Chapter 7
Model-Driven Automated Error Recovery in Cloud Computing

Yu Sun
University of Alabama at Birmingham, USA

Jules White
Virginia Tech, USA

Jeff Gray
University of Alabama, USA

Aniruddha Gokhale
Vanderbilt University, USA

ABSTRACT

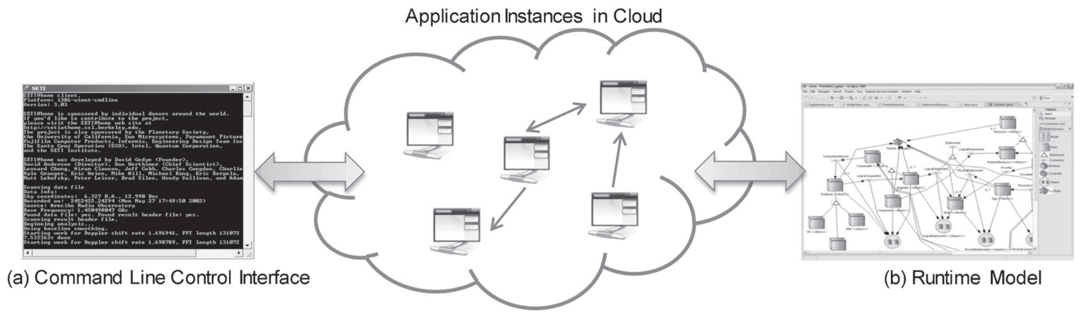
Cloud computing provides a platform that enables users to utilize computation, storage, and other computing resources on-demand. As the number of running nodes in the cloud increases, the potential points of failure and the complexity of recovering from error states grows correspondingly. Using the traditional cloud administrative interface to manually detect and recover from errors is tedious, time-consuming, and error prone. This chapter presents an innovative approach to automate cloud error detection and recovery based on a run-time model that monitors and manages the running nodes in a cloud. When administrators identify and correct errors in the model, an inference engine is used to identify the specific state pattern in the model to which they were reacting, and to record their recovery actions. An error detection and recovery pattern can be generated from the inference and applied automatically whenever the same error occurs again.

INTRODUCTION

With the increasing complexity of software and systems, domain analysis and modeling are becoming more important for software development and system applications. Applying domain-specific modeling languages and transformation engines is an effective approach to address platform complexity and the inability of third-generation languages to express domain concepts clearly (Schmidt, 2006). Building correct models for a specific domain can often simplify many complex tasks, particularly for distributed applications

DOI: 10.4018/978-1-61692-874-2.ch007

Figure 1. Two options to control application instances


Application Instances in Cloud
SETI
(a) Command Line Control Interface
(b) Runtime Model

```
SETI@home client.
Platform: i386-winnt-cmdline
Version: 3.03

SETI@home is sponsored by individual donors around the world.
If you'd like to contribute to the project,
please visit the SETI@home web site at
http://setiathome.ssl.berkeley.edu.
The project is also sponsored by the Planetary Society,
the University of California, Sun Microsystems, Paramount Picture
Fujifilm Computer Products, Informix, Engineering Design Team Inc
The Santa Cruz Operation (SCO), Intel, Quantum Corporation,
and the SETI Institute.

SETI@home was developed by David Gedye (Founder),
David Anderson (Director), Dan Werthimer (Chief Scientist),
Leonard Chung, Hiram Clawson, Jeff Cobb, Charles Congdon, Charlie
Kyle Granger, Eric Heien, Mike Hill, Michael Kang, Eric Korpela,
Matt Lebofsky, Peter Leiser, Brad Silen, Woody Sullivan, and Adam

Scanning data file
Data Info:
Sky coordinates:  6.327 R.A., 12.990 Dec
Recorded on:  2452422.24294 (Mon May 27 17:49:50 2002)
Source: Arecibo Radio Observatory
Base Frequency: 1.420498047 GHz
Found data file: yes. Found result header file: yes.
Scanning result header file.
Beginning analysis...
Doing baseline smoothing.
Starting work for Doppler shift rate 1.696941, FFT length 131072
7.532363% done
Starting work for Doppler shift rate 1.698789, FFT length 131072
```

based on cloud computing (Hayes, 2008) that offer several opportunities for customization and variability.

Cloud computing shifts the computation from local, individual devices to distributed, virtual, and scalable resources, thereby enabling end-users to utilize the computation, storage, and other application resources (which forms the "cloud") on-demand (Hayes, 2008). Amazon EC2 (Elastic Compute Cloud) (http://aws.amazon.com/ec2/, 2009) is an example cloud computing platform that allows users to deploy different customized applications in the cloud. A user can create, execute, and terminate the application instances as needed, and pay for the cost of time and storage that the active instances use based on a utility cost model (Rappa, 2004).

In the cloud computing paradigm, the large number of running nodes increases the number of potential points of failure and the complexity of recovering from error states. For instance, if an application terminates unexpectedly, it is necessary to search quickly through the large number of running nodes to locate the problematic nodes and states. Moreover, to avoid costly downtime, administrators must rapidly remedy the problematic node states to avoid further spread of errors.

Just like standard enterprise applications, cloud computing applications can suffer from a wide range of problems stemming from hardware failure to operator error (Oppenheimer et al., 2003). For example, Amazon EC2 provides limited guarantees about availability or reliability of hardware or virtual machine (VM) instances. Operators must be prepared to re-launch VM instances when failures occur, transfer critical data to newly provisioned VM images, start critical services on new VM instances, join new nodes to virtual LANs or security contexts, or update load balancers and elastic IP addresses to reference newly provisioned infrastructure.

Although Amazon EC2 provides a user-friendly and simple interface to manage and control the application instances (Figure 1a), administrators must still be experienced with the administrative commands, the configuration of each application, as well as some domain knowledge about each running instance. Administrators must therefore be highly trained to effectively and efficiently handle error detection and error recovery. The complexity of managing a large cloud of nodes can increase maintenance costs, especially when personnel are replaced due to turnover or downsizing.

Even with experienced administrators, the process of error recovery involves the following challenges:

- It is hard to locate errors accurately with a large number of running application instances.
- It may take too much time to detect and locate an error, causing a long period of service termination or further error propagation.