1

Introduction

Roy H. Campbell

Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA

COPYRIGHTED MATERIAL

Mission assurance for critical cloud applications is of growing importance to governments and military organizations, yet mission-critical cloud computing may face the challenge of needing to use hybrid (public, private, and/or heterogeneous) clouds and require the realization of "end-to-end" and "cross-layered" security, dependability, and timeliness. In this book, we consider cloud applications in which assigned tasks or duties are performed in accordance with an intended purpose or plan in order to accomplish an *assured* mission.

1.1 Introduction

Rapid technological advancements in global networking, commercial off-the-shelf technology, security, agility, scalability, reliability, and mobility created a window of opportunity in 2009 for reducing the costs of computation and led to the development of what is now known as *cloud computing* [1–3]. Later, in 2010, the Obama Administration [4] announced an

> "extensive adoption of cloud computing in the federal government to improve information technology (IT) efficiency, reduce costs, and provide a standard platform for delivering government services. In a cloud computing environment, IT resources—services, applications, storage devices and servers, for example—are pooled and managed centrally. These resources can be provisioned and made available on demand via the Internet. The cloud model strengthens the resiliency of mission-critical applications by removing dependency on underlying hardware. Applications can be easily moved from one system to another in the event of system failures or cyber attacks" [5].

In the same year, the Air Force signed an initial contract with IBM to build a mission-assured cloud computing capability [5].

Assured Cloud Computing, First Edition. Edited by Roy H. Campbell, Charles A. Kamhoua, and Kevin A. Kwiat.
© 2018 the IEEE Computer Society, Inc. Published 2018 by John Wiley & Sons, Inc.

Table 1.1 Model of cloud computing.

Service Models	*Deployment Models*			
Software as a Service	**Private Cloud** **Community Cloud** **Hybrid Cloud** **Public Cloud**			
Platform as a Service				
Infrastructure as a Service				
Essential Characteristics	Resource Pooling	Rapid Elasticity	Measured Service	Broad Network Access

Cloud computing was eventually defined by the National Institute of Standards and Technology (as finalized in 2011) as follows [6]: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models." That model of cloud computing is depicted in Table 1.1.

One of the economic reasons for the success of cloud computing has been the scalability of the computational resources that it provides to an organization. Instead of requiring users to size a planned computation exactly (e.g., in terms of the number of needed Web servers, file systems, databases, or compute engines), cloud computing allows the computation to scale easily in a time-dependent way. Thus, if a service has high demand, it can be replicated to make it more available. Instead of having two Web servers provide a mission-critical service, the system might allow five more Web servers to be added to the service to increase its availability. Likewise, if demand for a service drops, the resources it uses can be released, and thus be freed up to be used for other worthwhile computation. This flexible approach allows a cloud to economically support a number of organizations at the same time, thereby lowering the costs of cloud computation. In later chapters, we will discuss scaling performance and how to assure the correctness of a mission-oriented cloud computation as it changes in size, especially when the scaling occurs dynamically (i.e., is *elastic*).

1.1.1 Mission-Critical Cloud Solutions for the Military

As government organizations began to adopt cloud computing, security, availability, and robustness became growing concerns; there was a desire to use cloud computing even in mission-critical contexts, where a *mission-critical system* is one that is essential to the survival of an organization. In 2010, in response to

military recognition of the inadequacy of the then state-of-the-art technologies, IBM was awarded an Air Force contract to build a secure cloud computing infrastructure capable of supporting defense and intelligence networks [5]. However, the need for cloud computing systems that could support missions involved more numerous major concerns than could easily be solved in a single, focused initiative and, in particular, raised the question of how to assure cloud support for mission-oriented computations—the subject of this book. Mission-critical cloud computing can stretch across private, community, hybrid, and public clouds, requiring the realization of "end-to-end" and "cross-layered" security, dependability, and timeliness. That is, cloud computations and computing systems should survive malicious attacks and accidental failures, should be secure, and should execute in a timely manner, despite the heterogeneous ownership and nature of the hardware components.

End-to-end implies that the properties should hold throughout the lifetime of individual events, for example, a packet transit or a session between two machines, and that they should be assured in a manner that is independent of the environment through which such events pass. Similarly, *cross-layer* encompasses multiple layers, from the end device through the network and up to the applications or computations in the cloud. A survivable and distributed cloud-computing-based infrastructure requires the configuration and management of dynamic systems-of-systems with both trusted and partially trusted resources (including data, sensors, networks, computers, etc.) and services sourced from multiple organizations. For mission-critical computations and workflows that rely on such dynamically configured systems-of-systems, we must ensure that a given configuration doesn't violate any security or reliability requirements. Furthermore, we should be able to model the trustworthiness of a workflow or computation's completion for a given configuration in order to specify the right configuration for high assurance.

Rapid technological advances and computer-based weapons systems have created the need for net-centric military superiority. Overseas commitments and operations stretch net-centricity with global networking requirements, use of government and commercial off-the-shelf technology, and the need for agility, mobility, and secure computing over a mixture of blue and gray networks. (*Blue* networks are military networks that are considered secure, while *gray* networks are those in private hands, or run by other nations, that may not be secure.) An important goal is to ensure the confidentiality and integrity of data and communications needed to get missions done, even amid cyberattacks and failures.

1.2 Overview of the Book

This book encompasses the topics of architecture, design, testing, and formal verification for assured cloud computing. The authors propose approaches for

using formal methods to analyze, reason, prototype, and evaluate the architectures, designs, and performance of secure, timely, fault-tolerant, mission-oriented cloud computing. They examine a wide range of necessary assured cloud computing components and many urgent concerns of these systems.

The chapters of this book provide research overviews of (1) flexible and dynamic distributed cloud-computing-based architectures that are survivable; (2) novel security primitives, protocols, and mechanisms to secure and support assured computations; (3) algorithms and techniques to enhance end-to-end timeliness of computations; (4) algorithms that detect security policy or reliability requirement violations in a given configuration; (5) algorithms that dynamically configure resources for a given workflow based on security policy and reliability requirements; and (6) algorithms, models, and tools to estimate the probability of completion of a workflow for a given configuration. Further, we discuss how formal methods can be used to analyze designed architectures, algorithms, protocols, and techniques to verify the properties they enable. Prototypes and implementations may be built, formally verified against specifications, and tested as components in real systems, and their performance can be evaluated.

While our research has spanned most of the cloud computing phenomenon's lifetime to date, it has had, like all fast-moving technological advances, only a short history (starting 2011). Much work is still to be done as cloud computing evolves and "mission-critical" takes on new meanings within the modern world. Wherever possible, throughout the volume (and in the concluding chapter) we have offered reflections on the state of the art and commented on future directions.

- Chapter 2**: Survivability: Design, Formal Modeling, and Validation of Cloud Storage Systems Using Maude**, José Meseguer in collaboration with Rakesh Bobba, Jon Grov, Indranil Gupta, Si Liu, Peter Csaba Ölveczky, and Stephen Skeirik

  To deal with large amounts of data while offering high availability and throughput and low latency, cloud computing systems rely on distributed, partitioned, and replicated data stores. Such cloud storage systems are complex software artifacts that are very hard to design and analyze. We argue that formal specification and model checking analysis should significantly improve their design and validation. In particular, we propose rewriting logic and its accompanying Maude tools as a suitable framework for formally specifying and analyzing both the correctness and the performance of cloud storage systems. This chapter largely focuses on how we have used rewriting logic to model and analyze industrial cloud storage systems such as Google's Megastore, Apache Cassandra, Apache ZooKeeper, and RAMP. We also touch on the use of formal methods at Amazon Web Services. Cloud computing relies on software systems that store large amounts of data

correctly and efficiently. These cloud systems are expected to achieve high performance (defined as high availability and throughput) and low latency. Such performance needs to be assured even in the presence of congestion in parts of the network, system or network faults, and scheduled hardware and software upgrades. To achieve this, the data must be replicated both across the servers within a site and across geo-distributed sites. To achieve the expected scalability and elasticity of cloud systems, the data may need to be partitioned. However, the CAP theorem states that it is impossible to have both high availability and strong consistency (correctness) in replicated data stores in today's Internet.

Different storage systems therefore offer different trade-offs between the levels of availability and consistency that they provide. For example, weak notions of consistency of multiple replicas, such as "eventual consistency," are acceptable for applications (such as social networks and search) for which availability and efficiency are key requirements, but for which it would be tolerable if different replicas stored somewhat different versions of the data. Other cloud applications, including online commerce and medical information systems, require stronger consistency guarantees.

The key challenge addressed in this chapter is that of how to design cloud storage systems with high assurance such that they satisfy desired correctness, performance, and quality of service requirements.

- Chapter 3**: Risks and Benefits: Game-Theoretical Analysis and Algorithm for Virtual Machine Security Management in the Cloud**, Luke A. Kwiat in collaboration with Charles A. Kamhoua, Kevin A. Kwiat, and Jian Tang

  Many organizations have been inspired to move to the cloud the services they depend upon and offer because of the potential for cost savings, ease of access, availability, scalability, and elasticity. However, moving services into a multitenancy environment raises many difficult problems. This chapter uses a game-theoretic approach to take a hard look at those problems. It contains a broad overview of the ways game theory can contribute to cloud computing. Then it turns to the more specific question of security and risk. Focusing on the virtual machine technology that supports many cloud implementations, the chapter delves into the security issues involved when one organization using a cloud may impact other organizations that are using that same cloud. The chapter provides an interesting insight that a cloud and its multiple tenants represent many different opportunities for attackers and asks some difficult questions: To what extent, independent of the technology used, does multitenancy create security problems, and to what extent, based on a "one among many" argument, does it help security? In general, what, mathematically, can one say about multitenancy clouds and security? It is interesting to note that it may be advantageous for cloud applications that have the same levels of security and risk to be clustered together on the same machines.

- Chapter 4**: Detection and Security: Achieving Resiliency by Dynamic and Passive System Monitoring and Smart Access Control**, Zbigniew Kalbarczyk in collaboration with Rakesh Bobba, Domenico Cotroneo, Fei Deng, Zachary Estrada, Jingwei Huang, Jun Ho Huh, Ravishankar K. Iyer, David M. Nicol, Cuong Pham, Antonio Pecchia, Aashish Sharma, Gary Wang, and Lok Yan

  System reliability and security is a well-researched topic that has implications for the difficult problem of cloud computing resiliency. Resiliency is described as an interdisciplinary effort involving monitoring, detection, security, recovery from failures, human factors, and availability. Factors of concern include design, assessment, delivery of critical services, and interdependence among systems. None of these are simple matters, even in a static system. However, cloud computing can be very dynamic (to manage elasticity concerns, for example), and this raises issues of situational awareness, active and passive monitoring, automated reasoning, coordination of monitoring and system activities (especially when there are accidental failures or malicious attacks), and use of access control to modify the attack surface. Because use of virtual machines is a significant aspect of reducing costs from shared resources, the chapter features virtualization resilience issues. One practical topic focused on is that of whether hook-based monitoring technology has a place in instrumenting virtual machines and hypervisors with probes to report anomalies and attacks. If one creates a strategy for hypervisor monitoring that takes into account the correct behavior of guest operating systems, then it is possible to construct a "return-to-user" attack detector and a process-based "key logger," for example. However, even with such monitoring in place, attacks can still occur by means of hypervisor introspection and cross-VM side-channels. A number of solutions from the literature, together with the hook-based approach, are reviewed, and partial solutions are offered.

  On the user factors side of attacks, a study of data on credential-stealing incidents at the National Center for Supercomputing Applications revealed that a threshold for correlated events related to intrusion can eliminate many false positives while still identifying compromised users. The authors pursue that approach by using Bayesian networks with event data to estimate the likelihood that there is a compromised user. In the example data evaluated, this approach proved to be very effective. Developing the notion that stronger and more precise access controls would allow for better incident analysis and fewer false positives, the researchers combine attribute-based access control (ABAC) and role-based access control (RBAC). The scheme describes a flexible RBAC model based on ABAC to allow more formal analysis of roles and policies.
- Chapter 5**: Scalability, Workloads, and Performance: Replication, Popularity, Modeling, and Geo-Distributed File Stores**, Roy H. Campbell in collaboration with Shadi A. Noghabi and Cristina L. Abad

  Scalability allows a cloud application to change in size, volume, or geographical distribution while meeting the needs of the cloud customer. A

practical approach to scaling cloud applications is to improve the availability of the application by replicating the resources and files used, including creating multiple copies of the application across many nodes in the cloud. Replication improves availability through redundant resources, services, networks, file systems, and nodes but also creates problems with respect to whether clients observe consistency as they are served from the multiple copies. Variability in data sizes, volumes, and the homogeneity and performance of the cloud components (disks, memory, networks, and processors) can impact scalability. Evaluating scalability is difficult, especially when there is a large degree of variability. This leads one to estimate how applications will scale on clouds based on probabilistic estimates of job load and performance. Scaling can have many different dimensions and properties. The emergence of low-latency worldwide services and the desire to have higher fault tolerance and reliability have led to the design of geo-distributed storage with replicas in multiple locations. Scalability in terms of global information systems implemented on the cloud is also geo-distributed. We consider, as a case example, scalable geo-distributed storage.

- Chapter 6**: Resource Management: Performance Assuredness in Distributed Cloud Computing via Online Reconfigurations**, Indranil Gupta in collaboration with Mainak Ghosh and Le Xu

  Building systems that perform *predictably* in the cloud remains one of the biggest challenges today, both in mission-critical scenarios and in non-real-time scenarios. Many cloud infrastructures do not easily support, in an assured manner, reconfiguration operations such as changing of the shard key in a sharded storage/database system, or scaling up (or down) of the number of VMs being used in a stream or batch processing system. We discuss *online* reconfiguration operations whereby the system does not need to be shut down and the user/client-perceived behavior is indistinguishable regardless of whether a reconfiguration is occurring in the background, that is, the performance continues to be assured in spite of ongoing background reconfiguration. We describe ways to scale-out and scale-in (increase or decrease) the number of machines/VMs in cloud computing frameworks, such as distributed stream processing and distributed graph processing systems, again while offering assured performance to the customer in spite of the reconfigurations occurring in the background. The ultimate performance assuredness is the ability to support SLAs/SLOs (service-level agreements/objectives) such as deadlines. We present a new real-time scheduler that supports priorities and hard deadlines for Hadoop jobs.

  This chapter describes multiple contributions toward solution of key issues in this area. After a review of the literature, it provides an overview of five systems that were created in the Assured Cloud Computing Center that are oriented toward offering performance assuredness in cloud computing frameworks, even while the system is under change:

1) Morphus (based on MongoDB), which supports reconfigurations in sharded distributed NoSQL databases/storage systems.
2) Parqua (based on Cassandra), which supports reconfigurations in distributed ring-based key-value stores.
3) Stela (based on Storm), which supports scale-out/scale-in in distributed stream processing systems.
4) A system (based on LFGraph) to support scale-out/scale-in in distributed graph processing systems.
5) Natjam (based on Hadoop), which supports priorities and deadlines for jobs in batch processing systems.

We describe each system's motivations, design, and implementation, and present experimental results.

- Chapter 7**: Theoretical Considerations: Inferring and Enforcing Use Patterns for Mobile Cloud Assurance**, Gul Agha in collaboration with Minas Charalambides, Kirill Mechitov, Karl Palmskog, Atul Sandur, and Reza Shiftehfar

  The mobile cloud combines cloud computing, mobile computing, smart sensors, and wireless networks into well-integrated ecosystems. It offers unrestricted functionality, storage, and mobility to serve a multitude of mobile devices anywhere, anytime. This chapter shows how support for fine-grained mobility can improve mobile cloud security and trust while maintaining the benefits of efficiency. Specifically, we discuss an actor-based programming framework that can facilitate the development of mobile cloud systems and improve efficiency while enforcing security and privacy. There are two key ideas. First, by supporting fine-grained units of computation (actors), a mobile cloud can be agile in migrating components. Such migration is done in response to a system context (including dynamic variables such as available bandwidth, processing power, and energy) while respecting constraints on information containment boundaries. Second, through specification of constraints on interaction patterns, it is possible to observe information flow between actors and flag or prevent suspicious activity.
- Chapter 8**: Certifications Past and Future: A Future Model for Assigning Certifications that Incorporate Lessons Learned from Past Practices**, Masooda Bashir in collaboration with Carlo Di Giulio and Charles A. Kamhoua

  This chapter describes the evolution of three security standards used for cloud computing and the improvements made to them over time to cope with new threats. It also examines their adequacy and completeness by comparing them to each other. Understanding their evolution, resilience, and adequacy sheds light on their weaknesses and thus suggests improvements needed to keep pace with technological innovation. The three security certifications reviewed are as follows:

1) ISO/IEC 27001, produced by the International Organization for Standardization and the International Electrotechnical Commission to address the building and maintenance of information security management systems.
2) SOC 2, the Service Organization Control audits produced by the American Institute of Certified Public Accountants (AICPA), which has controls relevant to confidentiality, integrity, availability, security, and privacy within a service organization.
3) FedRAMP, the Federal Risk and Authorization Management Program, created in 2011 to meet the specific needs of the U.S. government in migrating its data on cloud environments.

References

1 "Cloud computing: Clash of the clouds," *The Economist*, October 15, 2009. Available at http://www.economist.com/node/14637206. (accessed November 3, 2009).

2 "Gartner says cloud computing will be as influential as e-business" (press release), Gartner, Inc., June 26, 2008. Available at http://www.gartner.com/newsroom/id/707508 (accessed August 22, 2010).

3 Knorr, E., and Gruman, G.,"What cloud computing really means," *ComputerWorld*, April 8, 2008. Available at https://www.computerworld.com.au/article/211423/what_cloud_computing_really_means/ (accessed June 2, 2009).

4 Obama, B.,"Executive order 13571: Streamlining service delivery and improving customer service," Office of the Press Secretary, the White House, April 27, 2011. Available at https://obamawhitehouse.archives.gov/the-press-office/2011/04/27/executive-order-13571-streamlining-service-delivery-and-improving-custom.

5 "U.S. Air Force selects IBM to design and demonstrate mission-oriented cloud architecture for cyber security" (press release), IBM, February 4, 2010. Available at https://www-03.ibm.com/press/us/en/pressrelease/29326.wss.

6 Mell, P. and Grance, T.,"The NIST definition of cloud computing: recommendations of the National Institute of Standards and Technology," Special Publication 800-145, National Institute of Standards and Technology, U.S. Department of Commerce, September 2011. Available at https://csrc.nist.gov/publications/detail/sp/800-145/final.