IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Issue-4 E-ISSN: 2347-2693

Selective Load Balancing System of Video Traffic in Wireless Networks

J.Sagana[1*], A.Ashlin Jeba[2]

[1] *M.E Scholar, Department of Electronics & Communication Engineering, A.R.J College of Engineering & Technology, Mannargudi.*

[2] *Assistant Professor, Department of Electronics & Communication Engineering, A.R.J College of Engineering & Technology, Mannargudi.*

www.ijcseonline.org

Received: Mar/23/2016 Revised: Apr /03/2016 Accepted: Apr/19/2016 Published: Apr/30/2016

Abstract— The state-of-art of the innovation focuses on data handling and, sharing to deal with enormous sum of data and, client's needs. Wireless network is a promising technology, which empowers one to achieve the aforesaid goal, leading towards enhanced business performance. Wireless network comes into focus of consideration immediately when you think about what IT constantly needs: a implies to increment capacity or add abilities on the fly without investing in new infrastructure, training new human resources, or licensing new software. The network should give resources on demand, to its customers with high availability, scalable and, with decreased cost. Wireless network Framework has widely been adopted by the industry, though there are numerous existing issues which have not been so far wholly addressed. Load balancing is one of the primary challenges, which is required to distribute the dynamic workload over distinctive hubs to ensure that no single hub is overwhelmed. This Paper gives an effective dynamic load balancing calculation for network workload administration by which the load can be dispersed not only in a adjusted approach, but moreover it dispenses the load systematically and, uniformly by checking certain parameters like number of demands the server is handling currently. It parities the load on the over-stacked hub to under stacked hub so that reaction time from the server will diminish and, execution of the framework is increased.

Keywords— *Load Balancing, Network Framework, Wireless network.*

I. INTRODUCTION

Wireless network promises to increment the velocity with which applications are deployed, enhance modernization, and, lower expenses, all at the same time increasing business agility. Wireless network is a concept that has numerous PCs interconnected through a genuine time system like internet. Wireless network fundamentally refers to dispersed computing. Wireless network empowers well-situated, on-demand, dynamic and, reliable use of dispersed figuring assets. The network is altering our life by giving clients with new kinds of services. Clients acquire administration from a network without paying consideration to the details. Wireless network is a on demand, administration in which shared resources work together to perform a assignment to get the results in slightest conceivable time by circulation of any dataset among all the connected handling units. Wireless network is moreover referred to allude the system based administrations which give an illusion of giving a genuine server equipment but in genuine it is simulated by the software's running on one or more genuine machines. Such virtual servers do not exist physically so they can be scaled up and, down at any point of time. Wireless network is high utility programming having the capacity to change the IT programming industry and, making the programming indeed more attractive. Hence, It helps to accommodate changes in demand, and, helps any association in avoiding the capital costs of programming and, equipment.

Wireless network exhibits several characteristics:

On demand, self-services: PC administrations like email, applications, system or server administration can be provided with no necessity of human collaboration with each administration provider. Network administration providers giving these administrations on demand, self-administrations are Amazon Web Administrations (AWS), Microsoft, Google, IBM and, Salesforce.com.. Gartner describes this quality as administration based. New York Times and, NASDAQ are examples of organizations utilizing AWS (NIST).

Broad system access: Network Abilities are offered over the system and, accessed through standard mechanisms that encourage use by mixed thin or thick customer platform such as mobile phones, laptops along with PDAs.

Asset pooling: The provider's figuring resources are pooled together to supply distinctive customers utilizing multiple-tenant model, with diverse physical and, virtual resources powerfully assigned and, reassigned according to end user

demand. The resources include among others storage space, processing, memory, system bandwidth, virtual machines moreover email services. The pooling collectively of the asset builds economies of scale (Gartner).

Rapid elasticity: Network administrations can be rapidly and, elastically provisioned, in some cases automatically, to swiftly scale out and, rapidly released to rapidly scale in. To the consumer, the abilities accessible for provisioning habitually emerge as unlimited and, can be purchased in any quantity at any time.

Measured service: Wireless network source utilization can be measured, controlled, and, reported given that transparency for both the supplier and, purchaser of the utilized service. Wireless network administrations apply a metering capacity which empowers to control and, optimize asset use. This implies that just similar to air time, electricity or municipality water IT administrations are charged per utilization metrics – pay per use. The additional you use the higher the bill. Just as utility organizations offer authority or power to subscribers, in addition to telephone organizations offer voice and, data services, IT administrations like system security management, data focus hosting or yet departmental billing can presently be effortlessly delivered as a contractual service.

Multi Tenacity: It is the 6th characteristics of wireless network advocated by the Network Security Alliance. It allude to the need for policy-driven enforcement, segmentation, separation, governance, administration levels, as well as chargeback/billing models for distinctive purchaser constituencies. Consumers may use a public network provider's administration offerings or actually be from the same organization, like distinctive business units rather than distinct organizational entities, however would still share infrastructure.

There are numerous problems predominant in wireless network, Such as:

- Ensuring fitting access control (authentication, authorization, as well as auditing)
- System level migration, so that it requires slightest fetched and, time to shift a work
- To offer correct security to the data in transit and, to the data at rest.
- Data availability issues in network
- Official quagmire and, transitive trust issues
- Data lineage, data origin and, inadvertent leak of sensitive data is possible.

And, the most predominant issue in Wireless network is the issue of Load Balancing.

II. NECESSITY OF LOAD BALANCING

Load balancing is a PC system strategy for circulating workloads over distinctive figuring resources, for case computers, a PC cluster, system links, focal handling units or disk drives. Load balancing plans to optimize asset use, maximize throughput, minimize reaction time, and, evade overload of any one of the resources. By the use of distinctive parts with load balancing instead of a single part may increment reliable through redundancy.

Load balancing in the network differs from classical thinking on load-adjusting architecture and, implementation by utilizing commodity servers to perform the load balancing since it's difficult to predict the number of demands that will be issued to a server. This provides for new opportunities and, economies-of-scale, moreover presenting its own unique set of challenges. Load balancing is one of the focal issues in wireless network. It is a component that distributes the dynamic neighborhood workload evenly over all the hubs in the entirety network to avoid a situation where some hubs are intensely stacked while others are idle or doing little work. It helps to accomplish a high customer satisfaction and, asset use ratio, consequently improving the overall execution and, asset utility of the system. It moreover makes sure that each figuring asset is dispersed efficiently and, fairly. It further prevents bottlenecks of the framework which may occur due to load imbalance. When one or more parts of any administration stop working, load balancing encourages in continuation of the administration by implementing fair-over, i.e. in provisioning and, de-provisioning of instances of applications without fail. Fig 1 depicts the Load balancing necessity in network when there are demands from distinctive clients. The existing load balancing strategies in networks, consider different parameters such as performance, reaction time, scalability, throughput, asset utilization, deficiency tolerance, relocation time and, related overhead. The emerging wireless network model attempts to address the explosive growth of web-connected devices, and, handle massive amounts of data and, customer demands. Thereby, giving rise to the question whether our network model is able to parity the ever-increasing load in an effective way or not.

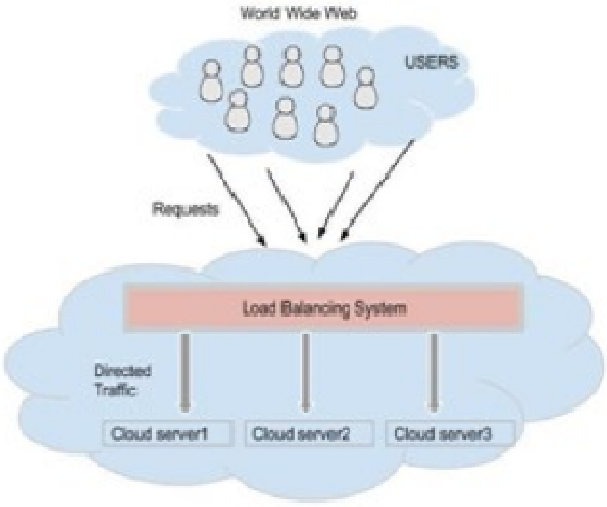

Fig 1. Load balancing framework in wireless network

III. LITERATURE REVIEW

Some of the chief goals of a load balancing algorithm, as pointed out by are:

- To accomplish a greater overall progress in framework execution at a realistic cost, e.g., diminish assignment reaction time while keeping acceptable delays;
- To treat all jobs in the framework similarly not considering their origin;
- To encompass a deficiency tolerance: execution survival under partial failure in the system;
- To have the capacity to alter itself in accordance with any changes;
- preserve framework stable

The important things to consider while developing such calculation are : 1)estimation of load 2)comparison of load 3)stable of distinctive systems 4)execution of framework 5)collaboration between the hubs 6)nature of work to be transferred 7) selecting of hubs and, numerous other ones. This load considered can be in terms of CPU load, sum of memory used, delay or else System load.

We can divide Load balancing calculations into 2 categories, Depending on the state of the system.

- *Static:* It doesn't depend on the current state of the system. Earlier data of the framework is essential.
- *Dynamic:* Choices on load balancing are based on current state of the system. No earlier data is required. Thus it is better than static approach.

In a dispersed system, dynamic load balancing can be done in two distinctive ways:

- Dispersed
- Non-dispersed

A dynamic load balancing calculation assumes no past data about work actions or the global state of the system, i.e., load balancing choices is only based on the existing or current status of the system. In the dispersed one, the dynamic load balancing calculation is executed by all hubs present in the framework and, the assignment of load balancing is shared among them. The collaboration among hubs to realize load balancing can take two forms: 1) helpful and, 2) non-cooperative. In the cooperative, the hubs work side by side to accomplish a common goal, for example, to advance the overall reaction time, etc. In the non-cooperative, each hub works independently in the direction of a objective neighborhood to it, for example, to advance the reaction time of a neighborhood task. Dynamic load balancing calculations having dispersed nature, habitually produce more messages than the non-dispersed ones because, each of the hubs in the framework is required to collaborate with each other node. The advantage, of this is that indeed if one or more hubs in the arrangement fail, it will not cause the total load balancing process to stop; it instead would influence the framework execution to a little extent.

In non-dispersed type, either one hub or a group of hubs perform the assignment of load balancing. Dynamic load balancing calculations of non-dispersed nature can get two forms: 1) unified and, 2) semi-distributed. In the centralized, the load balancing calculation is executed just by a single hub in the total system: the focal node. This hub is only in charge for load balancing of the entirety system. The other hubs collaborate merely with the focal node. However, in semi-dispersed form, hubs are partitioned into clusters, where the load balancing in each bunch is of unified form. A focal hub is chosen in each bunch by suitable election strategy which takes care of load balancing inside that cluster. Hence, the load balancing of the complete framework is done via the focal hubs of each cluster. Unified dynamic load balancing takes less messages to arrive at a decision, since the number of overall interactions in the framework diminishes drastically as analyzed to the semi-dispersed case. However, unified calculations can create a bottleneck in the framework at the focal hub and, moreover the load balancing process is rendered hopeless once the focal hub crashes. Therefore, this calculation is fundamentally suited for networks with

little size. Thus, fig 2 appears the summarization of dynamic load balancing technique.

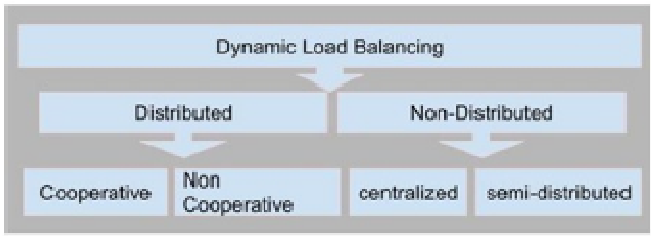

Fig 2: Summarizing Dynamic load balancing techniques

Round Robin: In this calculation, the frames are divided between all processors. Each process is handed over to the processor in a round robin order. The process circulation request is maintained in the vicinity autonomous of the allotments from remote processors. However the work load distributions between processors are the same but the work handling time for dissimilar frames are not same. So by any point of time some hubs may be greatly stacked and, others wait at leisure. This calculation is habitually utilized in web servers where http demands are of alike nature and, scattered likewise.

Association Mechanism: Load balancing calculation can as well be based on slightest association component which is a part of dynamic planning algorithm. It requires to count the number of connections for each server powerfully to approximate the load. The load balancer keeps track of the association number of each server. The number of link adds to when a new association is sent out to it, and, diminishes the number when association terminate or timeout happens.

A Assignment Planning Calculation Based on Load Balancing: This is discussed in a two-level assignment planning strategy based on load balancing to convene dynamic requirements of clients and, obtain high asset utilization. It accomplishes load balancing by first mapping tasks to virtual machines and, then virtual machines to host resources by this implies improving the assignment reaction time, asset consumption and, overall execution of the wireless network environment.

Randomized: Randomized calculation is of sort static in nature. In this calculation a process can be handled by a specific hub n with a probable p. The process allocation request is preserved for each processor autonomous of circulation from remote processor. This calculation encourages well in case of frames that are equal loaded. On the other hand, trouble arises when loads are of distinctive computational complexities. Randomized calculation does not keep up deterministic approach. It encourages well while Round Robin calculation generates overhead for process queue.

IV. PROPOSED SYSTEM

A dynamic load balancing calculation makes load circulation choices based on the current work load at each hub of the dispersed system. Accordingly, this calculation must give a implies for collecting and, managing framework status information.

The calculation handles the demands in a capable way. It starts by checking the counter variable of each server hub and, data center. After checking, it transfers the load accordingly by choosing the slightest esteem of the counter variable and, the request is handled effortlessly and, takes a littler sum of time, and, offers maximum throughput. The randomly transfer of load can cause some server to intensely stacked while other server is daintily loaded. If the load is similarly dispersed it not only improves execution moreover reduces the time delay. This calculation not only parities the load but moreover it improves the reaction time for the network. While taking into account the impact of fetched optimization one has to think on the subject of the solution to this difficulty.

A counter variable is related with each node. Counter variable is the number of demands that the specific server hub or data focus is presently handling. Each hub is having distinctive data focuses as appeared in fig 3. The esteem of counter variable of server hub will be equal to the sum of counter variables of its data centers.

This calculation essentially dispenses request which is coming from the customer hubs to the daintily stacked server bunch (Data Center) and, gives the reaction in a decreased sum of time by doing this, it makes the calculation capable for reaction to request ratio. We can see that the customers at a same time make demands to access the network application over the internet.

We proposed k-means technique for clustering. K means algorithm findout the distance between the clients. First findout the centroid point and form a cluster based on a distance. Then transmit the data for a clustered clients. Here data will be send as packets wise to destination node. Our load balancing technique using clustering can also increase network scalability and also performance will be efficient.

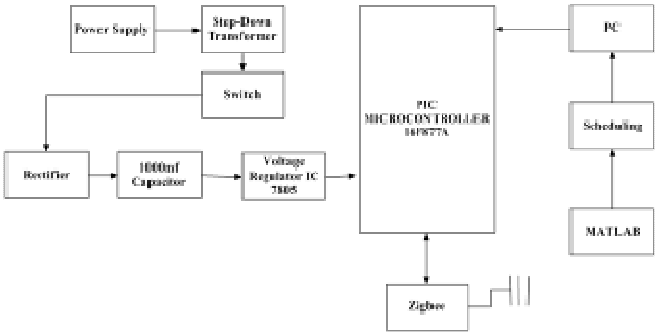

Fig 3: Base Station

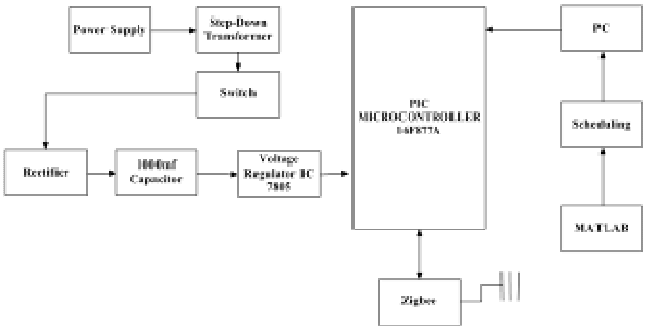

Fig 4. Block diagram of node 1

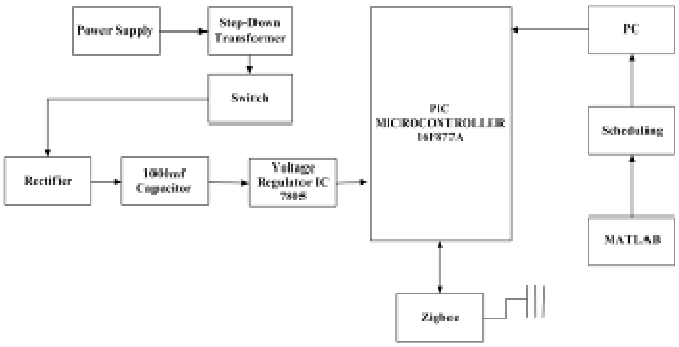

Fig 5. Block diagram of node 2

Algorithm:

1) Select the hub with slightest esteem of counter variable (i.e. slightest number of demands allotted)
2) Relegate the application request to the chosen node.
3) Select the data focus (of the chosen hub in step 1) that has slightest esteem of counter variable.
4) Relegate the customer request to the chosen data focus
5) Increment the counter variable of the data focus by 1
6) Execute the application request
7) Diminish the esteem of counter variable by 1

Presently in this calculation all the demands goes through the load balancer framework as appeared in fig 2 by which it checks the counter variable which is related with each server hub set to the maximum demands presently handled by a server node. Here cntA, cntB, cntA1, cntA2, cntB1, cntB2 are the counter variables of server hub A, server hub B, data focus A1, data focus A2, data focus B1, data focus B2 respectively as appeared in table 1. Let us assume that the hub A is having a counter esteem to 120, hub B is having the counter variable to 115. Hub B is handling the littler number of demands analyzed to hub A, so here the load balancer will stabilize the load (requests) to hub B as it is less consequently the adjusting is done at this level. Now, deciding which server bunch (data center) of hub B will handle the request. Suppose in hub B, bunch B1 is having counter variable set to 70, bunch B2 to 45. As bunch B2 is handling littler number of demands analyzed to other bunch of hub B, so the request will be allocated to bunch B2 in request to parity the overall load and, counter variable related with bunch B2 will be increased by 1. And, moreover counter variable related with server hub B will be increased by 1 i.e. presently it will become 116 as appeared in table 2. Figure 4 appears the workflow of the algorithm. Till presently we have handled the request however how the counter variable will get updated? The answer is the servers which the counter variable is related with, will simultaneously change (update) the counter variable. When a reaction is given back to the customer the server will consequently diminish its counter variable by the number 1 and, moreover the counter variable of related bunch will be decremented by 1, so that each time the calculation will have the upgraded esteem of counter variable. Therefore, demands are handled effortlessly by Server Clusters. The potency of server can be expanded or decreased by the administration supplier on request and, for data focuses too. So no necessity of Round Robin Adjusting or any other practice where time is consumed and, reaction to request proportion is little for enormous number of requests.

Component	Counter Variable	value
Server Node A	cntA	120
Data Center A1	cntA1	65
Data Center A2	cntA2	55
Server Node B	cntB	115
Data Center B1	cntB1	70
Data Center B2	cntB2	45

Table 1: Initial state of the system

Component	Counter Variable	value
Server Node A	cntA	120
Data Center A1	cntA1	65
Data Center A2	cntA2	55
Server Node B	cntB	116
Data Center B1	cntB1	70
Data Center B2	cntB2	46

Table 2: State of the framework after assigning customer request

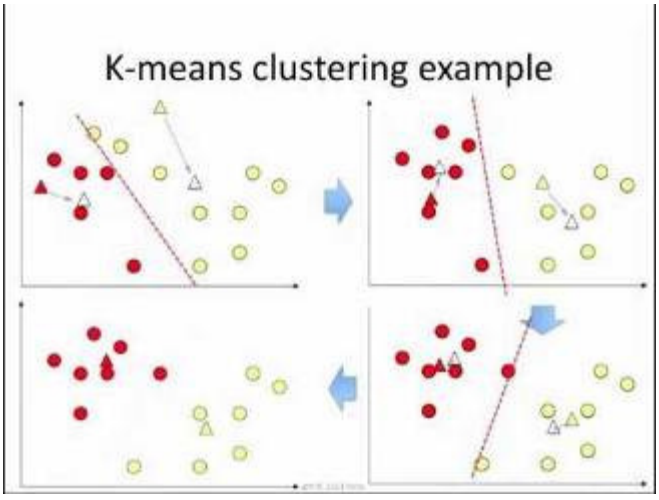

Fig 6. Workflow of the algorithm

V. CONCLUSION

Existing Load balancing techniques/Calculations that have been considered largely focus on decreasing overhead, decreasing the relocation time and, improving execution etc., but the reaction to request proportion is rarely considered. It is a challenge of each engineer to build up the network platform that can raise the throughput. In proposed algorithm, the request is allocated as early as conceivable to the fitting data center. As there are different server hubs having distinctive data centers, the reaction is given at the earliest, thereby circulating the load in a adjusted and, effective manner without any delay. Since of the dynamic nature of the algorithm, there is no need to have the earlier data of the state of the system; consequently the overhead for storing the past state of the framework is moreover eliminated.

References

[1] M. A. Kashem; V. Ganapathy; G. B. Jasmon, "A geometric approach for three-phase load balancing in distribution networks", Power System Technology, 2000. Proceedings. PowerCon 2000. International Conference on, Year: 2000, Volume: 1, Pages: 293 – 298.

[2] Jasween Kaur, Kiranbir Kaur, "A Study of Fuzzy Based Dynamic Load Balancing for Cell Networks", International Journal of Computer Sciences and Engineering, Volume-04, Issue-01, Page No (70-75), Jan -2016

[3] Nico Kruber; Mikael Högqvist; Thorsten Schütt, "The Benefits of Estimated Global Information in DHT Load Balancing", Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on, Year: 2011, Pages: 382 – 391.

[4] Zhihao Shang; Wenbo Chen; Qiang Ma; Bin Wu, "Design and implementation of server cluster dynamic load balancing based on OpenFlow", Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference on, Year: 2013,Pages: 691 – 697.

[5] Quentin Bragard; Anthony Ventresque; Liam Murphy, "Global dynamic load-balancing for decentralised distributed simulation", Proceedings of the Winter Simulation Conference 2014, Year: 2014, Pages: 3797 – 3808.

[6] Yingwu Zhu; Yiming Hu, "Efficient, proximity-aware load balancing for structured P2P systems", Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on, Year: 2003, Pages: 220 – 221.

[7] Hye-Seon Maeng; Hyoun-Su Lee; Tack-Don Han; Sung-Bong Yang; Shin-Dug Kim, "Dynamic load balancing of iterative data parallel problems on a workstation cluster" High Performance Computing on the Information Superhighway, 1997. HPC Asia '97, Year: 1997, Pages: 563 – 567.

[8] Vaishnavi Aher,Sayali Khairnar,Madhuri Shinde and Priyanka Shirole, "Load Balancing of node in Network using Ant Colony Optimization", International Journal of Computer Sciences and Engineering, Volume-03, Issue-01, Page No (105-108), Jan -2015

[9] Rajkumar Rajavel, "De-Centralized Load Balancing for the Computational Grid environment", Communication and Computational Intelligence (INCOCCI), 2010 International Conference on, Year: 2010, Pages: 419 – 424.

[10] Gengbin Zheng; Esteban Meneses; Abhinav Bhatele; Laxmikant V. Kale, "Hierarchical Load Balancing for Charm++ Applications on Large Supercomputers", 2010 39th International Conference on Parallel Processing Workshops, Year: 2010, Pages: 436 – 444.

