Vol. 7, Issue 5,February 2018

ISSN 2249-894X

REVIEW OF RESEARCH

An International Multidisciplinary Peer Reviewed & Refereed Journal

Impact Factor: 5.2331

UGC Approved Journal No. 48514

REVIEW OF RESEARCH

ISSN: 2249-894X
IMPACT FACTOR : 5.2331(UIF)
VOLUME - 7 | ISSUE - 5 | FEBRUARY - 2018

EFFICIENT DISTRIBUTION OF TASKS IN DISTRIBUTING SYSTEMS

Dr. Putti SrinivasaRao
Professor & HOD in CSE & Dean of PG Studies ,
JBIET Telangana ,Hyderabad, India .

ABSTRACT:

In this paper , A robust scalable and distributed job processing system is presented that adapts to the dynamic computing needs easily with efficient and dynamic distribution of tasks for heterogeneous systems with a de-centralised approach. The approach takes the drawbacks and limitations in the previous implementations of other job processing systems .The approach also provides a mechanism for improving the throughput.

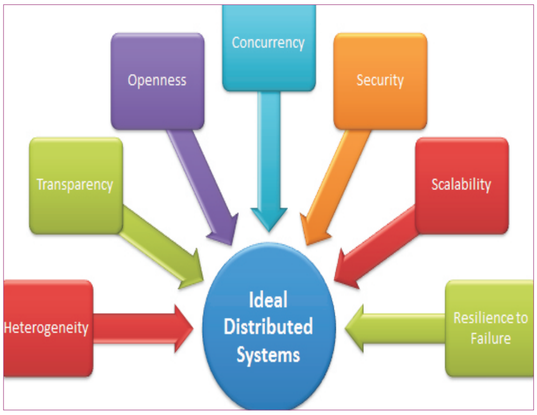
KEYWORDS: *dynamic computing , de-centralised approach , job processing systems .*

I. INTRODUCTION:

A Ddistributed System consists of several systems connected by a network. In a distributed computing environment, each node is independent, has its own resources. The nodes are however interconnected over a LAN or WAN. Such interconnection allows complete monitoring and coordination of the nodes/processors as well as the progress of the task execution. Distributed components need to be loosely coupled, yet they have to support certain common features like uniform protocol of communication, auto-discovery, periodic health-check, common management protocol etc.

II. EXISTING DISTRIBUTED JOB PROCESSING SYSTEM

A node is responsible for processing as well as job forwarding if it is loaded. When a node receives a job and is loaded beyond its capacity, it needs to query the status of other nodes to find out if any other node can share the load.Quite a good amount of time is wasted at each node to query other nodes. This time could have been utilized for processing the job. In a complex network, having multiple sub-networks, configuring each node for locating other nodes is a complex task.Assuming, the nodes use broadcast to announce their status, this also causes enormous load on the network.When every node requests the status of other nodes periodically, the network overhead increases many fold



Load Balancing in distributed systems is studied deeply for a long time. Bharadwaj and Zeng studied about different Dynamic Load Balancing adjustment Policies for Distributed systems like Queue adjustment and rate adjustment, combination of both queue and rate adjustment for different load conditions like high, moderate and low [1]. They focused to minimize the mean response time of the jobs.

Abhijit A. Rajguru presented analysis on performance and made a comparative study of Load Balancing Algorithms in Distributed computing System using Parameters like reliability, stability, response time and execution Time etc .

Iman Sadooghi, Ioan Raicu presented a framework which is a compact, scalable, light-weight, and distributed task execution framework (CloudKon) CloudKon is a distributed job management system that can support millions of tasks/jobs.

Andrei and Arjan showed that different technique of scheduling to lower cost than existing methods, without decreasing the performance . They proved that it will reduce the time complexity of a specific type of scheduling, considering the dynamic priorities.

Yan, Wang, presented different type of load balancing policy to have stability and performance and to maintain the execution performance of the system using static and dynamic load balancing .

Kevin Barker, Andrey Chernikov presented simple and effective load balancing framework [6]. It will support the application development on distributed-memory and parallel systems. The framework allows message forwarding automatic load balancing, global namespace and object migration.

Jorge E. Pezoa Sagar introduced a decentralized load balancing framework for heterogeneous distributed-computing system [8]. The aim of this frame work is to increase the reliability in the presence of failures. The reliability in the Distributed computing system is calculated based on the probability of specific workload service. For each submitted workload, Load balancing is done synchronously at runtime by all the nodes.

Young Joon Lee proposed an approach to reduce the simulation time in distributed system on multi-agent platform . They used different agents time management agent is introduced along with other agents to check the hardware resources for proper use and to reduce the simulation time by dynamic load balancing. The experimental results showed that the scheme greatly reduces the simulation time than the existing scheme.

Deng Huafeng proposed an algorithm for load balancing in distributed systems . Based on Min-min algorithm, jobs are distributed to the nodes. Min-min algorithm focuses on the accuracy. Their results show that this algorithm will have good performance and better load balancing.

Gábor Vincze, Zoltán Novák, Zoltán Pap, Rolland Vid proposed a information system for distributed applications with a new load balancing approach, to handle load balancing problem .

Azzedine Boukerche presented a hierarchical dynamic load balancing scheme and it is designed to facilitate the re-distribution of load in distributed systems. The balancing algorithm is split in to three phases that migrate load, re-distribute, and detects external loads and differences of loads in heterogeneous resources.

Sagar Dhakal, Majeed M Proposed a model to characterize the average overall completion time in a distributed computing system . This model considers the heterogeneity nodes and randomness in the delays due to communication channel. The proposed dynamic load-balancing policy performance is compared with performance of the existing static policies and dynamic load-balancing policies

Casavant and Kuhl studied about the behaviour and structure of decision-making policies, taking into account the performance and efficiency to the load sharing policies .

Shah proposed decentralized adaptive dynamic load balancing algorithms for grids. Many constraints like job migration cost, communication delay, processing delays are considered for load balancing.

III .PROPOSED APPROACH TO DISTRIBUTE THE JOBS

The various components of the system are

1.Job Dispatcher accepts new job requests validates them and places the jobs in the Job Queue for scheduling.

2.Job Scheduler accepts the job requests from the Dispatch queue and schedules jobs for processing based on the load .

3.Job processor picks up a job request from the Processing queue, processes

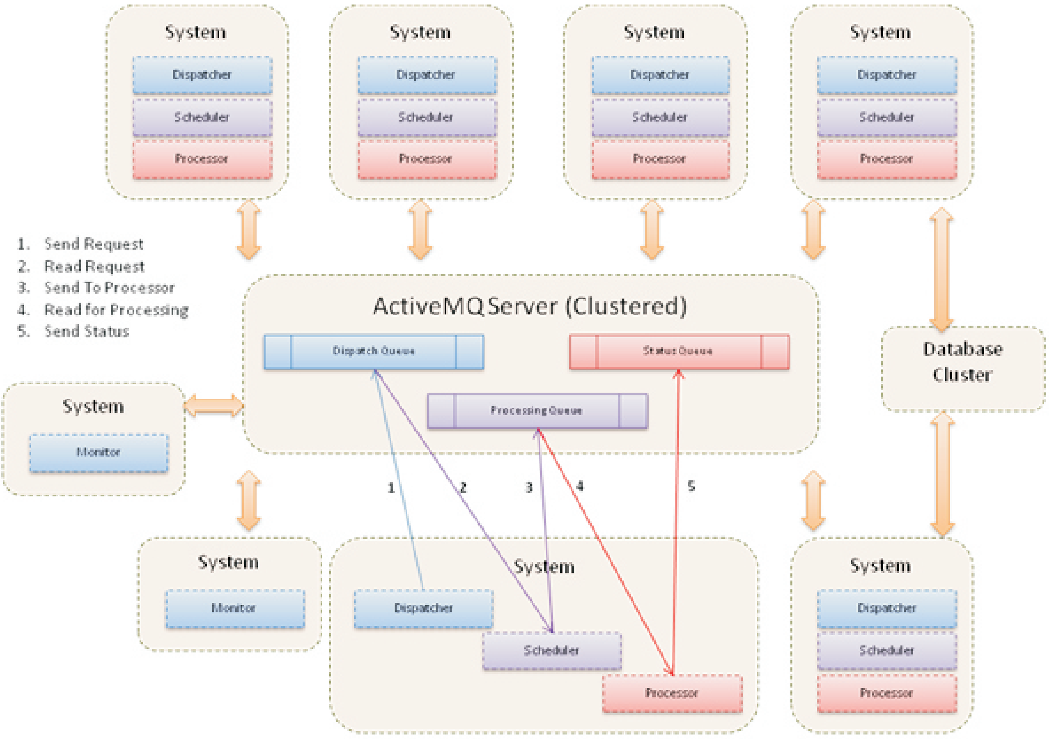
4.Job Monitor monitors the status messages and updates the database

5.Dispatch Queue stores the job requests dispatched until it is picked up by the scheduler

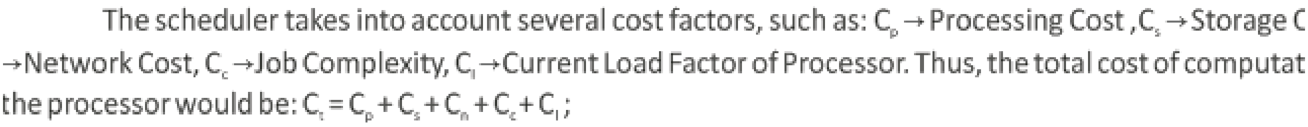
Processing queue that stores the scheduled jobs until a processor picks them up for processing

6.Progress / Status Queue store the job status sent by either dispatcher or processor

7.Database / Persistence All the information about the Job, the Processors, the state of processing and the availability of processors are maintained



When a job is requested, the scheduler checks for the target processor and its availability. It also checks if the job can be processed by alternate processors. Using such a technique, processor-affinity can be defined for jobs if necessary. It then checks for the processor, among all possible targets, having the least load and that is active


The scheduler takes into account several cost factors, such as: C_p → Processing Cost ,C_s → Storage C →Network Cost, C_c →Job Complexity, C_l →Current Load Factor of Processor. Thus, the total cost of computat the processor would be: $C_t = C_p + C_s + C_n + C_c + C_l$;

The proposed model avoids every processor (or sender/receiver) having to be concerned about identifying the load of other processors and routing the job requests. This minimizes the processing overhead on the processors and communication/network overhead on the network. The framework supports recovery of Jobs under processing at the time of a processor crash. This implementation will make the framework a completely safe and reliable

As compared to the existing implementations, the proposed model is quite flexible and can be scaled up and scale out easily by changing few configuration parameters. As explained earlier, new jobs can be added easily by writing a job that implements the interface defined.

VI CONCLUSION:

This paper presented a Effective Distribution mechanism of tasks in distributed systems. it also explained how the different component will help cooperate to distribute the task dynamically to improve the performance and throughput. This framework can be enhanced to middleware. It can easily be integrated with Cloud Computing platforms

REFERENCES :

[1] Zeng Zeng, and Bharadwaj Veeravalli "Design and Performance Evaluation of Queue-and-Rate-Adjustment Dynamic Load Balancing Policies for Distributed Networks", IEEE Transactions on computers, VOL. 55, NO. 11, NOVEMBER 2006.

[2] Abhijit , Rajguruand and S.S. Apte "A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters", International Journal of Recent Technology and Engineering (IJRTE) , ISSN: 2277-3878 Volume-1, Issue-3, August 2012.

[3] Iman Sadoop, Sandeep Palur and Ioan Raicu "Achieving Effiecient Distributed Scheduling with Mesage Queue in Cloud Computing for Many-Task Computing and High –Performance Computing".

[4] M. Mitzenmacher, "The power of Two Choices in Randomized Load Balancing", IEEE Trans. Parallel and Distributed Systems, vol. 12, no. 10, pp. 1094-1104, 2001.

[5] S. Penmatsa, Anthony and T. Chronopoulos, "Dynamic Multi-User Load Balancing in Distributed Systems". IEEE IPDPS, pp. 1-10, 2007.

[6] L. Ni, C. Xu, and T. Gendreau, "A Distributed Drafting Algorithm for Load Balancing", IEEE Transactions on Software Engineering, Vol. SE-11, No. 10, pp. 1153-1161, October 1985.

[7] Ardhendu Mandal and Subhas Chandra Pal, "An Empirical Study and Analysis of the Dynamic Load Balancing Techniques Used in Parallel Computing Systems", ICCS 2010.

[8] Cybenko G, "Dynamic load balancing for distributed memory multiprocessors", Journal of Parallel Distributed Computing, Vol 7, pp. 279-301, 2001.

[9] K. Aida, A. Takefusa, H. Nakada, S. Matsuoka, S. Sekiguchi, and U. Nagashima, Performance evaluation model for scheduling in a global computing system, The International Journal of High Performance Computing Applications, Vol. 14, No. 3, 2000.

[10] J. Skovira, W. Chan, H. Zhou, and D. A. Lifka. "The EASY - LoadLeveler API Project In Job Scheduling Strategies for Parallel Processing", volume 1162 of LNCS, pages 41{47. Springer, 1996.

[11] Orly kremien and Jeff Karmer,"Methodical Analysis of Adaptive Load Sharing Algorithms", IEEE Transactions on parallel and distributed system,Vol 3, No 6,nov 1992.

[12] M. V. Gopalachari, P. sammulal, and A. V. Babu, "Correlating Scheduling and Load Balancing to Achieve Optimal Performance from a Cluster", Proc. IEEE Int. Conf. Advance Computing (IACC), Mar. 2009, pp. 320-325.

[13] G. Cybenko, "Dynamic Load Balancing for Distributed Memory Multiprocessors", "J. Parallel and Distributed Computing", vol. 7, no. 2,pp. 279-301, 1989.

[14] M. Maheswaran, S. Ali, H. J. Siegel , D. Hensgen and R.F. Freund, "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems", Journal of Parallel and Distributed Computing, vol. 59, no.2, , 1999, pp. 107–131.