



Data Storing in Multitenant Clouds using Split and Merge Method

Aswini. S

Department of CSE, Mailam College of Engineering, Mailam College of Engineering, Villupuram, Tamil Nadu, India

Prasanna. S

Associate Professor, CSE, Mailam College of Engineering, Mailam College of Engineering, Villupuram, Tamil Nadu, India

ABSTRACT

Resource Accounting of Shared IT Resources is a technique that distributes the excess dynamic local workload evenly across all the nodes. It is used for achieving a better service provisioning and resource utilization ratio, hence improving the overall performance of the system. Incoming tasks are coming from different location are received by the load balancer and then distributed to the data center, for the proper load distribution. Also accounting of the shared resources is very important to process the new requests. Hence in this project, we have designed and implemented the resource accounting technique, called Rameter. Rameter consists of 2 parts : 1. Keeps track of the distributed requests. 2. Keeps an account of the resource usage. Also integrated Linear Regression (LR) to monitor the accountability of the CPU usage. For experimental results, we are implementing Apache server as the accounting load balancer to keep track of the virtual machine status. Tomcat server is installed in virtual machine to obtain the job status. For data secure repository or data recovery in cloud storage we proposed multi-cloud architecture, where the user data is split into 2 parts and both files are encrypted and stored in Cloud 1 and Cloud 2. During the request our technique will merge the 2 parts and provide the response to the requested user. For Cloud storage we are invoking public clouds namely Dropbox and CloudMe.

In this project, we are developing a load balancer for effective user request monitoring and file access. In our proposed system, apache server is used as a server for processing the user request and tomcat is used as

virtual machines. The status (idle, busy), session time, packet size, virtual machine name, type, hostname, port address and bytes read in each virtual machines are been monitored in apache server and based on the status the job is allocated to the virtual machines. Thus our proposed project provides an efficient load balancer to avoid congestion and overloading of server in the data center. For dynamic load balancing, we used ANT colony algorithm.

I. INTRODUCTION

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers that may be located far from the user—ranging in distance from across a city to across the world. Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over an electricity network.

Advocates claim that cloud computing allows companies to avoid upfront infrastructure costs (e.g., purchasing servers). As well, it enables organizations

to focus on their core businesses instead of spending time and money on computer infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables Information Technology (IT) teams to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This will lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.

II. LITERATURE SURVEY

Author – Risat Mahmud Pathan and Jan Jonsson

This paper proposes a fixed-priority partitioned scheduling algorithm for periodic tasks on multiprocessors. A new technique for assigning tasks to processors is developed and the schedulability of the algorithm is analyzed for worst-case performance. We prove that, if the workload (utilization) of a given task set is less than or equal to 55.2% of the total processing capacity on m processors, then all tasks meet their deadlines. During task assignment, the total work load is regulated to the processors in such a way that a subset of the processors are guaranteed to have an individual processor load of at least 55.2%. Due to such load regulation, our algorithm can be used efficiently as an admission controller for online task scheduling. And this online algorithm is scalable with increasing number of cores in chip multiprocessors. In addition, our scheduling algorithm possesses two properties that may be important for the system designer. The first one guarantees that if task priorities are fixed before task assignment they do not change during task assignment and execution, thereby facilitating debugging during development and maintenance of the system. The second property guarantees that at most $m/2$ tasks are split, thereby keeping the run-time overhead as caused by task splitting low.

Author - Divya Jain and Sushil Chandra Jain

With phenomenal growth of distributed real time control system in high speed applications, the usage of multiprocessor system has become essential. In order to obtain performance benefits from the resources available in multiprocessor system, real time task have to coordinate and share resources to their maximum use. Load balancing amongst

processors of multiprocessing system is one of the most important issues for task scheduling. The processors working on balanced load perform better compared to unbalanced load in terms of response time and resource utilization. Heat dissipation on individual processor is also reduced. In this paper, we propose an algorithm named LBPSA (Load Balanced Partitioning and Scheduling Algorithm) that offers load balancing amongst processors. The algorithm uses an upper threshold, a lower threshold and average utilization of processors for load balancing. Evaluations show that it offers good and almost stable load mean square error under all load conditions. Further the concept is enhanced to generalize it, so that it can be applied on other partition algorithm for incorporating load balancing.

Author - Davide Compagnin and Enrico Mezzetti

The Quasi-Partitioning Scheduling algorithm optimally solves the problem of scheduling a feasible set of independent implicit-deadline sporadic tasks on a symmetric multiprocessor. It iteratively combines bin-packing solutions to determine a feasible task-to-processor allocation, splitting task loads as needed along the way so that the excess computation on one processor is assigned to a paired processor. Though different in formulation, QPS belongs in the same family of schedulers as RUN, which achieve optimality using a relaxed (partitioned) version of proportionate fairness. Unlike RUN, QPS departs from the dual schedule equivalence, thus yielding a simpler implementation with less use of global data structures. One might therefore expect that QPS should outperform RUN in the general case. Surprisingly instead, our implementation of QPS on LITMUS^{RT} invalidates this conjecture, showing that the QPS offline decisions may have an important influence on run-time performance. In this work, we present an extensive comparison between RUN and QPS, looking at both the offline and the online phases, to highlight their relative strengths and weaknesses.

Author - Saurav Kumar and Aryabartta Sahu

Classical work stealing is an efficient dynamic load-balancing technique in shared memory multiprocessor or multicore system. But the performance of the same classical work scheduling on cluster chip multicore is not appreciable. So modification to this is necessary to improve performance. In this paper, we have discussed many earlier proposed modifications, and

also proposed some simplistic modifications to suite targeted clustered environment. We have described a methodology to evaluate all the variations of work stealing analytically and experimentally on multiprocessor simulator and on real platform. Our methodology of evaluation include designing of novel parametric synthetic benchmark, which can be used to mimic behavior (or profile) of many real life benchmarks. The designed synthetic benchmark caters a wide range of application profiles to evaluate the design space of both variations of work stealing algorithms and clustered chip multiprocessor. In this work, we found that if the number of available parallelism of the targeted application is higher and data sharing between tasks is high then one of the proposed modification of work stealing (probabilistic based victim search and threshold on size of migratable task) outperform the rest of the modifications.

Author - Abdus Samad and Qasim Rafiq

The key advantage of massively parallel systems is to allow concurrent execution of workload characterized by computation units known as processes or tasks, which can be independent programs or partitioned modules of a single program. The scheduling problem is to maintain a balanced execution of all the tasks among the various available processors (nodes) in a multiprocessor system. Cube based multiprocessors have been widely used for executing parallel applications with proper utilization of interconnected nodes. In this paper a new dynamic scheduling scheme considering the topological properties of cube based networks is proposed. The algorithm is designed through the comprehensive study and analysis of various existing dynamic scheduling schemes designed for different types of multiprocessor networks. The main idea behind the design is to check the effectiveness of the proposed algorithm on various cube based multiprocessor systems by scheduling the divisible load efficiently. Two existing algorithms have also been implemented on similar systems. The comparative simulation study shows that the proposed dynamic scheme gives better performance in terms of task scheduling on cube based multiprocessor networks.

III. PROPOSED SYSTEM

We proposed dynamic algorithm for resource planning. We proposed Ant Colony Optimization Based Load Balancing Algorithm.

Our proposed resource planning load balancer involves both request monitoring and file access. Also the load balancer will keep track of the virtual machine status i.e, Busy or Ideal, session time, packet size, virtual machine name, type, hostname, port address and bytes read in each virtual machines are been monitored in apache server and based on the status the job is allocated to the virtual machine. Our proposed system will help to analyse the HEAP memory space of the server (maximum request load).

Our proposed system uses hierarchical load balancing technique. These can be modeled using tree data structure wherein every node in the tree is balanced under the supervision of its parent node. Master or manager can use light weight agent process to get statistics of slave nodes or child nodes.

For data secure repository or data recovery in cloud storage we proposed multi- cloud architecture, where the user data is split into 2 parts and both are encrypted and stored in Cloud 1 and Cloud 2. During the request our technique will merge the 2 parts and provide the response to therequested user. For Cloud storage we are invoking public clouds namely Dropbox and CloudMe.

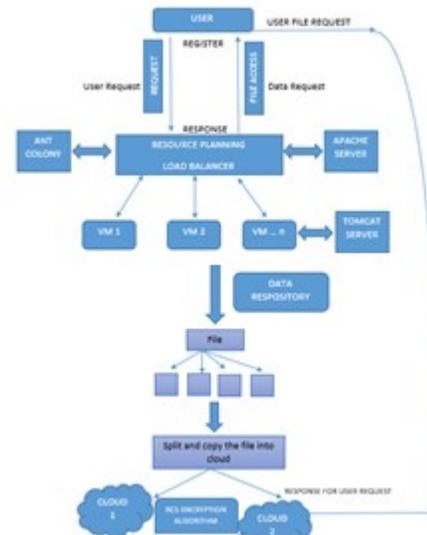


Fig 1: system architecture

USER AUTHENTICATION

In the Privacy preservation environments, a reasonable security protocol would be developed to achieve the following requirements. Identity Management may come under the privacy preserving .This module may be done in the user side. Initially while registering details in our system, we need to set

the unique ID, during registration time for uniqueness. To access the appropriate thing or to get the requested file after the file has been uploaded, Unique Id is needed.

Resource Planning in multi-tenant:

Load balancing is the process of improving the performance of the system by shifting of workload among the processors. Load balancing is one of the important factors to heighten the working performance of the cloud service provider. This module is used for verifying User complaints and their complaint are send to the server for rectifying the complaints.

Also get the request data from the brokers and data request send to the server when the load balancing have to balance the data suppose when the server is busy means it through the error message to the user as well as broker. For example, In our real world we are all getting our semester result in our Anna university Web portal. All over Tamilnadu engineering student could use the server at a time that time server will through the error message.

ANT COLONY

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. The first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants. From a broader perspective, ACO performs a model-based search and share some similarities with Estimation of Distribution.

Ant colony optimization algorithms have been applied to many combinatorial optimization problems, ranging from quadratic assignment to protein folding or routing vehicles and a lot of derived methods have been adapted to dynamic problems in real variables, stochastic problems, multi-targets and parallel implementations. It has also been used to produce near-optimal solutions to the travelling salesman problem. They have an advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically; the ant colony algorithm can be run continuously and

adapt to changes in real time. This is of interest in network routing and urban transportation systems

FILE SPLIT AND MERGE:

In proposed system, we use RC5 for encryption and then the files are split into equal fragments while uploading. We split the file in different portions then encode and store it on different cloud. Meta data necessary for decrypting and moving a file will be stored in metadata management server. File can club with another file.

The basic plan is to use many clouds at constant time to mitigate the risks of malicious knowledge manipulation, disclosure, and method meddling. This design changed targets the confidentiality of knowledge and process logic. It provides a solution to the subsequent question: however will a cloud user avoid absolutely revealing the information or process logic to the cloud provider. The information shouldn't solely be protected whereas within the persistent storage, however particularly once it's processed. The idea of this design is that the applying logic must be divided into fine-grained components and these components area unit distributed to distinct cloud. In coding technique, the user encrypts the information together with his key and uploads the cipher texts to the Cloud. Thus different fragments for a single file would be saved in 2 different clouds.

ENCRYPTION

Cryptography is dividing into two main encryption techniques, which are; asymmetric key encryption technique and symmetric key encryption technique. Within these techniques, the secret keys are used to perform encryption and decryption processes. On the other hand, there are many symmetric key algorithms that used to secure the data like; RC6, RC5, RC4 and RC2. In this project, RC5 algorithm has been used to perform the encryption process. RC5 is a fast block cipher, which is designed to be suitable for both software and hardware implementation. Also, it is a parameterized algorithm, where the block size, length of secret key and number of rounds are variable.

RC5 is a symmetric encryption algorithm developed by Ron Rivest in 1994. RC stands for "Ron's Code" or "Rivest Cipher". It is suitable for hardware and software implementation. The RC5 encryption algorithm is a w is word block cipher that converts plaintext data blocks of 16, 32 and 64 bits into the

cipher text blocks of the same length. RC5 uses a key selectable length b (0, 1, 2, ..., 255) byte.

The algorithm is organized as a set of iteration called rounds r that takes values in the range (0, 1, 2, ..., 255) as demonstrate .The operation performed on the blocks include bitwise XOR of words, data-dependent rotations by means of circular left and right rotations and Two's complement addition/subtraction of words, which is modulo addition/subtraction. RC5 is a fully parameterized family of encryption algorithm; it is more accurately specified as RC5- $w/r/b$ where the word size is w bits, encryption consists of a nonnegative number of rounds r and b denotes the length of the encryption key in bytes.

KEY SEED MECHANISM

The SEED key generation mechanism is a key generation mechanism for SEED. This is the techniques to generate and hide the key to or from the user. In this mechanism keys such as encrypt and decrypt keys are generated by the key seed mechanism and also it give the keys if the user forget or request the key from admin. The key seed mechanism mainly used for hiding the key from attacker. It will gives the keys only authorized users.

MULTIPLE CLOUD STORAGE

The user wants to store their file on cloud server in distributed manner. For this purpose, input files are subdivided into several fragments. These fragments are store on multiple servers on cloud storage, which are accomplished by single main cloud. In this module, user can upload the file on cloud. For this, file is fragmented and store on multiple servers in distributed manner. This process maintains the security and privacy of data stored on multiple servers and reduces the chances of data corruption.

IV. SYSTEM IMPLEMENTATION

In an infrastructure-as-a-service (IaaS) cloud, the cloud provider dynamically segments the physical machines, using virtualization technologies, to accommodate various virtual machine (VM) requests from its customers.

It is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded

while some others are under loaded. A load balancing algorithm which is dynamic in nature does not consider the previous state or behavior of the system, that is, it depends on the present behavior of the system.

The important things to consider while developing such algorithm are : estimation of load, comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones. This load considered can be in terms of CPU load, amount of memory used, delay or Network load.

Load balancing is the process of improving the performance of the system by shifting of workload among the processors. Workload of a machine means the total processing time it requires to execute all the tasks assigned to the machine. Load balancing is done so that every virtual machine in the cloud system does the same amount of work throughout therefore increasing the throughput and minimizing the response time. Load balancing is one of the important factors to heighten the working performance of the cloud service provider.

V. CONCLUSION AND FUTURE WORK

The major issues of file access through a server is Load Balancing. Overloading of a system may lead to poor performance which can make the technology unsuccessful, for the efficient utilization of resources, the efficient load balancing algorithm is required. Thus our project provides a complete solution for efficient load balancing along with accounting techniques named Rameter and Linear Regression. Also the proposed multiple cloud architecture provides secure data storage in cloud environment providing user data confidentiality and integrity.

Aim of the ant colony optimization to search an optimal path between the source of food and colony of ant on the basis of their behavior. This approach aims efficient distribution of work load among the node. When request is initialized the ant starts movement towards the source of food from the head node. Regional Load Balancing Node (RLBN) is chosen in Cloud Computing Service Provider (CCSP) as a head node.

Ants keep records the every node they visits ant record their data for future decision making .Ant deposits the pheromones during their movement for

other ants to select next node. The intensity of pheromones can vary on the bases of certain factors like distance of food, quality of food etc. When the job gets successful the pheromones is updated. Each ant build their own individual result set and it is later on built into a complete solution. The ant continuously updates a single result set rather than updating their own result set. By the ant pheromones trials, The solution set is continuously updated.

REFERENCES

1. Alibaba. Alibaba cloud computing [Online]. Available: [http:// www.aliyun.com/](http://www.aliyun.com/), Apr. 2015.
2. Amazon. Amazon elastic compute cloud (amazon ec2) [Online]. Available: <http://aws.amazon.com/cn/ec2/>, Apr. 2015.
3. L. Andrew, A. Wierman, and A. Tang, "Optimal speed scaling under arbitrary power functions," ACM SIGMETRICS Perform. Eval. Rev., vol. 37, no. 2, pp. 39–41, 2009.
4. A. Antoniadis and C.-C. Huang, "Non-preemptive speed scaling," J. Scheduling, vol. 16, no. 4, pp. 385–394, 2013.
5. Apache. Apache hadoop [Online]. Available: <http://hadoop.apache.org/>, Apr. 2015.
6. Maria Spinola, —An Essential Guide to Possibilities and Risks of Cloud Computing: a Pragmatic Effective and Hype Free Approach for Strategic Enterprise Decision Making. (white paper) 2009
7. Ratan Mishra and Anant Jaiswal, —Ant Colony Optimization: A solution of Load Balancing in Cloud, International Journal of Web & Semantic Technology (IJWesT), April 2012
8. Venubabu Kunamneni, "Dynamic Load Balancing for the cloud", International Journal of Computer Science and Electrical Engineering, 2012.
9. Pooja Samal, Pranati Mishra, |Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing| (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (3) , 2013, 416-419.
10. Che-Lun Hung¹, Hsiao-hsi Wang² and Yu-Chen Hu², —Efficient Load Balancing Algorithm for Cloud Computing Network. IEEE Vol. 9, pp: 70-78, 2012