ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

OPTIMIZE SELF-ORGANIZING CLOUD USING MULTI-ATTRIBUTE RESOURCE ALLOCATION METHOD

Ramesh
[*] Department of Information Science and Engineering, B M S College of Engineering,Bangalore, India

ABSTRACT

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 resource attributes that would satisfy the requirements of a submitted task. The System propose cloud architecture, namely self-organizing cloud (SOC), which can connect a large number of desktop computers on the Internet by a P2P network. In SOC, each participating computer acts as both a resource provider and a resource consumer. SOC 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. The System propose an algorithm to optimize the task execution time on a qualified resource node, given its pre-set budget and tolerable quality of service (QoS).

KEYWORDS: Cloud computing, VM- multiplexing resource allocation, Optimization, P2P Multialtribute range query.

INTRODUCTION

Resource Management is an important issue in cloud environment. Cloud computing is the delivery of computing and storage capacity as a service to a community of end-recipients. The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infra

structure it contains in system diagrams. Cloud computing entrusts services with a user's data, software and computation over a network. There are three types of cloud computing:

- Infrastructure as a service (IaaS),
- Platform as a service (PaaS), and
- Software as a service (SaaS).

Using software as a service, users also rent application software and databases. The cloud providers manage the infrastructure and platforms on which the applications run. End users access cloud-based applications through a The System b browser or a weight. The System weight desktop or mobile app while the business software and user's data are stored on servers at a remote location. Proponents claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand

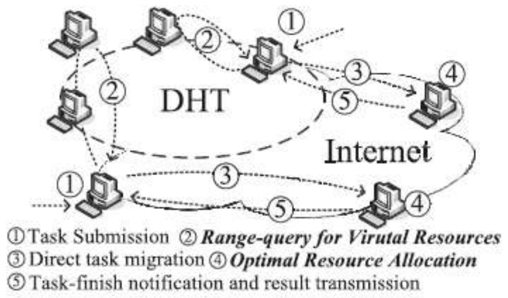
DHT
Internet
①
②
③
④
⑤
① Task Submission ② Range-query for Virutal Resources
③ Direct task migration ④ Optimal Resource Allocation
⑤ Task-finish notification and result transmission

Figure 1: The Entire Task Execution

In this paper discuss the dynamic optimization of self-formed cloud environment resources are denoted by the query based formation. To optimize the resource formation by its The System weight and budget. It provides the execution for resource evaluation by RAM and CPU utilization. The System can prove each rational participant gets its optimal payoff if their resource demands and expected prices are truthfully declared.

The problem of efficiently locate a qualified node in a large scale peer-to-peer network[2] for executing the submitted task with controllable message delivery overhead. Absence of convex optimization analysis, and then derive the optimal algorithm for the case with the constraint. The resource optimization dissent the evaluation pattern of the scheme

RELATED WORK

P2P desktop[2] Grids[3] [7] [8] are a decentralized distributed communication system with desktop application connectivity. Grids are a form of distributed computing whereby a "super virtual computer" is composed of many networked loosely coupled computers acting together to perform very large tasks. This technology has been applied to computationally intensive scientific, mathematical, and academic problems through volunteer computing.

Existing P2P desktop[2] Grids[3] [7] [8] favor CAN-based[1] or Chord-based resource discovery protocols. 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[2] Grid[3] [7] [8] because the resources of a selected node can only be used exclusively by a single task. Existing multiattribute range query solutions require to propagate multiple subqueries along multiple dimensions in parallel[4].

- Existing solutions which often generate bulky messages per request.
- Takes long time to process the query.
- Design is feasible only for a P2P desktop[2] Grid[3] [7] [8] for a single task.
- It is complex to apply the existing technique in dynamic resource provisioning.

PROPOSED SYSTEM

In proposed system, The System focus on the multi-attribute range query[10] problem and the resource allocation problem[6] for determining the amount of resources of a qualified node to the submitted task. The System propose a fully distributed, VM-multiplexing resource allocation scheme to manage decentralized resources[6]. Our approach not only achieves maximized resource utilization using the proportional share model, but also delivers provably and adaptively optimal execution efficiency[5].

The System also design a multi-attribute range query[4] protocol for locating qualified nodes The System achieve Optimization of task's resource allocation[6] under user's budget, Maximized resource utilization based on PSM[5] and Weight The System weight resource query protocol with low contention.

- Integrating volunteer computing into cloud architectures to envision a gigantic self-organizing cloud (SOC) to handle huge volume of task demand with dynamic resource provisioning.
- It is a fully distributed, VM-Multiplexing resource allocation scheme to manage decentralized resources.
- Achieves maximized resource utilization using the proportional share model (PSM)[5].
- This approach has optimal execution efficiency[5].

A novel multi-attribute range query protocol[10] produces only one weightThe System weight query message per task on the Content Addressable Network (CAN) for locating qualified nodes.

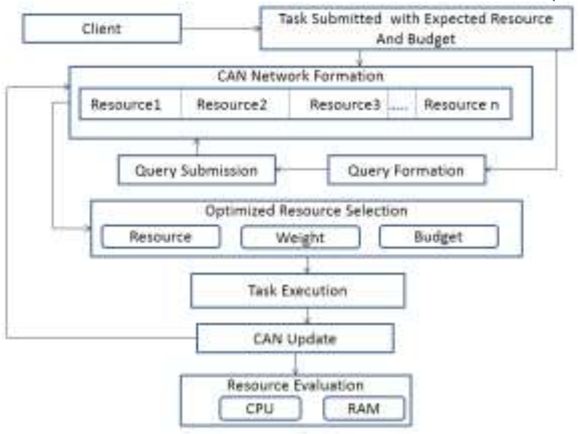

Figure 2. Architecture and Data Flow Diagram

Resource Capacity and Dimension

In self-organizing cloud (SOC), a resource can either acts as a service provider or service consumer. Resource capacity is computed for all of the available resources using the different dimensions (attributes) of the resources. Using the dimension values of the resources, resource price vector is computed. Resource capacity and resource price vector helps to allocate the optimized resource to execute task based on the user demand.

Task Submission with Budget

Service consumer in the self-organizing cloud (SOC) submits task for execution with expected resources, time and budget. Consumer loads the task with expected resource vector, The System weight vector and budget. Expected resource vector includes the list of resources that will be used by the submitted task and its expected amount of resource consumption. Expected The System weight vector is computed based on the expected resource consumption and task length. Budget is computed based on the price vector of each resource that is used by the task with its expected resource consumption.

CAN (Content Addressable Network)

CAN is designed to be scalable, fault tolerant, and self-organizing. A CAN[1] node maintains a routing table that holds the IP address and virtual coordinate zone of each of its neighbors. A node routes a message towards a destination point in the coordinate space. The node first determines which neighboring zone is closest to the destination point, and then looks up that zone's node's IP address via the routing table. Based on the available resources, its computed resource dimensions are used to form the CAN. Based on the task submission and execution the CAN[1] is updated by updating the available free dimensions of the resources.

Resource Optimization and Selection

Task submitted by consumer is received with its expected resources and budget. The system actual resource consumption and price vector are computed. For the given task with its budget and resources, an optimized resource is computed by computing the actual resource dimension and price and comparing it with its expected resource and budget. Finally, a resource which can execute the given task within the user’s demand and budget is selected for task execution.

Task Execution and CAN Update

The given task is executed in the selected optimized resource with minimized time, cost and resource consumption which provides user’s satisfaction as the given task is executed within the expected demand. After task execution, the resource dimension values are updated based on the resource usage. Once computing the resource dimensions, the position of the resource in the CAN[1] will be changed based on the available resource dimension or it will be removed from the CAN if the resource cannot be used anymore because of its load.

Evaluation

Performance Evaluation functions as a leading journal in the area of modeling, measurement, and evaluation of performance[9] aspects of computing and communication systems. As such, it aims to present a balanced. Finally, the self-organizing cloud (SOC) approach achieves maximum utilization of resources with minimum time consumption, resource consumption and cost reduction and also achieves optimized resource provision with Weight. The System Weight resource query protocol with low contention that effectively searches qualified resources.

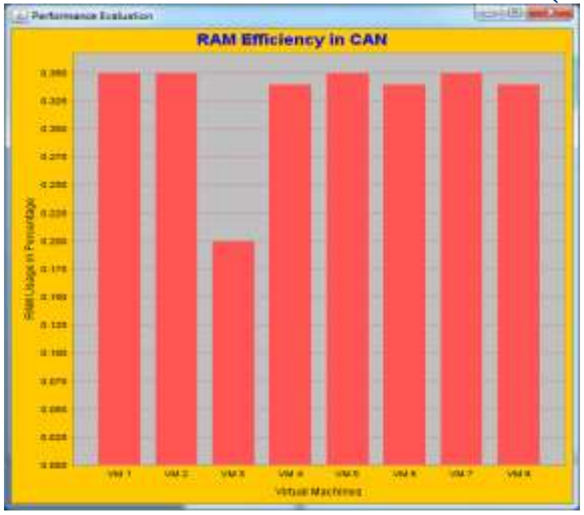

Figure. 3. RAM Efficiency for Different Users

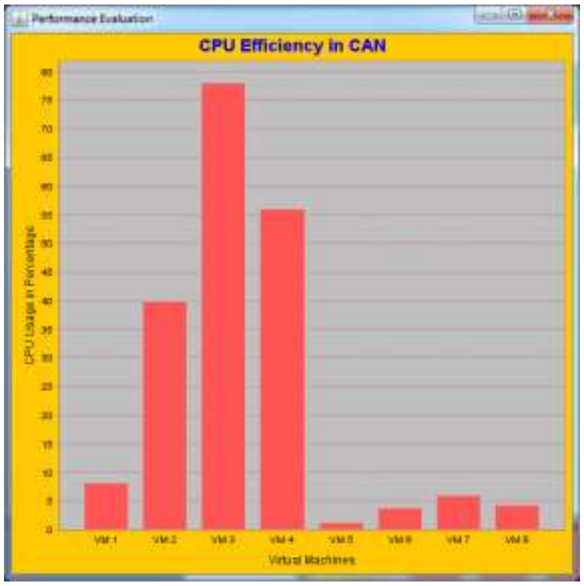

Figure. 3. CPU Efficiency for Different Users

CONCLUSION

This paper proposes a scheme dynamic optimal proportional-share (DOPS) for virtual resource allocation on a SOC, with three key contributions listed below.

Optimization of task's resource allocation under user's budget

- A solution which can optimize the task execution performance based[9] on its assigned resources under the user budget is proposed.
- Maximized resource utilization based on PSM this helps to utilize idle resources more efficiently. WeightThe System weight resource query protocol with low contention
- The resource searching request is processed as range query with sufficient and necessary conditions for getting the optimal resource allocation

ACKNOWLEDGEMENTS

The paper is successfully Done and supported by Department of ISE,BMSCE Bangalore, Special thanks to Professor's and My Friends.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

REFERENCES

[1] J.S. Kim et al., “Using Content-Addressable Networks forLoad Balancing in Desktop Grids,” Proc. 16th ACM Int’l Symp. High Performance Distributed Computing (HPDC ’07), pp. 189
[2] J. Cao, F.B. Liu, and C.Z. Xu, “P2pgrid: Integrating P2P Networks Into the Grid Environment: Research Articles,” vol. 19, no. 7, pp. 1023-1046, 2007.
[3] A.R. Bharambe, M. Agrawal, and S. Seshan, “Mercury: SupportingScalable Multi-Attribute Range Queries,” Proc. ACM SIGCOMM ’04, pp. 353-366, 2004.
[4] 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.
[5] Y. Drougas and V. Kalogeraki, “A Fair Resource Allocation Algorithm for Peer-to-Peer Overlays,” Proc. IEEE INFOCOM ’05, pp. 2853-2858, 2005.
[6] H. Abbes, C. Cerin, and M. Jemni, Bonjourgrid: Orchestration of Multi-Instances of Grid iddlewares on Institutional Desktop Grids,” Proc. IEEE 23rd Int’l Symp. Parallel & Distributed Processing (IPDPS ’09), pp. 1-8, 2009.
[7] G. Singh, C. Kesselman, and E. Deelman, “A Provisioning Model and its Comparison with Best-Effort for Performance-CostOptimization in Grids,” Proc.H 16th ACM Symp. High Performance Distributed Computing (HPDC ’07), 117-126, 2007.
[8] Q. Zheng, H. Yang, and Y. Sun, “How to Avoid Herd: A Novel Stochastic Algorithm in Grid Scheduling,” Proc. 15th ACM Int’l Symp. High Performance Distributed Computing (HPDC ’06), pp. 267-278, 2006.
[9] C.B. Lee and A.E. Snavely, “Precise and Realistic Utility Functionsfor User-Centric Performance Analysis of Schedulers,” Proc. 16th ACM Int’l Symp. High Performance Distributed Computing (HPDC’07), pp. 107-116, 2007.
[10] A.R. Bharambe, M. Agrawal, and S. Seshan, “Mercury: SupportingScalable Multi-Attribute Range Queries,” Proc. ACM SIGCOMM ’04, pp. 353-366, 2004. [40] D. Li, J. Cao, X. Lu, and