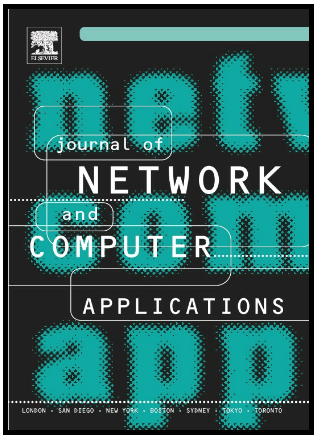

www.elsevier.com/locate/jnca

Compatibility-based Static VM Placement Minimizing Interference

Mahfuzur Rahman, Peter Graham

PII: S1084-8045(17)30065-6
DOI: http://dx.doi.org/10.1016/j.jnca.2017.02.004
Reference: YJNCA1860

To appear in: *Journal of Network and Computer Applications*

Received date: 3 July 2016
Revised date: 5 February 2017
Accepted date: 9 February 2017

Noname manuscript No.
(will be inserted by the editor)

Compatibility-based Static VM Placement Minimizing Interference

Mahfuzur Rahman · Peter Graham

Received: date / Accepted: date

Abstract The static (or initial) packing of VMs into a cloud provided host is done based on their expected resource requirements as specified in Service Level Agreements (SLAs). SLAs in Infrastructure as a Service (IaaS) clouds, however, capture neither changes in requirements over a VM's lifetime nor their dynamic characteristics (e.g. cache behaviour). Placing VMs for packing efficiency alone can result in "incompatible" VMs being co-located that interfere with one another's executions. This can result in the need for costly early VM migrations. In this paper, we address this problem by introducing Compatibility-based Static VM Placement (CSVP). CSVP contributes by exploiting easy-to-obtain information about VMs' expected load variation to co-locate compatible VMs within a scheduling batch together thereby improving their initial performance. We have implemented CSVP in CloudSim and done simulations using workloads derived from a subset of the Google traces. Our results show that even using only simple threshold information about VM behaviour CSVP provides better initial VM placements to avoid some VM interference. Using CSVP, VMs are thus more likely to execute effectively together from their start thereby decreasing the overhead of VM migration.

1 Introduction

Algorithms for the static (or initial) assignment of Virtual Machines (VMs) to Physical Machines (PMs) in a IaaS cloud are often focused on packing efficiency to minimize use of cloud resources. Packing is done based on resource requirements specified by cloud users via Service Level Agreements (SLAs). Such SLAs specify an expected requirement for each of a number of resource types (CPU, memory, network, I/O, etc.). In an IaaS cloud, SLAs do not include application-level information which must be inferred. Further, such SLAs do not capture variation in resource requirements over the execution lifetime of a VM and this can lead users to over- or under-estimate their VMs' resource requirements to achieve their performance and cost goals. This can be problematic by, for example, leading to poor utilization of cloud resources. Further, SLAs do not address all the potential sources of contention with other VMs they may be co-located with.

A strict focus on VM packing alone ignores some aspects of the "compatibility" of co-hosted VMs (i.e. which VMs will execute more effectively together given that they compete for resources). For example, it is possible that two VMs on the same PM may have been packed together based on SLA values that indicate they should run well together given the available resources but, due to unexpected load variation, they collectively over-consume one or more types of resources sometime during their execution. This form of VM interference, that we refer to as "resource-visible interference" must be addressed to ensure efficient cloud operation and user satisfaction.[1]

It is beneficial to initially co-locate VMs that are likely to be compatible in terms of their varying resource requirements as well as being able to be efficiently packed into a PM based on SLA values. If this can be done easily (without requiring the cloud provider

Mahfuzur Rahman, Peter Graham
Department of Computer Science, University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2
E-mail: {mahfuz,pgraham}@cs.umanitoba.ca

[1] Other forms of VM interference involve implicitly shared resources including those at the micro-architectural level such as last-level caches (that are commonly shared between cores) where the interference is not "visible" based on SLAs.