Are Clouds Ready for Large Distributed Applications?

Kunwadee Sripanidkulchai, Sambit Sahu, Yaoping Ruan, Anees Shaikh, and Chitra Dorai
IBM T.J. Watson Research Center

ABSTRACT

Cloud computing carries the promise of providing powerful new models and abstractions that could transform the way IT services are delivered today. In order to establish the readiness of clouds to deliver meaningful enterprise-class IT services, we identify three key issues that ought to be addressed as first priority from the perspective of potential cloud users: how to deploy large-scale distributed services, how to deliver high availability services, and how to perform problem resolution on the cloud. We analyze multiple sources of publicly available data to establish cloud user expectations and compare against the current state of cloud offerings, with a focus on contrasting the different requirements from two classes of users – the individual and the enterprise. Through this process, our initial findings indicate that while clouds are ready to support usage scenarios for individual users, there are still rich areas of future research to be explored to enable clouds to support large distributed applications such as those found in enterprises.

1. INTRODUCTION

Cloud computing has recently gained significant attention from both industry and academia. While there are many definitions of cloud computing, they all embrace these key characteristics: the ability to deliver IT services at a lower barrier to entry in terms of cost, risk, and expertise, with higher flexibility and better scaling on-demand. Many cloud early adopters that started from scratch without any pre-existing infrastructure have had great successes in leveraging these capabilities to deliver services [13]. In addition, clouds have also had much success in delivering short-term uses and batch applications such as indexing the New York Times' newspaper archive [17]. These successes demonstrate powerful cloud capabilities that could be leveraged to deliver services much faster than any of these users could have achieved if they had to build out their own infrastructure. Despite these successes, some potential users particularly enterprise are grappling with if and how they can use the cloud to deliver their services.

In this paper, we take a step towards understanding and resolving some of these uncertainties. We examine "Cloud Readiness" – whether or not clouds are ready to deliver meaningful IT services from two user vantage points: *individual/small* customers and *large scale commercial enterprise* customers. Individual customers typically have simple IT services that leverage standard components such as LAMP stacks and perhaps three-tiered architectures. To contrast, enterprise customers have additional performance, availability and scalability requirements on top of these simple architectures and as a result have much more complex larger-scale distributed services. For example, enterprises leverage clustering to provide scalability, load balancing, hot stand-bys and disaster recovery mechanisms to provide availability, and messaging infrastructures to support integration across multiple applications and external third-party services. In addition, security, privacy and isolation guarantees as well as full-scale ITIL-compliance [14] for infrastructure and application monitoring, problem determination, and change management are required. While some of these requirements may be desirable for individual users if made affordable, they are absolutely mandatory for enterprise users.

The approach we take in this paper is to assess and understand the readiness of cloud computing. Rather than enumerate a broad set of requirements, we focus on three key requirements. Our selection of requirements is driven by prioritizing the most important questions from the perspective of potential cloud users: how to deploy large-scale distributed services on the cloud, how to deliver high availability services using clouds, and what to do when there are problems with services running on the cloud. We note that business requirements such as cost are a common concern, but pricing models have been extensively established in previous work [11] and are not further discussed in this paper.

In order to answer our three questions, we use publicly available data to establish the current state of the cloud and compare it against cloud user expectations. In addition, we discuss areas of future research to accelerate the adoption of clouds for all classes of users. Our goal is to raise and discuss these research directions rather than provide specific answers or final conclusions. These are wide open areas that deserve attention from the systems community. In § 2, we look at how to deploy large-scale distributed services on the cloud and analyze the deployment building blocks available from virtual appliance marketplaces. Service availability expectations are explored in § 3 using Website uptime monitoring data. In § 4, we mine discussions on cloud user forums to better understand the problem resolution processes on today's clouds. § 5 summarizes our findings and provides initial answers to our bigger question: are clouds ready for individuals and enterprises to use?

2. SERVICE DEPLOYMENT ON THE CLOUD

In this section, we study the readiness of clouds for deploying large-scale IT applications. First we examine the current state of the art in service deployment, particularly looking at common application building blocks or images ready to be deployed in the cloud. We then discuss future work that could make the cloud ready for large distributed applications.

2.1 Current Practice

The deployment steps in a cloud are similar to the tradi-

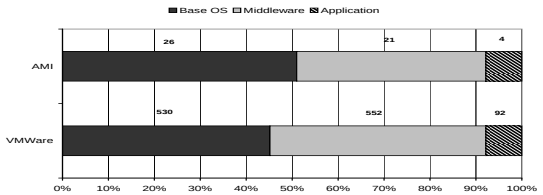

Figure 1: Amazon EC2 Public AMIs and VMWare Marketplace Virtual Appliance Categories on March 23, 2009.

tional non-cloud practices, usually including hardware and software procurement, base OS installation, middle-ware and application provisioning, customization and data loading. Clouds simplify these steps by providing automated provisioning. However, exactly which steps are automated and to what extent depends on the type of service abstraction used by the cloud provider. For example, infrastructure as a service providers like Amazon EC2 [1] automate procurement, base OS and some monolithic middle-ware provisioning, while platform as a service providers such as Google App Engine [3] include middle-ware by default. Going further up the stack, software as a service providers such as Salesforce [8] have applications ready for use. Depending on the type of user, the extent of the automation in each of these steps is also important. An enterprise user may want automated support for distributed multi-tier multi-node applications whereas an individual user may be satisfied with monolithic base image provisioning.

Applications are commonly deployed on a cloud using virtual appliances as building blocks. We study two prominent community sources, public EC2 Amazon Machine Images (AMI) and VMWare Virtual Appliance Marketplace [9], to understand what type of virtual appliances are available for deployment today. We took a snapshot of the images available on March 23, 2009 and categorized them based on their types. We find that most of the available images are for base OS and common applications as depicted in Figure 1. There are a total of 51 images offered by Amazon EC2. Based on image names, 51% are OS (Fedora core, Windows), and 41% are common middle-ware (Apache, TomCat, mySQL) or combination images such as Fedora core with Apache and mySQL (LAMP stack). In VMWare Marketplace, there are a total of 1105 virtual appliances. Using VMWare's categorization, base OS and middle-ware images also dominate similar to on EC2. While monolithic images satisfy many use cases particularly those required by individuals or small businesses, there is little automated support for putting together multiple images to form complex services.

In large-scale enterprise applications, multiple VMs are commonly deployed together to support a single application to address scalability, load balancing, and availability requirements. With growing interest of enterprise customers, we expect to see pressure on clouds to go beyond monolithic base image provisioning to support automated large distributed service provisioning. Therefore deploying such applications would need additional capabilities such as the ability to: (i) express relationships among these VMs denoting the dependencies at configuration time and at running time, (ii) compose complex deployment from single and already built set of VMs, and (iii) instantiate the deployment based on the above stated dependencies. The concept of provisioning a complex application based on single images has been commonly referred to as *Service Composition*. For example, RightScale [6] and 3Tera [5] provide provisioning capabilities beyond single VM deployment, specifically multi-tier image provisioning to support the composition of complex applications and customer data loading. While capabilities in this area are emerging in the right direction, it is still an open space to turn these capabilities into accepted common standards.

2.2 Challenges in Deploying Enterprise Applications

In this section, we look at two additional challenges that go beyond the above challenge in service composition. Particularly, we discuss one of the largest barriers for enterprises to transition their applications onto the cloud. In addition, once services are running on the cloud, we look at how to accommodate service changes by providing re-provisioning capabilities.

A major barrier for enterprise cloud adoption is how to transform an existing enterprise service deployment into a cloud-based deployment. Take an example of a multi-tiered web service deployment with extra components such as directory service and messaging service integration. Migrating such an application into the cloud presents a few technical challenges worth further exploration. First, prior to migration, a thorough understanding of the existing architecture is required, including what components are involved, their relationships, and how the configuration of such relationships are achieved. One would expect the application owner to know this information. In reality unfortunately, application owners may not keep track of all detailed information. Personnel changes also add to the complexity of keeping the architecture documentation up to date. Dependency graphs have been used in performance debugging for distributed systems [12, 10], however, dependencies for migration may need to be more definitive and may require discovering full configuration of all of relevant components. Second, migrating business-critical applications impose constraints on downtime as any downtime would cause a significant loss to the enterprise. Therefore, live migration for such an application is desirable. Though live migration in LAN environment has been a mature process, migrating into cloud from an existing data center may not necessarily meet this requirement. New technologies may need to be invented to overcome limitations in network bandwidth and latency as well as maintaining the liveness of client sessions.

Second, while provisioning is often thought of as a one-time event, re-provisioning to accommodate changes is a rather common operation. The complexity of enterprise applications usually evolves over time to accommodate new business requirements and new technologies. For example, a simple 3-tiered web deployment may change to a complex multi-tiered deployment when new components such as directory service for user verification and messaging service for collaborative questioning are added. Deploying such a system in a non-cloud environment usually requires manual dependency dis-

covery and manual configuration in each of the system components. However, once services are running on the cloud, such manual efforts are not in the spirit of cloud's automated provisioning capabilities. The challenge for cloud providers to support these evolving deployments is to enable simple ways to express service-level relationships and logic, support changes, and automatically re-provision or map services to a standard set of system-level components (i.e., base virtual machine images). This will likely require that cloud providers maintain a set of standard images that can be customized according to meta-data/configuration specified by cloud users. However, what qualifies as a standard base image, a customization, and a way to put together these base images remains an open area.

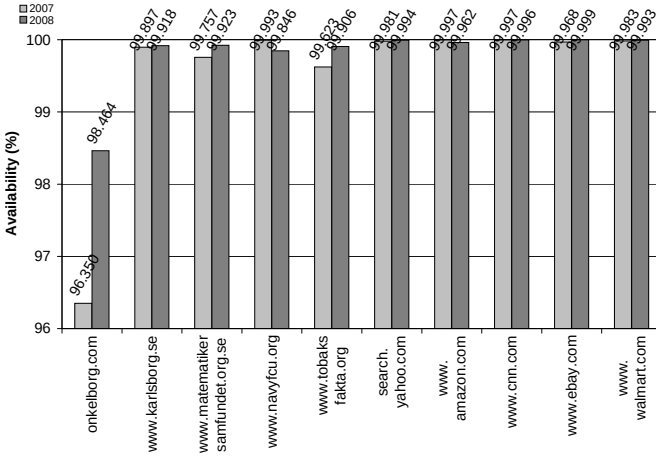

Figure 2: User expectations of service availability.

3. SERVICE AVAILABILITY ON THE CLOUD

A key concern for users when considering deploying services on clouds is the availability of their services. In this section, we look at cloud providers' service level agreement (SLA) and establish cloud users' availability expectations. Then we discussion future directions to enable cloud users to achieve higher availability.

3.1 Cloud SLA Coverage

In order to better understand the cloud provider's scope of responsibility, we map cloud SLAs to known classification of root causes of Internet service failures: human operator (i.e., configuration error), hardware, software, environmental failures, server node failures, and network failures [20]. EC2 exposes a virtual hardware abstraction to its users with an SLA that covers all hardware, network, and environmental failures. This leaves two causes of failures that need to be managed by the cloud user: operator node failures and software node failures. Projecting what would likely be covered in a future Google App Engine SLA based on its programming language API that does not allow users to directly control the infrastructure, Google would likely manage all causes of failures except for those made by the cloud user in developing the software running on the cloud. Putting this into context, if the three Internet services studied by Oppenheimer, et al. [20] were to be delivered on EC2, the percentage of failures that would have had to be resolved by the cloud user would have been significantly reduced from all failures to 24%-79% of the failures. Cloud users can take full advantage of cloud SLAs and focus on resolving a much smaller set of failures.

3.2 User Expectations of Availability

Given that many availability problems could be handled by cloud providers, we next examine whether or not the level of availability in today's cloud SLAs meet users' expectations. We first establish users' expectations by looking at the current state of service availability of two classes of potential cloud users: individual and enterprise users. We analyze monitoring data collected by Pingdom [7] that tracks Website performance from many different vantage points every monitoring interval (i.e., 5 minutes). For the average individual user, we randomly selected as representatives five sites that use 'org' in their names and have at least 2 years of monitoring data to avoid bias against any start up hiccups. For enterprise-class cloud users, we look at the availability data of five well-known Internet services that have large-scale multi-tier infrastructures: Amazon, CNN, Ebay, Walmart and Yahoo. We expect enterprise services to have similar high availability requirements to these enterprise Websites. We compute downtime based on the method defined in the EC2 SLA as the total of number of minutes the site is unavailable for events lasting longer than 5 minutes over a 1-year period. Shorter events are not considered as downtime. Note that Pingdom measures the end-to-end service availability, of which cloud SLAs only provide partial coverage and are at best an optimistic upper-bound to the overall end-to-end availability.

Figure 2 depicts the availability of each of these sites for the past two years 2007-2008. The first five sites are individual sites, having 96.350% to 99.993% availability with an average of 99.368% or over 55 hours of downtime in a year. The EC2 SLA provides higher availability at 99.95% or roughly 4 hours of downtime. If infrastructure problems are the main causes of failures, the average site could substantially benefit from a higher level of availability on the cloud than what they may have using their own infrastructure.

On the other hand, there is a significant gap in service availability between well-provisioned enterprise services and individual services. This is not surprising given that enterprises invest significant resources in managing their services. The availability of the last five enterprise-class services in Figure 2 range from 99.962% to 99.999%, with an average of 99.987% availability or only an hour downtime in a year. Enterprise users could see up to 4-times their typical downtime if they were to deploy their services on a cloud. This could be a *deterrent* for enterprise users from adopting the cloud until SLAs are a closer match to their current operating level.

3.3 New Directions for Higher Availability

In this section, we look at what would be needed to get cloud-based services up to and beyond the level of availability required by enterprise users, by discussing the need for new

and easy-to-use high availability solutions.

First, we explore the technical differences between individual vs. enterprise sites that result in the observed gap in service availability. The key difference is enterprises' extensive use of scaling architectures such as content delivery networks, HTTP front-ends, and load-balancing at various levels [16]. Cloud providers have taken large steps to reduce the barrier by providing ready-to-use building blocks such as content delivery networks [4], load balancing components [6, 5, 1], and automatic scaling [1] to all users. However, the key challenge is how to make the enterprise-level expertise to architect and put together those blocks equally widely accessible. For example, under which system conditions and workloads would a content delivery network be a superior architectural solution than automatic scaling? Or when does it make sense to use load balancing and what kind of load balancing? If one could encode such logic into a set of templates and rules based on system conditions to automatically leverage the appropriate architectural solution in a way that it could be easily reused by different cloud users for their services, we would be getting closer to commoditizing the delivery of highly available services on the cloud.

A second direction going beyond the scope of one cloud is to extend high availability architectures across different cloud providers. Compared to the first research direction discussed above, this direction is in its infancy both in terms of the technical components and the expertise. Individual cloud providers are likely to focus on increasing the performance and stability of their own clouds in order to provider higher levels of availability in their SLAs. However, cloud users would like the flexibility to take advantage of multiple providers to increase their service availability. For example, when one cloud has a problem, services can automatically fail-over to a different cloud without any impact on overall availability. The key challenge is to establish mechanisms and provide a common API or framework that would commoditize the construction of high availability services delivered across multiple clouds.

Lastly, an equally important research direction is to develop new virtualization technologies that support higher availability. Cloud providers have yet to take advantage of recent advances in virtualization technology that could be used to bypass problems in a manner that is transparent with little or no impact on service availability, eventually leading to providing higher SLAs. For example, live migration [19, 15] enables the movement of a running virtual machine to a different physical location. This capability can be applied to many of the problems reported on the EC2 cloud user forum to avoid down time. For example, today when cloud providers need to perform fixes or maintenance on a physical machine, they inform users to shutdown and reattach their instances to a different location. Instead of incurring service down time, cloud providers or users could use live migration to move the instance while still running. Another example is contention on a physical machine where one user's instance could be over-consuming shared resources [18]. Other users could migrate to different physical machines to avoid poor performance. While migrating to a destination in the same cloud is supported by today's technology, migrating to a different cloud is a rich open area. We believe that there are interesting design and architectural issues that could be further explored to enable capabilities such as live migration on the cloud, whether they are to be used by providers under the covers or by users through cloud APIs.

4. SERVICE PROBLEM RESOLUTION

Problem determination and resolution in IT environments has been a long-standing challenge. Cloud computing adds to the complexity because cloud providers are handling all of the infrastructure management tasks and giving users an abstraction that hides operational details from higher-level applications. Therefore, this limits users' visibility and capability to perform root cause analysis for applications when incidents happen. In this section, we look at the current state of the art and future areas of research to see through the layers of abstraction in order to perform problem resolution. We focus on EC2 as the primary data point as it has a strong user community that is creating early best practices in this area. We note that our discussion applies to both class of users, individuals or enterprises, as both require problem resolution capabilities.

4.1 Current Problem Resolution Practice

EC2 provides two options for problem resolution: premium paid support or free user discussion forums [2]. With premium service, users are able to access a Web-based ticketing system and get one-on-one support. Amazon also provides client-side diagnostic tools for premium users which collects information about the client environment. Output files of these tools can be share with Amazon's support team on a problem ticket. However, the actual problem determination practice still requires manual information sharing and collaborative debugging between users and Amazon. Moreover, cloud APIs lack automated report or query mechanisms that would allow users to get more information about their instances and their related hardware status. The most relevant EC2 API – the *ec2-describe-instances* call provides instance running state. The recently-introduced Amazon CloudWatch provides customers with monitoring metrics such as resource utilization and operational performance of individual instances. However, this information has limited use for problem resolution as it does not expose information about the instances' external dependencies on the shared infrastructure such as the physical server or the network.

In order to understand the problem resolution process, we mine all active discussion threads from the EC2 forum between April 1-7, 2009. We classify the 139 threads into one of three possible categories: Feature Request, How To, and Problem. Table 1 shows the number of threads in each category. Feature requests such as "instance type needed" account for 10% of the threads. More than half of the threads fall under HowTo/Info where known solutions or best practices are discussed. And about 34% of the threads fall under the Problem category. Each problem thread starts with a clear problem statement such as the symptoms listed in Table 2. Of the 47 problem threads, 5 had no follow-up resolution, 30 were user

errors resolved by suggestions from other users or the Amazon support team, and the remaining 12 problems were EC2 errors including hardware problems, system bugs, or outage ultimately solved by Amazon's support team.

Feature Request	HowTo/ Info	Problem		
		Cloud Err	User Err	Unknown
14(10%)	78 (56%)	12 (25%)	30 (64%)	5 (11%)

Table 1: Categories of EC2 forum threads in one week

In Table 2, we categorize the type of problem based on the manifestation of the symptom and present the top three areas: (i) problem with specific user instance, (ii) elastic block storage service that provides storage for the instance, and (iii) security problem. Next, we make several observations about the nature of the problems: when the problems happen, how the problem determination process works, and what additional data is needed to perform the diagnosis.

We see problems happen at multiple phases in the service management lifecycle. Several problems occur at the *service on-boarding stage* such as unable to launch instance, failure to attach EBS, or unable to login. In addition, there are failures during *day-to-day operations* as well such as instance unreachable, instance died, or authorization problem. Lastly, we see problems with deleting or shutting down instances at the *termination phase* which are unique problems for clouds. Note that users are charged based on usage, therefore, supporting clean deletion is a requirement.

We also make observations about the problem resolution process. First, similar symptoms such as failure to launch instance could be manifestations of different root causes that could be on the user side (wrong key file used) or the cloud side (EC2's configuration service failed). Without sufficient *visibility* on both ends, problem determination is left to users and cloud providers to request additional information from each other back-and-forth with some amount of trial and error until a root cause is determined. As an example, we walk through one thread reporting that an EBS volume is failing to attach to an instance. The thread starts with the problem description by one user and evolves into a 23-posting discussion involving 9 users, 5 of which observed similar problems, and 2 members of the EC2 support team. The EC2 team requested additional information from users such as commands that were used to attach the EBS volumes, specific volume names, and their locations. From that, we speculate that they used that information to guide their examination of the internal EC2 logs. They eventually identified that the problem was localized to a subset of compute hardware and advised users to launch new instances as a work-around.

4.2 Peeling Through the Layers of Abstractions

Given that the risk is shared between cloud providers and users, it is desirable to be able to determine clear ownership of responsibility when problems occur. Next, we discuss potential research areas that could help to ensure ownership and provide stream-lined support for problem resolution.

First, we discuss what information is needed for problem resolution. At the most basic level, an answer to the question

Problem	Symptom	Root cause / Solution
Instance problems		
Shut down	• Could not shut down, status hangs at "shutting down"	• Hardware problem • Reason unknown
Inaccessible	• Instance down unreachable	• Access point bug • Reason unknown
Died	• Instance died, up after reboot	• Reason unknown
Launching	• Console hangs at "running local boot scripts..." • Status shows "running" but no console output	• Used wrong key file • ec2config failed • Reason unknown
Elastic Block Storage(EBS) problems		
Attach	• Fail to attach but status shows available	• EBS software bug
Delete	• Cannot delete, status is "deleting" for hours	• Connection problem • Communication delay • Dead instance
Snapshot	• Pending for hours	• Status not updated correctly without cause
Security problems		
Authorization	• All ports opened	• Reason unknown
Login	• Unable to login	• Reason unknown

Table 2: Description and solution for top problems.

"is it my problem or is it your problem?" would greatly simplify the process for both parties. In fact, for 64% of the reported problems which are user errors, cloud providers could even skip participation in those discussions if users could themselves verify that the pieces of the cloud infrastructure that are running the users' instances are operating without incident. In addition, this verification makes it faster for users to do problem determination. More detailed information is useful particularly when there may be underlying problems with the infrastructure. However, the challenge is that the information has to come from both the user and the provider. In almost all of the discussions, user activity logs were useful in providing specific information about user operations that did not succeed. In addition, the topology of the users' services which could span multiple instances would also be needed in order to map to the underlying infrastructure. From the provider side, status information of the infrastructure associated with the user's services is also critical. This information may include hardware status, network capacity, connectivity, etc.

Another important issue is to develop mechanisms to enable sharing of such information between the multiple parties. The information in its raw state may be sensitive. Exploring how to cleanse that information such that it could be appropriately shared is an interesting area of future research. Another possible direction is to develop new mechanisms or APIs to enable cloud users to directly and automatically query and correlate application level events with lower level hardware information to better identify the root cause of the problem.

5. SUMMARY

In this paper, we have looked at answering three pressing issues that are important to cloud users to determine if and how they will embrace clouds as the new delivery platform for their IT needs: service deployment, service availability, and service problem resolution. We use publicly available data, and look at these issues from the perspective of two very different types

of potential cloud users – individuals and enterprises.

We find that current deployment mechanisms focus on monolithic systems, with some initial support for some enterprise-style application underway. Future work for better support of large-scale distributed architecture including migration of legacy applications onto the cloud, and more flexible composition of basic system components especially when re-provisioning of an existing system is needed.

Service availability is also a developing area as clouds are beginning to offer SLAs to its users. For individual users, the availability levels in the SLAs may already be reasonably sufficient. However, raising the availability levels is crucial to attract enterprise use. Exploring mechanisms to enable higher levels of availability such as advanced virtualization/live migration capabilities could be beneficial to both classes of users.

Lastly, for problem resolution, we find that the current manual process both classes of users faces scaling challenges. It is critical for cloud users to obtain enough *visibility* into the cloud infrastructure to identify the root cause of problems. We hope to see providers offer more automated support mechanisms to facilitate better problem determination, and in turn spur larger-scale adoption of cloud computing.

6. REFERENCES

[1] Amazon Elastic Compute Cloud (Amazon EC2). http://www.amazon.com/gp/browse.html?node=201590011, July 2008.

[2] Forum: Amazon Elastic Compute Cloud. http://developer.amazonwebservices.com/connect/forum.jspa?forumID=30, April 2008.

[3] Google App Engine. http://code.google.com/appengine/, July 2008.

[4] Amazon CloudFront. http://aws.amazon.com/cloudfront/, April 2009.

[5] Cloud Computing For Web Applications — 3tera. http://www.3tera.com, April 2009.

[6] Cloud Computing Management Platform by RightScale. http://www.rightscale.com, April 2009.

[7] Pingdom. http://www.pingdom.com, March 2009.

[8] Platform as a Service (Paas) - Powering On-Demand SaaS Development. http://www.sales.com, April 2009.

[9] Virtual Appliance Marketplace. http://www.vmware.com/appliances/, 2009.

[10] Marcos K. Aguilera, Jeffrey C. Mogul, Janet L. Wiener, Patrick Reynolds, and Athicha Muthitacharoen. Performance debugging for distributed systems of black boxes. In *SOSP '03*, pages 74–89, New York, NY, USA, 2003. ACM.

[11] Michael Armbrust, Armando Fox, Rean Griffith, and et. al. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[12] Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula, David A. Maltz, and Ming Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. In *SIGCOMM '07*, pages 13–24, New York, NY, USA, 2007. ACM.

[13] "Amazon Web Services Blog". Animoto - Scaling Through Viral Growth. http://aws.typepad.com/aws/2008/04/animoto—scali.html, April 2008.

[14] David Cannon and David Wheeldon. *Service Operation ITIL, Version 3*. The Stationery Office, 2007.

[15] Christopher Clark, Keir Fraser, Steven Hand, and et. al. Live migration of virtual machines. In *NSDI'05*, pages 273–286, Berkeley, CA, USA, 2005. USENIX Association.

[16] Jeremy Elson and Jon Howell. Handling Flash Crowds from your Garage. In *USENIX'08*, June 2008.

[17] Derek Gottfrid. The New York Times Archives + Amazon Web Services = TimesMachine. http://open.blogs.nytimes.com/2008/05/21/the-new-york-times-archives-amazon-web- services-timesmachine/, May 2008.

[18] Oren Michels. The amazon ec2 outage no one noticed. http://oren.blogs.com/praxis/2008/04/the-amazon-ec2.html, April 2008.

[19] Michael Nelson, Beng-Hong Lim, and Greg Hutchins. Fast transparent migration for virtual machines. In *USENIX'05*, Berkeley, CA, USA, 2005. USENIX Association.

[20] David Oppenheimer, Archana Ganapathi, and David A. Patterson. Why do internet services fail, and what can be done about it? In *4th USITS*, March 2003.