J Internet Serv Appl (2012) 3:87–94
DOI 10.1007/s13174-011-0044-9

SI: FOME - THE FUTURE OF MIDDLEWARE

A middleware for assured clouds

Roy H. Campbell · Mirko Montanari · Reza Farivar

Received: 2 November 2011 / Accepted: 9 November 2011 / Published online: 29 November 2011

Abstract This paper considers mission assurance for critical cloud applications, a set of applications with growing importance to governments and military organizations. Specifically, we consider applications in which assigned tasks or duties are performed in accordance with an intended purpose or plan in order to accomplish an assured mission. Mission-critical cloud computing may possibly involve hybrid (public, private, heterogeneous) clouds and require the realization of "end-to-end" and "cross-layered" security, dependability, and timeliness. We propose the properties and building blocks of a middleware for assured cloud computing that can support critical missions. In this approach, we assume that mission critical cloud computing must be designed with assurance in mind. In particular, the middleware in such systems must include sophisticated monitoring, assessment of policies, and response to manage the configuration and management of dynamic systems-of-systems with both trusted and partially trusted resources (data, sensors, networks, computers, etc.) and services sourced from multiple organizations.

Keywords Cloud computing · Mission assurance · Security · Middleware · Monitoring

1 Introduction

Rapid technological advancements, global networking, commercial off-the-shelf technology, security, agility, scalability, reliability, and mobility create a window of opportunity for reducing the costs of computation. But mission-critical cloud computing across hybrid (public, private, heterogeneous) clouds requires the realization of "end-to-end" and "cross-layered" security, dependability, and timeliness. That is, computations and computing systems should survive malicious attacks and accidental failures; they should be secure, and they should execute in a timely manner. End-to-end implies that the properties should hold throughout the lifetime of individual events, e.g., 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 at the data center. 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 (data, sensors, networks, computers, etc.) and services sourced from multiple organizations. To assure mission-critical computations and workflows that rely on such dynamically configured systems-of-systems, we must ensure that a given configuration does not 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 assurances.

This paper discusses the architecture and design for middleware platforms to support assured cloud computing. We describe our implementation of such a middleware centered on policy-based event monitoring and dynamic reactions, and we highlight the other important research areas whose development is fundamental for creating assured cloud computing systems.

R.H. Campbell (✉) · M. Montanari · R. Farivar
Thomas M. Siebel Center for Computer Science, University of Illinois, MC258 201 N. Goodwin Avenue, Urbana, IL 61801-2302, USA
e-mail: rhc@illinois.edu

2 Assured cloud computing

The paradigm of cloud computing is changing the way in which organizations use storage and computational resources to support their activities. Cloud computing solutions have been introduced for data analysis [10, 15], for large-scale distributed storage, for running high-traffic websites, and for high-performance computing (HPC) [32]. The ability of scaling, the limited capital investment, and economy of scale are continuing to increase the types of applications that run on cloud computing resources. A recent outage of a small part of Amazon Web Services (AWS) showed how several small and large organizations rely on a such infrastructure to provide services to their customers [26].

Interactions between cloud users and cloud providers are regulated by agreements. Cloud users specify the amount of resources that they require, and cloud providers agree to provide a minimum level of quality of services when giving access to such resources. These agreements are represented by Service Level Agreements (SLAs) and, more recently, by regulatory compliance [3]. SLAs generally use metrics such as availability, response time, and error-rate to define the level of service. Regulatory compliance, on the other hand, is a process ensuring that cloud providers implement a minimal level of security by implementing specific security processes and security configurations on their infrastructure. Security-focused cloud solutions are already being introduced in the market [21].

However, the guarantees provided by SLAs and compliance are defined in general for entire classes of services. Many modern applications require strong guarantees on the ability of cloud systems to provide reliable and secure services with requirements that are dynamic and might change for each service call. For example, services such as critical communication systems [7], real-time image analysis for Unmanned Air Vehicles (UAVs) and stock exchanges require cloud systems that have strict real-time, availability, and security requirements and need to be negotiated and guaranteed for the duration of the service. The Air Force defines such strict requirements as the base for "mission assurance." DoD Directive 3020.40 [11] defines Mission Assurance (MA) as "a process to ensure that assigned tasks or duties can be performed in accordance with the intended purpose or plan."

A cloud computing solution able to provide mission assurance [16] enables a wide range of applications that current cloud systems are not yet able to support. To provide the necessary guarantees, a mission-assured cloud computing solution requires processes that ensure continued performance in face of real world security threats, and compute real-time risk assessments. It should then utilize this real-time insight within the fabrics of the cloud to ensure continued availability of services at satisfactory service level agreements.

Services in the cloud are provided through the composition of several independent applications. A middleware solution would provide the proper support for the dynamic reorganization and monitoring of these applications to ensure meeting the minimum requirements. Given the request for a service, the sequence of interactions that produce the response is organized in a workflow. Each request provides a specific SLA and a set of security properties. The goal of a middleware for assured cloud computing is to compose this information and guarantee that the overall system is able to meet the mission requirements even in the presence of faults and of security problems.

We identify three properties that assured cloud computing needs to provide: security, availability, and real-time guarantees. Research has been performed in each individual area, but a complete integration of these techniques in an overall framework is still missing.

Security: The critical nature of mission-oriented computing requires assurances on the protection of the system from malicious users. Several thread models can be considered when looking at security in the cloud. Previous work analyses the security guarantees that can be provided when the cloud provider itself is untrusted. Techniques such as the proof of storage allow verifying that cloud providers are storing data correctly [18], and that the minimal level of redundancy in the storage is guaranteed [8]. Other work introduces mechanisms for preserving the confidentiality of data. For example, data can be separated in private data and public data. Public data can be processed in a public cloud, while private data are processed in a private cloud [33]. Cryptography, with homomorphic encryption, provides the theoretical ability of preserving confidentiality during data processing by allowing computation over encrypted data [14]. Other techniques consider the case in which the cloud provider is trusted, but other users are potentially malicious. Techniques have been introduced to increase the isolation between Virtual Machines running on the same host [29]. Other techniques are used for increasing the security of VMs by monitoring the operation of systems to detect compromises [13].

A middleware managing assured cloud applications should be able to combine these techniques to provide the required level of confidentiality and integrity to the service. For example, the security requirements of a request could mandate the confidentiality of the data from cloud providers. In such a case, the middleware should use encryption over the data. In another example, a cloud user might request the creation of a virtual machine for providing a critical service. Such a virtual machine should be subject to detailed analysis by the cloud provider to ensure that it has not been

compromised. The middleware should be able to estimate the performance tradeoffs between applying such additional security protections and the real-time characteristics of the system.

Availability: Faults are inevitable in large systems. Hence, guaranteeing availability in the presence of faults is a critical property for assured cloud systems. Faults can be simple such as the failure of a hard drive, or can arise from complex and unexpected interactions between services. Replication is a solution for providing the required level of availability, however, naive replication strategies might not be sufficient for surviving complex faults. For example, Amazon's postmortem descriptions of their availability problems provide us with interesting insights on the causes of large-scale outages. In 2008, a single bit error in a gossip message caused a large-scale availability problem in S3 [2]. More recently, a network configuration change triggered a large and unnecessary data replication that caused the EBS service to become unavailable. The problem cascaded and affected Amazon's Computation Cloud (EC2) and their Relational Database Service (RDS) [4]. Since preventing such issues is challenging, an assured cloud computing middleware needs to provide fault tolerance mechanisms for reducing the effects of such problems, for example, by isolating faults or providing checkpointing support.

Real-time: A wide range of applications such as communication, voice processing, or image analysis require time guarantees on the service response time. Previous work in the area of Web Services shows several solutions to the problem of monitoring the performance characteristics specified in a SLA [28], how to negotiate the SLAs of multiple services [34], and how these violations can be predicated automatically so that the service provider can act accordingly [19]. An assured computing middleware solution needs to integrate these capabilities and extend them to function at large scale and in a multi-tenancy environment (i.e., multiple cloud users sharing the same cloud infrastructure). In the field of data analysis, previous work introduced systems providing deadline driven scheduling for data processing in map/reduce clusters. These scheduling systems introduce guarantees on when a particular computation is going to be completed [31]. More research is needed to extend these guarantees to different types of distributed applications and evaluate how real-time guarantees can be provided even when faults and attacks are present.

Software is the fundamental piece that enables assured cloud computing. The security, availability, and real-time components are reflected in the software that manages the cloud. Such a software substrate is going to include software-defined networking as a management mechanism for cloud networks [30]. There is a strong need for a middleware solutions that supports such an integrated management.

3 Monitoring in assured cloud computing

A middleware for supporting assured cloud computing needs to enable applications to run in a secure, dependable, and timely fashion. In order to obtain this goal, the middleware needs to support applications for the management of the configuration of dynamic system-of-systems, the detection of security problems, and the dynamic mapping of workflow tasks. Supporting these applications requires having mechanisms for acquiring information about the system.

Hence, monitoring is a fundamental task for both cloud users and cloud providers of assured cloud computing. Without monitoring, a system is blind and unable to react to ensure meeting all the security and timeliness requirements. Monitoring ensures that the cloud system is operating within an assured level of service by providing information about the operations and about the state of each component.

Policies are an important tool for describing security and reliability requirements of the cloud infrastructure. Policies specify allowed configurations on the infrastructure and aim at guaranteeing a minimal level of security, reliability, and service. For example, simple policies are used today for expressing the security requirements of PCI-DSS [27] or FISMA [25]. These policies are expressed as access control policies, or they are expressed over the configuration of the infrastructure itself. While access control policies can be enforced at the application level, a large set of security controls need to be enforced at the infrastructure level: valid system configurations, network access control rules, access to systems, redundancy guarantees, or the presence of vulnerable software are properties that need to be monitored below the application level. These characteristics of the system are accessible through network security management tools such as SNMP, network scanners, or can be accessed through dedicated monitoring tools added to applications or operating systems. Information provided by these tools can be integrated by a middleware software running at the application level (e.g., to acquire workflow task placements, website traffic), at the OS level (e.g., running processes, network connection), and at the VM level (e.g., load of VMs, IO, and trusted information about VMs behavior).

While this information provides a low-level view of what is happening in the system, more processing is needed to make sense of such information in term of compliance to policies. *Events* that represent changes in the state of the system might need to be correlated across the entire system in order to detect violations of these policies. Such a middleware needs to be scalable and secure from attacks. Moreover, as cloud systems are not completely managed by a single organization, such a monitor middleware needs to provide functionalities for sharing monitoring information across organization boundaries. While current network

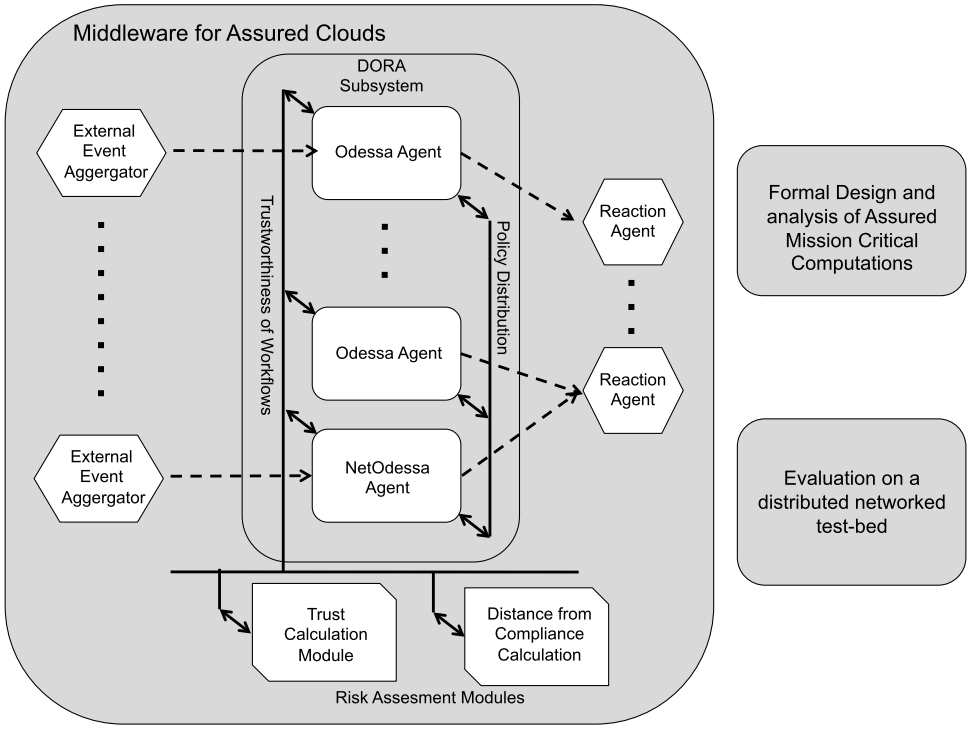

Fig. 1 The general architecture of the Middleware for Assured Clouds (Sect. 4), and the related research tracks (Sect. 6)

management solutions [12] and event-based systems [20] provide a partial solution to some of the aspects of these problems, verifying that they are able to satisfy all the requirements of this new type of middleware remains a challenge.

Real-time and scalability: The distribution of monitoring load is at the base of obtaining a scalable and real-time monitoring. Cloud operators are already managing systems composed of tens of thousands of machines. Detailed monitoring of the security state of an infrastructure could require accessing information such as running programs, network connections, Syslog events, and application-level events like requested pages in a web servers, logins in SSH servers, or database queries. Millions of events can be generated each second and need to be processed for detecting violations from the assured characteristics of the system. The monitoring system should be able to detect such violations in near real-time in order to initiate a proper response.

Availability and security: Since the information collected by the monitoring system is at the base of the assured operation of cloud systems, attacks that compromise it directly affect the operations of the infrastructure. Attacks on the availability of the system can reduce the ability to detect attacks or to perform operations such as workflow placement. Attacks on the confidentiality of the monitoring system allow an attack to acquire critical information about the security state that can enable additional attacks. Attacks on the integrity of the system can hide malicious behavior and inject false information. If this information is then used for reaction, it can create problems for the integrity and availability of the provided services. Monitoring systems should be designed with "need-to-know" and "separation-of-duty" principles because such a design can limit the effects that security compromises have on the infrastructure they monitor.

Information sharing: The composition of services for the execution of the workflow requires the integration of information about the infrastructure managed by cloud providers and the infrastructure managed by cloud users. For example, the validation of policies imposed on the cloud-user systems might depend on information provided by the cloud provider such as colocation with other virtual machines, load of other nodes in the same physical node, or security state of the machines provided services. A monitoring middleware needs to be able to share the necessary information for validating policies without providing access to the entire state of the system.

4 A "middleware for assured clouds"

Our research in Assured Cloud Computing involves three main tracks, as depicted in Fig. 1. In this section, we describe the general architecture of the Middleware for Assured Clouds (MAC). The MAC middleware is comprised

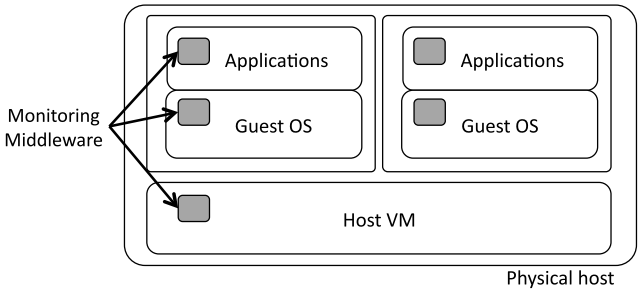

Fig. 2 Monitoring capabilities need to be deployed at multiple levels in the cloud architecture: application, guest VMs, and host VMs

of four major components, which will be described in detail in the rest of this section.

4.1 DORA subsystem

The core of the monitoring system is composed of a set of software agents that receive events and correlate them to detect when policies are violated. These agents operate and acquire information at different levels in the cloud-computing architecture, as shown in Fig. 2. Additionally, special agents (such as the NetOdessa agent [6]) run on special devices such as OpenFlow controllers.

We propose several techniques that exploit the policy-based nature of compliance monitoring in the cloud to address several of the challenges we presented in distributing monitoring load and introducing redundancy. The architecture of each middleware component is shown in Fig. 3. The local software agent running on each VM monitors the state of the local system by acquiring information through different mechanisms such as SNMP, Syscalls, or VM introspection. Agents perform a partial processing of policies using a local inference engine, and use the network to exchange information with the other monitoring agents that are part of the middleware.

Policies are expressed as rules over the configurations and state of *resources* in the system, such as hosts, network devices, or applications. The policies and the state information are expressed using Datalog [9]. Policies are represented as Datalog rules, and the state is represented using facts. Events mark changes in the state of the system by providing information concerning changes in the state of a resource. Agents monitor only events relevant to the purpose of validating compliance to the given policies. Examples of events are establishing a new connection, running a new service, or creating a new guest VM. As an example of policy, a simple policy could specify that critical services should not be run on an insecure machine. We express this policy in Datalog as follows:

```
critical_service(C), runs_program(H, C),
insecure(H) → fail(C, H)
```

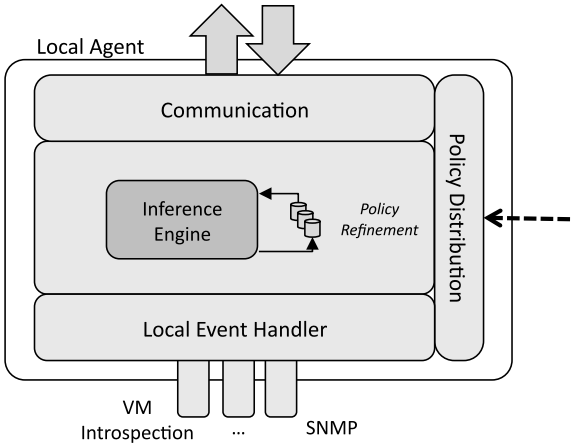

Fig. 3 High-level architecture of the monitoring agent, part of the middleware running on guest VMs and host VMs

In this rule, C and H are variables representing resources, and `runs_program` is a statement that is part of the system's state and changed by events. The two statements `critical_service` and `insecure` can either be events generated by local agents or can be inferred using policies from other events.

By taking advantage of the intuition that events describe information about the state of resources in the system (e.g., `critical_service`) or their relations (e.g., `runs_program`), our middleware performs optimizations aimed at reducing the monitoring load and at increasing security.

First, each local agent identifies the portion of each policy that relates to a single type of resource. If events related to such a resource are generated only locally on the device monitored by the agent, the validation of compliance to this policy is partially processed within the agent itself [23]. Second, a similar analysis of the relations between policies and resources is used to define a distributed protocol for validating compliance when events are generated by multiple agents. Resources are mapped to different nodes in the system, and policies are interpreted as describing undesirable relations between resources having specific states. The detection of violations is performed by having nodes exchange information in a way consistent with the relation described by the policies. This algorithm does not require aggregating information in any static centralized or hierarchical structure [22]. This leads to high-scalability and low overhead in monitoring.

Our architecture addresses the security challenges in two ways. We use redundancy in monitoring to validate critical parts of the policy (e.g., by acquiring information from both guest VMs and host VMs), and we use byzantine replication at multiple levels for ensuring that a compromised component cannot compromise the integrity of the entire monitoring infrastructure [23]. The local processing of the policy

and the distributed policy validation algorithm implement a "need-to-know" principle. Events are sent to different nodes only if such an action is required for validating policies. This reduces the information about the system stored in each node. Consequently, it reduces the effects of compromised nodes on the confidentiality of the monitoring system [22].

4.2 External event aggregators

OpenFlow switches, routers, intrusion detection systems (IDS), or legacy applications generate events and might not be able to run on top of our middleware infrastructure. However, acquiring such events is important to validate policies that encompass their state. The *external event aggregators* translate events from different sources and wrap them into a common format and send them to the DORA subsystem.

4.3 Risk assessment modules

The MAC middleware provides continuous risk assessments on the state of the system.

The first module, the "trust calculation module," is modeled after [17], where the authors present an approach to a formal-semantics-based calculus of trust, and is used in MAC for real-time estimation of trustworthiness as a risk assessment mechanism. Trust is defined as "a conditional belief, represented as a probability distribution over three states: trust, distrust, and untrust." The module provides a systematic method of assessing the trustworthiness of a workflow by considering trustworthiness of its components and propagating them. It also provides mechanism to use quantified uncertainty to choose whether to accept the risks that trust implies.

The second module, the "distance from compliance calculation," estimates the effects of malfunctioning in the monitoring system [24]. If a part of the monitoring system is compromised or unavailable, changes in the state might be not detected. This module uses the current state of the system to compute a risk proportional to the number of events that, if undetected, would lead the system to operate in an undesirable state.

4.4 Reaction agents

Once the middleware detects a violation of the requirements that might impact the ability of the system to operate successfully, the system should react in a timely manner to correct the situation. The monitoring system can communicate with specific reaction agents and provide information such as the type of violation that has been detected and the set of events, configurations, and devices which are responsible for the presence of the violation. Using such information, the reaction modules perform changes in the state of the system. In our implementation, we use OpenFlow to adapt network access control policies to the state of the infrastructure. Our reaction agent [6] receives information about specific security violations and authorizes network flows consequently.

5 Open issues and challenges

There are still open research issues which we have not addressed in our current MAC design. These issues are related to the design of the middleware architecture and the definition of security policies for assured cloud environments.

The current MAC middleware needs to be extended to address completely the problem caused by the system-of-systems aspects of cloud environments. A real-world cloud computing infrastructure is a complex, heterogeneous system-of-systems. Aside from performance issues, the heterogeneity means that different components of the system might be vulnerable to different threats and, therefore, the risk assessment and management processes should consider these differences.

Risk assessments and security of the MAC middleware itself are still open research issues. Research is needed to integrate techniques that increase our confidence that the information provided by the monitoring agent is correct. This issue is similar to the classic problem of "who checks the checker." One possible approach is the use of hardware-based root-of-trust techniques based on commercial solutions such as Intel® V-pro and Intel® Trusted Execution Technology (Intel® TXT) that utilizes Intel® trusted platform module (TPM). Additionally, the system could use possible redundancies in the monitoring information to "monitor itself" using separation-of-duty principles to ensure that monitoring components are behaving correctly.

Metrics for the security of the monitoring itself are needed for comparing different monitoring systems. Such measures should express quantitatively the effects that attacks on the monitoring systems have on the monitoring itself and on the system being monitored. For example, these metrics should account for the presence of critical components in the system being monitored and the effects that a compromise of the monitoring system would cause on such components.

Using these measures, we would be able to express properties such as that the difficulty of attacking a system should scale with the size of the system and cost of a successful attack (i.e., attacking the monitoring of a large, complex system should be as hard as attacking the one of a small system, given equal cost of a successful attack).

Other research issues are related to the identification of an appropriate set of security policies for the assured cloud environment. More studies are needed to study the nature

of the policies that should be enforced by the MAC middleware. This problem is complicated by the fact that currently it is challenging to identify the effectiveness of policies in increasing the overall security of the system. Using numerical definition of assurance, it might be possible to verify their effects on the actual assurance values. Another solution for the identification of policies could be provided by data mining techniques. These techniques could be used for gaining knowledge from massive amounts of system logs.

Last, distributing policies throughout the different parts of the middleware might present several challenges. Signing policies and distributing them might not be sufficient for ensuring that policies are applied correctly, as the PKI architecture would become the central point of failure of the entire monitoring process.

6 Other research projects involved in assured clouds

Research in the design of a middleware for assured cloud computing should analyze a wide range of issues that go beyond the system level issues described in this paper [5]. Research is investigating the use of formal methods to design survivable dynamic distributed architecture that are reconfigured and deployed flexibly to adapt to changing situations, support environments with varying levels of trust, and monitor, detect, and respond to threats by deploying appropriate subarchitectures and protocols. In addition, formal methods are utilized in designing real-time assured algorithms and techniques to enable cloud-based heterogeneous systems take on mission-critical tasks.

Similarly, formal methods are utilized for analysis of assured clouds properties. This task is achieved by formally analyzing properties of independent components of the architecture, e.g., protocols and algorithms. For example, cryptographic protocols need to be formally analyzed for their security properties before placing trust in them and utilizing them as building blocks of assured clouds. Formal analysis tools such as Maude-NPA (used in the design and analysis of cryptographic protocols of the assured cloud) and Real-Time Maude (used to analyze network protocols and sensor network systems) are utilized to achieve our goals.

Finally, formal methods are used to evaluate quantitative properties on performance of clouds. This includes various Quality of Service (QoS) properties, particularly availability properties. Questions pertaining to this class of properties are not amenable to a "true" or "false" answer. Instead, they require quantitative answers (for example, an interval of estimated real-time values for the time that it will take to receive a response in answer to a request). For such quantitative properties two tools are used in tandem: probabilistic models specified with probabilistic rewrite rules [1], and the VesTa statistical model checker and its Maude interface. VesTa supports statistical model checking of properties in a quantitative probabilistic temporal logic, whose evaluation is obtained by Monte Carlo simulation, and hence scalable in cloud systems.

7 Conclusions

This paper considers the properties and building blocks of a middleware for critical cloud applications where mission assurance is a necessity. Such applications include complex dynamic systems-of-systems, with both trusted or partially trusted resources (data, sensors, networks, computers, etc.) and services sourced from multiple organizations. In particular, this middleware should include sophisticated monitoring, assessment of policy, and handling of the configuration and management of such complex systems. This paper describes a distributed monitoring middleware designed using the principles of need-to-know, separation of duty and redundant verification, and scalability of real-time detection and response.

Acknowledgements The authors would like to acknowledge the contribution of the following individuals (in alphabetical order): Professor Gul Agha, Dr. Rakesh Bobba, Dr. Jingwei Huang, Dr. Zbigniew T. Kalbarczyk, Professor José Meseguer and Professor David Nicol. We also would like to thank the Air Force Research Laboratory (AFRL) and Air Force Office of Scientific Research (AFOSR). This material is based on research sponsored by the Air Force Research Laboratory and the Air Force Office of Scientific Research, under agreement number FA8750-11-2-0084. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copy-right notation thereon. This work was also partially supported by the Boeing Company.

References

1. Agha G, Meseguer JE, Sen K (2006) PMaude: Rewrite-based specification language for probabilistic object systems. In: 3rd wksp quantitative aspects of programming languages (QAPL)
2. Amazon AWS (2008) Amazon S3 availability event: July 20, 2008. http://status.aws.amazon.com/s3-20080720.html
3. Amazon AWS (2011) AWS risk and compliance. Amazon Whitepapers
4. Amazon AWS Summary of the Amazon EC2 and Amazon RDS service disruption in the US East Region. http://aws.amazon.com/message/65648/
5. Assured Cloud Computing University Center of Excellence. http://assured-cloud-computing.illinois.edu/
6. Bellessa J, Kroske E, Farivar R, Montanari M, Larson K, Campbell RH (2011) NetODESSA: resilient policy enforcement for cloud networks. In: RACOS 2011, in conjunction with the 30th IEEE symposium on reliable distributed systems (SRDS)
7. Binning D (2011) Cloud computing may be the saviour of true unified communications. CIO Mag, May
8. Bowers KD, Van Dijk M, Juels A, Oprea A, Rivest RL (2011) How to tell if your cloud files are vulnerable to drive crashes. In: ACM conference on computer and communications security

9. Ceri S, Gottlob G, Tanca L (1989) What you always wanted to know about Datalog (and never dared to ask). IEEE Trans Knowl Data Eng 1(1):146–166
10. Dean J, Ghemawat S (2004) MapReduce: simplified data processing on large clusters. In: OSDI
11. Department of Defense (2010) Directive 3020.40: DoD policy and responsibilities for critical infrastructure. January
12. DMTF, Web-based enterprise management. http://www.dmtf.org/standards/wbem
13. Garfinkel T, Rosenblum M (2003) A virtual machine introspection based architecture for intrusion detection. In: Annual network and distributed systems security symposium
14. Gentry C (2009) Fully homomorphic encryption using ideal lattices. In: ACM symposium on theory of computing (STOC)
15. Isard M, Budiu M, Yu Y, Birrell A, Fetterly D (2007) Dryad: distributed data-parallel programs from sequential building blocks. In: Proceedings of the 2nd ACM SIGOPS/EuroSys European conference on computer systems
16. Jabbour K, Muccio S (2011) The science of mission assurance. J Strateg Secur 4(2):61–74
17. Jingwei H, Nicol D (2010) A formal-semantics-based calculus of trust. IEEE Internet Comput
18. Juels A, Kaliski BS Jr (2007) PORs: Proofs of retrievability for large files. In: ACM conference on computer and communications security
19. Leitner P, Michlmayr A, Rosenberg F, Dustdar S (2010) Monitoring, prediction and prevention of sla violations in composite services. In: IEEE international conference on web services
20. Li G, Jacobsen HA (2005) Composite subscriptions in content-based publish/subscribe systems. In: ACM/IFIP/USENIX middleware
21. Lockheed Martin (2011) Lockheed Martin announces blackcloud solution based on trusted infrastructure technologies from cyber security alliance partners. Lockheed Martin Press Release
22. Montanari M, Campbell R (2011) Attack-resilient compliance monitoring for large distributed infrastructure systems. In: 5th international conference on network and system security (NSS)
23. Montanari M, Chan E, Larson K, Yoo W, Campbell R (2011) Distributed security policy conformance. In: Future challenges in security and privacy for academia and industry (SEC)
24. Montanari M, Chaugule A, Campbell R (2011) Robustness of compliance to infrastructure security policies. Computer Science Research and Technical Reports. University of Illinois
25. National Institute for Standard and Technology (2011) Federal information security management act (FISMA) implementation project. http://csrc.nist.gov/groups/SMA/fisma/
26. New York Times, April 2011. http://bits.blogs.nytimes.com/2011/04/21/amazon-cloud-failure-takes-down-web-sites/
27. PCI Security Standard Council (2011) PCI-DSS v2.0. https://www.pcisecuritystandards.org
28. Sommers J, Barford P, Duffield N, Ron A (2007) Accurate and efficient SLA compliance monitoring. In: ACM SIGCOMM
29. Szefer J, Keller E, Lee RB, Rexford J (2011) Eliminating the hypervisor attack surface for a more secure cloud. In: ACM conference on computer and communications security
30. Vaughan-Nichols SJ (2011) OpenFlow: the next generation of the network? IEEE Comput 44(8)
31. Verma A, Cherkasova L, Campbell RH (2011) ARIA: automatic resource inference and allocation for MapReduce environments. In: International conference on autonomic computing (ICAC), June 2011
32. Wang L, Tao J, Kunze M, Castellanos AC, Kramer D, Karl W (2008) Scientific cloud computing: early definition and experience. IEEE HPCC
33. Zhang K, Zhou X, Chen Y, Wang X, Ruan Y (2011) Sedic: Privacy-aware data intensive computing on hybrid cloud. In: ACM conference on computer and communications security
34. Zulkernine F, Martin P, Craddock C, Wilson K (2009) A policy-based middleware for web services sla negotiation. In: IEEE international conference on web services