Developing Robust Cloud Applications

Yuanyuan(YY) Zhou
University of California, San Diego

1. Introduction

Cloud Computing is a way of providing dynamically scalable and available resources such as computation, storage etc as a service to users who can use it to deploy their applications and store data. Despite its possible security and privacy risks, cloud computing has become an industry trend. No matter what forms of cloud, public or private cloud, raw cloud infrastructure or applications (software) as a service, they all provide the benefit of utility-based computing, i.e. pay for what you have used. It can help reduce IT cost as cloud service is paid incrementally and scales with demands. It can also support larger scale in terms of computation power and data storage without the configuration and set-up hassle required for installing and deploying a local large-scale cluster. It also has more mobility and provides access from wherever with internet. All these benefits allow IT to shift focus and focus on their own domain specific problems and innovations.

More and more applications are ported or developed to run on clouds. For example, Google News, Google Mail, Google doc are all applications running on clouds. Of courses, the cloud platforms in these cases are owned and controlled by the application service provider, namely Google, which make some of the challenges discussed below easier to address. Also many applications, especially those require cheap cost and are less sensitive to security issues, also moved to public cloud such as Amazon Elastic Computing Cloud (EC2) or Amazon Machine Images (AMIs). For example, in Feb 2009, IBM and Amazon Web Services have partnered to allow developers to use Amazon EC2 to build and run a range of IBM platform technologies. Specifically, this new "pay-as-you-go" model provides development and production instances of IBM DB2, Informix Dynamic Server, WebSphere Portal, Lotus Web Content Management and Novell's SUSE Linux operating system on EC2. Developers can use their existing IBM licenses on Amazon EC2, enabling software developers to quickly build applications based on IBM software within Amazon EC2.

With this new paradigm shift in computation, the tangible cost savings and other benefits also bring unique challenges to building robust and reliable applications on clouds. The first major challenge is the need for a mindset change: considering the unique characteristics (elasticity of scale, transparency of physical devices, unreliable components, etc) of deploying and running an application in clouds. The second challenge is the lack of appropriate frameworks and toolsets to support developing, testing and diagnosing applications in clouds.

In the following section, I will discuss what have changed from traditional application development and execution environment, what are the unique challenges and characteristics with clouds, their implications to cloud computing to application development, and what need to be done to make this paradigm shift and develop robust applications for clouds.

2. What are different in cloud?

While there are many commonalities between traditional in-house /local execution platform and clouds, there are several characteristics and challenges that are either unique or more pronounced in clouds. Some are more of a short-term characteristic and should improve when cloud computing is becoming mature. The following lists a few such characteristics:

a) **Statelessness and server failures are givens in the cloud**

As one of the major benefits of cloud computing is to reduce cost, so cloud service provider is likely to use more cost-effective hardware/software that are less robust and reliable than what people would purchase for in-house/local platforms. In particular, the underlying infrastructure is used to support various applications with different reliability and robustness demands, it may not be configured to target applications with high demands on reliability and robustness.

In the past 2-3 years, there have been many service outages in clouds, and some of them are well known and caused some major damages or at least significant inconvenience to end users. For example, When Google's Gmail faltered on Sept. 24, 2009, although it was down for less than a couple of hours, it was the second outage during the month, following a disturbing sequence of outages for Google's cloud-based offerings for search, news and other apps in the past 18 months. Various explanations have been provided, from routing errors to server maintenance issues. Another example is the big Twitter outage in early August 2009 that lasted throughout the morning and into early afternoon. While such outage may only cause some fume amount serious twitterers, Ebay's PayPal online payments system failed a couple of times in August 2009 and lasted for between 1 and 4.5 hours, leaving millions of customers unable to complete transactions. A network hardware issue was reported as the culprit for the outage. It costs PayPal millions of dollars in lost business; it's unclear how much it cost merchants. As CIO.com's Thomas Wailgum reported in January 2009, Salesforce.com suffered a service disruption for about an hour on Jan. 6 2009 due to a core network device failing because of memory allocation errors. The general public service providers are also not free from outages.

For example, Rackspace was forced to pay out between $2.5 million and $3.5 million in service credits to customers in the wake of a power outage that hit its Dallas data center in late June. Amazon S3 storage service was knocked out in summer 2008 and lasted and followed another outage earlier last year caused by too many authentication requests.

b) Lack of transparency and control (virtual vs. physical)
Different from local platforms, cloud computing involves imposing a layer of abstraction between the applications and physical machines/devices. Applications also have little control and even knowledge about the underlying physical platform and also other applications running sharing the same platform. As clouds are based on virtualization, applications need to be first virtualized before being moved to any of the cloud environments. Many assumptions or dependencies on the underlying physical systems need to be removed first.

c) Network, I/O contention with other applications
For in-house data grid deployments, getting a separate set of network cards and putting them on a dedicated VLAN or even their own switch is a really good idea, because of the broadcast traffic between the nodes. Application developers on a cloud may not have this option. To maximize utilization, cloud service provider may place many virtual machines on the same physical machine and may design a system architecture that groups significant amount of traffic going through a single file server, database machine or a load balancer. For example, so far there is no equivalent of a nice network attached shared storage on Amazon. In other words, cloud application developers may no longer be able to assume dedicated network channels and storage devices.

d) Less individualized platform support for reliability and robustness
Similar to lack of dedicated network, I/O devices, it is also less likely for cloud platforms to provide individualized guarantees for reliability and robustness. Although it is very possible some advanced, mature clouds may provide several levels of reliability support in the future, but it won't be fine-grained enough to match individual applications. Therefore

e) Elasticity
The main motivation and driver for cloud computing in the first place was to be able to grow as you need and pay for what you use, i.e. elasticity. Therefore, applications that can dynamically react to workload changes are good candidates for clouds since the cost of running on clouds is much lower than having to buy more hardware that stays idle most of the time until the peak loads come along. If you already have a good percentage of your workloads virtualized, then they are good candidates for

clouds. If you simply port the static images of existing applications to clouds, you are not taking advantage of cloud computing because your application will be over-provisioned based on the peak load and you'll end up with poorly utilized environment. Taking existing enterprise applications to the cloud can be very difficult simply because most of those applications were not designed to take advantage of the cloud's elasticity. If you, you will

f) **Large-scale, elasticity, distributed bugs**
Developing distributed applications are prone to bugs such as deadlocks, incorrect message ordering, etc, all of which are very hard to detect, test and debug. The new elasticity makes it even more challenging. Developers have to think dynamically to allocate/reclaim resources based on workloads. This can easily introduce bugs such as resource leaks or tangling links to reclaimed resources, etc. To address this problem, it would require software development tools to help testing and detect these types of bugs; or new application development model such as mapreduce that would relieve developers to worry about dynamically scaling up and down.

g) **Lack of development, execution, testing and diagnosis support**
Finally, one of the most severe but fortunately short-term challenges is lack of development, testing and diagnosis support on clouds. Most of today's enterprise applications were built using frameworks and technologies not yet supported as first class citizen by cloud providers. An application that works on a local platform may not work well on a cloud environment since local platforms are hard to test the interference from other applications and also the elasticity aspect of the application. Also when an application fails or suffers from some serious performance bottlenecks, due to the transparency of physical configuration/layout and other applications running on the same physical device/hardware, diagnosing and debugging the failure or performance issue can be a challenging task.

3. What are needed for cloud applications?

Cloud computing likely will bring transformational potential to IT industry. But such transformation can't happen overnight -- and it certainly can't happen without a plan. To development robust applications, both application developers and cloud platform providers need to make some effort.

For application developers, mindset adjustment is important. First, they need to understand the reason and the benefit of moving to clouds so they can evaluate whether their applications are best suited or at least have been revised properly to take advantage of the elasticity of clouds. Second, since each cloud platform may be different, it is important for application developers to understanding the platform's elasticity model and dynamic configuration method. It is also useful for them to stay abreast of the provider's evolving monitoring services and service level agreements, even potentially engaging the service provider as an on-going operations partner to make sure your applications' demands are addressed properly.

For cloud platform providers, the most important thing is to provide application developers a testing, deployment, execution, monitoring and diagnosis support. In particular, it would be useful if applications developers have a good local debugging environment. There is also a lack of good IDE's that can help with programming and debugging programs written for the cloud. The providers that do provide a local debug experience, do not simulate real cloud like conditions. Both from my personal experience and from conversations with other developers, I have come to realize that most people face problems when moving code from their local development servers to the actual cloud. This is only due to inconsistencies in the behavior of the local dev environment compared to the cloud.

We end the discussion with the cloud computing adoption model proposed by XXX, which provides an incremental, pragmatic approach to cloud computing. This is loosely modeled after the Capability Maturity Model (CMM) from the Software Engineering Institute (SEI) at Carnegie Mellon University, the Cloud Computing Adoption Model proposes five steps: (1)Virtualization: leveraging hypervisor-based infrastructure and application virtualization technologies for seamless portability of applications and shared server infrastructure. (2) Experimentation based on controlled and bounded deployments utilizing Amazon Elastic Compute Cloud (EC2) for compute capacity and as the reference architecture. (3) Foundations. Governance, controls, procedures, policies, and best practices begin to form around the development and deployment of cloud applications. (4) Advancement. Governance foundations allow organizations to scale up the volume of cloud applications through broad-based deployments in the cloud. (5) Actualization. Dynamic workload balancing across multiple utility clouds