REVIEW PAPER ON SCALABLE BIG DATA PROCESSING ON CLOUD COMPUTING

[1]Gitanjali Sinha,[2]Dr. Asha Ambhaikar

[1]Ph.D. scholor,[2]Professor & HOD (CSE) and Dean Student Welfare

[1]Department of Computer Science,
[1]KalingaUnivercity Naya Raipur ,(C.G.),India

Abstract : Large scale data analytics is emerging as a huge consumer of computational resources due to its complex, data-hungry algorithms. Especially graph/networked data analysis are becoming increasingly important for solving multiple problems in diverse fields. As these problems grow in scale, parallel computing resources are required to meet the computational and memory requirements. Notably, the algorithms, software and hardware that have worked well for developing mainstream parallel applications are not usually effective for massive-scale data from the real world, which exhibits more complex structure. Research into large scale data processing is currently at a fragmented stage.

***Index Terms*: Parallel Computing, Scalability, Big Data, Cloud computing etc.**

I. INTRODUCTION

Scalability has long been a distress for business, company or organization decision-makers, but now it's taking on new scope. The Information age has established beyond our wildest dreams, and our principles need to evolve with it. Big data analytics is becoming increasingly tangled with domains like industry intelligence, customer relationship management, and even diagnostic medicine. Enterprises that want to expand must incorporate growth-capable IT strategies into their operating plans[1].

"A **solid base for scaling Big Data communications and infrastructure that helps us to address each crucial factor associated with optimizing feat in scalable and dynamic Big Data clusters[2]."**

Here we explore some insights and techniques proven with all types of database engines and environments, including SQL, NoSQL, and Hadoop. Two start-to-finish case studies walk you through planning and implementation, offering specific lessons for formulating your own scalability strategy[3]. Those are

- Understanding the genuine reasons for database execution debasement in the present Big Data situations.
- Scaling easily to petabyte-class databases and past.
- Defining database groups for most extreme adaptability and execution.
- Integrating NoSQL or columnar databases that aren't "drop-in" trades for RDBMS.
- Scaling application parts: arrangements and choices for every level.
- Recognizing when to scale your information level—a choice with gigantic ramifications for your application condition.
- Why information connections might be considerably increasingly critical in non-social databases.
- Why practically every database versatility execution still depends on shading, and how to pick the best methodology.
- How to set clear targets for architecting superior Big Data executions

It's one thing to implement a data storage or analysis framework that scales. Scaling the vital connections that deliver information to your system is another story[4]. One potential versatility combination workaround could lie in buying a total framework rather than only an apparatus. Numerous business structures are intended to interface easily with outsider devices. Devices like Salesforce Marketing Cloud use MongoDB to allow scaling locally as you go[5].

These models certainly utilize enormous information investigation to convey customized content, yet there are endless different applications. There are a wide range of approaches to make a framework that collects experiences from enormous information[6].

From web-based social networking locales, to web crawler results, to publicizing, organizations hoping to exploit customer/client data, have a fortune trove readily available[7]. However, with the exponential increments in the volume of information being delivered and prepared, numerous organizations' databases are being overpowered with the storm of information they are confronting[8].

These stages use added equipment or programming to build yield and capacity of information. At the point when an organization has an adaptable information stage, it additionally is set up for the capability of development in its information needs [9].

How can organizations know they have an infrastructure that is strong enough to handle big data? The architecture is at the core of successfully implementing a big data initiative[10]. Unlimited data scalability systems have four key elements in common:

- Shared-nothing architecture
- Software data flow
- Data partitioning for linear data scalability
- Design isolation

1.1 Shared-nothing architecture

Architected software is designed from the ground up to capitalize on a shared-nothing, massively parallel processing (MPP) architecture (see Figure 1 *A contention-free, shared-nothing architecture*). Data sets are partitioned across computing nodes and a single application is executed with the same application logic implemented against each data partition. As a result, there is no single point of contention, or processing bottleneck, anywhere in the system. And there is no upper limit on data volume, processing throughput, number of processors, and nodes[11]

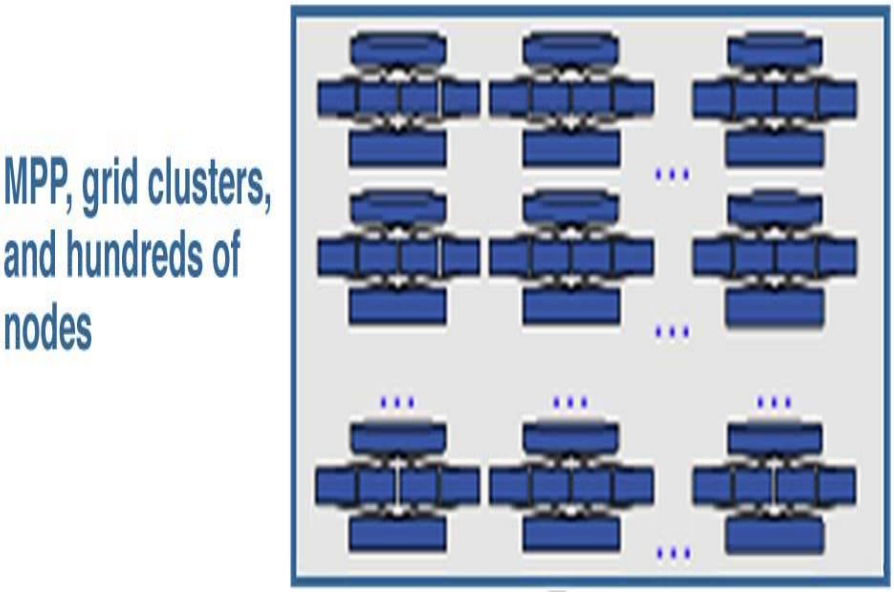

Figure 1 A contention-free, shared-nothing architecture

1.2 Software data flow

A shared-nothing architecture can be fully exploited by a software data flow that easily implements and executes data pipelining and data partitioning within a node and across nodes (see Figure 2 *Parallel software data flow with automatically repartitioning data*). Software data flow also hides the complexities of building, tuning, and executing parallel applications from end users[12].

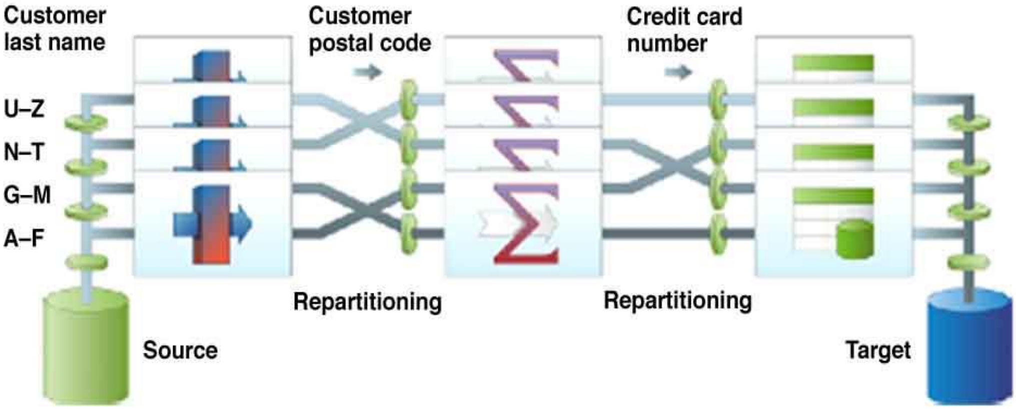

Figure 2 Parallel software data flow with automatically repartitioning data

1.3 Data partitioning

Large data sets can be partitioned across separate nodes, and a single job—for example, deploying the IBM® InfoSphere® Information Server integration platform—can execute the same application logic against all partitioned data (see Figure 3 *The same application logic executed for all data partitioned across nodes*). Other approaches such as task partitioning are not capable of delivering linear data scalability as data volumes grow because the amount of data that can be sorted, merged, aggregated, and so on is limited to what can be processed on one node[13].

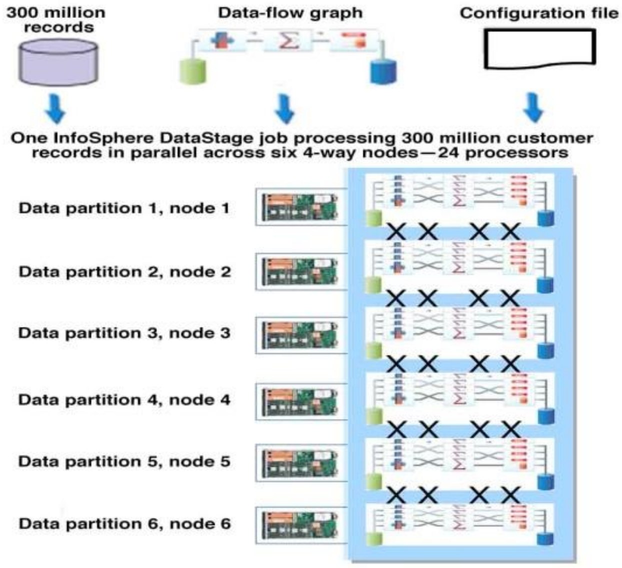

Figure 3 The same application logic executed for all data partitioned across nodes

The following characteristics apply to systems with data partitioning:

- Data partitions are distributed across nodes.
- One job is executed in parallel across nodes.
- Pipelining and repartitioning occur between stages and between nodes without landing to disk.
- Grid hardware for big data is cost-effective.

1.4 Design isolation

The capability for a developer to design a data processing job once, and use it in any hardware configuration without needing to redesign and retune the job[14], is called design isolation (see Figure 4 *One-time data processing design for any hardware configuration*). This approach allows a job to be built once and run without modification anywhere. It offers one unified mechanism for parallelizing data flow[15]. A single configuration file provides a clean separation between the development of a job and the expression of parallelization at runtime. In addition, performance tuning is not required every time the hardware architecture changes, and there is no upper limit for data scalability as hardware is added[16].

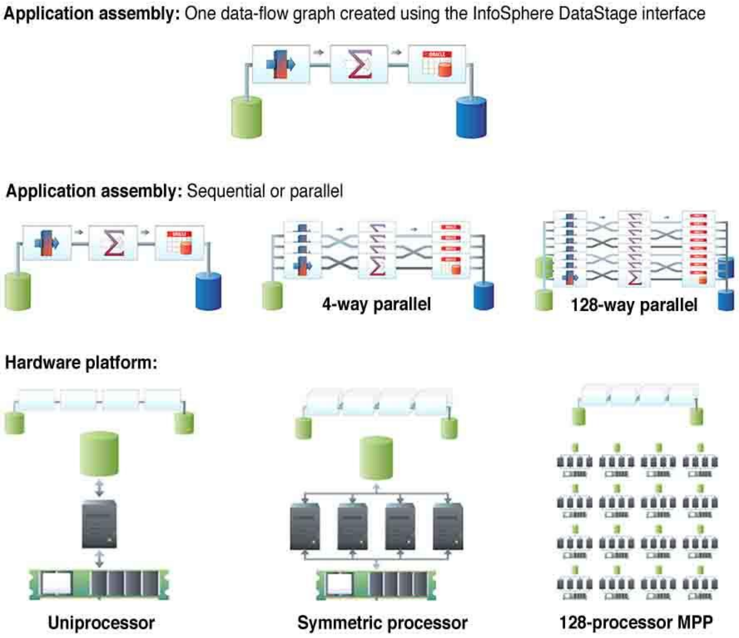
Application assembly: One data-flow graph created using the InfoSphere DataStage interface
Application assembly: Sequential or parallel
4-way parallel
128-way parallel
Hardware platform:
Uniprocessor
Symmetric processor
128-processor MPP

Figure 4 One-time data processing design for any hardware configuration

II. Related Work

Link-based investigation of the Web gives the premise to numerous essential applications like Web seek, Web-based information mining, and Web page arrangement that convey request to the gigantic measure of circulated Web content. They presented a parameterized structure for SLA, investigated a few basic parameters, and directed the primary huge scale relative investigation of SLA over various substantial genuine Web informational collections and numerous contending targets. They locate that cautious tuning of these parameters is indispensable to guarantee accomplishment over every goal and to adjust the execution over all targets[17].

Skyline is a critical task in numerous applications to restore a lot of fascinating focuses from a conceivably enormous information space[18]. Given a table, the activity discovers all tuples that are not overwhelmed by some other tuples. SSPL uses arranged positional file records which require low space overhead to lessen I/O cost essentially. During recovering the rundowns in a round-robin design, SSPL performs pruning on any competitor positional record to dispose of the hopeful whose relating tuple isn't horizon result. Phase 1 closes when there is a competitor positional file found in the majority of the included records. It is demonstrated that SSPL performs well when the span of horizon criteria is little. Especially if the qualities are emphatically related, the execution of SSPL is very palatable. They additionally assess SSPL against tree-based calculations (ZSearch and SUBSKY)[18].

The new ages of cell phones have high preparing force and capacity, yet they linger behind as far as programming frameworks for enormous information stockpiling and handling. Hadoop is a versatile stage that gives dispersed capacity and computational abilities on bunches of ware equipment. Building Hadoop on a portable system empowers the gadgets to run information concentrated processing applications without direct learning of basic dispersed frameworks complexities. These applications have extreme vitality and unwavering quality requirements. The assessment results demonstrate that our framework is proficient for huge information examination of unstructured information like media records, content and sensor information. Their execution results look extremely encouraging for the arrangement of our framework in true bunches for enormous information expository of unstructured information like media records, content and sensor information[19].

Although Web Search Engines record and give access to colossal measures of reports, client inquiries ordinarily return just a direct rundown of hits. They have appeared at disintegrate a successive Named Entity Mining calculation into an identical disseminated MapReduce calculation and convey it on the Amazon EC2 Cloud. To achieve the most ideal burden adjusting, they structured two MapReduce methodologies and examined their adaptability and by and large execution under various setup/tuning parameters in the hidden stage (Apache Hadoop)[20].

Generalized meager non-negative lattice factorization (SNMF) has been demonstrated valuable in extricating data and speaking to inadequate information with different kinds of probabilistic distributions from mechanical applications, e.g., recommender frameworks and informal communities. To beat these issues, an on the web, versatile and single-string put together SNMF for CUDA parallelization with respect to GPU (CUSNMF) and multi- GPU (MCUSNMF) is proposed. Meanwhile, the model has a streamline-like processing style, which can take into account the figuring qualities of the standard big information and mechanical stages, e.g., GPU, Spark, and Flink. The streamline-like processing style can understand the internet learning and fine-grained parallelization with CUDA parallelization capacity on GPU (CUSNMF) and multi GPU[21].

They create web based learning calculations that empower the operators to agreeably figure out how to boost the general reward in situations where just boisterous worldwide input is accessible without trading any data among them. They examined a general multi-specialist choice making issue in which decentralized specialists gain proficiency with their best activities to boost the framework compensate utilizing just boisterous perceptions of the general reward. The difficult part is that individualized criticism is missing, correspondence among operators is inconceivable and the worldwide criticism is liable to singular perception blunders[22].

Strong State Drives (SSDs) were at first created as quicker stockpiling gadgets proposed to supplant regular attractive Hard Disk Drives (HDDs). This offloads Map undertakings from the host MapReduce framework to the ISC SSDs. They furthermore improve the host Hadoop framework to utilize our proposed ISC Hadoop framework. This paper connected the SSD-situated In-Storage Computing (ISC) model to the Hadoop MapReduce system – an accepted standard appropriated registering structure for enormous information preparing. After examination, we offloaded the Mapper work from a host Hadoop framework to SSDs and incorporated the current Hadoop MapReduce framework with our ISC gadgets [23].

They give new answers for handle the issue of parallel hash-based conglomeration, particularly focusing at spaces of amazingly expansive cardinality. They likewise present another productive conglomeration calculation (EAA), to total the apportioned information in parallel with low reserve lucidness miss and bolting costs. Theoretical examination just as experimental investigation on an IBM X5 server demonstrate that our recommendations are no less than multiple times quicker than existing techniques. In-memory accumulation calculation of huge information with huge gathering via cardinality on current servers under NUMA designs [24].

This paper presents a successful preparing structure assigned Image Cloud Processing (ICP) to capably adapt to the information blast in picture handling field. The proposed ICP system comprises of two instruments, i.e., Static ICP (SICP) and Dynamic ICP (DICP). In particular, SICP is gone for preparing the huge picture information pre-put away in the circulated framework, while DICP is proposed for dynamic input. To achieve SICP, two novel information portrayals named P-Image and Big-Image are intended to collaborate with MapReduce to accomplish more upgraded design and higher effectiveness. From the attractive outcomes, they accept that huge picture information handling is a promising bearing, which calls for undertaking in framework, processing structure, demonstrating, learning calculation, applications, and varying backgrounds[25].

The customary way to deal with putting away information have confronted difficulties because of the quick development of data. A half breed reserve design utilizing a multi-layer store design. Rather than utilizing adaptable NoSQL database, attempt to use the common memory reserve to improve the question productivity of customary SQL database frameworks. The multi-layer storing design, UUID, and a reserve substitution strategy are utilized to accelerate the inquiry adequacy. The UUID is connected to information question to disentangle the inquiry activity of joining different tables [26].

This paper archives the noteworthy advancement accomplished in the field of conveyed registering structures, especially Apache Hama, a top dimension venture under the Apache Software Foundation, in view of mass synchronous parallel preparing. The outcomes demonstrate that the execution of Hama is superior to Graph regarding versatility and computational speed. This paper additionally depicts these difficulties, breaks down arrangements proposed to conquer them, and features inquire about circumstances. Among accessible systems, Apache Hama is a developing and unmistakable open source structure with various attributes, for example, quick preparing rate, effective correspondence and hindrance synchronization instruments, and a wide application area[27].

This paper presents PISCES (Pipeline Improvement Support with Critical chain Estimation Scheduling), a basic chain enhancement (a basic chain alludes to a progression of occupations which will make the application run longer if any of them is deferred), to give better help to multi-work applications[28]. PISCES stretches out the current MapReduce structure to permit planning for numerous employments with conditions by powerfully developing work reliance DAG for current running occupations as indicated by their information and yield registries. At that point utilizing the reliance DAG, it gives a creative component to encourage the information pipelining between the yield stage (map stage in the Map-Only occupation or diminish stage in the Map-Reduce work) of an upstream activity and the guide period of a downstream activity. Examinations demonstrate that PISCES can expand the level of framework parallelism by up to 68% and improve the execution speed of utilizations by up to 52%. PISCES stretches out the MapReduce system to permit booking for numerous occupations with conditions by structure a reliance DAG for the running employment bunch progressively[28].

They propose an expectation based and territory mindful assignment booking strategy for parallelizing video trans coding over heterogeneous MapReduce group. They endeavor to plan sub-errands on machines that contain the related information, which is alluded to as information territory, to decrease expansive scale information development and information exchange amid the mapping stage. An expectation based and area mindful assignment planning strategy for parallelizing video trans coding over heterogeneous MapReduce bunch. They dissect and anticipate the video transcoding multifaceted nature as our booking establishment. The exploratory outcomes demonstrate that PLTS can successfully lessen the complete video trans coding time [29].

Large measures of geospatial information are every day produced by numerous perception forms in various application spaces. Existing perception information the executives arrangements need explanatory particular of spatio-fleeting investigation. Then again, current information the executives innovations miss perception information semantics and neglect to incorporate the administration of substances and samplings in a solitary information demonstrating arrangement. Perception information semantics are fused into the model with proper metadata structures. A framework for the examination of spatial perception data was planned and subjectively assessed and contrasted and related information the board advancements and methodologies [30].

Stockroom Scale Computers (WSC) are frequently utilized for different huge information employments where the enormous information under handling originates from an assortment of sources. They demonstrate that diverse information parcels, from the equivalent or distinctive sources, have distinctive significances in deciding the ultimate result of the calculation, and thus, by organizing them and doling out more assets to preparing of progressively critical information, the WSC can be utilized all the more productively as far as time just as expense. They give a straightforward low-overhead instrument to rapidly survey the centrality of every datum segment, and demonstrate its viability in finding the best positioning of information partitions [31].

This paper demonstrated a model execution of EPIC Real-Time which utilizes occasion driven and responsive programming procedures. They additionally present an exhibition assessment on how productively the constant and group arranged inquiries perform, how well these questions address the issues of our experts, and give knowledge into how EPIC Real-Time performs along various measurements including execution, convenience, versatility, and unwavering quality. The scope of framework types ranges bunch ETL frameworks to constant examination of spilling information. They introduced a lightweight, administration based stage considered EPIC Real-Time that was intended to help [32].

Contemporary systems for information investigation, for example, Hadoop, Spark, and Flink try to enable applications to scale execution adaptably by including equipment hubs.They found that when the calculation on every individual hub is enhanced, fringe exercises, for example, making information allotments, informing and synchronizing between hubs reduce the speedup reachable from including more equipment. They investigate outstanding tasks at hand which convey activities on corresponded information, for example, joins and accumulation found in SQL, content closeness quests, and picture uniqueness calculations. In the wake of enhancing calculation on proficient, custom processors, they find difficulties in scaling the applications to many hubs on a high-data transfer capacity arrange [33].

In this paper, They proposed another huge information structure for handling huge measures of remote detecting pictures on distributed computing stages. Notwithstanding exploiting the parallel handling capacities of distributed computing to adapt to extensive scale remote detecting information, this structure consolidates task planning procedure to additionally misuse the parallelism amid the circulated preparing stage. Utilizing a calculation and information serious container honing strategy as an examination case, the proposed methodology begins by profiling a remote detecting application and describing it into a coordinated non-cyclic diagram (DAG). Exploratory outcomes exhibit that the proposed structure accomplishes promising outcomes as far as execution time as contrasted and the conventional (sequential) preparing approach [34].

In this work, They present a novel technique that profits half breed process models to connect this hole. Besides, to adapt to the present enormous occasion logs, we propose a productive strategy, called f-HMD, goes for versatile cross breed display disclosure in a distributed computing condition. In this paper, we have presented a productive methodology, called f-HMD, to find half breed process models from expansive occasion logs. They have depicted the itemized usage of our methodology in a distributed computing condition (i.e., over Spark) and our exploratory outcomes have exhibited that the proposed 2f-HMD calculation is exceptionally proficient and versatile [35].

They hypothetically and observationally demonstrate that one hub in sans scale organize has a set number of neighbors with a higher degree. Utilizing this finding, a novel parallel triangle disclosure calculation called degree parcel (DePart) is structured. Since just the part of neighbors with higher degrees is stacked into inner memory in the reducers, DePart understands "the scourge of the last reducer" issue and works effectively in groups with restricted inward memory. They create DePart utilizing MapReduce structure and look at the execution of DePart on five expansive true systems. They tended to the triangle revelation issue for expansive scale true systems utilizing MapReduce and illuminated "the scourge of the last reducer" in MR-NI, which is related with inner memory multifaceted nature[36].

The shrewd home physical layer incorporates all the detecting advancements and keen gadgets inside the savvy home, which screens the home condition and its inhabitants. The information of these sensors will be sent to the shrewd home haze figuring layer that can do constrained information stockpiling and handling. At that point, all the required information will be sent to the distributed computing layer utilizing brilliant home system layer. The proposed shrewd home design can give a pervasive and shared information condition as the key part of Internet-of-Things frameworks. The shrewd home application layer incorporates every one of the applications which can trade information with savvy home administration layers [37].

III. Problem Analysis

After referring many papers few of them are listed above in literature survey, we found our problem statement that can we summarize as a problem statement written below.

In today's network society, the big data processing frameworks is continuously highly demanded and growing. In Big Data processing multiple job operation and scalability of operations using single program is main issues in distributed cluster.

Big Data and Hadoop is a popular Technology and framework for analyzing massive and huge amount of datasets. Big Data, MapReduce typically operates on static data by scheduling batch jobs stored in workstation, servers and terminals. This process is complex enough, time taking and difficult to continuous unbounded streams of data in real time. To address this problem, we will design a scalable data processing on large distributed processing method based on Hadoop, Big Data, MapReduce, Big Table and Big Query framework. We will design an algorithm which is based on 3-tiers or n-tier architecture where different types of data are stored in different distributed clusters and may use for different domain analysis for multiple purposes simultaneously. Our design will do the parallel approach to carry out high performance parallel big data analysis. BigTable will be used for storing and managing large scale data distributed system. The concept of sharing resource used in traditional distributed system will be used as it is in this work also and try to develop a model that will give observation of our application workloads and technological environment. Our model support working set of data from the multiple parallel operations. This research work includes Hadoop Framework, Big Data Technology, Big Table and Big Query.

IV. Methodology

1. Design and implement the multiple node based Big Data Infra- structure using CLOUD services:

Cloud computing is also turning into the storage for Big Data needs. At the Infrastructure as a Service (IaaS) level, Enormous Data can use the Storage capacities of Clouds, also in the meantime, it can depend on calculation inside Virtual Machine(VM). Additionally Hadoop, introduced into VMs, is upgraded for preparing Big Data. It is vety interesting to see that VM instance and their configuration setups unequivocally influence this sort of handling. Utilizing Cloud assets in connection to BigData operation is a direct objective. Hadoop is the bigger utilized open source system received for overseeing Big Data with Map/Reduce approach

2. Design and develop an algorithm to combine BIGDATA and BIGTABLE:

Bigtable is used to provide fast access and retrieval of structure data. Framework will support to access the data from big data using the bigtable.

3. Design and implement BIG Table over the BIG DATA for fast retrieval of information:

Algorithm will support the mechanism that can be give data faster comparatively to the BigData.

4. Design and implement BIGQUERY that gives the instruction to BIGTABLE for desired output:

For query purpose, Bigtable data source using a permanent table, An algorithm that are able to query to big table database that is linked to our Bigtable data source.

5. Identify the process to represent output to the end user.

The result should reflect and show the original fruitful data and should be a presentable format. So that the end user will use the output data to desired purpose.

V. Proposed Outcome

In this research we will get first a model will be designed based on Hadoop BIG DATA Framework, Distributed Data processing system, BigTable and Big Query which works on different data stream. Second a server based framework will be designed that will generalize many different domain analysis in a single one and last one third to design a model to deploy data to different clusters using partitioning method. To develop a system that is capable to adjust run-time resource allocation to reduce the analysis tasks.

VI. CONCLUSION

A framework for the analysis of spatial observation data was designed and qualitatively evaluated and compared with related data management technologies and approaches. An Integrated environment based on BIG Data and Cloud. BigData Processing based Distributed Data processing system, BigTable and Big Query which works on different data stream. A system that implements an innovative pipeline optimization among dependent jobs and a chain job scheduling model in an existing MapReduce system to minimize the application execution time and maximize resource utilization and global efficiency of the system. Scalability of Big data analysis can be achieved. MapReduce program will able to do multiple analyses. Different domain analysis can be generalize with this framework can obtain. Clustering can be utilized through portioning method to overcome the data load. Capable of doing runtime resource allocation to reduce the analysis task.

VII. ACKNOWLEDGMENT

This research was supported by [UR Name]. We thank our supervisor from [Name of supervisor Institute] who provided insight and expertise that greatly assisted the research paper.

REFERENCES

[1]. Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2014). Data mining with big data. IEEE Transactions on Knowledge and Data Engineering, 26(1), 97–107. https://doi.org/10.1109/TKDE.2013.109

[2]. Malarvizhi, S. P., & Sathiyabhama, B. (2016). Frequent pagesets from web log by enhanced weighted association rule mining. Cluster Computing, 19(1), 269–277. https://doi.org/10.1007/s10586-015-0507-z

[3]. Sun, D., & Huang, R. (2016). A stable online scheduling strategy for real-time stream computing over fluctuating big data streams. IEEE Access, 4(c), 8593–8607. https://doi.org/10.1109/ACCESS.2016.2634557

[4]. Wu, Y., Yan, C., Ding, Z., Liu, G., Wang, P., Jiang, C., & Zhou, M. C. (2016). A Multilevel Index Model to Expedite Web Service Discovery and Composition in Large-Scale Service Repositories. IEEE Transactions on Services Computing, 9(3), 330–342. https://doi.org/10.1109/TSC.2015.2398442

[5]. Wu, Y., Yan, C., Ding, Z., Liu, G., Wang, P., Jiang, C., & Zhou, M. C. (2016). A Multilevel Index Model to Expedite Web Service Discovery and Composition in Large-Scale Service Repositories. IEEE Transactions on Services Computing, 9(3), 330–342. https://doi.org/10.1109/TSC.2015.2398442

[6]. Dong, L., Lin, Z., Liang, Y., He, L., Zhang, N., & Chen, Q. (2016). Framework for Big Image Data, 2(4), 297–309.

[7]. Lin, Y. Te, Hsiao, Y. H., Lin, F. P., & Wang, C. M. (2016). A hybrid cache architecture of shared memory and meta-table used in big multimedia query. 2016 IEEE/ACIS 15th International Conference on Computer and Information Science, ICIS 2016 - Proceedings. https://doi.org/10.1109/ICIS.2016.7550809

[8]. Anadiotis, A. C. G., Morabito, G., & Palazzo, S. (2016). An SDN-Assisted Framework for Optimal Deployment of MapReduce Functions in WSNs. IEEE Transactions on Mobile Computing, 15(9), 2165–2178. https://doi.org/10.1109/TMC.2015.2496582

[9]. Larkou, G., Mintzis, M., Andreou, P. G., Konstantinidis, A., & Zeinalipour-Yazti, D. (2016). Managing big data experiments on smartphones. Distributed and Parallel Databases, 34(1), 33–64. https://doi.org/10.1007/s10619-014-7158-6

[10]. Liu, Z., Zhang, Q., Ahmed, R., Boutaba, R., Liu, Y., & Gong, Z. (2016). Dynamic Resource Allocation for MapReduce with Partitioning Skew. IEEE Transactions on Computers, 65(11), 3304–3317. https://doi.org/10.1109/TC.2016.2532860.

[11]. Zhang, Y., Chen, S., & Yu, G. (2016). Efficient Distributed Density Peaks for Clustering Large Data Sets in MapReduce. IEEE Transactions on Knowledge and Data Engineering, 28(12), 3218–3230. https://doi.org/10.1109/TKDE.2016.2609423

[12]. Ahmadvand, H., & Goudarzi, M. (2016). Using Data Variety for Efficient Progressive Big Data Processing in Warehouse-Scale Computers. IEEE Computer Architecture Letters, 6056(c), 1–1. https://doi.org/10.1109/LCA.2016.2636293

[13]. Heintz, B., Chandra, A., Sitaraman, R. K., & Weissman, J. (2016). End-to-End Optimization for Geo-Distributed MapReduce. IEEE Transactions on Cloud Computing, 4(3), 293–306. https://doi.org/10.1109/TCC.2014.2355225

[14]. Bharill, N., Tiwari, A., & Malviya, A. (2016). Fuzzy Based Scalable Clustering Algorithms for Handling Big Data Using Apache Spark. IEEE Transactions on Big Data, 2(4), 339–352. https://doi.org/10.1109/TBDATA.2016.2622288

[15]. Zhang, Y., Chen, S., Wang, Q., & Yu, G. (2016). I2MapReduce: Incremental mapreduce for mining evolving big data. 2016 IEEE 32nd International Conference on Data Engineering, ICDE 2016, 27(7), 1482–1483. https://doi.org/10.1109/ICDE.2016.7498385

[16]. Mesmoudi, A., Hacid, M. S., & Toumani, F. (2016). Benchmarking SQL on MapReduce systems using large astronomy databases. Distributed and Parallel Databases, 34(3), 347–378. https://doi.org/10.1007/s10619-014-7172-8

[17]. Caverlee, J., Webb, S., Liu, L., & Rouse, W. B. (2009). A parameterized approach to spam-resilient link analysis of the web. IEEE Transactions on Parallel and Distributed Systems, 20(10), 1422–1438. https://doi.org/10.1109/TPDS.2008.227

[18]. Han, X., Li, J., Yang, D., & Wang, J. (2013). Efficient skyline computation on big data. IEEE Transactions on Knowledge and Data Engineering, 25(11), 2521–2535. https://doi.org/10.1109/TKDE.2012.20

[19]. Johnu George, Chien-An Chen, Radu Stoleru, Geoffrey G. Xie Member, IEEE, Hadoop MapReduce for Mobile Clouds IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 3, NO. 1, JANUARY 2014

[20]. Kitsos, I., Magoutis, K., & Tzitzikas, Y. (2014). Scalable entity-based summarization of web search results using MapReduce. Distributed and Parallel Databases, 32(3), 405–446. https://doi.org/10.1007/s10619-013-7133-7

[21]. Li, H., Member, S., Li, K., Member, S., An, J., & Li, K. (2015). An Online and Scalable Model for Generalized Sparse Non-negative Matrix Factorization in Industrial Applications on Multi-GPU, 14(8), 1–11.

[22]. Xu, J., Tekin, C., Zhang, S., & van, der S. (2015). Distributed Multi-Agent Online Learning Based on Global Feedback. IEEE Transactions on Signal Processing, 63(9), 2225–2238. https://doi.org/10.1109/TSP.2015.2403288

[23]. Dongchul Park, Member, IEEE, Yang-Suk Kee, Member, IEEE, In-Storage Computing for Hadoop MapReduce Framework: Challenges and Possibilities, IEEE TRANSACTIONS ON COMPUTERS, JULY 2015

[24]. Wang, L., Zhou, M., Zhang, Z., Shan, M. C., & Zhou, A. (2015). NUMA-Aware Scalable and Efficient In-Memory Aggregation on Large Domains. IEEE Transactions on Knowledge and Data Engineering, 27(4), 1071–1084. https://doi.org/10.1109/TKDE.2014.2359675.

[25]. Le Dong, Member, IEEE, Zhiyu Lin, Yan Liang, Ling He, Ning Zhang, Qi Chen, Xiaochun Cao, and Ebroul Izquierdo, Senior Member, IEEE, A Hierarchical Distributed Processing Framework for Big Image Data, IEEE TRANSACTIONS ON BIG DATA, VOL. 2, NO. 4, OCTOBER-DECEMBER 2016

[26]. Lin, Y. Te, Hsiao, Y. H., Lin, F. P., & Wang, C. M. (2016). A hybrid cache architecture of shared memory and meta-table used in big multimedia query. 2016 IEEE/ACIS 15th International Conference on Computer and Information Science, ICIS 2016 - Proceedings. https://doi.org/10.1109/ICIS.2016.7550809

[27]. Siddique, K., Akhtar, Z., Yoon, E. J., Jeong, Y. S., Dasgupta, D., & Kim, Y. (2016). Apache Hama: An emerging bulk synchronous parallel computing framework for big data applications. IEEE Access, 4(c), 8879–8887. https://doi.org/10.1109/ACCESS.2016.2631549

[28]. Chen, Q., Yao, J., Li, B., & Xiao, Z. (2016). PISCES: Optimizing Multi-job Application Execution in MapReduce. IEEE Transactions on Cloud Computing, 7161(c), 1–1. https://doi.org/10.1109/TCC.2016.2603509

[29]. Zhao, H., Zheng, Q., Zhang, W., & Wang, J. (2016). Prediction-based and Locality-aware Task Scheduling for Parallelizing Video Transcoding over Heterogeneous MapReduce Cluster. IEEE Transactions on Circuits and Systems for Video Technology, 8215(c), 1–1. https://doi.org/10.1109/TCSVT.2016.2634579

[30]. Villarroya, S., Viqueira, J. R. R., Regueiro, M. A., Taboada, J. A., & Cotos, J. M. (2016). SODA: A framework for spatial observation data analysis. Distributed and Parallel Databases (Vol. 34). Springer US. https://doi.org/10.1007/s10619-014-7165-7

[31]. Ahmadvand, H., & Goudarzi, M. (2016). Using Data Variety for Efficient Progressive Big Data Processing in Warehouse-Scale Computers. IEEE Computer Architecture Letters, 6056(c), 1–1. https://doi.org/10.1109/LCA.2016.2636293

[32]. Ahmadvand, H., & Goudarzi, M. (2016). Using Data Variety for Efficient Progressive Big Data Processing in Warehouse-Scale Computers. IEEE Computer Architecture Letters, 6056(c), 1–1. https://doi.org/10.1109/LCA.2016.2636293

[33]. Govindaraju, V., Idicula, S., Agrawal, S., Vardarajan, V., Raghavan, A., Wen, J., … Sedlar, E. (2017). Big Data Processing : Scalability with Extreme Single-Node Performance. https://doi.org/10.1109/BigDataCongress.2017.26

[34]. Jin Sun, Member, IEEE, Yi Zhang, Zebin Wu , Senior Member, IEEE, Yaoqin Zhu, Xianliang Yin, Zhongzheng Ding, Zhihui Wei, Javier Plaza , Senior Member, IEEE, and Antonio Plaza , Fellow, IEEE, An Efficient and Scalable Framework for Processing Remotely Sensed Big Data in Cloud Computing Environments, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING

[35]. Cheng, L., Dongen, B. F. Van, Aalst, W. M. P. Van Der, & Member, S. (2019). Scalable Discovery of Hybrid Process Models in a Cloud Computing Environment. IEEE Transactions on Services Computing, PP(c), 1. https://doi.org/10.1109/TSC.2019.2906203

[36]. Zhou, X., Liang, X., Member, S., & Tang, Z. (2018). Scalable Triangle Discovery Algorithm for Large Scale-free Network with Limited Internal Memory, 0(0), 1–12.

[37]. Homes, D. S., & Zhang, Q. (2019). A New Layered Architecture for Future Big. IEEE Access, 7, 19002–19012. https://doi.org/10.1109/ACCES.2019.2896403

[38]. Yang, C., Xu, X., Ramamohanarao, K., & Chen, J. (2019). A Scalable Multi-Data Sources based Recursive Approximation Approach for Fast Error Recovery in Big Sensing Data on Cloud. IEEE Transactions on Knowledge and Data Engineering, PP, 1. https://doi.org/10.1109/TKDE.2019.2895612

[39]. Sun, P., Wen, Y., Duong, T. N. B., & Xie, H. (2016). MetaFlow: a Scalable Metadata Lookup Service for Distributed File Systems in Data Centers, 7790(c), 1–14. https://doi.org/10.1109/TBDATA.2016.2612241

[40]. Sun, P., Wen, Y., Duong, T. N. B., & Xie, H. (2016). MetaFlow: a Scalable Metadata Lookup Service for Distributed File Systems in Data Centers, 7790(c), 1–14. https://doi.org/10.1109/TBDATA.2016.2612241

[41]. Lin, K., Luo, J., Hu, L., Hossain, M. S., & Ghoneim, A. (2017). Localization based on social big data analysis in the vehicular networks. IEEE Transactions on Industrial Informatics, 13(4), 1932–1940. https://doi.org/10.1109/TII.2016.2641467.