ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

An Efficient Resource Allocation for Improving Resource Utilization in Self Organizing Clouds

S. Saranya[1], N. Saranya[2]

PG Scholar, Department of CSE, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamilnadu, India[1]

Assistant professor, Department of CSE, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamilnadu, India[2]

ABSTRACT- By leveraging virtualization which is a key technology in cloud computing, cloud resource can be provisioned on demand in a fine grained multiplexed manner while enabling fault isolation and scalability. This paper proposes a new resources allocation scheme to utilize the unused computing powers that is efficient enough to allocate resource. Envisioning self-organizing cloud (SOC) by integrating the volunteer computing into cloud architecture in order to utilize the huge computing power. Each participant may autonomously act as both resource consumer and provider. Different from previous works, a novel multi attribute range query protocol called Pointer Gossiping Protocol (PG-CAN) has been designed to locate the qualified nodes under a randomized policy that mitigates the contention among requesters. Our scheme also produces a lightweight query message per task on the Content Addressable Network (CAN) and maximizing resource utilization using Proportional Share Model (PSM). In addition SOC with optimized algorithm make an improvement by 15-60 percent in system throughput.

KEY WORDS - virtualization, CAN, resource contention, multiattribute range query, volunteers computing.

I. INTRODUCTION

Cloud computing is an emerging technology which uses internet to work. Data and applications are maintained using Central remote servers in cloud. Applications can be used in Cloud computing without any installation of software. The users can access the Internet and send messages anywhere in the world. Centralized storage, memory, processing and bandwidth are more efficient computing that are allowed in Cloud computing. The best example is Google mail. Cloud helps consumers to outsource computation, storage and other tasks to third party cloud providers and pay only for the resources used. Today's cloud architectures are not without problems. Most cloud services built on top of a centralized architecture may suffer denial-of-service (DoS) attacks [3], unexpected outages, and limited pooling of computational resources. On the contrary, volunteer computing systems (or Desktop Grids) can easily aggregate huge potential computing power to tackle grand challenge science problems [4]. In view of this, we propose novel cloud architecture, namely self-organizing cloud (SOC), which can connect a large number of desktop computers on the Internet by a P2P network. They operate autonomously for locating nodes with more abundant resource or unique services in the network to offload some of their tasks; meanwhile they could construct multiple VM instances for executing tasks submitted from others whenever they have idle resources. We focus on two key issues in the design of SOC: 1) the multiattribute range query problem in a fully decentralized environment for locating a qualified node to satisfy a user task's resource demand with bounded delay and 2) how to optimize a task's execution time by determining the optimal shares of the multiattribute resources to allocate .We further propose a dynamic optimal proportional-share (DOPS) resource allocation algorithm, by incorporating the proportional-share model (PSM) [12]. The key idea is to dynamically scale the amount of resources at each dimension among running tasks proportional to their demand, such that these tasks could use up the maximum capacity of each resource type at a node.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

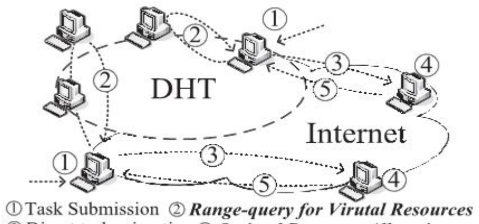
Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

The rest of the paper is organized as follows: We formulate the resource allocation problem in a VM-multiplexing environment in Section 2. In Section 3 details the proposed range query protocol along with our DOPS resource allocation scheme. In section 4 and 5 proposed work is enumerated and concluded with the work of this paper.

II. PROBLEM DESCRIPTION

In this work, we only focus on the multiattribute range query problem (Step 2) and the resource allocation problem for determining the amount of resources of a qualified node to the submitted task (Step 4). Suppose there are n nodes in SOC, each is denoted as pi , where 1<i<n. Each node owns R different resources managed by a Virtual Machine Monitor (VMM). We denote ∏ to be the set of resource attributes owned by node pi and c(pi)=c1(pi),c2(pi),...,c_R(pi))T as Pi's capacity vector .Let mi denote the total number of tasks submitted to pi . A task submitted to node pi is denoted as tij, where 1<j<mi. Each task is associated with an expected resource vector e(tij)=(e1(tij),e(tij),.....,e_R(tij))^T.



① Task Submission ② *Range-query for Virutal Resources*
③ Direct task migration ④ *Optimal Resource Allocation*
⑤ Task-finish notification and result transmission

Fig 1. Task Execution Procedure

The user-specified expected resource vector is a rough estimation of the needed amount of resources with respect to the R resource attributes for a submitted task to be completed within a tolerable execution time . Each task has a load vector indicating the amount of the R attributes for completing the task. For simplicity, we assume the execution of a task cannot be done concurrently among different resources at the same node.

Preferential vector w(tij) indicating the relative importance of a resource that might affect the execution time of a task according to the property. In essence, w(tij)acts as a more relaxed requirement in using our model as we assume the user does not know the exact load vector, but only needs to specify the preferential weight vector. we follow a simple monetary model to analyze the economic implications between consumers and providers. For any node pi, its resource price vector is denoted as b(pi)=(b1(pi),b2(pi);...;b_R(pi))^T, which is designated by the resource provider. Let bk (pi) 1<k<R represent the per-time-unit price for using the kth resource attribute on pi.

III. POINTER GOSSIPING CAN PROTOCOL

The PG-CAN range query protocol aims to find the qualified resources with minimized contention among requesters based on task's demand. It is unique in that for each task, there is only one query message propagated in the network during the entire course of discovery. This is different from most existing multi-attribute range query solutions that require propagating multiple sub queries along multiple dimensions in parallel. To mitigate the contention problem due to analogous queries in CAN, our range query protocol proactively diffuses resource indexes over the network and randomly route query messages among nodes to locate qualified ones that satisfy tasks' minimal demands. To locate qualified nodes in the SOC environment, we design a fully decentralized range query protocol, namely pointer-gossiping CAN(PG-CAN), tailored for DOPS. Every joining node registers its static resource attributes (e.g., CPU architecture, OS version) or maximum capacity on the CAN/Chord overlay, so that other users could find the most matched node within a logarithmic (or sublinear) number of routing steps. Such a design is feasible for a P2P desktop Grid because the resources of a selected node can only be used exclusively by a single task.

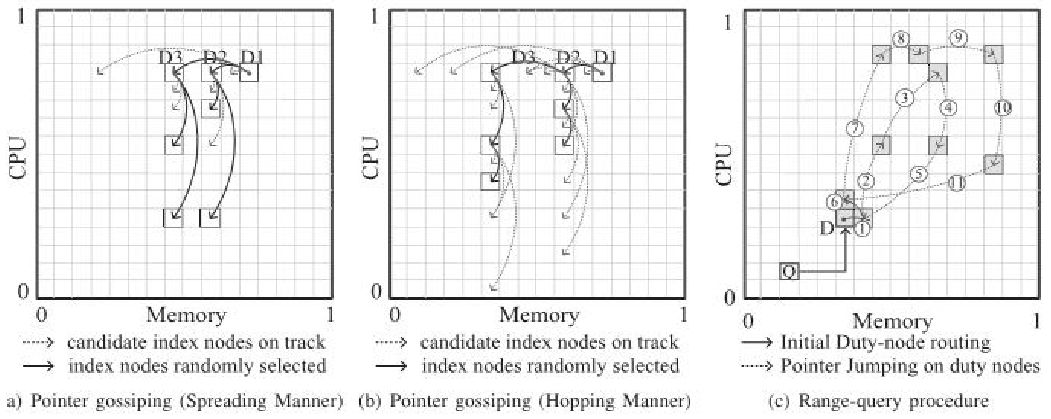

CPU
Memory
D3 D2 D1
D
Q
0
1
candidate index nodes on track
index nodes randomly selected
Initial Duty-node routing
Pointer Jumping on duty nodes
a) Pointer gossiping (Spreading Manner)
(b) Pointer gossiping (Hopping Manner)
(c) Range-query procedure

Fig 2. Resource Matching

However, due to dynamic resource provisioning technologies used in cloud, the frequent resource repartitioning and reallocation (e.g., upon task arrival or completion) make it a challenging problem to locate a node containing a combination of available resources along all the R resource attributes that would satisfy the requirements of a submitted task.

Also, we discuss the design of our dynamic optimal proportional-share resource allocation method, which leverages the proportional share model. The key idea to redistribute available resources among running tasks dynamically, such that these tasks could use up the maximum capacity of each resource in a node ,while each task's execution time can be further minimized in a fair way. DOPS consists of two main procedures: 1)Slice handler: It is activated periodically to equally scale the amount of resources allocated to tasks, such that each running task can acquire additional resources proportional to their demand along each resource dimension. 2)Event handler: It is responsible for resource redistribution upon the events of task arrival and completion.

IV. PROPOSED ALGORITHM

In the proposed work we describe the multi-environment routing of query message to find the required resources to execute the job submitted by the end user. Migrating the cloud resources among various environments provides the availability of multiattribute resource at anytime to the any number of users. PG CAN protocol is used to identified the duty nodes among different environment where scheduling of resource are performed automatically at the duty nodes based on the query message routed to the duty nodes. The loads are distributed eventually at the respective nodes. At each duty nodes matching the resource on the arrival of queries are performed first before allocating the resource for the task execution. Various range query procedure are executed for matching the resource. The situation arises where the requested resource are not able to find in the available nodes. During this situation the queries are routed to the other environment to acquire the resource. This concept overcomes the fault tolerance as it finds multiattribute resources in any environment which preferably eliminates the resource contention problem, reduces the processing time of the job. During the process the queries are routed to the various environment and resources are selected based on the location of duty nodes from the node where the resources are required for processing and the resource are shared among other user when they are idle .This scheme enhances the resource allocation and improves the end to end throughput.

It achieves higher throughput than other routing approaches and effectively utilizes the idle cloud resources. It also provides better resilience from the dynamic interruption of primary users. And exhibits fairly high adaptability in a dynamic node-churning environment.

IV. CONCLUSION

In this paper, execution time of the submitted task and resource utilization is highly optimized by making sure that every computing resource is distributed efficiently and fairly. A solution is formulated which can optimize the task execution performance based on its assigned resources under the user budget. Resources contention problem is also minimized. Our algorithm achieves both optimized resource utilization and allocates the resources on multiattribute constraints.

REFERECNES

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A.Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M.Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB/EECS-2009-28, Feb. 2009.
[2] D.P. Anderson, "Boinc: A System for Public-Resource Computing and Storage,"Proc. IEEE/ACM Fifth Int'l Workshop Grid Computing,pp. 4-10, 2004.
[3] P. Crescenzi and V. Kann,A Compendium of NP Optimization Problems. ft p://ftp.nada. kth.se/Theory/Viggo-Kann/ compendium.pdf, 2012.
[4] O. Sinnen,Task Scheduling for Parallel Systems,Wiley Series on Parallel and Distributed Computing. Wiley-Interscience, 2007.
[5] O.H. Ibarra and C.E. Kim, "Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors," J. ACM, vol. 24,pp. 280-289, Apr. 1977.
[6] X. Meng et al., "Efficient Resource Provisioning in Compute Clouds via vm Multiplexing,"Proc. IEEE Seventh Int'l Conf.Autonomic Computing (ICAC '10),pp. 11-20, 2010.
[7] J. Sonneck and A. Chandra, "Virtual Putty: Reshaping the Physical Footprint of Virtual machines,"Proc. Int'l HotCloud Workshop in Conjunction with USENIX Ann. Technical Conf.,2009.
[8] M. Feldman, K. Lai, and L. Zhang, "The Proportional-Share Allocation Market for Computational Resources,"IEEE Trans.Parallel and Distributed Systems,vol. 20, no. 8, pp. 1075-1088, Aug.2009.
[9] S. Soltesz, H. Poetzl, M.E. Fiuczynski, A. Bavier, and L. Peterson,"Container-Based Operating System Virtualization: A Scalable,High-Performance Alternative to Hypervisors,"Proc. Second ACM Int'l European Conf. Computer Systems (Euro '07),pp. 275-287. 2007,
[10] L. Cherkasova, D. Gupta, and A. Vahdat, "Comparison of the Three cpu Schedulers in Xen,"SIGMETRICS Performance Evalua-tion Rev.,vol. 35, no. 2, pp. 42-51, 2007.
[11] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network,"Proc. ACM SIGCOMM '01, pp. 161-172, 2001.
[12] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrish-nan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications,"Proc. ACM SIGCOMM '01,pp. 149-160, 2001.
[13] J.S. Kim et al., "Using Content-Addressable Networks for Load Balancing in Desktop Grids,"Proc. 16th ACM Int'l Symp.High Performance Distributed Computing (HPDC '07),pp. 189-198, 2007.
[14] A. Leite, H. Mendes, L. Weigang, A. de Melo, and A. Boukerche,"An Architecture for P2P Bag-of-Tasks Execution with Multiple Task Allocation Policies in Desktop Grids,"Proc. IEEE Int'l Conf. Cluster Computing,pp. 1-11, Feb. 2011.
[15] Y. Drougas and V. Kalogeraki, "A Fair Resource Allocation Algorithm for Peer-to-Peer Overlays,"Proc. IEEE INFOCOM '05,pp. 2853-2858, 2005.
[16] S. Di, C.-L. Wang, W. Zhang, and L. Cheng, "Probabilistic Best-Fit Multi-Dimensional Range Query in Self-Organizing Cloud,"Proc. IEEE 40th Int'l Conf. Parallel Processing,pp. 763-772, 201