ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

Fault Tolerant and Multiplexing Resource Allocation model for Self Organizing Cloud

[1]B.Sasikala, Asst.Prof /CSE,[2]M.Sasikala, Asst.Prof /CSE,
[3]I.SibiyaAsst.Prof /CSE, [4]P.Dhivya, Asst.Prof/CSE Bharathiyar
Institution of engineering for women,Salem,Tamilnadu
Sasipathi.b@gmail.com

***Abstract* –self organizing cloud can be formed by connecting large number of desktop computers on the internet by peer to peer network. Towards this new architecture where each participant will act as either provider or consumer. In multiplexing environment, computing resources to be dynamically partitioned and reassembled to meet the needs of end users. We present a fully distributed, VM-multiplexing resource allocation scheme in decentralized resources. We also design a novel multi-attribute range query protocol for providing schedule priority to the requesters.**
Contrary to existing solution which often provides mutual resource sharing between providers, our protocol produces the payoff method to improve the profit of providers. We also propose fault tolerant method and sensitivity analysis to improve the efficiency of Self Organizing cloud.

***Index terms*-Cloud computing, Fault tolerant, Schedule priority, Resource Allocation.**

I.INTRODUCTION

Cloud Computing [2],[3] has emerged as a compelling distributed paradigm with VM's resource isolation technology[4],[5],[6].Resource could be partitioned and reassembled to meet end users actual needs[7],[8].A dividable resource allocation scheme is gaining more attention in recent years. As an example, the Proportional Share Model (PSM)[9] allows resource shares be allocated proportional to the users bids and it has been leveraged in several cloud computing systems[10],[11],[12]. An efficient resource discovery protocol in Self Organizing Cloud (SOC) such that each individual host could autonomously find a qualified volunteer computer in the Internet for its task execution via multi dimensional range query. In every joined host, either a public server or desktop computer serves as individual node on a structured p2p overlay network. To perform a multi multi-dimensional range query, task's resource demand is expressed as a vector specifying its minimal requirements along each resources type (e.g., CPU, memory, network, storage) such that the task can be completed on time. The discovery process is conducted by propagating the query message hop by hop towards the peer nodes that keep the qualified resource records on different attribute dimensions. After the qualified resource node is found and determined, its split resource shares can be allocated to the task.

However, two users may simultaneously request for nodes with similar resources and with same capacities, the same candidate nodes could be returned as their query results without proper coordination. This situation may cause the task schedulers to dispatch and run their tasks on the same node, resulting in contention problem and making all of them cannot meet expected execution times. Such an issue is very challenging due to the fact that the overall performance of any virtual execution environment is closely related to its allocated resource shares along many dimensions, so that we cannot use existing single dimensional contention free models [13].In Such a framework, the participants will have high motivation to contribute their resources, because of two pillars.1)Any resource contributor will earn an amount of payment or income, which benefit them in turn based on the relation between motivation and market[14] ,2) O.Nov et al.[15] shown that many of the volunteers

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

providing about 5.5 petaFLOPS every day with more than 6,00,000 computers in BONIC project[16].

Recently, there already exist many projects being designed based on this framework. A typical example is the on-going project Cloud @Home [17].Its main aim is to design a predictive model of resource availability for internet resources and a set of strategies for checkpoint allocation using VM in low bandwidth .Another example is Community Cloud [18] ,in which each participating host is considered the resource supplier and the whole system is organized based on the principal of ecosystem. This system is claimed to provide openness of the usage (removing the dependence on vendors0, avoid the system wide cascade failure, and quite high environmental sustainability with significantly smaller carbon footprint .Z.Xu et al. implemented a Self Organizing Cloud prototype and make use of hole-punching technology to support VM alive migration on the WAN, which is transparent to cloud customers. Wuala Cloud [20] can make use of the disk space and network bandwidth from the distributed volunteer desktop computers to provide the large capacity storage. Its prominent features include guarantee of highly secure online storage using improved encrypting technologies, distributed file synchronization, real-time versioning and so on.

Based on the above cases, designing the Self Organizing Cloud system can be definitely benefit a lot:1)resource utilization can be improved with fine granularity resource allocation in VM technology;2)distributed idle resources can be used while the substrate details are still transparent to the users as if in a single point of access manner;3)it owns high robustness and reliability by minimizing the impact of one single node's failure or malicious DDoS attack to the whole system;4)it suffers much less cost on central management and maintenance than that of traditional vendor clouds.

In addition to the opportunities mentioned above, there are some problems to solve, in order to achieve the user agreed quality of services based on flexible customization. Since there are no central server to coordinate the global resource status. Costs, constructing a fault tolerant situation all users will get efficient resource allocation and can use idle resources also. Specially there are few desired features :(1) Fault tolerant: User who using cloud sometimes may suffer by failures. Mostly resource failure only will suffer the customer most. In order to provide efficient resource allocation fault tolerant method is handled.(2)Analysis Method: In this method sensitivity analysis is done since provider rate must be increased to gain more number of customers in Self Organizing cloud.(3)Profit: In previous work ,mutual resource between was exist, but we introduced the Economic model between providers to improve the profit.(4)Payoff: Payoff is introduced to scheduling priority if any of the customers' needs resource in short time period.(5)Win Win Effect:

Recently, most existing system focused social optimized feature to the resource allocation in their Grid/Cloud Platform, but no one of them guarantee this effect to the best of our knowledge. Christo Ananth et al. [19] proposed a secure hash message authentication code. A secure hash message authentication code to avoid certificate revocation list checking is proposed for vehicular ad hoc networks (VANETs). The group signature scheme is widely used in VANETs for secure communication, the existing systems based on group signature scheme provides verification delay in certificate revocation list checking. In order to overcome this delay this paper uses a Hash message authentication code (HMAC). It is used to avoid time consuming CRL checking and it also ensures the integrity of messages. The Hash message authentication code and digital signature algorithm are used to make it more secure . In this scheme the group private keys are distributed by the roadside units (RSUs) and it also manages the vehicles in a localized manner. Finally, cooperative message authentication is used among entities, in which each vehicle only needs to verify a small number of messages, thus greatly alleviating the authentication burden.For instance, B. An et al.[10] proposed an automatic negotiation approach for maximizing the social welfare in Cloud's resource allocation, yet such a mechanism rarely discuss the resource owner's payoff. Instead our solution can make sure both sides are satisfied with payoff. Different resource providers may assign the different costs on their resources, introducing a potentially competitive situation. If they are no robust pricing policy and auction method to coordinate the interest of resource providers and consumers, some participants may tend to lie on their real demands in order to maximize their selfish gains, weakening other ones' motivation. We propose double sided auction method to reveal their true expectation.

The design of Dynamic Optimal Proportional Share resource allocation method, which leverages the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

proportional share model. The idea to redistribute available resources among running tasks dynamically, such that these tasks could use up the maximum resource capacity of each resource in a node, while each task's execution time can be further minimized in a fair way.

DOPS consist of two main procedures: 1)Slice Handler: This is activated periodically to equally scale the amount if resources which allocated to the tasks, such that each running tasks, such that each running task can acquire additional resources proportional to their demand along each resource dimension.2)Event Handler : This is responsible for resource redistribution upon the events of task arrival and completion.

Suppose that there are M tasks running on particular node $p(s)$, denoted as $t\ 1\ ,t\ 2\ \ldots\ldots,t\ M$ based on their arrival order. Accordingly w(i), B(i), and r(i) denote t(i)'s preferential weight vector, budget and resource share vector, respectively.

Assume that when t(i) is to be scheduled on $p(s)$,the available resource vector is denoted as a(i),then which can be calculates as component wise C(p(s)).Resource node $p(s)$ found by discovery protocol is "qualified" ,i.e ., for $t\ j\ = i, 2\ldots. M$, $b\text{^}T(p(s)).e(i)<=B(i)$.Based on the above analysis ,it is possible that the resource along certain dimensions may not be fully used. It improves the resource utilization by redistributing the remaining resource at the k th dimension. By leveraging PSM, each running task can acquire its resource proportional to their computed optimal shares along every dimension. Since task will be executed faster with the augmented resource, while the payment will still be calculated based on optimal share. This means the user will not be charged more even with any extra resources allocation.Another issue is how to determine the optimal resource allocation upon arrival and completion. When a new task is scheduled in$p(s)$,it will get the optimal shares of resources based on the availability of resources. Then Slice Handler will be activated based on DOPS design. When the task is finished, it is possible for other running task to share the newly released resources.

B.PONITER-GOSSIPING CAN

Resource allocation approach relies on the assumption that all qualified node must satisfy the constraints.

$$b\ p\ T.e(tij) \leq B(tij), \quad (1)$$

$$e\ tij\ \leq a\ ps\ . \quad (2)$$

To meet this requirement, there was a design of resource discovery protocol, namely Poniter-Gossiping CAN, to find this qualified nodes.CAN [17] is taken as DHT overlay to adapt to the dimensional feature.

Like traditional CAN, each node under PG-CAN is responsible for a unique multidimensional range zone randomly selected when it joins the overlay. Figure 1 illustrate the CAN overlay network.

Figure 1.CAN Topology

Suppose there are 25 joined nodes, each taking charge of a single zone. If a new node (26) joins, a random point will be generated and its zone will be set as the new zone evenly split along dimension from the existing zone (node 25)that contains this point. If there is only one non overlapped range dimension between two nodes (e.g .,i and j) and they are adjacent at this dimension, call them as neighbors to each other. Furthermore, if the non overlapped range of I is always no less than j's, i is called j's positive neighbor and j is i's negative neighbor. Node 9,12 and 20 are positive neighbor of node 1.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

All nodes will periodically propagate the state update message about its available resource vector to the Duty node whose zone encloses this vector. After a task generates the query, the query message will be routed to duty node containing the expected vector. Then justify that the state message of all qualified nodes must be kept in those onwards nodes.

Obviously, the searching area may still be too large for the complete resource query without flooding .However, according to observation, this will significantly reduce the likelihood of finding qualified resources, finally degrading the system throughput and user's QoS. Alternatively, improved the mechanism by periodically diffusing a few pointer messages for any du nodes owning the state update message to the distinct nodes towards negative directions, so that these duty nodes could be more easily found. For instance, Node4's negative pointer nodes along CPU dimension are Nodes 14,3 and 23.By periodically sending the pointer-recovery messages, each with empty payload outward, each node can easily maintain the connection to the negative pointer node. Each query routed to the duty node will check its stored record and the pointed duty nodes. If this finds qualified resource records on the current or other pointed duty nodes, it will return that information to the requesting node. Otherwise it will continue the searching next neighbor duty nodes.

Every duty node D1 will catch state-update messages received from its neighbors, which are checked periodically and removed if updated. It propagates its identifier to a few randomly selected pointer nodes toward its negative direction. For those duty nodes containing valid state message, can call then as nonempty-cache nodes.

Basically there are two manners to propagate the duty nodes' identifiers backward-spreading manner and hopping manner. From hopping manner can easily convert to spreading manner.

Consequently upon receiving the query result, the requesting node will randomly choose one out of them as the final resource node for executing the submitted task. Now easily mitigate the decision conflict among different task.

Disadvantages:

- This system is not focused on satisfactory of both sides with payoff.
- Sensitivity analysis is not performed
- The system supports mutual resource sharing scheme only.
- Resource failures are not handled

III.PROPOSED SYSTEM

In Self Organizing Cloud will be formed by connecting number of volunteer computers. Users of cloud can request for resource. Figure 2 shows the flow of the SOC process. If user/customer wants to request for resource they should generate the query first. Then query will be sent in the network in which all the status record is diffused. According to the status record message also will be diffused. After finding the qualified node set of results is send to customers to choose particular resource.

In this work, *payoff* should be considered. When resource is shared between two providers profit of providers can be generated as follows

Profit = Rending cost - Providing cost

User who using cloud sometimes may suffer by failures. Mostly resource failure will suffer the customer most. In order to provide efficient resource allocation *fault tolerant* method is handled. After proving the resource to the user should scale whether resource is under good condition. If condition goes below threshold value it must be taken into consideration. In our method *sensitivity analysis* is done since provider rate must be increased to gain more number of customers in Self Organizing cloud Scheduling priority is introduced for payoff if any of the customers' needs resource in short time period. This effect is used for both the sides (Providers and consumers).*Resource constraints* could be of two types: (1) On-Demand resource request, (2) Reservation Plan in which resource is reserved in advance and can acquire it when it is need.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
nternational Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
/ol. 3, Special Issue 2, March 2016

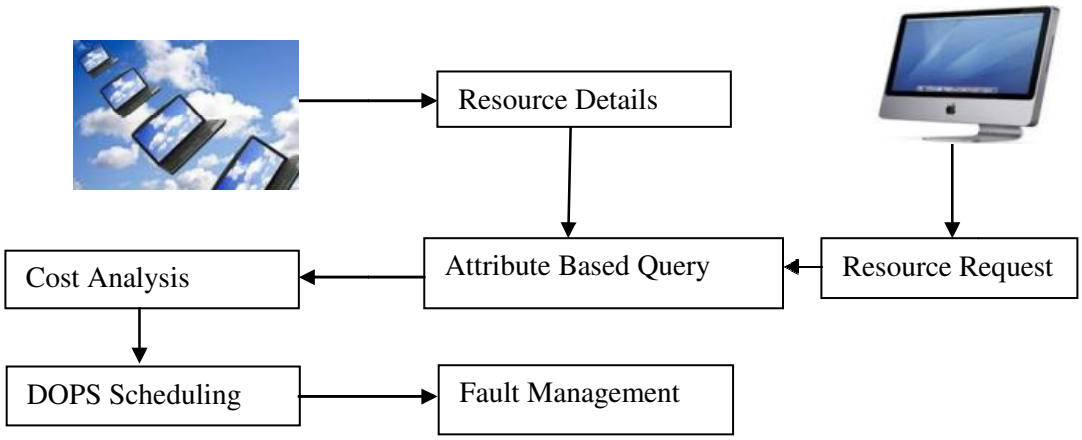

Figure 2 Process of Resource Allocation

IV.SIMULATION RESULT

To conduct the simulation, we first build a proportional share scheduler .We used cloudsim to implement the proposed CAN-based range query protocols on large network containing 2000 nodes. The configuration is selected randomly from the Table 1.

Parameter	Value
# of nodes	2000~12000
# of processor per node	1,2,4,8
Computation rate per processor	1,2,2.4,3.2 Hz
I/O speed per node	20,40,60,80 Mbps
Memory size per node	512,1024,2048 MB
Disk size per node	20,60,120,240 Gb
LAN network bandwidth	5~10 Mbps
WAN network bandwidth	0.2~2 Mbps

Table 1.System Settings

The throughput radio between SOC and P2P is defined as per the radio of the number of finished tasks and total number of generated task in the whole system over time. We observed that the SOC would achieve around 60 percent improvement as the task size are relatively small on average (λ=1/6 or 1/8).When all the task sizes are relatively large, SOC could still get about 15 to 20 percent improvement. We evaluate the system scalability of the PG-CAN protocol. All the values in Table 2 are recorded after the first duration test. With the increasing system scale, the performance metric (throughput, efficiency) will not change notably.

Metric	1000	1500	2000
Throughput	0.598	0.592	0.572
Failed radio	23%	26.4%	30%
Average Eff.	2.72	2.89	3.0
Fairness index	0.665	0.672	0.701
Msg delivery	2347	2456	2789

Table 2.System Scalability

V.CONCLUSION

This paper propose a novel fault tolerant which could guarantee that resource customers could use resources for their task efficiently and without any trouble and resource constraints mechanism which ensures that consumer are satisfied with their task execution and the resource contributors are also content with their profit for their resource provisioning. Moreover, by extending the price bidding policy, our algorithm can include both strategic resource consumers and contributor to truthfully reveal their demands. Finally we confirm our efficiency of our design via simulation. In the future, we may introduce more algorithms in order to improve the effectiveness of system.

REFERENCES

[1]. Sheng Di and Cho-Li Wang (2013), 'Dynamic Optimization of Multiattribute Resource Allocation in Self-Organizing Clouds' IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 3, pp. 2101-2110.

[2]. Varuero L.M., Rodero-Merino L.,Caceres J.and Lindner M.(2009), ' A break in the clouds: towards a cloud definition' SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50–55.

[3]. Amazon elastic compute cloud: http://aws.amazon.com/ec2/.

[4]. Gupta D., Cherkasova L., Gardner R. and Vahdat A.(2006), 'Enforcing performance isolation across virtual machines in xen' in Middleware, pp. 342–362.

[5]. Cherkasova L., Gupta D. and Vahdat A. (2007), 'Comparison of the three cpu schedulers in xen' SIGMETRICS Perform. Eval.Rev., vol. 35, no. 2, pp. 42–51.

[6]. Walters J. P., Chaudhary V., Cha M., Jr S. G. and Gallo S. 'A comparison of virtualization technologies for hpc' In AINA'08: 25th International Conference on Advanced Information Networking and Applications, pp. 861–868, 2008.

[7]. Cloud desktop: http://www.gladinet.com/.

[8]. icloud project: http://www.icloud.com/en.

[9]. Feldman M., Lai K. and Zhang L. (2009), 'The proportional share allocation market for computational resources' IEEE Transactions on Parallel and Distributed Systems, vol. 20,pp. 1075–1088.

[10]. Grit L. E. and Chase J. S.(2008), 'Weighted fair sharing for dynamic virtual clusters' SIGMETRICS Perform. Eval. Rev., vol. 36,pp. 461–462.

[11]. Raghavan B., Vishwanath K., Ramabhadran S., Yocum K. and Snoeren A. C.(2007), 'Cloud control with distributed rate limiting' SIGCOMM Comput. Commun. Rev., vol. 37, pp.337–348.

[12]. Barker S. K. and Shenoy P.(2010) 'Empirical evaluation of latency sensitive application performance in the cloud' in Proceedings of the first annual ACM SIGMM conference on Multimedia systems, ser. MMSys '10. New York, NY, USA: ACM, pp. 35–46.

[13]. Di S. and Wang C.-L. (2010), 'Conflict-minimizing dynamic load balancing for p2p desktop grid' in Grid'10: The 11th IEEE/ACM International Conference on Grid Computing, pp. 137–144.

[14]. Anderson D. P.(2004) 'Boinc: a system for public-resource computing and storage' in Proceedings of the 5th IEEE/ACM

[15]. International Workshop on Grid Computing, pp. 4–10.

[16]. MacLeod W. B. and Malcomson J.(1997), 'Motivation and markets' Boston College Department of Economics, Boston College Working Papers in Economics 339.

[17]. Nov O., Anderson D. and Arazy O. (2010) 'Volunteer computing: a model of the factors determining contribution to community-based scientific research' in WWW, pp. 741–750.

[18]. Cloud@home: http://clouds.gforge.inria.fr/pmwiki.php.

[19]. Christo Ananth, M.Danya Priyadharshini, "A Secure Hash Message Authentication Code to avoid Certificate Revocation list Checking in Vehicular Adhoc networks", International Journal of Applied Engineering Research (IJAER), Volume 10, Special Issue 2, 2015,(1250-1254)

[20]. Xu Z., Di S., Cheng L. and Wang C.-L.(2011) 'Wavnet: Wide-area network virtualization for elastic cloud computing' in 40th International Conference on Parallel Processing (IEEE ICPP2011), pp. 285–294.

[21]. Wuala: http://www.wuala.com/.

[22]. Ratnasamy S., Francis P., Handley M., Karp R. and Shenker S.(2001), 'A Scalable Content-Addressable Network," Proc. ACM SIGCOMM '01, pp. 161-172.