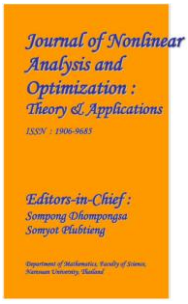
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.4 : 2024
ISSN : **1906-9685**



THE METHODOLOGICAL APPROACH OF GRID COMPUTING EXPLIOTING TECHNIQUES OF CLUSTERING WITHIN THE BOUNDS OF RECENT ACTIVITES

Dr. D. ELANGOVAN, Assistant Professor Department of Computer Science SRM Institute of Science and Technology, Kattankulathur, Chennai, INDIA
Dr.G MANOHARAN Assistant Professor Department of Computer Science St. Thomas College of Arts & Science, Chennai, INDIA

Abstract:
Recent advancements in communication and networking technologies have significantly enhanced network capacity and quality, making high-quality video and audio transmission feasible. This has facilitated the widespread adoption of multimedia transmission and related network applications, contributing to the growth and popularity of e-learning methods such as Distance Education, Video Conferencing, and Video on Demand. Distance Education, particularly, plays a crucial role in e-learning applications, proving to be an effective learning approach whether implemented in traditional educational settings or corporate training programs. Grid computing, on the other hand, allows users to efficiently share data storage, transfer resources, and computing power across a global network. Recognizing the evolving economics of computing and networking, Grid computing offers cost-cutting solutions for technology expenditures and has successfully transitioned from academic and research environments to commercial applications. This paper explores load balancing techniques for media parameters within learning grid clusters, aiming to enhance the viewing experience of e-content and improve user comfort in learning and developing domain knowledge.
Keywords: Grid computing, Load-Balancing, K-mean, clustering.

1. INTRODUCTION

Grid computing technology is widely adopted for deploying e-learning content within the internet as any e-learning setting would demand significant computing resources significantly for a vast pool of concurrent e-content users. Computer code and hardware area unit required to be updated or revived again and again for this setting. [1] A large learning grid could be a machine-cooperative setting for effectively managing massive pools of e-learners online that are turning into vogue. One of the basic problems in such grid environments is job programming which is required to achieve higher performance. As the grid setting is mostly suburbanized and it consists of heterogeneous systems, an economical programming technique would be noticeably required for sound acceptable resources for relevant e-content which says massive or little or the e-content.

1.1 Multicast: This setup enables educators and students to be situated in different places. Utilizing networking technology, classroom video/audio and multimedia teaching materials can be broadcasted in real-time to lecture rooms at remote locations. Moreover, it facilitates two-way interactive communication between instructors and students in these remote lecture rooms.

1.2 Virtual Classroom: This approach employs a management system to replicate the classroom experience, encompassing teacher lectures, written exams, assignments, Q&A sessions, student inquiries, oral quizzes, and more. Both educators and students can connect to the management system online at any given time.

1.3 Video on Demand: Video on Demand (VOD) technology is utilized in this method. Students access teaching materials via the internet using computers or television sets equipped with set-top boxes. This allows the distance learning process to accommodate each student's individual learning pace by controlling the broadcasting process. The teaching approach under examination here integrates both Multicast and VOD methods. [5] Period video teaching is performed at fastened times, with the remainder of the time reserved for net teaching. Grid portals permit communication between grids and also the outside world, and area units perpetually immense, complicated frameworks. The most portal area units are the Appliance Portal (AP) and the User Portal (UP). AP portals permit specific grid operations for specific applications, like the Astronomy Simulation Collaborators (ASC) portal [2] and the Diesel Combustion Collaborators (DSC) portal. User portals offer special services to specific members of the general public and researchers, like the Host Page portal user portal, the entryway project, and UNICORE. From a user stance, the first needs of a Grid vascular system embrace the following:

1.4 Security: Users visit portals victimization net browsers and area units and suggest user IDs and passwords. Whereas higher authentication technologies exist, this one is demanded by users. Safer systems, like sensible cards, and area units are potentially, however unlikely to be deployed at any time presently.

1.5 Remote Access: Tools for accessing file information directories and remote file archives area unit is based on the central portal demand. Grid FTP tools are essential, however several files area unit are mandatorily managed by virtual knowledge systems during which knowledge is catalogued, curettage, and staged by back-end grid services.

1.6 Remote Execution: The power to submit jobs to the Grid for execution and watching could be a customary portal demand. Users allocating specific resources need to be able to see the duty queues on those resources and consult programming assistants. They additionally have to be able to scan logs to stay track of job execution and grasp once operations fail.

1.7 User Data Services: Access to directories and index tools is a vital portal operates. All users ought to have non-public, persistent stores of references to special data they need to keep on the Grid.

1.8 Application interfaces: The key to scientific portals is having the ability to cover Grid details behind helpful application interfaces. Users have to be able to launch, configure, and manage remote applications in the same manner they use desktop applications. [5]

II. METHODOLOGY

2.1 LOAD BALANCING AND CLUSTERING

The thought of Load-Balancing (Figure 2.1) is that tasks or requests are unit-distributed onto multiple computers. For example, [1] I create a regular communications protocol request from a shopper to access an online application. It gets directed to multiple internet servers. Primarily happening here is that the application's workload is distributed among multiple computers, creating it additional scalable.

It additionally helps in providing redundancy, therefore let’s say one server fails in a very cluster, and the load balancer distributes the load among the remaining servers. But once a miscalculation happens, and therefore the request is moved from a failing server to a practical server it is referred to as "failover".

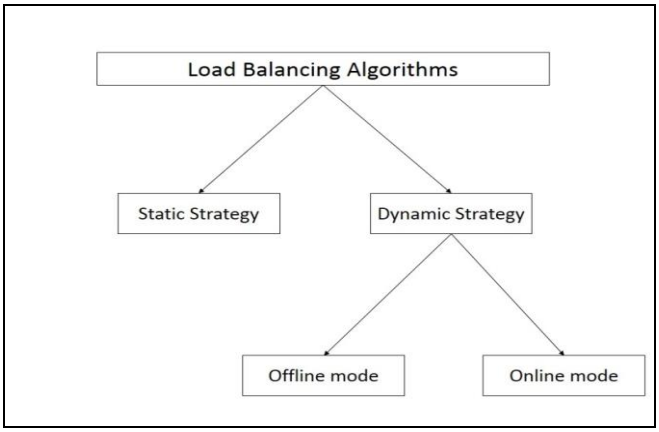


Figure 2.1 Load Balancing Algorithms

Cluster (Figure 2.1) is often a collection of servers running identical applications and its purpose is twofold. To distribute the load onto totally different servers and supply redundancy/failover mechanism, it is referred to as a "Round Robin". [4] Here the tasks area unit is equally distributed between the servers in a cluster. Here every task is equally distributed to a server. It works primarily once all servers have identical capabilities, and every task would like an identical quantity of effort. The problem here is that this technique doesn't take into account that every task, might have a special effort to be processed. Therefore, you will have a scenario where, let us say all server area units are given three tasks such as T1, T2, and T3. On the face of it looks equal, however, T1, and T2 on Server One desire additional effort to the method, then T1, and T2 on Server a group of three tasks. Therefore, in impact Server, one is bearing a heavier load here, than Server a pair of and three, despite good distribution.

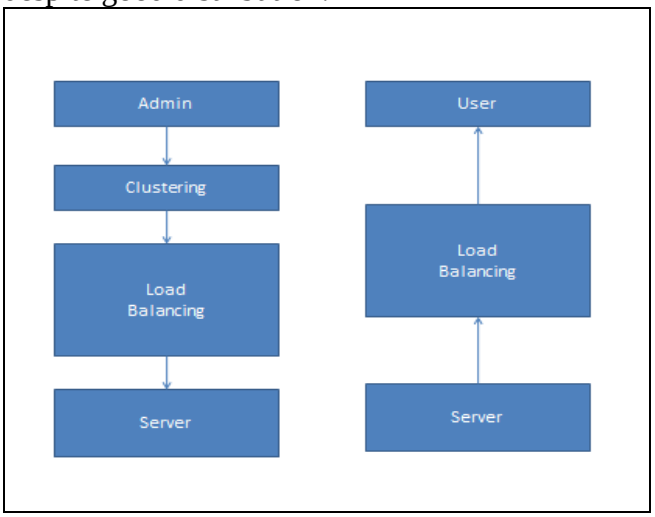


Figure 2.2 Clustering

With a massive set of knowledge and a large pool of user area units concerned in a very dynamic set of learning grids, virtual repositioning may well be tried out for storing and retrieving information. Virtual repositioning is a vital technology that is adopted in e-commerce. It provides flexibility in handling massive data. However, the biggest advantage of the virtual warehouse is value reduction. This can be because of the very fact that structuring of the individual information set for several concurrent users may well be avoided, [2] whereas permitting several such learners to pool from one information warehouse. A mere adaptation of a knowledge warehouse for storing e-content

III. RESULTS AND DISCUSSION

3.1 GRID PROCESSING TIME

The sample data for the proposed experiments contain e-content objects that are in split up form of an integrated lot (both are used in the experiments). Based on the file size and memory utilization of an object, files are clustered into two categories:

- Instructional Parameter

- Media Parameter

The advantages of clustering e-content objects into Instructional Parameter and Media Parameter categories include:

Efficiency: Clustering allows for better organization and management of e-content objects, which can lead to more efficient processing and retrieval.

Resource Allocation: By categorizing objects based on their instructional or media nature, resources such as processing power, memory, and bandwidth can be allocated more effectively, optimizing system performance.

Customization: It enables tailored processing approaches for different types of content, allowing for customized treatment suited to the specific needs of instructional and media parameters.

Scalability: Clustering facilitates scalability, as systems can be designed to handle increasing volumes of content while maintaining efficient processing and retrieval times.

Enhanced User Experience: By efficiently managing instructional and media parameters separately, user experiences such as content delivery, streaming, and interaction can be optimized, leading to improved satisfaction and engagement.

Performance Optimization: By clustering similar types of content together, it becomes easier to implement performance optimization strategies tailored to the specific characteristics of instructional and media content, leading to faster processing times and better overall system performance.

The following shows the instructional objects based on the file size and the memory utilization and the processing times are calculated using Grid-Sim 5.0. File size and Memory are measured in terms of Kilo Bytes. Grid processing time is measured in terms of milliseconds.

- The file size of the 'definable' instruction object in storage is 188KB, and its size in memory is 186.6KB; processing time by the Grid-Sim 5.0 would be around 2300 ms (excluding user retention time).
- The file size of the 'demonstrable' instruction object in storage is 654KB, and its size in memory is 651KB; processing time by the Grid-Sim 5.0 would be around 14300 ms.
- The file size of the 'solvable' instruction object in storage is 362KB, and its size in memory is 660.7KB; processing time by the Grid-Sim 5.0 is around 8600 ms.
- The file size of the 'perceivable' instruction object in storage is 160KB, and its size in memory is 158.8KB; processing time by the Grid-Sim 5.0 would be around 1800 ms.
- The file size of the same content integrated into a single document is about 1346KB; size the same in memory is 1340KB; processing time by the Grid-Sim 5.0 would be around 28350 ms (excluding user retention time).
- With authorized research support [Kala-Devi 2013], the average computational ratio of Definable, Demonstrable, Solvable, and Perceivable has been empirically worked out to be about: 1.00, 4.00, 3.00, 0.75 (i.e. 11.5%: 46%: 34.5%: 8%), which is more or less matching with size and computational time. however, on the appliance of freelance and Poisson's possibilities for avoiding spare trust computations on the clusters of the training grid, economizing machine resources for many thousands of concurrent e-users [5].
- In a similar empirical study on media categories with authorized research support [Jagadeesan, B 2014], the average empirical computational ratio of Textual: Graphical: Animation is about 1.00, 1.10, 1.30 (i.e. 28%: 33%: 39%) respectively.

Specification	**Trials & range**
Tasks for massive users (Experiment 1).	10, 50, 100, 200 and 400
Clusters (resources) requested for creation of tasks for massive users (Experiment 2).	1000 – 10000 in steps of 100
Massive clusters that would be grouped into resources.	Varies and will be decided through first experiments
Parameters for selection of clusters (Experiment 1)	Trust, performance, redundancy removal

Table 3.1 Experimental Setup for Grid Resources

The above Table 3.1 shows the summarized specification, trials, and range for the samples and inputs for the proposed experimental setup. Tasks of massive users take 10, 50, 100, 200, and 400 experimental trials of tasks. Clusters or Resources for the creation of task for massive users takes trial ranges from 1000 to 10000. Massive clusters would be grouped into resources that can be varied in different ranges and will be decided through the first experiments. Parameters for the selection of clusters for experimental trials are trust-based performance-based and to remove the redundancy.

TRIALS	DEFINABLE (CLUSTER 4)	DEMONSTRABLE (CLUSTER 8)	SOLVABLE (CLUSTER 5)	PERCEIVABLE (CLUSTER 7)
1100	2958	15965	8840	1967
2100	2946	15120	8994	1945
3100	2918	16780	8995	1998
4100	2940	17130	9103	2004
5100	2978	16920	9254	2165
6100	2993	16192	9458	2137
7100	2940	17996	9574	2276
8100	2980	17770	9887	2397
9100	2995	18105	9998	2403
10100	2916	18975	10046	2555

Table 3.2 Grid Processing Time for Instructional Parameter

The above table 3.2 shows the grid processing time of the Instructional parameter. For Each category of instructional parameters, the experimental trials of tasks are taken from 1100 to 10100. Instructional Parameters Are Definable, Demonstrable, Solvable, and Perceivable.

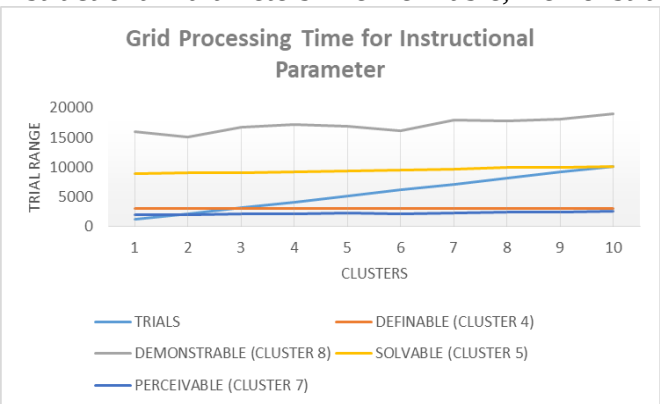


Figure 3.1 Line Graph of Grid processing time

IV. CONCLUSION AND THE FUTURE WORK

The proposed methodology is more efficient for viewing the e-content based on the load harmonization techniques on media parameters in massive clusters of learning grids. This proposed methodology makes the user or customer more comfortable in learning and developing their domain knowledge, of particular media parameters such as files can be Animation, Graphics, or Video.

The proposed methodologies are compared with the existing two algorithms such Recursive partition based K- mean algorithm and G/S/M algorithms. The First Grid process running time-based on Poisson distribution using a grid is fewer than the Recursive partition-based k-means algorithm and G/S/ M algorithms. Based on the file size of the media parameter and memory utilization of the Massive clusters, harmonization techniques are very effective compared with the existing two algorithms. Clustering acts as an intermediate process using load harmonization techniques, whereas the RPKM and G/S/M model reflects as an independent process.

Reference:

[1] H.D. Karatze [1994], "Job Scheduling in Heterogeneous Distributed Systems", Journal of Systems and Software 1994, pp: 203-212.

[2] SaeedFarzi [2009], "Efficient Job Scheduling in Grid Computing with Modified Artificial - Fish SWARM Algorithm", IJCTE, Vol-1, No.1, April-2009.

[3] Abdul Rahman A, Hailes S [2000], "Supporting Trust in Virtual Communities", Hawaii International Conference on System Sciences.

[4] Ani Brown Mary, Saravanan K [2013], "Performance Factors of Cloud Computing Data Centres Using M/G/1/GD Model Queuing Systems", IJGCA Vol. 4, No. 1.

[5] BelabbasYagoubi and Yahya-Slimani [2006], "Dynamic Load Balancing Strategy for Grid Computing", Proceedings of World Academy of Science and Engineering and Technology Volume 13 May 2006 ISSN 1307-6884.