Peeking into the Cloud: Toward User-Driven Cloud Management

Theophilus Benson, Aditya Akella
University of Wisconsin, Madison

Sambit Sahu, Anees Shaikh
IBM Research, TJ Watson

1. INTRODUCTION

The emergence and growing popularity of cloud computing signals an evolution in the way IT infrastructure and services are delivered and consumed. Cloud services have a number of essential characteristics, such as self-service on-demand consumption, location independence, and rapid elasticity. There are also evolving cloud delivery models, including infrastructure-as-a-service (IaaS), platform-as-a-service, and software-as-a-service, which provide IT infrastructure, application development services, and software applications from the cloud, respectively [2].

The IaaS delivery model is perhaps the best known cloud service type. With IaaS, users have the ability to easily acquire and release infrastructure resources on-demand with an accompanying pricing model in which users pay for only what they use. The IaaS model of cloud computing is best exemplified in services such as Amazon's Elastic cloud computing (EC2) [1]. These providers give users easy and inexpensive access to virtual computing resources. However, this high degree of virtualization comes with a significant drawback: it limits users' ability to efficiently diagnose problems with their applications in the cloud. By hiding information about the underlying infrastructure, IaaS providers leave users with little alternative except trial-and-error troubleshooting. These approaches are effective, but take several hours to days to resolve problems.

Our aim is to develop a framework for IaaS cloud offerings which provides support for more streamlined, informed and automated user-driven troubleshooting and management. To design this framework we investigate the extent to which information hiding in IaaS clouds impacts the problem diagnosis and resolution process. We studied 3 years worth of user problem data containing over 9000 reported problems.

As a result of our study, we found that the problems can be broadly classified into 4 groups based on the symptoms identified by the users; **Instance unreachable:** users are unable to log in or perform other actions on their instance, **Instance unbootable:** users are unable to boot up an image on their instance, **Component unattachable:** users are unable to detach virtualized infrastructure from an instance, and **Performance problems:** users face abnormally long wait times. Of the 4 symptoms, we observed that the "Instance unreachable" symptom was the most prevalent symptom, constituting 44% of the reported problems. In evaluating the resolution times, we discovered that about 60% of the problems were resolved in under 25 hours while for the next 20% the resolution time took as long as addtitional 100 hours. We believe a large number of these resolutions can be significnatly sped up by simply exposing a small amount of additional information about the cloud's infrastructure.

2. CLOUD PRIMITIVES

Building on the problems observed in our study, we developed a framework for IaaS clouds, that enables better management by providing users with a set of primitives that expose hidden information about the virtualized infrastructure. The framework implements the following primitives:

Virtual Console: The current abstraction of IaaS implementations limit user access to the cloud only via their in-band channel, i.e. via SSH. In our study, we observed that due to the circular dependency between management and the liveness of the instance, users have to rely on the cloud provider entirely if an instance is not loaded successfully, or the guest operating system crashes. We propose that the cloud provider equipt the cloud with an out-of-band channel to a user's instance. Using this primitive, users can better debug "Instance unreachable" problems.

Component Health State: In our study, we saw several instances where the user was unsure why the instance was not responding or malfunctioning. This ambiguity was caused by the virtualization of physical resources which strips the users of visibility into the health of the resources that can affect the user's instance We propose that having binary information on the dependent set of resources would provide the user the required information for making decisions such as whether to restart a new instance or debug the application. Using this primitive, users can better debug "Instance unreachable" and "Instance unbootable" problems.

Component Status Logs: We observed many cases where the user was unable to attach virtual components due to issues with a previous instance. The root cause of many of these problems is that there is currently no way for a user to know if a submitted task succeeded or not as the cloud provider does not provide the status of task that cross the user-infrastructure boundary. We propose that the cloud provider logs the status of tasks required to complete a user's requests and expose these logs to the user. Using this primitive, users can better debug "Component unattachable" problems.

Performance Benchmarks: Our study of the corresponding performance problem tickets indicated that, more often than not, the root cause resided with the provider; however in some cases, user errors could also lead to performance problems. To disambiguate between user and provider side problems, we propose that the providers expose a set of benchmark numbers for operating system and network actions that the users depend on. For example, this set of performance counters would include time to attach virtualized infrastructure, or time to transfer blocks to disk. Using this primitive, users can better debug "Performance" problems.

3. REFERENCES

[1] Amazon ec2. http://aws.amazon.com/ec2/.
[2] P. Mell and T. Grance. Draft NIST working definition of cloud computing, June 2009.