Simulation Modelling Practice and Theory 75 (2017) 17–28

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

Optimal provisioning of servers for hosting services of multiple types

Paul Ezhilchelvan*, Isi Mitrani

School of Computing Science, Newcastle University, Claremont Tower, Claremont Road, Newcastle upon Tyne, NE1 7RU, UK

ARTICLE INFO

Article history:
Received 18 December 2016
Revised 24 March 2017
Accepted 25 March 2017

Keywords:
Service provisioning
Multiple job classes
Revenue optimization
Job migration
Erlang-type models

ABSTRACT

Services of different types are provided to paying customers by instantiating Virtual Machines on servers hired from a cloud. Different VMs can share a server, subject to one or more resource constraints. Incoming jobs whose resource requirements cannot be satisfied are lost. The objective is to maximize the long-term average profit per unit time. A single-server model is analyzed exactly and the results provide approximations for the system with n servers. The latter is also solved exactly when the servers are dedicated and when the VMs can migrate instantaneously. Numerical examples and comparisons with simulations are presented.

1. Introduction

This paper is concerned with the provision of services of different types, with different patterns of demand, resource requirements and revenue streams. The service provider hires servers from a Cloud, incurring certain costs. To run a job of a given type, a Virtual Machine (VM) of that type is instantiated on one of the servers. The resource availability on a server is bounded, so that whether a VM can be allocated to it or not, depends both on the type of the new job and on the numbers and types of the other jobs already running. When an incoming job cannot be started on any of the servers, it is rejected and the revenue that it would bring is lost.

The problem is to decide how many servers to hire so as to maximize the average long-term profit (revenues minus costs) per unit time. To that end, we examine first a model of a single server with either a single shared resource or multiple shared resources. The exact solution of that model, which is known, is then used to provide accurate estimates for the profit achieved by n servers, provided that n is not very large.

Another model that is solved exactly concerns moveable VMs that can migrate from server to server and be packed efficiently according to some simple algorithm. It turns out that such packing does not produce significant improvements in the achievable profit.

For large-scale problems, such as deciding how many servers to power on, out of the thousands that are typically available in a service center, we propose two simpler approximations. One is based on aggregating the different job types into a single type, with appropriately chosen parameters. The other treats unit resource requests as separate and independent of each other. Both approximations make acceptable predictions about the optimal number of servers, but the one using aggregation is more accurate away from the optimal region.

* Corresponding Author.
E-mail addresses: paul.ezhilchelvan@ncl.ac.uk (P. Ezhilchelvan), isi.mitrani@ncl.ac.uk (I. Mitrani).

http://dx.doi.org/10.1016/j.simpat.2017.03.011
1569-190X/

We assume that the demand parameters are given, and the system reaches steady state during a period where those parameters remain fixed. In practice, the hiring policies would have to be supplemented by some monitoring and parameter estimation technique that would detect when the traffic parameters change. Such techniques exist (see below).

1.1. Related work

The resource sharing and optimization problems described here have not, to our knowledge, been addressed before in the context of server sharing by multiple job types. There has been quite a lot of work on server allocation with a single job type. Perhaps the closest to the present study is the paper by Ezhilchelvan and Mitrani [4], where it was found that dynamic allocation policies do not bring significant benefits over static ones. The trade-off between performance and energy consumption, again for a single job type, was examined by Mazzucco et al. [11,12], using models and empirical observations. Their focus, and also that of Bodík et al. [2], is on estimating the traffic and reacting to changes in the parameters.

A number of early works have considered the multi-class resource sharing problem in the context of circuit-switched networks, see Kelly [10], Hampshire et al. [8] and Ross [16]. In the telephony field, the resources are the circuits available on various links, and the job types are indexed by the set of links that can be reserved for a call. Mapping those models to the cloud and VM area, as was done in Tan et al. [18], results in what we call here the 'single server model' (section 2). The concept of a number of servers, each with bounded resources, and a policy for allocating VMs to servers, has not been examined in the presence of multiple job types.

The studies by Wood et al. [22], Singh et al. [17], Weijia et al. [21], and Arzuaga and Kaeli [1], assume a given set of jobs currently present in the system, together with their resource requirements, and aim to allocate the corresponding VMs so as to minimize the number of servers and satisfy certain performance constraints. There are similar works concerned with power management (eg. Tang et al. [19] and Moore et al [14]). None of these papers take into account the processes of job arrivals and services.

A preliminary version of the present paper, which contained neither the large-n approximations nor their evaluation, was presented at MASCOTS 2016, [5].

The single server models and their solutions are described in Section 2. The profit maximization problem is introduced and solved in Section 3. An evaluation of a system where servers are dedicated to particular job types is also presented there. Section 4 covers the model with moveable VMs and packing. The large-scale approximation is introduced and evaluated in Section 5. Some conclusions and directions for future work are summarized in Section 6.

2. Models of a single server

A server may be shared by VMs of K different types, numbered $1, 2, \ldots, K$. The service provided by a VM of type i during its lifetime is referred to as a 'job of type i'. Jobs of type i arrive in an independent Poisson stream with rate λ_i. Their service times may be general IID random variables with mean $1/\mu_i$ $(i = 1, 2, \ldots, K)$.

Assume, to begin with, that the resource requirement of a VM is measured by a single number. More precisely, a job of type i consumes b_i units of resource. In order that the jobs running in parallel do not interfere with each other unduly, an upper bound B is imposed on the total amount of resource used by the jobs in the server. An incoming job that would cause that bound to be exceeded is rejected and is lost.

This model is of a type introduced and solved some decades ago in connection with circuit-switching networks. There, a number of circuits are allocated to calls of different types (e.g., see Ross [16]). The product-form solution was shown to be insensitive to the service time distribution.

The state of the server is described by the integer vector $\mathbf{j} = (j_1, j_2, \ldots, j_K)$, where j_i is the number of jobs of type i in progress. Denote by $S(K, B)$ the set of admissible state vectors. The resource restriction implies that this set is defined by

$$S(K,B) = \left\{ \mathbf{j} \ : \ \mathbf{j} \geq 0 \ , \ \sum_{i=1}^{K} j_i b_i \leq B \right\}. \tag{1}$$

The dependence of $S(K, B)$ on the individual resource requirements b_i is left implicit in order to keep the notation simple.

Let $\pi(\mathbf{j})$ be the steady-state probability that the server is in state $\mathbf{j}$. These probabilities are given by

$$\pi(\mathbf{j}) = \frac{1}{G(K,B)} \prod_{i=1}^{K} \frac{\rho_i^{j_i}}{j_i!}; \ \ \mathbf{j} \epsilon S(K,B) \ , \tag{2}$$

where $\rho_i = \lambda_i/\mu_i$ is the offered load of type i. The normalizing constant $G(K, B)$ is chosen so that the sum of all probabilities is 1. That is,

$$G(K,B) = \sum_{\mathbf{j} \epsilon S(K,B)} \prod_{i=1}^{K} \frac{\rho_i^{j_i}}{j_i!}. \tag{3}$$

Computing the normalization constant $G(K, B)$ can be a non-trivial task. A simple way to accomplish it is to use recursion. Let $m_i = \lfloor B/b_i \rfloor$ be the largest possible number of type i jobs that can be admitted into the server. Consider a particular job