CCIT Report #888 September 2015

Links as a Service (LaaS): Feeling Alone in the Shared Cloud

Eitan Zahavi[*§], Alex Shpiner[§], Ori Rottenstreich[¶], Avinoam Kolodny[*] and Isaac Keslassy[*]
[*] Technion [§] Mellanox [¶] Princeton University

ABSTRACT

The most demanding tenants of shared clouds require complete isolation from their neighbors, in order to guarantee that their application performance is not affected by other tenants. Unfortunately, while shared clouds can offer an option whereby tenants obtain dedicated servers, they do not offer any network provisioning service, which would shield these tenants from network interference.

In this paper, we introduce *Links as a Service (LaaS)*, a new abstraction for cloud service that provides physical isolation of network links. Each tenant gets an exclusive set of links forming a virtual fat tree, and is guaranteed to receive the exact same bandwidth and delay as if it were alone in the shared cloud. Under simple assumptions, we derive theoretical conditions for enabling LaaS without capacity over-provisioning in fat-trees. New tenants are only admitted in the network when they can be allocated hosts and links that maintain these conditions. Using experiments on real clusters as well as simulations with real-life tenant sizes, we show that LaaS completely avoids the performance degradation caused by traffic from concurrent tenants on shared links. Compared to mere host isolation, LaaS can improve the application performance by up to 200%, at the cost of a 10% reduction in the cloud utilization.

1. INTRODUCTION

Many owners of private data centers would like to move to a shared multi-tenant cloud, which can offer a reduced cost of ownership and better fault-tolerance. But it is vital for these tenants that their applications will will not be affected by other tenants, and will keep exhibiting the same performance [1–3]. (By *performance*, we refer to the inverse of the total application run-time, including both the computation and communication times.)

Unfortunately, distributed applications often suffer from unpredictable performance when run on a shared cloud [4, 5]. This unpredictable performance is mainly caused by two factors: *server sharing* and *network sharing* [6–22]. The first factor, *server sharing*, is easily addressed by using bare-metal provisioning of servers, such that each server is allocated to a single tenant [23]. However, the second factor, *network sharing*, is much more difficult to address. When network links are shared by several tenants, network contention can significantly worsen the application performance if other tenant applications consume more network resources, e.g. if they simply want to benchmark their network or run a heavy backup [24]. This can of course prove even worse when other tenants purposely generate adversarial traffic for DoS or side-channel attacks [25].

As detailed in Section 2, current solutions either (a) require tenants to provide the traffic matrix in advance, which often proves impractical [11, 21]; (b) provide enough throughput for any set of admissible traffic matrices using a hose model, but then fail to provide a predictable latency that does not depend on other tenants [4, 16]; or (c) attempt to track the current traffic matrix, but then cannot guarantee the same performance [14, 15, 17, 19, 22].

In Section 3, we establish experimentation and simulation environments to better understand the impact of network contention as a function of the number of tenants and of the cluster size. We show that concurrent tenants with similar traffic can degrade MapReduce performance by 25%, and reduce the performance of scientific computing jobs by up to 65%.

In this paper, we introduce a simple and effective approach that eliminates any interference in the cloud network. Keeping with the notion that good fences make good neighbors, we argue that the most demanding tenants should be provided with exclusive access to a subset of the data center links, such that each tenant receives its own dedicated fat-tree network. We refer to such a cloud architecture model as *Links as a Service (LaaS)*. Under the LaaS model, we guarantee that tenants can obtain the exact same bandwidth and delay as if they were alone in the shared cloud, independently of the number of additional tenants.

While the LaaS abstraction sounds attractive, Figure 1 illustrates why it can be a challenge to provide it given any arbitrary set of tenants. First, Fig. 1(a) illustrates a bare-metal allocation of distinct hosts (servers) to two tenants that does not satisfy the LaaS abstraction, since