Future Generation Computer Systems 56 (2016) 51–63

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Decentralized cloud datacenter reconsolidation through emergent and topology-aware behavior

Mina Sedaghat *, Francisco Hernández-Rodriguez, Erik Elmroth

Department of Computing Science, Umeå University, Sweden

HIGHLIGHTS

- A formulation of the VM consolidation problem as a distributed optimization problem.
- A topology-aware resource management framework for VM consolidation.
- A VM consolidation algorithm for durable consolidations and cheap migration plans.
- An in-depth simulation-based evaluation of the system behavior under different settings and configurations.
- Accounting the topology constraints and performance factors reduces migration costs.

ARTICLE INFO

Article history:
Received 27 April 2015
Received in revised form
24 July 2015
Accepted 17 September 2015
Available online 23 October 2015

Keywords:
Cloud computing
Peer to peer
VM consolidation
Resource management

ABSTRACT

Consolidation of multiple applications on a single Physical Machine (PM) within a cloud data center can increase utilization, minimize energy consumption, and reduce operational costs. However, these benefits come at the cost of increasing the complexity of the scheduling problem.

In this paper, we present a topology-aware resource management framework. As part of this framework, we introduce a Reconsolidating PlaceMent scheduler (RPM) that provides and maintains durable allocations with low maintenance costs for data centers with dynamic workloads. We focus on workloads featuring both short-lived batch jobs and latency-sensitive services such as interactive web applications. The scheduler assigns resources to Virtual Machines (VMs) and maintains packing efficiency while taking into account migration costs, topological constraints, and the risk of resource contention, as well as the variability of the background load and its complementarity to the new VM.

We evaluate the model by simulating a data center with over 65,000 PMs, structured as a three-level multi-rooted tree topology. We investigate trade-offs between factors that affect the durability and operational cost of maintaining a near-optimal packing. The results show that the proposed scheduler can scale to the number of PMs in the simulation and maintain efficient utilization with low migration costs.

1. Introduction

Cloud providers offer an infrastructure to be shared by multiple applications, which is usually expensive and needs to be wisely utilized. Utilization can be improved by running an appropriate mix of application workloads on each individual machine, which is known as consolidation. While consolidation can be used to increase utilization, it also increases the complexity of the scheduling problem [1]. Complexity comes from the fact that application workloads are often heterogeneous with respect to their size, lifetime, performance sensitivity, and the type of resources they use, i.e. whether they are CPU- or memory-intensive. A degree of sub-optimal application placement is inevitable due to load changes and the fact that it would be impractically expensive to completely re-map every component of every running application across all of the available servers each time a load change occurred. Consequently, there is a need for a scheduler that can respond rapidly to changes in demand, producing efficient and durable packing in a way that accounts for the heterogeneity of the cloud's workloads, imposes low costs of maintaining the packing efficiency, and can scale up to tens of thousands of servers per data center. We consider a packing to be durable if it does not necessitate frequent migrations in order to maintain the usage efficiency of allocations.

Consolidation is intrinsically a computationally hard problem. Several groups have formulated consolidation as an Integer

* Corresponding author.
E-mail addresses: mina@cs.umu.se (M. Sedaghat), hernandf@cs.umu.se (F. Hernández-Rodriguez), elmroth@cs.umu.se (E. Elmroth).

http://dx.doi.org/10.1016/j.future.2015.09.023
0167-739X/

Linear Programming (ILP) problem, which can be solved relatively quickly [2–5]. However, the ILP approach does not scale well and becomes unfeasible when dealing with larger data centers and/or more severe packing constraints. To achieve scalability, an ILP formulation must either compromise on the quality of the solution in order to maintain a response time that is within acceptable limits or alternatively impose a static partitioning scheme on the infrastructure, which limits the efficiency of resource utilization because one fixed partition may be underutilized while another is over-utilized [1].

Here, we propose a new P2P consolidation framework. Some of this framework's basic functionality has previously been verified in prototype form [6]. The proposed framework is a general computational model for cooperatively optimizing a global system objective through local interactions and computations in a multi-agent system over a semi-random connectivity. We also introduce a scheduling heuristic designed to provide and maintain durable packing with low maintenance costs for a data center with a dynamic workload. A scheduler based on this heuristic is shown to achieve such durable packing in a way that avoids costly reconfigurations, and to offer cheap migration plans (i.e. schemes that specify which workloads should be migrated to which resources) to maintain packing efficiency. The framework's P2P structure provides parallelization with a high degree of concurrency, and also helps to minimize the time required for computations while improving scalability. The random overlay used in the proposed P2P structure allows the system to create a logical dynamic connectivity among a large pool of resources, dynamic cells, and reduces the negative impacts of static partitioning (which can lead to low utilization). Formulating the consolidation as a distributed optimization problem allows the system to factor in more sophisticated trade-offs than are possible with the ILP approach because it avoids the need for a highly loaded centralized scheduler. For example, the scheduler can determine whether to migrate a VM to a remote Physical Machine (PM) on another cluster or deploy it on a PM that is more nearby but subject to a higher risk of resource contention. The local decision making employed within the P2P framework also reduces the amount of monitoring data that must be collected and transferred over the network.

The main contributions of this paper are:

- A formulation of the VM consolidation problem as a distributed optimization problem.
- A topology-aware resource management framework for VM consolidation.
- A heuristic algorithm for VM consolidation that factors in the risks of resource contention, packing efficiency, migration costs, and migration locality to produce durable consolidations and offer cheap migration plans to maintain packing efficiency and reduce resource stranding.
- An in-depth simulation-based evaluation of the system behavior under different settings and configurations. The results obtained in this evaluation show that the proposed scheduling framework can produce durable consolidations for large numbers of VM requests with varying demands, arriving over a simulation time of 24 h at a data center with over 65 000 PMs. The framework scales to the tested number of PMs and maintains efficient resource utilization with low migration costs.

The remainder of the paper is organized as follows. In Section 2, we discuss the requirements and challenges of scheduling a mixed workload. Section 3 presents problem statements. In Section 4, we present the full P2P framework and the proposed heuristic algorithm. Section 5 describes the experimental setup, and Section 6 reports the evaluation and analysis of the results. Finally, we discuss related works and offer some concluding remarks in Sections 7 and 8, respectively.

2. Challenges and requirements

When scheduling VMs to run different services or batch jobs, the scheduler must meet several requirements and it faces a number of challenges in meeting them. A summary of these challenges are:

1. **Resource contention caused by consolidation**: Co-locating different applications can cause performance variability or degradation due to resource contention when resources are being shared [7]. The scheduler should therefore identify complementary workloads and place them together to improve packing efficiency and minimize resource contention.
2. **Job heterogeneity**: A data center will be required to run different types of applications. In broad terms, two classes of application can be distinguished: long-running interactive services and batch jobs, which perform a specific computation and then finish. Batch jobs that are run in cloud data centers are usually shorter and less latency-sensitive than interactive services, involve constant resource utilization, and do not usually require careful scheduling [1,8]. It would thus be best to devote most of the scheduler's available time and resources to the placing of interactive services and to spend relatively little time on scheduling batch jobs [1]. In addition, some tactics that can be applied to batch jobs in order to reduce the burden on an overloaded server could not acceptably be applied to interactive jobs. For example, a batch job could be stopped and restarted later, or the VM on which it is running could be transferred to another server via a cold migration. Neither approach would be possible for a VM running a latency-sensitive interactive service. Jobs can be further distinguished on the basis of other characteristics such as their lifetime, size, and performance sensitivity in order to develop effective strategies for fixing sub-optimal allocations that lead to the over- or under-loading of individual servers.
3. **Migration cost**: VM migration is a widely used technique for achieving consolidation once the decision on which jobs to consolidate has been made [9]. However, migrations are often costly. Particularly important costs to consider include the cost of double resource utilization during the migration, the costs of SLA violations caused by migration downtime, the cost of network traffic, and the potential network contention issues that may arise during the migration. The scheduler should produce a cheap migration plan with a minimal impact on the performance of the running applications. A migration plan may specify which type of migration is to be performed (cold or live), a candidate destination PM, and a list of VMs (selected based on their migration costs) to be migrated.
4. **Topological constraints**: The scheduler should consider the network topology to avoid high migration costs due to network traffic, contentions, or redundant configurations. Most existing works on scheduling treat the data center as an unstructured pool of resources, but real data centers Virtual LANs (VLANs), Access Control Lists (ACLs), broadcast domains, and load balancers that impose constraints and create barriers that reduce the scope for agility in migration [10].
5. **Risk of load change and contention**: The scheduler should factor the risk of change and contention into its decision function so as to avoid frequent migrations and produce durable decisions.
6. **Computation time**: The scheduler should produce a solution within an acceptable time-frame, and before the solution becomes disparaged due to load changes.