OPTIMIZATION OF RESOURCE PROVISIONING COST IN CLOUD COMPUTING

[1]S.Ezhildevi, [2] H.Shabuddeen

[1]PG Scholar, Department of Information Technology, Jayam College of Engineering and Technology, Dharmapuri.

[2]Assistant. Professor, Department of Information Technology, Jayam College of Engineering and Technology, Dharmapuri.

Abstract: In instruction to additional variety use of the shiftless resources we strategy an energetic and marrying the overhead algorithm with PSM and the entrance achievement of new responsibilities. This can give inducements to users by achievement an additional segment of unexploited resource without additional imbursement. Experimentations approve realizing a fantastic finest implementation effectiveness of their responsibilities is conceivable. They could change to a development on system quantity by 15 percent 85 percent than the traditional approaches used in P2P Grid model conferring to the reproduction. We recapitulate the resource penetrating request as assortment interrogation constrictions. We evidence them to be the satisfactory and required settings for receiving the finest resource distribution. Experimentations authorize the considered PG-CAN protocol within consequential interrogation upstairs is intelligent to exploration competent possessions very effectively.

Keywords: Cloud computing, VM-multiplexing resource allocation, convex optimization, P2P multi-attribute range query.

1. INTRODUCTION

Cloud computing has emerged as a compelling paradigm for deploying distributed services. Resource allocation problem in cloud systems emphasizes how to harness the multi-attribute resources by multiplexing operating systems. With virtual machine (VM) technology its are able to multiplex several operating systems on the same hardware and allow task execution over its VM substrates without performance interference.

Fine-grained resource sharing can be achieved as each VM substrate can be configured with proper shares of resources dynamically. The balloon driver, difference engine, joint-VM, and virtual putty, can dynamically adjust the memory resource among collocated virtual machines. These advanced techniques enable computing resources to be dynamically partitioned or reassembled to meet the elastic needs of end users.

Such solutions create an unprecedented opportunity to maximize resource utilization, which are not possibly applied in most Grid systems. Its propose a fully distributed, VM-multiplexing resource allocation scheme to manage decentralized resources. Our approach not only achieves maximized resource utilization using the proportional share model (PSM). PSM also delivers provably and adaptively optimal execution efficiency. It's also design a novel Multi attribute range query protocol for locating qualified nodes. Contrary to existing solutions which often generate bulky messages per request, our protocol produces only one lightweight query message per task on the Content Addressable Network (CAN).

2. RELATED WORK

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. 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.

Fig. 1 shows the entire journey of a task from its submission to completion over the SOC system. In this work, we only focus on the multi attribute range query problem and the resource allocation problem for determining the amount of resources of a qualified node to the submitted task.

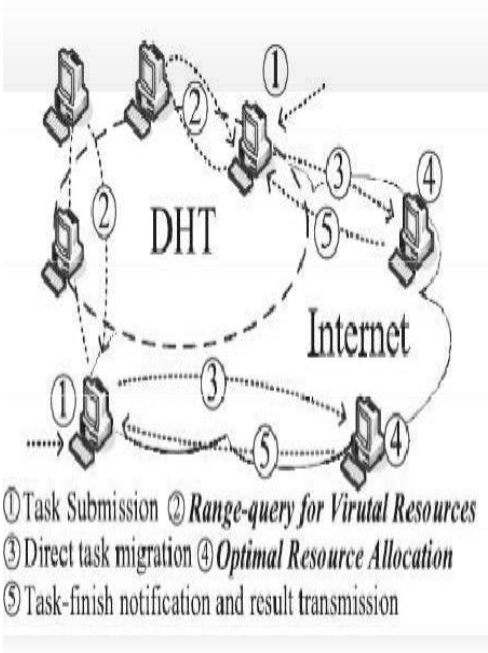

Figure 1: The entire task execution procedure

We focus on two key issues in the design of SOC:

- The multi attribute range query problem in a fully decentralized environment for locating a qualified node to satisfy a user task's resource demand with bounded delay.
- To optimize a task's execution time by determining the optimal shares of the multi attribute resources to allocate to the tasks with various QoS constraints, such as the expected execution time and limited budget.

SOC is different from the traditional Grid model (including P2P desktop Grid) in the resource consumption manner. Grids generally assume exclusive resource usage to ensure user QoS. The problem of job scheduling in Grids is usually categorized as a multiprocessor scheduling (MPS) problem (a kind of combinatorial optimization problem), which has been proved to be NP-complete.

Accordingly, many approximation algorithms as well as (Meta) heuristics applied to various versions of the MPS problem in the Grid environment. Met heuristic for solving the fixed job scheduling problem where processors are subject to spread time constraints, i.e., the time spent between the submission time and the completion time should not exceed a given duration.

Generalized External Optimization (GEO) is another metaheuristic for solving the MPS problem. The Grid scheduling problem through a cost-based provisioning model and a multi objective genetic algorithm for getting approximately optimized performance (such as throughput). In P2P desktop Grids, a heuristic load balancing method for improving the task scheduling throughput on desktop Grids over CAN overlay. Similarly the problem to be a bins-and-balls model with herds phenomenon and tried to get the approximately optimal performance using a stochastic algorithm atop a DHT overlay.

To maximize the VM-multiplexing resource utilization by analyzing VM-pairs' compatibility in terms of the forecasted workload and estimated VM sizes.

However, two significant drawbacks still remain:

- Poor scalability due to the central management of VM-correlation matrix;
- Restrictive constraints on implementation since them only identify the compatibility of VM pairs.

To overcome these problems, we formulate multi attribute resource allocation as a convex optimization problem and devise a resource allocation algorithm to minimize the task execution time with O (R3) time complexity. Since the node identifiers over the DHT are often generated based on some hash functions, it is uneasy to directly perform range queries. Some existing strategies have to build an extra layer to reorganize all of nodes over the DHT, whereas others leverage a CAN topology. Many other existing works mainly focus on how to locate the duty nodes that satisfy the user-specified range in all dimensions with limited delays.

3. PROBLEM FORMULATION

The entire journey of a task from its submission to completion over the SOC system. In this work, we only focus on the multi attribute range query problem and the resource allocation problem for determining the amount of resources of a qualified node to the submitted task. Suppose there are n nodes in SOC, each is denoted as pi, where 1 _ i_ n. Each node owns R different resources (or resource attributes) 1 managed by a Virtual Machine Monitor (VMM). We denote _ to be the set of resource attributes owned by node pi pi's capacity vector. For example, if a computer owns a

2.4 G flops single-core CPU, 1 GB memory, and a 10 Mbps network bandwidth, its capacity vector is (2.4, 1, 10)T . 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. 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.

After tij gets scheduled, we denote its actual allocated resource as T, where. Here, _ means the component wise inequality between two vectors. For short, we denote as r and rk, respectively, in the case without causing ambiguity. Each task has a load vector, denoted as, indicating the amount of workload on each of the R resource 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. Hence, if a task tij is executed at pi, its execution time is equal which indicates the relative importance of a resource that might affect the execution time of a task according to its property (e.g., CPU bound or IO bound).

In essence, 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.

4. POINTER-GOSSIPING CAN

Our resource allocation approach relies on the assumption that all qualified nodes must satisfy Inequalities. To meet this requirement, we design a resource discovery protocol, namely pointer- gossiping CAN, to find these qualified nodes. We choose CAN as the DHT overlay to adapt to the multidimensional feature. Like traditional CAN, each node (a.k.a., duty node) under PG-CAN is responsible for a unique multidimensional range zone randomly selected when it joins the overlay. Suppose there are 25 joined nodes, each taking charge of a single zone. If a new node (node 26) joins, a random point such as (0.6 Gflops, 0.55 GB) will be generated and its zone will be set as the new zone evenly split along a dimension from the existing zone that contains this point.

If there is only one non overlapped range dimension between two nodes (e.g., pi and pj) and they are adjacent at this dimension, we call them neighbors to each other. Furthermore, if the non-overlapped range of pi is always no less than pj's, pi is called pj's positive neighbor and pj is called pi's negative neighbor. Every node will periodically propagate the state-update messages about its available resource to the duty node whose zone encloses this vector. After a task tij generates a query with the constraints, the query message will be routed to the duty node containing the expected vector.

We could justify that the state messages (or state records) of all qualified nodes must be kept in those onward nodes of the duty node. Obviously, the searching area may still be too large for the complete resource query without flooding, so the existing solutions usually adopt random walk to get an approximated effect. However, according to our observation (to be presented), this will significantly reduce the likelihood of finding qualified resources, finally degrading the system throughput and user's QoS.

Basically, there are two manners to propagate the duty nodes' identifiers (or pointers) backward—spreading manner and hopping manner, thus the PG-CAN can also be split into two types, namely spreading manner based PG-CAN (SPG-CAN) and hopping manner-based PGCAN (HPG-CAN)., the duty node D1 sends a pointer-message containing D1's identifier to its selected pointer nodes (such as D2 and D3), notifying them that D1 has records. Upon receiving the message, the pointer nodes (D2 and D3) will further gossip D1's identifier to their negative direction pointer nodes along next dimension. In the identifier of any nonempty-cache node will be forwarded from pointer node to pointer node along each dimension. Obviously, the former results in fewer number of hops for message delivery, but its identifiers cannot be diffused as widely as the latter's. In fact, we can prove that the delay complexity of identifier delivery for the hopping manner, so the hopping manner is likely to be better than the spreading manner (to be confirmed in our simulation).

5. WORKING PROCESS

A system architecture or systems architecture is the conceptual design that defines the structure and/or behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or

building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system. System may enable one to manage investment in a way that meets business needs. The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.

The composite of the design architectures for products and their life cycle processes. A representation of a system in which there is a mapping of functionality onto hardware and software components, a mapping of the software architecture onto the hardware architecture, and human interaction with these components.

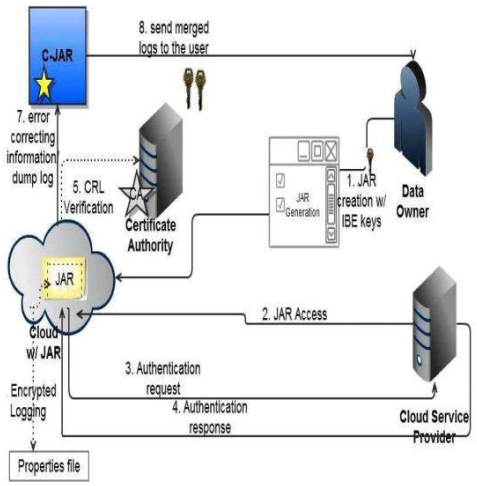

Figure 2.System Architecture

6. SYSTEM IMPLEMENTATION

6.1SELF-ORGANIZING CLOUD

SOC propose a self-managed key-value store that dynamically allocates the resources of a data cloud to several applications in a cost efficient and fair way. Our approach offers and dynamically maintains multiple differentiated availability guarantees to each different application despite failures.

6.2 PEER-TO-PEER NETWORK.

A peer-to-peer (P2P) network is a type of decentralized and distributed network architecture in which individual nodes in the network act as both suppliers and consumers of resources, in contrast to the centralized client–server model where client nodes request access to resources provided by central servers.

In a peer-to-peer network tasks are shared amongst multiple interconnected peers who each make a portion of their resources directly available to other network participants without the need for centralized coordination by servers.

6.3 QUERY DELAY TRAFFIC NETWORK

The data replication solutions in either wired or wireless networks aim at either reducing the query delay or improving the data availability but not both. As both metrics are important for mobile nodes.

Proposed schemes to balance the trade-offs data availability and query delay under different system settings and requirements. Extensive simulation results show that the proposed schemes can achieve a balance these two metrics and provide satisfying system performance.

6.4 RESOURCE MEASUREMENT

Quantity is computed from dimensions revealed in out crops trenches workings or drill holes grade and/or quality are computed from the results of detailed sampling. The sites for inspection sampling and measurements are spaced so closely and the geologic character is defined that size shape depth, and mineral content of the resource are established.

Quantity and grade and/or quality are computed from information similar to that used for measured resources but the sites for inspection, sampling measurement are farther apart or are otherwise less adequately spaced.

7. CONCLUSION

Prototype supporting live movements of VMs cloud that have any two nodes on the Internet are built successfully. It will also conduct sensitivity analysis of how violation of our model assumptions would impact the optimal resource allocation in future.

The resource identical frequency may transpire while limiting request interruption and network traffic. The resource detection protocol specifically Practical Catalogue Dispersion proactively disperses resource catalogues over the nodes and arbitrarily route query messages among them.

This paper proposes a novel scheme (DOPS) for virtual resource allocation on a SOC, with three key contributions listed below. . Optimization of task's resource allocation under user's budget: With a realistic monetary model, we propose a solution which can optimize the task execution performance based on its assigned resources under the user budget

REFERENCES

[1] J.E. Smith and R. Nair, Virtual Machines: Versatile Platforms for Systems and Processes. Morgan Kaufmann, 2005. 476 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 3, MARCH 2013

[2] D. Gupta, L. Cherkasova, R. Gardner, and A. Vahdat, "Enforcing Performance Isolation across Virtual Machines in Xen," Proc. Seventh ACM/IFIP/USENIX Int'l Conf. Middleware (Middleware '06), pp. 342-362, 2006.

[3] 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.

[4] 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.

[5] P. Crescenzi and V. Kann, A Compendium of NP Optimization Problems.ftp://ftp.nada.kth.se/Theory/Viggo-Kann/ compendium.pdf, 2012.

[6] O. Sinnen, Task Scheduling for Parallel Systems, Wiley Series on Parallel and Distributed Computing. Wiley-Interscience, 2007.

[7] 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.