www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 12, Dec.2013 Page No. 3345-3352

A STUDY ON TRADE-OFFS AMONG SERVICE EXCELLENCE ATTRIBUTES OF CLOUD COMPUTING

P.Sundaramoorthy [#1], S.P. Santhoshkumar [#2], B. Santhoshkumar [#3], Dr. M. Selvam[#4]

[#1]*Assistant Professor, Department of CSE, SNS College of Technology, Coimbatore, India.*
E-Mail ID: sundarme08@gmail.com
[#2]*Assistant Professor, Department of CSE, SNS College of Technology, Coimbatore, India.*
E-Mail ID: spsanthoshkumar16@gmail.com
[#3]*Assistant Professor, Department of CSE, SNS College of Technology, Coimbatore, India.*
E-Mail ID: b.santhoshkumar@gmail.com
[#4]*Principal, ACT College of Technology, Coimbatore, India.*
E-Mail ID: ammselvam@gmail.com

Abstract: In this paper infrastructure as a service (IaaS), development and runtime platforms as a service (PaaS), and software and business applications as a service (SaaS). Clients not having any resources and data are guaranteed to be available and ubiquitously easy to get to by means of Web services and Web APIs "in the Cloud". In the condition of cloud computing, this implies a significant, principle business decision whether to own and preserve a data center or outsource operations to the cloud. In Challenges and the opportunities are bond with the availability or performance of software running in the cloud, as well as privacy and data control. IT resources that is actually needed; for the service provider, better resource utilization of existing infrastructure is achieved through a multi-tenant architecture. In This paper suggest in trade-offs among service quality attributes, such as availability, distributed data consistency, service runtime presentation, and privacy in cloud computing.

Keywords: Trade-offs, PaaS, SaaS, Distributed Systems, Service Quality, Cloud Computing

1 INTRODUCTION:

In recent years, Cloud Computing has become an emerging technology that gains wide influence on IT systems. Cloud Computing is a distributed computing model for enabling service-oriented, on-demand network access to rapidly scalable resources [9]. Such resources include infrastructure as a service (IaaS), development and runtime platforms as a service (PaaS), and software and business applications as a service (SaaS). Clients do not own the resources, yet applications and data are guaranteed to be available and ubiquitously accessible by means of Web services and Web APIs "in the Cloud".

1.1 Value Proposition

The main value proposition of Cloud Computing is to provide the clients a cost-effective, convenient means to consume the amount of IT resources that is actually needed; for the service provider, better resource utilization of existing infrastructure is achieved through a multi-tenant architecture. From a business perspective, Cloud Computing is about improving organizational efficiency and reducing cost, often coupled with the objective of achieving a faster timeto- market. Centrally hosted services with self-service interfaces can help to reduce lead times between organizational units who use the cloud as a collaborative IT environment. Re-usable components, packaged on virtual machines, provide a way to exchange working IT solutions. Capabilities to allocate and de-allocate shared resources on demand can significantly decrease overall IT spending. Low-cost access to data centres in different geographical regions may further reduce market entry barriers and enable new business models. From a technology and engineering perspective, Cloud Computing can help to realize or improve scalability, availability, and other non-functional properties of application architectures. In this paper, we focus on the survey on technology perspective, and in particular on challenges and opportunities of Cloud Computing research related to quality-driven software service architectures. These include aspects of availability,

runtime performance and power management, as well as privacy and distributed data usage.

1.2 Challenges & Research Questions

Not all desired architectural properties can be achieved at the same time. Trade-off decisions have to be made between several (sometimes contradictory) goals, such as:

1. Increase availability & reliability
2. Increase performance (latency, throughput)
3. Increase security and ensure privacy

1.3 Virtualization

Which metrics are useful to describe and analyze these trade-off decisions? Based on specific software architecture styles and solutions: How are these goals correlated? How can trade-offs be accounted for during application design, how can they be adapted during run-time? Since several cloud providers succeeded to introduce scalable, highly available software components such as messages queues or data stores as a service, building a software application (as a service) to be deployed in the cloud requires new architectural decisions and decision-making processes. Which cloud services can be adapted as components of a new software/service? Using system virtualization, multiple virtual machines, which may run various operating systems, can be run on a single physical machine. In a similar way, storage virtualization provides access to a logical storage that abstracts from (possibly heterogeneous) physical storage devices. Application virtualization provides a virtualized environment that runs inside an OS process and provides a platform-independent environment for applications (e.g. Java applications running in a Java virtual machine).

1.4 Decision Support

When building new software applications or services that might potentially be deployed in the cloud, some decisions are inevitable in different stages of the software engineering process. This starts with initial build-or-buy decisions which lead to subsequent questions of how services are operated and who has control over and holds responsible for service delivery. For example, use of an in-house IT department may be compared to the use of third party service providers. In the context of cloud computing, this implies an important, principle business decision whether to own and maintain a data center or outsource operations to the cloud. For the different objectives, i.e. availability and performance of software running in the cloud, as well as questions related to privacy and distributed data usage, we give an overview on the state of the art, highlight important research questions, and outline approaches to tackle the presented challenges.

2 A SURVEY ON CHALLENGES AND OPPORTUNITIES:

Cloud Computing introduces a number of technology and engineering challenges, many of which relate to “traditional” requirements of distributed systems, which now must be revisited in the context of virtualized environments.

2.1 Availability

Generally, availability is the degree to which a system is operable, that is, capable of producing responses to submitted requests. Stronger definitions of availability may include objectives with regard to the time window allowed for any response to arrive, or the time window allowed for the system not to be operable.

2.1.1 State of the Art

Availability is a major challenge in the face of massive numbers of servers constituting an environment where frequent failures are a fact to be coped with. Replication of servers and storage is the key technique to achieve high availability. CAP states that only two of the three properties, transactional consistency (C), high availability (A), and resiliency to network partitions (P) can be achieved at the same time. In widely distributed systems – typical cloud computing environments – partitions are considered inevitable, leaving the trade-off between consistency and availability. For example, strong consistency can be achieved by pessimistic replication mechanisms at the cost of availability.

2.1.2 Goals and Approach

Cloud computing is widely perceived as a disruptive technology shift from on-premise infrastructure, platforms andapplications to Internet-centric infrastructure, platform, and software services. Moving on-premise applications and services into the cloud can improve availability, particularly along with the following properties:

1. Worldwide access
2. Workload elasticity
3. Fault-tolerance & disaster-tolerance

Desktop applications and applications deployed in small-scale networks (LANs) are designed to be accessible to a group of local users. Although access can be granted to remote users in different geographic regions, this is not the natural modus operandi of such applications. Moving an application into the cloud can simplify access of worldwide users.

Workload elasticity means adapting system resources to changing workloads in realtime, i.e. growing under increasing workload and shrinking under decreasing workload. Vertical scalability – replacing old hardware with

new hardware – is not well-suited for providing workload elasticity. Horizontal scalability, i.e. partitioning and replicating homogeneous system components, on the other side, allows incrementally adapting system resources to changing workloads.

Cloud-based application architecture can provide improved levels of fault-tolerance compared to locally hosted applications – and even disaster-tolerance. Infrastructure as a service (IaaS) allows allocating and de-allocating system resources on demand.

2.1.3 Research Questions and Challenges

Cloud computing provides techniques and infrastructure for building highly available, Internet-scale applications and services.

1. What are the major trade-off decisions when moving into the cloud? The trade-off between "availability" and "consistency" serves as a prominent example. Other trade-off decisions must be identified and evaluated, particularly taking into consideration "scalability", "reliability", "performance" (latency and throughput), and "cost-efficiency".
2. Benchmarking cloud services requires a new set of benchmarking tools that consider cloud-specific properties, such as practically infinite scalability, relaxed consistency guarantees, on demand resource allocation and accounting, et cetera.
3. How can Desktop applications and traditional network-based application architectures, such as Java EE architectures, profit from the integration of cloud services or being moved into the cloud altogether?
4. Which new tools are necessary to facilitate service (re-)engineering and migration?
5. Heterogeneous cloud services and application delivery channels demand for integration technology. How should a "Cloud Service Bus" be designed?

2.2 Design-time Performance Prediction

Reasoning about the performance of a software system is a key factor that has to be taken into account in software development. If performance flaws of a software are detected early in the software development process, costs and efforts of changing the system to increase the performance are lower compared to later changes.

2.2.1 State of the Art

Reflecting performance-relevant virtualization properties would also allow for choosing between different virtualization solutions, as the choice of virtualization solutions may have an impact on the application's performance. As the performance heavily depends on the hardware resources and the system environment software is running on, these factors have to be taken into account for performance prediction. Such models are available for analyzing the performance of non-virtualized software systems, but reusing the models for analyzing software running in virtualized and cloud-computing environments may lead to imprecise or wrong prediction results.

2.2.2 Goals and Approach

To allow for accurate performance predictions of software running in virtualized environments, performance-relevant properties of the virtualization layer have to be taken into account. However, integrating such properties into an analysis model is a cumbersome task, as it has to be done manually and requires domain knowledge

To apply the approach to virtualized environments, different performance-relevant properties have to be regarded, and thus different measurements have to be designed.

2.2.3 Research Questions and Challenges

To predict the performance of applications running in virtualized environments during design-time, the following challenges have to be addressed:

1. What are the performance-relevant properties of virtualized environments that can be taken into account during design-time?
2. How to model software for design-time performance prediction, and how to include performance-relevant properties of virtualized environments into the model?
3. How to integrate such performance-relevant properties into performance analyses?

To automatically detect performance-relevant properties of virtualized environments through goal-oriented measurements, additional challenges arise:

1. How to design measurements to detect performance-relevant properties in a technology-independent way?
2. How to deal with additional load on the system which might lead to disturbed measurement results?

2.3 Run-time Performance and Power Management

Cloud computing and virtualization promise substantial reduction of IT operating costs resulting from higher energy efficiency and lower system management costs. Today, only 12% of x86 server workloads are running in virtual machines, however, by 2013 that number is expected to rise to 61%.

2.3.1 State of the Art

A meta-model for modeling the virtualization layer in detail is not provided. The authors concentrate on allocating resources dynamically based on a ”Monitor-Analyze-Plan- Execute” pattern which takes into account evolving system workloads. This approach, however, does not provide means to model the mapping of virtualized resources to physical resources explicitly.

2.3.2 Goals and Approach

To address the challenges of predictable and efficient resource management in virtualized data centers comprising a Cloud Computing environment, we advocate the development of novel techniques for self-adaptive management of application performance and energy efficiency.

The goal of these techniques will be to continuously optimize the performance and energy efficiency of the computing infrastructure by automatically reconfiguring resource allocations in response to changes in application workloads.

The specific goals that will be pursued are listed below:

1. Extend existing performance meta-models for component-based software architectures to support modeling applications running in cloud computing environments.
2. Develop meta-models for capturing the energy efficiency of virtualized data center infrastructures taking into account the resource allocations of individual hosted applications as well as the utilization of system resources.
3. Develop efficient analysis techniques for solving instances of the meta-models.
4. Develop methods and tools for automatic model maintenance during operation through continuous monitoring of the service infrastructure.
5. Develop efficient techniques for self-adaptive system reconfiguration at run-time to reflect dynamic changes in application workloads.

2.3.3 Research Questions and Challenges

1. Can design-time performance prediction techniques be simplified and adapted so that they can be used for performance prediction at run-time?
2. At what level of abstraction should services and infrastructure components in a cloud computing environment be modeled to enable predictability at run-time?
3. What model solution techniques are suitable for performance prediction at runtime providing a good trade-off between prediction accuracy and overhead?
4. What time-scales are reasonable to apply the various online reconfiguration mechanisms?
5. What workload forecasting techniques are appropriate to model the evolution of complex workloads composed of multiple usage profiles of independent applications running on a shared physical infrastructure?
6. What utility functions are suitable to evaluate the quality of alternative system configurations in terms of their performance and energy efficiency?
7. At what level of granularity should application workloads and system components be monitored at run-time in order to detect changes in workloads and operating conditions early enough to be able to proactively adapt the system configuration accordingly while minimizing the monitoring overhead?

2.4 Privacy in Service-Oriented Computing

2.4.1 Privacy Problems Introduced through Service Orientation

Due to its advantages, cloud computing will replace traditional computing in many fields. Clients will not need to maintain a costly computing center and obtain software and computing resources as a Service.

Despite the benefits Service orientation has to offer, there are inherent privacy problems. By using services, clients lose control over their data. They can not control if their data gets copied or misused on the server and have to trust the Service provider. Current security mechanisms focus on protecting the data from external adversaries, for example an adversary eavesdropping data transfer between the client and the service.

2.4.2 State of the Art

Cryptography offers methods with strong security guarantees for scenarios involving different parties while one or more party is not fully trusted. There are methods for different security models. It is possible to apply these methods to services in order to enhance privacy. There are cryptographic solutions for two or more parties cooperatively computing a certain function over a set of data without any party learning anything about the input of other parties except what is learned by the output. Using an interactive protocol, these secure multiparty computations can thus solve all computation related privacy problems.

The problem is that for each party, the computation cost is higher than computing the whole function on the complete input without any other party. This makes the concept of multiparty computation for outsourcing services too expensive and in fact pointless if the client is the only one with private input.

2.4.3 Goals and Approach

A major aim of this project is the development of novel methods that provide provable privacy guarantees, yet are efficient enough to be used in a service scenario. In general, these privacy guarantees have to be weaker than classical cryptographic notions, but provide a sufficient level of protection. Combinations of architectural and cryptographic approaches are a promising direction. We proposed separations of duties that can be used to enhance the privacy of services.

2.4.4 Research Questions

We have identified the following challenges:

1. Security guarantees
2. What can practical security guarantees that can be used in a Software as a Service scenario look like?
3. What are achievable protection requirements?
4. How can they be formalized in order to prove the level of privacy a service provides?
5. Realization
6. How can security guaranties achieved practically?
7. How can architectural and cryptographic approaches be combined in order to enhance privacy?
8. Are there architectures that favor the realization of privacy guarantees?
9. Can reasonable assumptions about hardware (TPM, USB smart cards) be used to achieve a certain level of privacy that could not be achieved otherwise?
10. How can services be adapted to achieve certain privacy guarantees?
11. What privacy guarantees can be provided for data in services using standard techniques?
12. How can these techniques be combined efficiently?
13. How can the level of privacy not well understood methods provide be formulated?
14. What privacy guarantees can be achieved using dummy data or dummy queries?
15. How can dummy queries and dummy data be generated in order to achieve the best possible practical level of privacy?

2.5 Distributed Data Usage

Today's cloud computing infrastructures usually require customers who transfer data into the cloud to trust the providers of the cloud infrastructure. This trust extends to both confidentiality and integrity of the data. Depending on the value of the data, however, not every customer is willing to grant this trust without any justification.

We plan to provide a framework for data-driven usage control in the cloud, i.e., the extension of usage control by data flow detection concepts. We want to enforce usage control properties, or at least detect their violation, not for one precisely specified object (one specific data container), but rather for all representations of the data, i.e., all containers that actually or potentially contain the respective object.

2.5.1 State of the Art

Enforcement Mechanisms Enforcement mechanisms for requirements such as "delete after thirty days," "do not copy," "notify me when giving away," "at most three copies," etc., have, for a variety of policy languages, been implemented at single layers of abstraction: at the operating system level, at the X11 level, for Java the .NET CIL and machine languages; workflow systems; service-oriented architectures [4]; the level of an enterprise service bus; for dedicated applications such as the Internet Explorer or in the context of digital rights management. From a slightly different perspective, comparable monitors are also investigated in grids where resources are dynamically assigned and freed, and they are considered in the domain of intrusion detection systems.

2.5.2 Goals and Approach

The general problem of distributed data usage control is concerned with the problem of how to manage data once it has been given away. Application domains include privacy, compliance with regulations, data management in distributed business processes, digital rights management, eGovernment, the management of intellectual property and, in general, that of secrets. Typical requirements include "don't disseminate," "notify me when giving away my data," "delete after thirty days," "don't delete within five years," etc. The more distributed a system is, the more complex the challenge becomes.

We are convinced that any cloud technology needs support and builtin approaches to provide mechanisms for the enforcement of distributed usage control policies both at the service and the infrastructure levels.

2.5.3 Research Questions and Challenges

Requirements and specification of policies Usage control policies - at different levels of granularity - need to address data, data representations, actions on data or representations, and, related, different systems or services on which the data representations are stored.

For instance, profile data in a social network may come as data base record, file, or Java object. It may be stored in file servers, backup servers, web servers, billing systems, etc.

Enforcement of policies Once policies have been specified, they need to be enforced at the different levels of abstraction, both vertically within one system and horizontally across different systems.

3 CONCLUSIONS:

In this paper, we presented A Survey on Challenges and Opportunities of Cloud Computing technology. Such challenges and opportunities deal with the availability or performance of software running in the cloud, as well as privacy and data control. For these research fields, we highlighted the current state of the art, and presented approaches to mitigate the open problems. We argue that Cloud Computing introduces new trade-off decisions in the context of quality-driven software service architectures. These decisions include trade-offs between service quality attributes, such as availability, distributed data consistency, service runtime performance, and privacy. We envision a structured decision support framework for cloud-based architectures that explicitly addresses these trade-offs.

REFERENCES:

[1] Multimedia framework (MPEG-21) – Part 5: Rights Expression Language, 2004. ISO/IEC standard 21000-5:2004.

[2] Adobe livecycle rights management http://www.adobe.com/products/livecycle/rightsmanagement/indepth.html, Aug. 2010.

[3] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, and Y. Xu. Two can keep a secret: A distributed architecture for secure database services. CIDR 2005.

[4] B. Agreiter, M. Alam, R. Breu, M. Hafner, A. Pretschner, J.-P. Seifert, and X. Zhang. A technical architecture for enforcing usage control requirements in service-oriented architectures. In SWS, pages 18–25, 2007.

[5] G. Amanatidis, A. Boldyreva, and A. O'Neill. Provably-secure schemes for basic query support in outsourced databases. In DBSec, pages 14–30, 2007.

[6] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter. Enterprise Privacy Authorization Language (EPAL 1.2). IBM Technical Report, 2003. http://www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/.

[7] B Santhosh Kumar, Suresh M, Shafi Ullah Z, "An Analysis of Load Balancing in Cloud Computing", International Journal of Engineering Research & Technology (IJERT), Vol 2: No:10, 2013, 428-433

[8] S. Balsamo, A. Di Marco, P. Inverardi, and M. Simeoni. Model-Based Performance Prediction in Software Development: A Survey. IEEE Transactions on Software Engineering, 30(5):295–310, May 2004.

[9] C. Baun, editor. Cloud Computing : Web-basierte dynamische IT-Services. Informatik im Fokus. Springer, Heidelberg [u.a.], 2010.

[10] S. Becker, H. Koziolek, and R. Reussner. The Palladio Component Model for Model-driven Performance Prediction. J. Syst. Softw., 82(1):3–22, 2009.

[11] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In CRYPTO, pages 535–552, 2007.

[12] S. Berger, R. C´aceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn. vtpm: virtualizing the trusted platform module. In USENIX-SS'06: Proceedings of the 15th conference on USENIX Security Symposium, pages 305–320, Berkeley, CA, USA, 2006. USENIX Association.

[13] K. Birman, G. Chockler, and R. van Renesse. Toward a cloud computing research agenda. SIGACT News, 40(2):68–80, 2009.

[14] D. Boneh, A. Joux, and P. Nguyen. Why textbook elgamal and rsa encryption are insecure (extended abstract), 2000.

[15] J. Brodkin. Gartner's data on energy consumption, virtualization, cloud. IT world, The IDG Network, 2008. http://www.itworld.com/green-it/59328/gartners-data-energy-consumption-virtualization-cloud.

[16] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6):599–616, June 2009.

[17] A. Ceselli, E. Damiani, S. D. C. D. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Modeling and assessing inference exposure in encrypted databases, 2005.

[18] D. Chaum, C. Cr´epeau, and I. Damgard. Multiparty unconditionally secure protocols. In STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing, pages 11–19, New York, NY, USA, 1988. ACM.

[19] J. Clause, W. Li, and A. Orso. Dytan: a generic dynamic taint analysis framework. In Proc. Intl. Symp. on software testing and analysis, pages 196–206, 2007.

[20] S. Curry, J. Darbyshire, D. W. Fisher, B. Hartman, S. Herrod, V. Kumar, F. Martins, S. Orrin, and D. E. Wolf. Infrastructure security: Getting to the bottom of compliance in the cloud. http://www.rsa.com/innovation/docs/CCOM_BRF_0310.pdf, March 2010.

[21] M. Dam, B. Jacobs, A. Lundblad, and F. Piessens. Security monitor inlining for multithreaded java. In Proc. ECOOP, pages pp. 546–569, 2009.

[22] E. Damiani, S. D. C. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Balancing confidentiality and efficiency in untrusted relational DBMSs, 2003.

[23] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. The Ponder Policy Specification Language. In Proc. Workshop on Policies for Distributed Systems and Networks, pages 18–39, 1995.

[24] L. Desmet, W. Joosen, F. Massacci, K. Naliuka, P. Philippaerts, F. Piessens, and D. Vanoverberghe. The S3MS.NET Run Time Monitor: Tool Demonstration. ENTCS, 253(5):153–159, 2009.

[25] B. Devlin, J. Gray, B. Laing, and G. Spix. Scalability Terminology: Farms, Clones, Partitions, Packs, RACS and RAPS. CoRR, cs.AR/9912010, 1999.

[26] R. Dowsley, J. M¨uller-Quade, and A. C. A. Nascimento. A cca2 secure public key encryption scheme based on the mceliece assumptions in the standard model. In CT-RSA, pages 240–251, 2009.

[27] R. I. (ed.). Open Digital Rights Language v1.1, 2008. http://odrl.net/1.1/ODRL-11.pdf.

[28] M. Egele, C. Kruegel, E. Kirda, H. Yin, and D. Song. Dynamic spyware analysis. In Proceedings of USENIX Annual Technical Conference, June 2007.

[29] U. Erlingsson and F. Schneider. SASI enforcement of security policies: A retrospective. In Proc. New Security Paradigms Workshop, pages 87–95, 1999.

[30] A. Fox and E. A. Brewer. Harvest, yield, and scalable tolerant systems. In HOTOS '99: Proceedings of the The Seventh Workshop on Hot Topics in Operating Systems, page 174, Washington, DC, USA, 1999. IEEE Computer Society.

[31] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh. Terra: a virtual machine-based platform for trusted computing. SIGOPS Oper. Syst. Rev., 37(5):193–206, 2003.

[32] W. Gasarch. A survey on private information retrieval. Bulletin of the EATCS, 82:72–107, 2004.

[33] C. Gentry. Fully homomorphic encryption using ideal lattices. In STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing, pages 169–178, New York, NY, USA, 2009. ACM.

[34] G. Gheorghe, S. Neuhaus, and B. Crispo. xESB: An Enterprise Service Bus for Access and Usage Control Policy Enforcement. In Proc. Annual IFIP WG 11.11 International Conference on Trust Management, 2010.

[35] S. Gilbert and N. Lynch. Brewer's conjecture and the feasibility of consistent available partition-tolerant web services. In In ACM SIGACT News, page 2002, 2002.

[36] R. Goldberg. Survey of Virtual Machine Research. In IEEE Computer Magazine, volume 7, pages 34–45, 1974.

[37] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing, pages 218–229, New York, NY, USA, 1987. ACM.

[38] H. Hacig¨um¨us, B. Iyer, C. Li, and S. Mehrotra. Executing SQL over encrypted data in the database-serviceprovider model. In Proceedings of the 2002 ACM SIGMOD international conference on Management of data, pages 216–227. ACM, 2002.

[39] H. Hacig¨um¨us, B. Iyer, and S. Mehrotra. Providing database as a service. In ICDE '02: Proceedings of the 18th International Conference on Data Engineering, page 29, Washington, DC, USA, 2002. IEEE Computer Society.

[40] M. Harvan and A. Pretschner. State-based Usage Control Enforcement with Data Flow Tracking using System Call Interposition. In Proc. 3rd Intl. Conf. on Network and System Security, pages 373–380, 2009.

[41] M. Hauck, J. Happe, and R. H. Reussner. Automatic Derivation of Performance Prediction Models for Loadbalancing Properties Based on Goal-oriented Measurements. In Proceedings of the 18th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'10), 2010. To appear.

[42] M. Hauck, M. Kuperberg, K. Krogmann, and R. Reussner. Modelling Layered Component Execution Environments for Performance Prediction. In Proc. 12th International Symposium on Component-Based Software Engineering, 2009.

[43] C. Henrich, M. Huber, C. Kempka, J. Mueller-Quade, and R. Reussner. Technical report: Secure cloud computing through a separation of duties. https://sdqweb.ipd.kit.edu/huber/reports/sod/technical_report_sod. pdf, 2010.

[44] M. Hilty, A. Pretschner, D. Basin, C. Schaefer, and T. Walter. A policy language for distributed usage control. In Proc. ESORICS, pages 531–546, 2008.

[45] B. Hore, S. Mehrotra, and G. Tsudik. A privacy-preserving index for range queries. In VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases, pages 720–731. VLDB Endowment, 2004.

[46] M. Huber. Towards secure services in an untrusted environment. In Proceedings of the Fifteenth International Workshop on Component-Oriented Programming (WCOP) 2010, 2010. to appear.

[47] G. Hunt and D. Brubacher. Detours: Binary Interception of Win32 Functions. In Proc. 3rd USENIX Windows NT Symposium, 1999.

[48] I. Ion, B. Dragovic, and B. Crispo. Extending the Java Virtual Machine to Enforce Fine-Grained Security Policies in Mobile Devices. In Proc. Annual Computer Security Applications Conference, pages 233–242. IEEE Computer Society, 2007.

[49] R. Iyer, R. Illikkal, O. Tickoo, L. Zhao, P. Apparao, and D. Newell. Vm3: Measuring, modeling and managing vm shared resources. Computer Networks, 53(17):2873 – 2887, 2009.

[50] B. Jansen, H. Ramasamy, and M. Schunter. Flexible integrity protection and verification architecture for virtual machine monitors. In Proc. Second Workshop on Advances in Trusted Computing, August 2006.

[51] M. Kantarcioglu and C. Clifton. Security issues in querying encrypted data. Technical report, 2004.

[52] R. Keeney and H. Raiffa. Decisions with multiple objectives. Cambridge Books, 1993.

[53] M. Klems, J. Nimis, and S. Tai. Do clouds compute? a framework for estimating the value of cloud computing. In Proceedings 7th Workshop on e-Business, LNBIP. Springer, Dezember 2008.

[54] M. Klems, L. Shwartz, G. Grabarnik, and S. Tai. Automating the delivery of it service continuity management through cloud service orchestration. In Proceedings of the 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010). IEEE, April 2010.

[55] H. Koziolek. Performance Evaluation of Component-based Software Systems: A Survey.