A Lightweight Approach Multiplexing Resource Allocation Scheme by Virtualization based on Time Series in SOC

C.SARAVANAN¹ and C.ANURADHA²
¹PG Scholar, SCAD Engineering College, Tirunelveli, India.
²HOD, SCAD Engineering College, Tirunelveli, India.
(Received: February 16, 2014; Accepted: March 28, 2014)

ABSTRACTS

By leveraging virtual machine (VM) technology which provides performance and fault isolation, Cloud resources can be provisioned on demand in a fine-grained, multiplexed manner rather than in monolithic pieces. By integrating volunteer computing into Cloud architectures, we envision a gigantic Self-Organizing Cloud (SOC) being formed to reap the huge potential of untapped commodity computing power over the Internet. Towards this new architecture where each participant may autonomously act as both resource consumer and provider, we 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), but also delivers provably and adaptively optimal execution efficiency. We 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). It works effectively to find for each task its qualified resources under a randomized policy that mitigates the contention among requesters. This paper extends our previous work and proposes a lightweight mathematical model to estimate the energy cost of live migration of an idle virtual machine quantitatively. A series of experiments were conducted on KVM to profile the migration time and the power consumption during live migration. Based on these data we derived an energy cost model that predicts the energy overhead of live migration of virtual machines with an accuracy of higher than 90%.

Key words: Cloud Computing, VM-multiplexing Resource Allocation, Convex Optimization, P2P Multi-attribute Range-query, Lightweight Model.

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) technology¹, we 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 (such as CPU, memory, storage, network bandwidth) dynamically.
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 avoid possibly uneven load distribution and abrupt resource over-utilization caused by un-coordinated node selection process from autonomous participants, we investigate three node selection policies, namely double-check policy [21], queuing policy [22], and extra-virtual-dimension policy [19].

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, we prove that optimal resource allocation does exist and show that our solution is optimal.

This paper proposes a lightweight model to quantify the energy cost of live migration of virtual machines. A series of experiments were conducted on KVM15 and we derived such a model through linear regression on recorded data. The migration cost model is able to estimate the energy overhead of live migration of an idle virtual machine within an accuracy of 90%. This paper addresses a lightweight energy cost model that copes without a dedicated power or time model. The model is able to quantify energy overhead of live migration with respect to the RAM size of the virtual machine as well as with respect to the servers available network bandwidth. It completes the study which focuses on performance cost of live migration of virtual machines and estimates energy cost within an accuracy of higher than 90%.

**Optimal resource allocation**

Given a task \( tij \) with its weight vector \( w(tij) \) and a budget \( B(tij) \), we first prove that the optimal resource allocation on a qualified node \( ps \) with its price vector \( b(ps) \) does exist.

**Lemma 1**

The optimal allocation (denoted \( r'(tij) \)) exists iff (i.e., \( D'(tij) \)) Inequalities (6) and (7) are met. \( b(ps)T \, \hat{0} \, e(tij) \, d' B(tij) \) (6)

**Proof**

To prove \( D'(tij) \): (transport property of inequalities)

If \( r'(tij) \) exists, it must satisfy Inequalities (3) and (5), thus the Inequalities (6) and (7) should hold.

To prove \( B'(tij) \): (to satisfy Slater’s condition [23])

If \( b(ps)T \hat{0} e(tij) = B(tij) \) or \( e(tij) = a(ps) \), then \( e(tij) \) is a unique solution, which can be regarded as an optimal one. If \( b(ps)T \hat{0} e(tij) < B(tij) \) and \( e(tij)z' a(ps) \), other than \( e(tij) \), there must exist another better solution (denoted \( r2(tij) \) such that \( b(ps)T \hat{0} r2(tij) < B(tij) \) and \( e(tij)z' a(ps) \), thus \( r'(tij) \) must exist according to Slater’s condition [23]. Similarly, if \( b(ps)T \hat{0} e(tij) = B(tij) \) and \( e(tij)z' a(ps) \), Slater’s condition can also hold by excluding the equations from the constraints (7).

We assume the qualified node \( ps \) that satisfies Inequalities (6) and (7) can be found based on \( tij \)’s expected resource vector \( e(tij) \) by a resource discovery protocol (to be discussed in Section 5). Thus, we could rewrite the constraint (5) to be Inequality (8) and construct a Lagrangian function \( F1(\pi(tij)) \) That is, the optimal resource vector \( r' \) could be found as long as we could satisfy the above conditions simultaneously.

In order to solve the above simultaneous equations and inequalities, there are two traditional candidate strategies: (1) brute-force method and (2) interior point method is converged with them.

If we replace Constraint (5) with Constraint (11), we could find an optimal solution through a few convex optimization steps. That is, via such a constraint relaxation, we could optimize the resource allocation for task \( tij \) on node \( ps \) without exhausting all \( 3R \) possible combinations like the brute-force method or worrying about the convergence problem in the interior point method.

**Pointer-gossiping can**

Our resource allocation approach relies on the assumption that all qualified nodes must satisfy Inequalities (6) and (7) (i.e., Lemma 1). To meet this requirement, we design a resource discovery protocol, namely pointer-gossiping CAN (PG-CAN), to find these qualified nodes. We
choose CAN [17] as the DHT overlay to adapt to the multi-dimensional feature.

Like traditional CAN, each node (a.k.a. duty node) under PG-CAN is responsible for a unique multi-dimensional range zone randomly selected when it joins the overlay. Fig. 2 (a) illustrates an example of CAN overlay network. 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, 0.55GB) will be generated and its zone will be set as the new zone evenly split along a dimension from the existing zone (node 25 in Fig. 2 (a)) 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. In Fig. 2 (a), Node 9, 12 and 20 are positive neighbors of node 1.

Each duty node (such as D1) will cache state-update messages received from its neighbors, which are checked periodically and removed if outdated (i.e., beyond their TTL). In the meanwhile, it propagates its own identifier (such as IP) to a few randomly selected pointer nodes towards it negative direction. For those duty nodes containing valid state messages, we call them non-empty-cache nodes.

Basically, there are two manners to propagate the duty nodes’ identifiers (or pointers) backward - spreading manner (Fig. 3 (a)) and hopping manner (Fig. 3 (b)), thus the PG-CAN can also be split into two types, namely spreading manner based PG-CAN (SPG-CAN) and hopping manner based PG-CAN (HPG-CAN). In Fig. 3 (a), 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 Fig. 3 (b), 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 is O(log2 n) (Theorem 5), so the hopping manner is likely to be better than the spreading manner (to be confirmed in our simulation).

CONCLUSIONS

This paper proposes a novel scheme (DOPS) for virtual resource allocation on a Self-Organizing Cloud (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. We prove its optimality using the KKT conditions in the convex-optimization theory.

- **Maximized Resource Utilization based on PSM**: In order to further make use of the idle resources, we design a dynamic algorithm by combining the above algorithm with PSM and the arrival/completion of new tasks. This can give incentives to users by gaining an extra share of un-used resource without more payment. Experiments confirm achieving a super-optimal execution efficiency of their tasks is possible. DOPS could get an improvement on system throughput by 15%<"60% than the traditional methods used in P2P Grid model, according to the simulation.

**Lightweight Resource Query Protocol with Low Contention**

We summarize the resource searching request as two range query constraints, Formula (6) and Formula (7). We prove them to be the sufficient and necessary conditions for getting the optimal resource allocation. Experiments confirm the designed PGCAN protocol with light-weight query overhead is able to search qualified resources very effectively. So far, we have successfully built a prototype supporting live migration of VMs between any two nodes on the Internet (even though they are behind different
NATs). In the future, we will study fault-tolerance support for a (DOPS-based, PG-CAN-enabled) SOC system; we will also conduct sensitivity analysis of how violation of our model assumptions would impact the optimal resource allocation.

In this paper we investigated the energy cost of live migration and analyzed the role of the VM size and the role of the available network bandwidth in the energy consumption of hosting servers. We migrated virtual machines of sizes between 800 MB and 1700 MB if network bandwidth is between 20 MBps and 100 MBps. We measured the migration time and the power consumption of source and destination host and compared these values with the power consumption of the hosts in an idle state to capture the energy cost of live migration. Based on the profiled data, we deduced a lightweight energy cost model through linear regression. Experimental results demonstrate that this model is able to estimate energy overhead of live migration of virtual machines with an accuracy of more than 90%.

REFERENCES

