Poster Abstract - Application-Agnostic Batch Workload Management in Cloud Environments

Seyyed Ahmad Javadi, Shalini Bhaskara, Rahul Doshi, Prashanth Soundararapandian, Muhammad Wajahat, Anshul Gandhi
Stony Brook University - {sjavadi, shbhaskara, radoshi, psoundarapan, mwajahat, anshul}@cs.stonybrook.edu

ABSTRACT

We present Scavenger, a reactive batch workload manager that opportunistically runs containerized batch jobs next to customer Virtual Machines (VMs) in a public cloud like setting to improve utilization. Scavenger dynamically regulates the resource usage of batch jobs, including CPU usage, memory capacity, and LLC capacity, to ensure that the customer VMs’ resource demand is met at all times. We experimentally evaluate Scavenger and show that it considerably increases resource usage without compromising on the resource demand of customer VMs. Importantly, Scavenger does so without requiring any offline profiling or prior information about the customer workloads.

1 PROBLEM BACKGROUND

Servers in cloud data centers often have low resource utilization. A study focused on Amazon EC2 observed that cloud server usage is often below 10% [5]. To increase server utilization, prior works have proposed running provider workloads, such as Hadoop or Spark batch jobs, next to customer VMs to leverage idle resources [1, 3]. While effective, the key challenge with this approach is interference – the performance degradation of the colocated customer VMs due to resource contention with batch workloads at the underlying host server. This interference can be caused by contention for several resources simultaneously [4].

Ideally, in a public cloud, provider (or background (bg)) workloads should run next to customer (or foreground (fg)) workloads or VMs in such a way that their resource utilization complements that of the customer VMs. In particular, the dynamic demand of the customer VMs, across all resources, should be met at all times and the bg workloads should consume the remaining resources to do useful work. Thus, the goal is to maximize resource usage and bg workload throughput in a public cloud while satisfying the customer VM workloads’ resource demands at all times.

While there has been considerable prior work in this important area, there are still several shortcomings that must be addressed. Existing solutions often rely on historical usage patterns to predict the resource demand of fg VMs [6] or benchmark customer VM performance to carefully colocate bg workloads [2]; such solutions cannot always be deployed in public clouds where customer VMs should not be instrumented and there is often significant variation in VM loads [3, 4].

REFERENCES