing, SASTRA University, Thanjavur - 613401, Tamil Nadu, India; eyalarasi20@gmail.com, shanthip@cse.sastra.edu

Abstract

Objectives: Cloud computing has evolved in recent days and is applied in various fields for effective resources and infrastructure in a distributed environment. Data analysis is the core functionality in cloud computing where large amount of data called Big Data is processed over clusters. Methods: MapReduce is one of the solutions for handling big data in the cloud environment because of its scalability and fault tolerance in a phased manner. Multiple data sets are joined to do complex data analysis for computation on certain aggregates. A common problem is whether MapReduce could be customized to get a scalable system, when the jobs are split and reduced. Also most of the systems do not consider the issue of security in MapReduce phases. Findings: The proposed solution uses multilevel threshold secret sharing to perform MapReduce operations providing secure processing. The solution extends MapReduce framework to improve security and also results in higher efficiency. The mechanism presents lower overhead costs when compared to the existing ones and has essential application in Big Data cloud environment. Applications: These approaches are lower cost and higher efficiency in cloud environment.

Keywords: Big Data, Cloud Computing, MapReduce, Security

1. Introduction

The technology press has been specialized in uprising “cloud computing,” a concept that entails connecting of huge number of processors operating in parallel to solve computing issues. This requires building a data center comprising large number of huge servers. Many tools have evolved to program such a clustered model. MapReduce is one such tool, a good choice for many problems as it provides an easy model that allows users to express relatively complicated distributed programs. Presently interest in MapReduce model is commercially and academically high and it is natural that MapReduce systems may put back the parallel DBMSs. The database is the first accessible and is available commercially from nearly twenty years ago. As robust, superior computing platforms, they provide a higher level programming set that is naturally parallelizable. Though it appears that MapReduce and the database management systems are totally different, it is feasible that parallel tasks can be done either by as a group of query or a cluster of MapReduce jobs.

1.1 Cloud Computing

Cloud computing is an evolving paradigm that provides access for on-demand networks and configures the computing resources (e.g. network, server) that may be fast provisioned. Also resources are released without provider interaction. The client can provision computing capabilities, such as server storage and the network when required automatically without need for human interactions. Pool of computing resources is served to multiple customers using multi-tenant models. Clients are given completely
different virtual and the physical resources automatically on demand. Location independency shows that the client typically has no knowledge and control over the precise location of the provided resources, however could specify the location at a higher level.

### 1.2 Big Data

Big Data associates degree extensive term for any assortment of data sets that are so massive and difficult to method them using traditional data processing applications. Then the challenges are storage, transfer, analysis, visual images. Data sets are huge due to the additional information derived from analysis of one huge dataset of connected data, as compared to separate small set with the constant total of the amount of data permitting correlations to be established to “spot business trends, forestall diseases and combat crime and so on”. Multiple data sets are joined to do complex data analysis for computation on certain aggregates. A common problem is whether MapReduce could be customized to get a scalable system, when the jobs are split and reduced. Most of the systems do not consider the issue of security in MapReduce phases. Hence this paper is to find a solution using multilevel threshold secret sharing to perform MapReduce operations thereby providing secure processing. The remaining section of this paper is sorted out as follows. Related work is discussed in Second Section. The system model is formulated in Third Section. Experimental results and future works are examined in Fourth Section. Conclusions are provided in Fifth Section.

In\(^1\) proposed that processing data is a simplified manner on huge cluster to a large variety of real world jobs using MapReduce paradigm. Here the users mention computation in cost of map and reduce functions, handles the machine disorders by parallelizing computation across huge-scale clusters of machines automatically and also schedule communication of inter-machine for making effective usage of disks and network. In\(^2\) proposed that relational data processing on huge clusters in simplified method using MapReduce-merge framework. This enables that ease integration of scalable distributed applications to operate a massive cluster. The existing framework won’t support directly processing multiple related heterogeneous datasets. Their work improved the framework by adding a merge phase. This method shows relational algebra operations and also constructs various join algorithms. In\(^3\) introduced Clustera, an integrated computing environment and data management system. This mainly differs from old cluster management systems which target unique sort of workloads methods. Experimental results show that the clusters have best scale-up property for processing and also deliver performance comparable with MapReduce process. Further it supports high throughput jobs than other methods. In\(^4\) proposed the storage of huge data (warehousing) over a MapReduce paradigm. In this method the large size of data are gathered and evaluated in industry for the business intelligence. Though the traditional data storage solutions are valuable, this method is useful to store and evaluate highly massive datasets on trade good hardware. Further, in\(^5\) proposed a new Not-so-foreign language (pig Latin) for processing the data for the ad-hoc analysis of largely huge datasets, especially to evaluate terabytes of internet based data collected every day. Pig Latin described here as a new language which fit in sweet point between the SQL declarative style and low-level processing way of MapReduce. Then\(^6\) discussed the future high performance based database systems, by using parallel database systems. This work mainly surveys the methods using some systems and reviews about recent mercantile and research systems. This process has evolved from the utilization of hardware to software dataflow architecture in parallel manner. Following the previous concept, in\(^7\) presents foremost performance analysis of Gamma, their design and development techniques. This mainly employs query processing techniques for dataflow. This work explains how the parallel processing could be pressurized with lowest control elevation through a combination of hashing and pipelining of data between processes. A comparison based approach has been discussed by in\(^8\) for large scale data evaluation. Here they describe and compare both parallel SQL DBMS and MapReduce framework for large scale data analysis, in order to measure the performance of the system and time consumption for execution of each tasks. In later\(^9\) discussed based on the previous concept with MapReduce and friends or foes. This is mainly focused on the large data warehouse system (e.g. Facebook data) are implemented using the MapReduce technique rather the DBMS. This work also argued that the performance of MR is best suited for massive datasets than that of parallel DBMS, mainly in the case of Extract-Transform-Load (ETL) of huge data. In\(^10\) proposed a model, which extracts useful information from huge amount of data, aggregate the result from output and present it to the user. This method concentrates on Big Data, where the extraction of infor-
mation from huge amount of data is difficult. Proposed
method is based on data partitioning and aggregation
process that has implemented with less error rate and high
accuracy of data. When discussed a MapReduce frame-
work which introduces the parallel data processing using
the method Bottom-Up Generalization (BUG) for large
scale data optimization problem. Proposed model based
introduction of new generalized nodes after a number
of generalization rounds and it will not affect the other
nodes in return. This approach results in high security,
minimum data loss compared existing methods.

2. System Model

The functions performed by the cloud are dispersed geo-
graphically and it is hard to execute experiments in all the
dispersed units. A malicious computer possibly im-
personates or declares as a Mapper (Reducer). Consequently,
impersonated nodes initiate MapReduce operations to
achieve the spoofing attacks. To overcome this problem
a novel data-aware framework is proposed for Big Data
applications that provide security by exploiting Shamir’s
threshold secret scheme as well as concentrates on
completing the works in an efficient manner. It performs
MapReduce efficiently and this proposed mechanism sig-
ificantly improves the MapReduce jobs finishing time
and saves major part of CPU execution time. When the
works are completed, the intermediate results from the
mappers are submitted and finally the results are col-
lated. Collated result is stored in cloud and the complete
execution is saved. This framework causes less overhead
to the users. By using the map and reduce functions bet-
ter results are achieved. Thus the data is secure and the
MapReduce functions are performed optimally.

2.1 Proposed Architecture

The proposed architecture shown in Figure 1 is based on
the map and reduce operations. Information available is
huge and only powerful data tools have the ability to col-
late, classify and cluster such data stored in distributed file
systems. The common model is to map the data to servers
and process the data. – MAP PHASE. Next is to collect the
resultant information and collate (aggregate) it – REDUCE
PHASE. Loading the data is the first process. The data vary
from small size to large size. The following are the steps
involved in the processing the dispersed huge jobs.

- Mapping stage to split and assign data to mappers.
- Worker stage to process data.
- Reducer stage to aggregate the intermediate data
  from the worker.
- Finally the result is stored in cloud and the ex-
  ecution is saved.

Figure 1. MapReduce architecture showing data flow.

2.2 Data Loading

Data is collected and stored into databases. The data
sources are connected to the cloud application server. In
reality they are not interconnected to every alternative
however connected with the main application itself. The
main purpose of exploitation of the datasets is enhancing
security by splitting and storing the data into four parts
and storing each in a separate entity in the database.

2.3 Data Split to Nodes

The data split and subsequent encryption processes are
discussed and is shown in Figure 2. Further the mode
of record extraction, reduce and joining the individual
record to form meaningful data are also done. This pro-
posed system uses symmetric encryption for securing
operations on data. Master node splits the data and assigns
the tasks to mappers. Further the master node generates
a Secret Key (SK) and splits into n shares and distrib-
utes them to the mappers at the first level. The shares
are further divided into sub-shares at the second level to
improve security. The threshold value is chosen dyna-
mically. Master node chooses an arbitrary polynomial
\( f(x) \) and indicates \( f(0) = SK \mod p \), where \( p \) is arbitrary prime.
By implementing \((t,n)\) threshold scheme, the system
ensures that at most \( t \) mappers having the shares of the
secret regenerate the secret by Lagrange interpolation and
can perform the operations securely. The data of any one
dataset even if hacked will be no use as without the other
counterpart data stored in the database.
Multilevel Threshold Secret Sharing Scheme to Secure MapReduce Computations in Cloud Computing Environment

2.4 MapReduce tasks

MapReduce jobs are based on the inputs from the clients. The huge data is split into the mappers and the mappers are sending the tasks to workers. Figure 3 shows the processing of jobs. The reducer is used to aggregate the result from the mappers. Mapper delivers the intermediate results to the reducer. Reducer before performing the task checks the integrity to improve security.

2.5 Aggregation phase

This module focuses on aggregating the results from the worker. Other processors in the system are not allowed to hack the execution of particular set of jobs. The reducer checks the integrity of data by comparing the hash. Thus delivered data integrity is validated and keeps it secure from impersonation attacks. These jobs allocated to the server are then aggregated.

3. Experimental Results

Figure 4 shows the comparison of MapReduce MR and Secure MapReduce SMR model. The experimental result shows that when the number of nodes (mapper/reducers) increases the time to complete the MapReduce processing incorporating security functions increases steadily. Thus module overcomes the security issues such as spoofing attacks and impersonation. Pipelining method is used for secure MapReduce operations and thus speed of execution of jobs is improved.

4. Conclusion

The MapReduce is a powerful process used to composite data analytical tasks on huge clusters. The MapReduce introduces the aggregation model using joining operations. This model allows the users to identify the data, join the datasets and collect it. It thus processes multiple jobs in easy and fast manner. This framework causes less overhead. By using the map and reduce functions better results are achieved. Thus the data is secure and the MapReduce functions are performed optimally. As future work, this can be extended by applying the proposed scheme to deal with attacks in cloud environments or various networks to strengthen the platform security.

5. References