Future Generation Computer Systems 28 (2012) 371–378

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

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

Model-driven auto-scaling of green cloud computing infrastructure

Brian Dougherty [a,*], Jules White [b], Douglas C. Schmidt [a]

[a] Institute for Software Integrated Systems, Vanderbilt University, Campus Box 1829 Station B, Nashville, TN 37235, United States

[b] ECE, 302 Whitemore Hall, Virginia Tech, Blacksburg, VA 24060, United States

ARTICLE INFO

Article history:
Received 1 November 2010
Received in revised form
12 April 2011
Accepted 9 May 2011
Available online 17 May 2011

Keywords:
Cloud computing
Auto-scaling
Power optimization
Model-driven engineering

ABSTRACT

Cloud computing can reduce power consumption by using virtualized computational resources to provision an application's computational resources on demand. Auto-scaling is an important cloud computing technique that dynamically allocates computational resources to applications to match their current loads precisely, thereby removing resources that would otherwise remain idle and waste power. This paper presents a model-driven engineering approach to optimizing the configuration, energy consumption, and operating cost of cloud auto-scaling infrastructure to create greener computing environments that reduce emissions resulting from superfluous idle resources. The paper provides four contributions to the study of model-driven configuration of cloud auto-scaling infrastructure by (1) explaining how virtual machine configurations can be captured in feature models, (2) describing how these models can be transformed into constraint satisfaction problems (CSPs) for configuration and energy consumption optimization, (3) showing how optimal auto-scaling configurations can be derived from these CSPs with a constraint solver, and (4) presenting a case study showing the energy consumption/cost reduction produced by this model-driven approach.

1. Introduction

Current trends and challenges. By 2011, the power consumption of computing data centers is expected to exceed 100,000,000,00 kilowatt hours (kW h) and generate over 40,568,000 tons of CO_2 emissions [1–3]. Since data centers operate at only 20–30% utilization, 70–80% of this power consumption is lost due to over-provisioned idle resources, resulting in roughly 29,000,000 tons of unnecessary CO_2 emissions [1–3]. Applying new computing paradigms, such as cloud computing with auto-scaling, to increase server utilization and decrease the idle time is therefore paramount for creating greener computing environments with reduced power consumption and emissions [4–8].

Cloud computing is a computing paradigm that uses virtualized server infrastructure to provision virtual OS instances dynamically [9]. In enterprise computing environments, however, the application demand often fluctuates rapidly and by a large margin. In some cases, drastic load increases may occur with such speed that new virtual machine (VM) instances cannot be booted quickly enough to meet response time requirements, even if automated techniques are applied. To overcome this problem and ensure that response time requirements are satisfied, VM instances can be prebooted to handle periods of high demand and remain idle during periods of light demand. When the application demand spikes, these VM instances can be allocated automatically without incurring the delay required for booting. This technique, however, always requires a number of idle VM instances standing by in the queue, leading to wasted power consumption and increased operating cost.

Rather than over-provisioning an application's infrastructure to meet peak load demands, an application can *auto-scale* by dynamically acquiring and releasing VM instances as the load fluctuates. Auto-scaling increases server utilization and decreases the idle time compared with over-provisioned infrastructures, in which superfluous system resources remain idle and unnecessarily consume power and emit superfluous CO_2. Moreover, by allocating VMs to applications on demand, cloud infrastructure users can pay for servers incrementally rather than investing large up-front costs to purchase new servers.

Devising mechanisms for reducing power consumption and environmental impact through cloud auto-scaling is hard. Auto-scaling must ensure that VMs can be provisioned and booted quickly to meet response time requirements as the load changes. If auto-scaling responds to load fluctuations too slowly, applications may experience a period of poor response time awaiting the allocation of additional computational resources. One way to mitigate this risk is to maintain an auto-scaling queue containing prebooted and preconfigured VM instances that can be allocated rapidly, as shown in Fig. 1.

* Corresponding author.
E-mail addresses: briand@dre.vanderbilt.edu (B. Dougherty), schmidt@dre.vanderbilt.edu (J. White), julesw@vt.edu (D.C. Schmidt).

0167-739X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2011.05.009

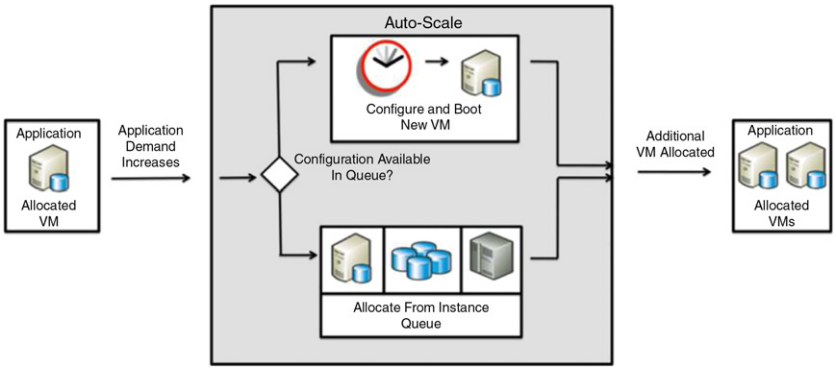

Fig. 1. Auto-scaling in a cloud infrastructure.

When a cloud application requests a new VM configuration from the auto-scaling infrastructure, the auto-scaling infrastructure first attempts to fulfill the request with a prebooted VM in the queue. For example, if a VM with Fedora Core 6, JBoss, and MySQL is requested, the auto-scaling infrastructure will attempt to find a matching VM in the queue. If no match is found, a new VM must be booted and configured to match the allocation request.

Open problem → determining green settings, such as the size and properties of the auto-scaling queue shared by multiple applications with different VM configurations [10]. The chosen configurations must meet the configuration requirements of multiple applications and reduce power consumption without adversely impacting the quality of service (QoS). For example, a web application may request VM instances configured as database, middle-tier Enterprise Java Beans (EJB), or front-end web servers. Determining how to capture and reason about the configurations that comprise the auto-scaling queue is hard due to the large number of configuration options (such as MySQL and SQL Server databases, Ubuntu Linux and Windows operating systems, and Apache HTTP and IIS/Asp.Net web hosts) offered by cloud infrastructure providers.

This article presents a technique for minimizing the number of idle VMs in an auto-scaling queue to reduce the energy consumption and operating costs without sacrificing response time. Although having a queue of prebooted VMs provides faster response time, determining the quantity and configuration of VMs to fill the queue is hard. This demonstrates that this problem can be phrased as a feature selection problem from software product lines and can be optimized with constraint solvers. Applying this optimization process yields the minimal set of VM configurations that should be prebooted to ensure that response time requirements are satisfied.

Solution approach → auto-scaling queue configuration derivation based on feature models. This paper presents a model-driven engineering (MDE) approach called *Smart Cloud Optimization for Resource Configuration Handling* (SCORCH). SCORCH captures VM configuration options for a set of cloud applications and derives an optimal set of VM configurations for an auto-scaling queue to provide three green computing contributions.

- An MDE technique for transforming feature model representations of cloud VM configuration options into constraint satisfaction problems (CSPs) [11,12], where a set of variables and a set of constraints govern the allowed values of the variables.
- An MDE technique for analyzing application configuration requirements, VM power consumption, and operating costs to determine what VM instance configurations an auto-scaling queue should contain to meet an auto-scaling response time requirement while minimizing power consumption.
- Empirical results from a case study using Amazon's EC2 cloud computing infrastructure (`aws.amazon.com/ec2`) that shows how SCORCH minimizes the power consumption and operating costs while ensuring that auto-scaling response time requirements are met.

2. Challenges of configuring virtual machines in cloud environments

Reducing unnecessary idle system resources by applying auto-scaling queues can potentially reduce the power consumption and resulting CO_2 emissions significantly. QoS demands, diverse configuration requirements, and other challenges, however, make it hard to achieve a greener computing environment. This section describes key challenges of capturing VM configuration options and using this configuration information to optimize the setup of an auto-scaling queue to minimize power consumption.

2.1. Challenge 1: capturing VM configuration options and constraints

Cloud computing can yield greener computing by reducing power consumption. A cloud application can request VMs with a wide range of configuration options, such as type of processor, OS, and installed middleware, all of which consume different amounts of power. For example, the Amazon EC2 cloud infrastructure supports five different types of processor, six different memory configuration options, and several different OS types, as well as multiple versions of each OS type [13]. The power consumption of these configurations ranges from 150 to 610 W/h.

The EC2 configuration options cannot be selected arbitrarily and must adhere to myriad configuration rules. For example, a VM running on Fedora Core 6 OS cannot run MS SQL Server. Tracking these numerous configuration options and constraints is hard. Sections 3.1 and 3.2 describe how SCORCH uses feature models to alleviate the complexity of capturing and reasoning about configuration rules for VM instances.

2.2. Challenge 2: selecting VM configurations to ensure auto-scaling speed requirements

While reducing idle resources results in less power consumption and greener computing environments, cloud computing applications must also meet stringent QoS demands. A key determinant of auto-scaling performance is the types of VM configuration that are kept ready to run. If an application requests a VM configuration and an exact match is available in the auto-scaling queue, the request can be fulfilled nearly instantaneously. If the queue does not have an exact match, it may have a running VM configuration that can be modified to meet the requested configuration more quickly than provisioning and booting a VM from scratch.