Can time-shared over-subscription improve resource utilization?

Suhas J Subramanya
suhasj@cs.cmu.edu

Aanand Nayyar
aanandn@andrew.cmu.edu

Nithin R
nithinra@andrew.cmu.edu

Abstract

A recent study [5] shows a glaring problem in cloud computing – 75% of provisioned virtual machines (VMs) used less than 25% of their allocated CPU resources. Resource over-subscription – a process by which same resources are offered to multiple customers – is a practical solution to this problem. However, existing approaches tradeoff either availability[4, 2] or performance[3, 1] to obtain higher resource utilization from a providers perspective. We propose a new approach to resource over-subscription based around time-sharing. In this project, we study utility of this approach by simulating a simple datacenter with real user-workloads with distinct characteristics [5].

1 Motivation

Cloud computing provides a cheap alternative to provision resources on demand. The usual contract between a cloud provider and a *third-party* customer stipulates the resources to be provisioned, the duration to provision resources for, and availability guarantees for the provisioned resources for this duration. A recent study[5] of workloads on a public cloud provider reveals a glaring problem in resource utilization. From the study, a large majority of the provisioned VMs utilized less than 25% of their allocated resources. The relevant graph is shown in Figure 1. This problem is relevant to all involved parties – (a) customers pay for more resources than utilized, increasing their expenditure, and (2) cloud providers earn a lower ROI due to lower utilization of resources. Having recognized this problem,

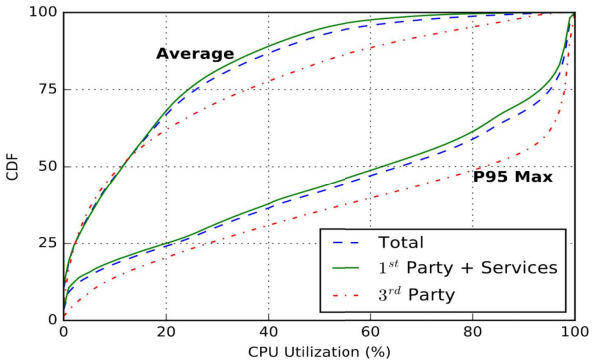

Figure 1: CPU Utilization in Azure VMs (2017)

the current approaches to mitigating resource under-utilization can be broadly divided into 3 categories.

- **Resource bidding**: In this model, a provider provisions resources to a customer with (almost) **no availability guarantees**. AWS Spot Instances[2] and Azure Spot VMs[4] are examples of this approach. This model enables a customer to acquire highly performant resources for a fraction of the usual price through a market bidding process, and enables the cloud providers to maximize resource utilization. For deep learning-like workloads, frequent checkpointing and restarts degrade performance for long running jobs, while workloads like automated software testing (CI/CD) pipelines can handle such interruptions more efficient.
- **Resource multiplexing**: In this model, a provider multiplexes available resources between different customers, resulting in only a **guaranteed base performance**, with promise