Dynamic Balancing of Complex Event and Data Recovery in Cloud Environment

Pranjali Dhore
PG Student, Department of Information Technology,
MITCOE, Pune, India
dhore.pranjali08@gmail.com

Kishor Kolhe
Associate Professor, Department of Information Technology
MITCOE, Pune, India
krkolhe@gmail.com

***Abstract*- Distributed computing is a developing innovation deals with dispersed figuring system. . One of the most challenging issues in cloud Computing is efficient scheduling of tasks. Cloud computing offers a variety of dynamic load balancing methods. Load balancing is the methodology of distributing the load among different node of a distributed framework to enhance both resource usage and reaction time while likewise keeping away from a circumstance where a percentage of the node are intensely stacked while different node are sit out of gear or doing next to no work. An solution for unbalance circumstance is to utilize parallelization approaches yet at the same time node will stay overwhelming. In this paper, we propose an integrated Dynamic Load balancing algorithm to attain scalability even in heavy queries, avoid fault tolerance when system crash, event generation and handling, job submission control, Overload control on cloud environment.**

Keywords— Cloud Computing, Complex Event processing, Distributed database, Dynamic Load balancing, Scalability.

I. INTRODUCTION

Now days, distributed database is spread over the system where application is required. High rate transforming of information is most urgent in appropriated environment.. Complex Event Processing is event processing that combines data from multiple sources. With the help of this event processing system thousands of incoming queries can be register, processed and that will produces result periodically, this will lead to scalability, failure of node because of heavily loaded node problem when big database is used.

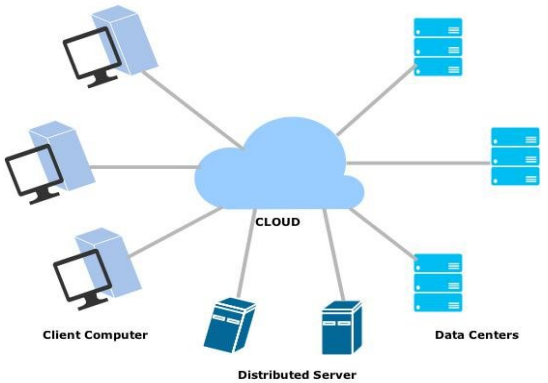

Fig 1: Components of Cloud Computing Solution

Fault tolerance occurred because of load unbalancing condition means a situation where some nodes are heavily loaded and some nodes are idle**.** To avoid fault tolerance nodes, the method is required to allocate dynamic workloads equally to all the nodes across cloud network hence load balancing approach is used. Load balancing is the method which makes sure that every processor within the system or every node in the network consume equal amount of power and finish approximately equal amount of work at any instant of time. The load can be identified as data uploading capacity, CPU load or network delay.

In this paper, we proposed a robust architecture which schedules the queries for scalability and distribution of equal work load. Registered incoming CEP queries in system are transferred to idle nodes or not heavily node. To improve both resource utilization and performance scheduler is responsible to distribute the load among various nodes and make it balance. Queries are applied to scheduler module of system, then scheduler schedules the query to lightly weighted node with respect to CPU capacity, memory consumption. Incoming queries are delivered to queue; if queue is full or reaching to limit size in queue measure will be taken. Overload situation can kill the nodes and results into data loss but this will be avoided by using replication method. The proposed system maintain ready queue, use it to queue incoming query when nodes are loaded and none of node is free. At last event is generate if no selection is conceivable.

The rest of the paper is organized as follows. Section II discusses literature review, Section III describes the problem statement, and Section IV discusses proposed method. We conclude the paper in last section.

II. LITERATURE REVIEW

There are numerous methodologies taken inside the writing for learning load balancing. In appropriated systems adjusting the node in a versatile way enhances the system execution impressively. Further, a definitive objective of load balancing is as following [5]

- Even appropriation of load to every node.
- Minimization of handling time for every node.
- Maximum utilization of each resource.

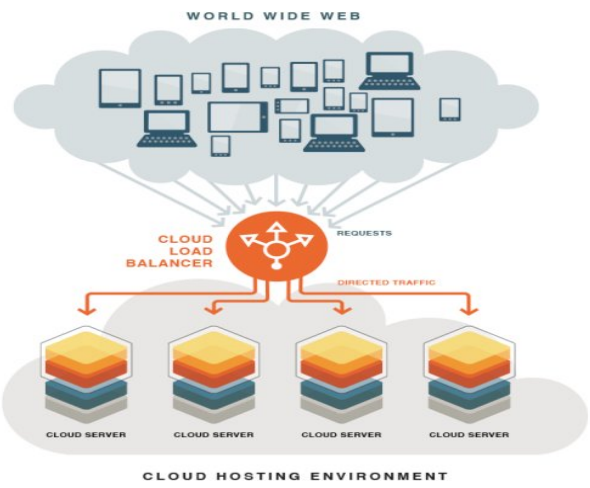

Fig 2: Cloud Hosting Environment

Load balancing is a system to upgrade assets, using parallelism, misusing throughput impromptu creation, and to decrease reaction time through a proper dispersion of the application [4]. Parallelization methodologies are utilized for load balancing as a part of which load is parallelize through numerous machines or centers. Indeed with parallelization approach still database questions are calm overwhelming obligation forms [1]. The essential things to consider while growing such calculation are : estimation of load, examination of load, soundness of diverse framework, execution of framework, cooperation between the nodes, way of work to be exchanged, selecting of node and numerous different ones. This heap considered can be regarding CPU load, measure of memory utilized, postpone or Network load. Load Balancing calculations are of two sorts relying upon current condition of condition of framework as Static and Dynamic. In static calculation earlier learning of framework parts is essential and in this manner does not rely on upon current framework. Then again Dynamic Algorithm is reliant on present condition of framework segment and is superior to static component. Static load balancing calculations dispense the undertakings of a parallel system to workstations[12]. In Static methodology work load is timetable by utilizing predefine execution arrangement which helps in gathering information, indexing and questioning. Complex Event Processing (CEP) is already a mature technology in specific markets. So far CEP systems are applied in the financial industry: stock trading [1], stream monitoring, information integration workflows, and so on. In contradiction of traditional database systems, CEP systems can directly execute processing over infinite data streams using a fixed amount of memory, accepting several thou- sands of events per second and sub-millisecond latency [6].

Different examination had created for load balancing in cloud and circulated environment. Dong [3] came up with Self Acting Load Balancing Algorithm for parallel file system based on distributed architecture. In Parallel file system Data management plays vital role as data is constantly transferred between memory and storage devices. Various factors like scalability, availability of system, Load Migration and Data Transmission followed by continuous workload variation. Thus this calls for central decision maker and to explore decision making algorithm. Thus one node act as decision maker and others follow the assigned task accordingly. If central node fails then total system will fail hence this method is not followed dominantly. This mechanism provided load balancing mechanism but as task migration will frequently takes place, possibility of system degradation increases. Self Acting Balancing Algorithm (SALB) provides better load forecasting algorithm, Load collection, Distribution and migration on in effective manner. [3]

Dhinesh Babu [4] proposed an alternate Load Balancing system in view of Honey Bee Behavior. The strategy is gotten from conduct of honey bees. The bumble bees are of two sorts as scout bumblebees and forager honey bees. Scout honey bees figure out the nourishment and offers message to forager bumblebees through move. After this forager bumblebees take after the scout honey bees and track the area of sustenance. Same idea of used for load balancing and over-load Virtual Machine (VM) go about as bumble bees. On task of any undertaking the under stacked Virtual machine will overhaul the data and if high stacked or need errand is submitted for handling it will be assigned to VM having minimum number of high priority task. As the task assignment is on the basis of ascending order under loaded VM's will get the high priority task and thus load balancing is carried out. Load of specific server is computed on the basis if information accessed from Data centre/server. Honey bee approach reduced waiting time and high priority task are processed quickly and scalability is achieved without further extension of resources. This suffers from drawback as high priority task are processed first it forms queue and low priority task remain in queue for further processing for quite long period[4]

Raj, G. Punjab Tech. Univ, Jalandhar, India Singh [5] proposed a system in which load adjusting is figured out how to attain to better reaction time. The system exchanges node of one server to an alternate server in round robin manner and after that scheduling is done in jobs time sharing among jobs in equal slice or quantum and in circular queue without priority so it is simple and easy to implement, it focuses on fairness of jobs distribution. The benefit of this algorithm is no job has to wait for another one to be completed. Again this technique is not a decent decision for employments that are generally changes in their size and necessities, by means, an occupation is never been fulfilled which thusly prompts starvation or uncertain blocking. With Round Robin scheduling have proposed a changed methodology of load balancing component which is a mix of two sorts of scheduling i.e. round robin planning and Batch Mode Heuristic need Algorithm. In this manner this procedure is utilized to upgrade the round robin planning and supported the framework by adding Heuristic Priority to streamline throughput. [5]

Pedro Martins, Maryam Abbasi, Pedro Furtado [6] further extended the architecture for load balancing and scaling with

queries and heavy data processing. The paper concentrate on recognizing load balancing issue, rescheduling queries controlling query, transforming and accomplishing elasticity. Thus this paper tries to integrate all the factors essential for speedy processing of queries without adding up of additional nodes. It aims at attending maximum throughput with available nodes and resources. This paper is constrained onto database and paper suggested that in future it can be utilized on cloud environment thus we proposed a framework which will utilized as a part of opportune execution of query in CEP and databases together on cloud environment.

III. PROBLEM DEFINITION

Proposed System will balance Workload by queuing the queries based on capacity of Node also Enhancing Scalability and throughput in dynamic environment by scheduling the load , control the load, to attain optimum scalability even in heavy queries without loss of data even in system crash, and alert system and automatic transfer of queries to node with available capacity with lowest turnaround time. Proposed System aims to achieve scalability and efficiency through architecture for load balancing and query processing in Distributed Database.

IV. PROPOSED SYSTEM

Proposed System consists of following phases:

- Replicating incoming Query
- Scheduling the Queries
- Overload detection.
- Event generation and handling

REPLICATING INCOMING QUERY

Incoming CEP from ERP system are replicated to all nodes. Each processing node has queue so new data is added to it. Replicate data to improve data availability and query response time. Performance is improved because the fragment replica is stored at the nodes where they are frequently needed.

SCHEDULING THE QUERY

Proposed System schedules the incoming data, decision of distribution of query are depend upon load balancing algorithm. A number of load balancing algorithm are there like Round Robin (RR), Least Weighted (LW) etc. Proposed system is based on least work Least Work based on the number of queries running. This algorithm requires knowledge about the number of queries running at each node, and chooses the node with less queries at the assignment instant. Finally, the Least Weight (LW) algorithm needs to measure current load in terms of parameters such as CPU, memory and IO in order to determine the less loaded node, then it assigns the query to the less-loaded node.

OVERLOAD DETECTION

When a new query arrives at the scheduler it is send to the node with less load. If the queue of the processing node reaches a limit size, then the query is removed from it, and put to run in the ready node, ready node becomes a processing node. Elasticity and scalability is achieved by adding new nodes to the set of ready-nodes.

When a node has a small minimum number of queries and minimum load, the resource is deprovisioned. The node tries to free resources by submitting the queries to the scheduler. If it gets free, the node will be set on standby as a ready-node.

EVENT HANDLING AND ALERT

Every time a P/C queue reaches the maximum size (configurable parameter), queries removal or load shedding decisions need to be made. If all the previews options are exhausted and the system is still overloaded it will alert the administrator, indicating the node and queries in overload condition. The administrator can decide to add more ready-nodes, remove more queries.

V. FLOW CHART

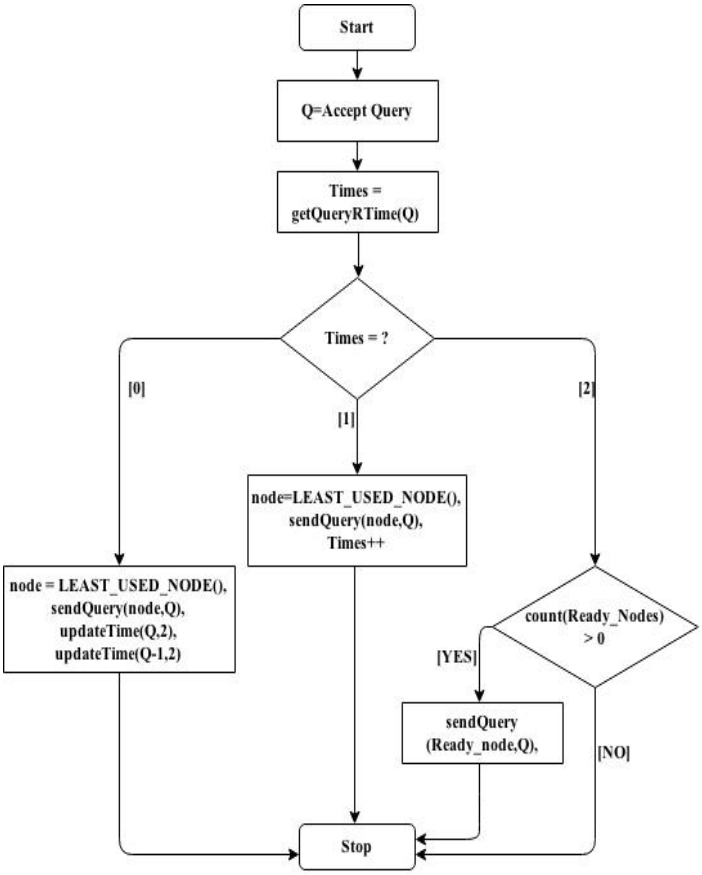

Fig 3: Flow of Scheduler

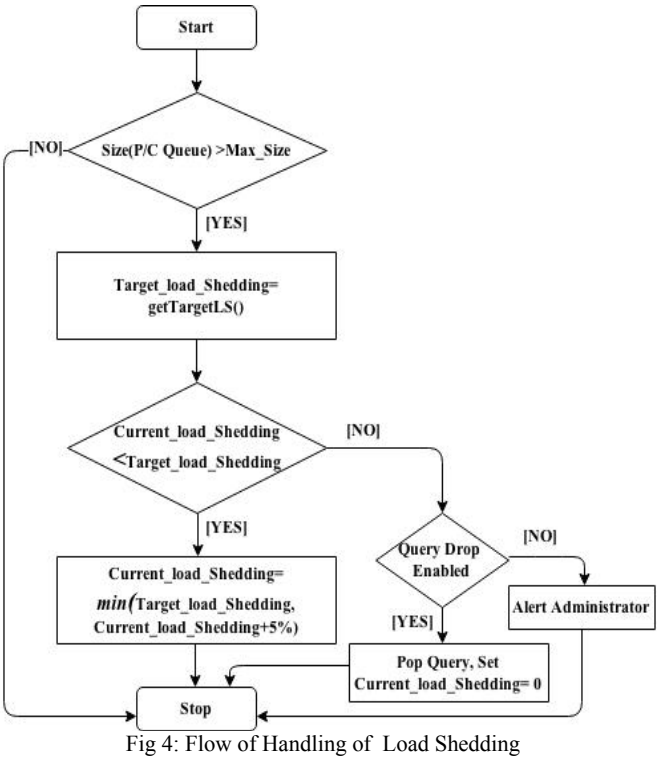

Fig 4: Flow of Handling of Load Shedding

Flow of load balancing and shedding are shown in above flowchart. The system is comprised by:

1. A data distribution node, for replication by all nodes of the incoming data. Each processing node has a Producer-Consumer (P/C) queue, so the new data is added to it.
2. Scheduler node, for scheduling queries by the nodes accordingly with their current load. Load assessment is based on either round robin, least work remaining based on the number of queries or least weight based on collecting and analyzing memory, CPU and disk I/O of all nodes.
3. CEP processing nodes running queries submitted, each node contains a P/C queue for overload detection. When a node queue reaches a limit size (configured by the user) it is assumed that the node is getting overloaded, thus queries should be relocated. Then the system tries to rebalance queries.
4. Ready-Nodes can be represented by one or more nodes that are in standby, ready to accommodate query relocation. When a query is registered into this node, it is also left running in the node that was getting overloaded until results are produced. Using this strategy, scalability can be obtained without loss of results. At the same time, nodes that are lightly loaded can automatically become ready-nodes. Ready- nodes can be added or removed by an administrator without any problem regarding running workloads. Those queries are (re)submitted to the scheduler.
5. A database system data is inserted in a batch bulk fashion, thus no data is lost.

VI. CONCLUSION

A solution for any loaded system is to parallelize the load though many machines or cores. However nodes can still overload. Hence an integrated approach is proposed to increase scalability of query processing with automated architecture for overload mitigation, scalability, Query processing control, Alerts in case of overload situation and node failure. Various researchers has put forward mechanism for load balancing in networking and cloud environment but this approach provides unique and integrated approach and considers many factors in unison to provide best possible results.

REFERENCES

[1] N. Leavitt, “Complex-event processing poised for growth”. IEEE Computer, 2009.
[2] Yunhua Deng, Rynson W.H. Lau, “Heat diffusion based dynamic load balancing for distributed virtual environments”, in: Proceedings of the17th ACM Symposium on Virtual Reality Software and Technology, ACM, 2010.
[3] Markus Esch, Eric Tobias, “Decentralized scale-free network construction and load balancing in Massive Multiuser Virtual Environments”, in: Collaborative Computing: Networking, Applications and Work sharing, Collaborate Com, 2010, 6th International Conference on, IEEE, 2010.
[4] Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao, Li Ruan, “A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers”, J. Parallel Distribution Computing,2012.
[5] Dhinesh Babu L.D, P. VenkataKrishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments”,Applied Soft Computing 13,2013.
[6] Raj, G. Punjab Tech. Univ., Jalandhar, India Singh, D. Bansal, A. "Load balancing for resource provisioning using Batch Mode Heuristic Priority in Round Robin (PBRR) Scheduling" IEEE Confluence 2013: The Next Generation Information Technology Summit (4th International Conference).
[7] Pedro Martins, Maryam Abbasi, Pedro Furtado , “AuDy: Automatic Dynamic Least-Weight Balancing for Stream Workloads Scalability”, 2014 IEEE International Congress on Big Data.
[8] U. Karthick Kumar , “A Dynamic Load Balancing Algorithm in Computational Grid Using Fair Scheduling”, IJCSI International Journal of Computer Science Issues, Vol. 8, September 2011.
[9] Godfrey B, Lakshminarayanan K, Surana S, Karp R, “Load balancing in dynamic structured P2P systems”, IEEE.”
[10] Article by Mayanka Katyal, Atul Mishra, “A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment” Article can be accessed online at http://www.publishingindia.com
[11] Article by Anisn Nasir on **“**Load Balancing in Stream Processing Engines**”.**
[12] D. C. Luckham. The power of events - an introduction to complex event processing in distributed enterprise systems. ACM, 2005.
[13] Pranjali Dhore, Kishor Kolhe, “A Survey for Dynamic Balancing of workload & Scalability in Cloud Environment”, International Journal of Allied Practice, Research and Review, Vol II, Issue I, p.n. 19-24, 2015.