Infrastructure Management Support in a Multi-Agent Architecture for Internet of Things

Bogdan Manațe*, Teodor-Florin Fortiș*†
*West University of Timişoara,
Faculty of Mathematics and Informatics
bvd. V. Pârvan, 4, 300223
Timişoara, Romania
Email: {bogdan.manate,fortis}@info.uvt.ro

Viorel Negru†*
†Institute e-Austria Timişoara,
bvd. V. Pârvan, 4, 300223
Timişoara, Romania
Email: vnegru@info.uvt.ro

***Abstract*—This paper examines the cloud resources management for a multi-agent IoT architecture. The resources tenancy is a costly operation, thus their allocation and management should be approached in an effective manner. On the other hand, the infrastructure should not be affected by the deployment or maintenance life cycle, operations that could put parts of the system offline, or even the entire system. We emphasize the need for infrastructure audit, which offers a good insight of how the resources are used, the geographical areas with an increased number of failures and where the allocation of supplementary resources is mandatory. Also, the security audit and its impact over a distributed multi-agent architecture that handles a large number of heterogeneous devices is discussed.**

***Keywords*–multi-agent architecture; infrastructure management; Internet of Things; cloud computing**

I. Introduction

The recent advances in Internet of Things (IoT) and cloud computing have led to an increased usage of this two technologies to create solid architectures that is able to handle hundred of thousands of concurrent connections, at the same time offering a good Quality of Service (QoS). The usage of cloud computing for creating the backbone infrastructure for an IoT architecture is beneficial for companies or research centers which are trying to reach a large number of clients, because there is no need to upgrade or maintain the physical infrastructure. Even though the cloud client is not completely aware of the exact location of the hardware, there are mechanisms available that can determine the best geographical region for optimal performance [1].

In order to handle a big number of connections from a myriad of devices, an Internet of Things architecture should employ a fast and reliable infrastructure for its services. Because of different usage patterns, the best solution for an infrastructure for IoT framework relies on cloud computing resources [2][3][4][5]. Also, the cost for keeping the architecture up and running can be significantly reduced by releasing unused resources at time intervals where low network traffic is measured.

From the reliability and operational standpoint, the solution needs to bypass any bottlenecks introduced by the on demand created infrastructure. Therefore, any replacement or restart of the virtual machines should not affect the system's response time.

On a small application, the gateway application and the database server can reside on the same physical server, thus the business logic is very easy to manage. In an Internet of Things framework, the business logic is separated on multiple physical servers, because it needs to process and store a large amount of information sent by the clients. Considering the tremendous amount of data that needs to be stored, the database instances are separated on multiple virtual machines, so that the database instances can be grouped in clusters for a better distribution of stored data.

The same need for intensive infrastructure management is identified in social networks, like Facebook [6], Twitter [7], LinkedIn [8] and MySpace [9], where data integrity and services up time play an important role.

This paper is structured as follows: in Section II is presented a literature review of existing infrastructures, in Section IV is presented a solution for a cloud infrastructure management used to scale an IoT architecture, in Section V we emphasize the importance of infrastructure audit and underline the benefits of a proper infrastructure audit and finally, in Section VI we conclude this paper.

II. Background

By unifying the Internet of Things, Cloud Computing and network edge services a new paradigm emerges called Fog Computing[10]. The researchers from CISCO have proposed this term to exemplify a system that is characterize by low latency, ample geographical distribution, location awareness, mobility and heterogeneity. In the context of Fog Computing the services are closer to the consumer delivering this way the required information in a fast and reliable manner. The support for bi-directional data flow (from devices to the cloud and conversely) allows the system to have a greater control over the data sources and the cloud resources. In case

978-1-4799-7412-2/14 $31.00 © 2014 IEEE
DOI 10.1109/EMS.2014.36

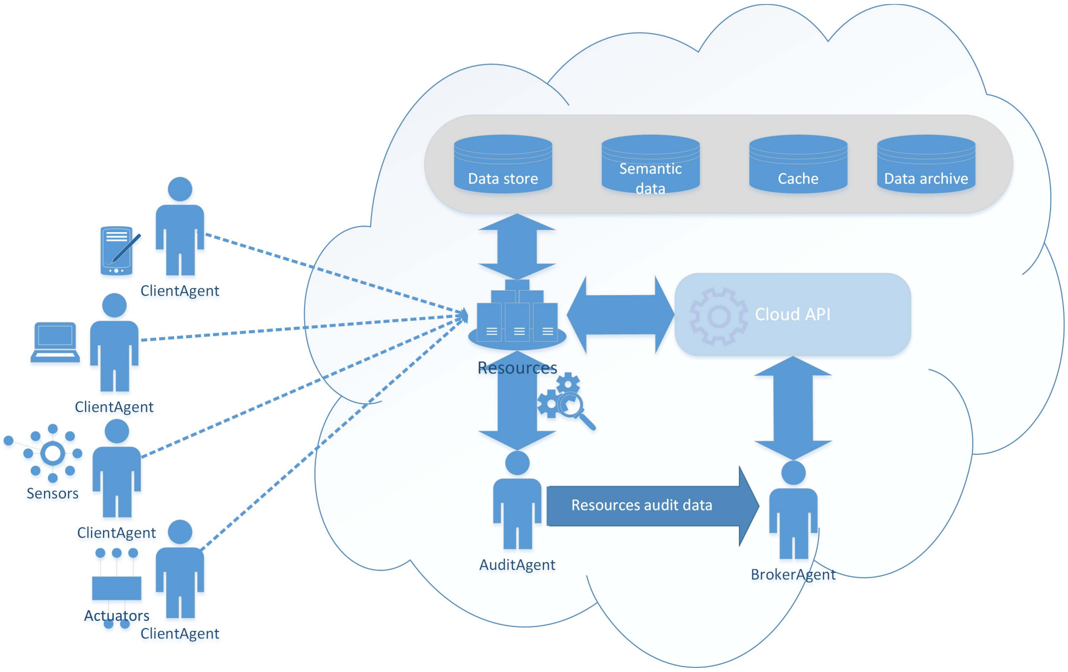

Figure 1. The actors that interact with the multi-agent architecture.

of a data source failure, the system can isolate the affected data sources and any important information is inferred from the neighboring devices based on the localization service.

Beside moving some of the logic to the edge of the network it is imperative to take into account the geographical distribution of the resources that need these services [11]. The services distribution depends on the audit performed on that parts of the network characterized by same location and the number of clients that the system should service. In the edge computing (EC) a important role is played by context awareness, so that a resource it is aware of its current location, the surrounding resources and where is the case the resource can be aware of the entire surrounding context.

Even though the services are brought as close as possible to the end-users, parts of the data requested by the users is sometimes dispersed on the main data store. In [12] is proposed a solution based on the data replication mechanism provided by distributed databases. The important data sets stored in the main data center is replicated on cheap commodity hardware situated at the edge of the network closer to the edge services, this way reducing the network latency and core network utilization.

By utilizing the multi-agent system the end-users should benefit from a good quality of service (QoS) enhanced by a pleasant experience which respects high standards of quality of experience (QoE) [13]. The QoE term is usually used to express the overall experience of using a multimedia system, however, considering the multitude of devices that are encompassed by the Internet of Things framework it is worth noting that these devices are used to improve the user's experience in an environment.

In order to scale the architecture, a large number of instances are launched to distribute the system load. The system can be scaled as long as the cloud provider has the necessary resources available, otherwise additional resources may be launched using another public cloud provider or a private cloud. A solution for this problem is offered by the open-source platform mOSAIC [14], which offers an application programming interface and a platform for developing applications that can be hosted on multiple cloud providers. The resources provisioning mechanism is based on a multi-agent architecture [15], that enables the platform to rent cost effective heterogeneous resources from different cloud providers based on a service level agreement (SLA) provided by the user. The best solution is matched by a reasoning module that operates on a Cloud Ontology.

The Aneka platform [16] is another cloud computing platform that supports services oriented applications. The platform offers a configurable services container, services discovery and load balancing, therefore the developers can focus on the application development and less on the infrastructure management.

Given the fact that it is impossible for a single cloud provider to scatter its data centers around the globe to offer computing power as close as possible to their clients, a viable solution for this problem is a federation of clouds [17]. By using a federation of clouds, the resources are rented

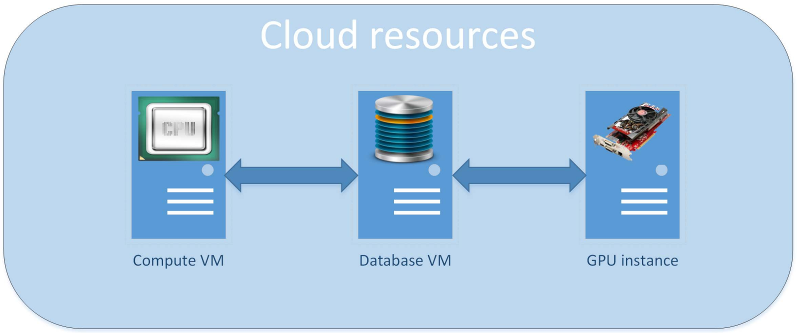

Figure 2. The types of virtual machines used in the IoT infrastructure.

from the closest available cloud provider thus enabling the infrastructure to offer fast and localized support.

III. Architecture

The proposed architecture for the multi-agent system uses the agents as independent and mobile entities, which are specialized to solve specific problems introduced by the Internet of Things paradigm. As seen in Figure. 1, a part of the agents work together to gather, annotate, process and store data in a cloud environment and the other part of the agents are distributed on client devices. By having the agents distributed both on the client devices and on the cloud infrastructure, the communication and the management of the agents is delegated to the core implementation of the multi-agent platform, thus the resulting system can be regarded as an integrated environment.

The information, which my be sent as processed or raw data, flows from the clients to the cloud endpoints that are routing the information to the available resources. When the information reaches one of the end-points of the cloud, it can be semantically annotated and sent for further processing. When the context aware component is active, the information related to a particular context can be used for updating the current context if the information received contains relevant values that depict a modification of the actual state.

An important role in a system that handles massive quantities of information is played by data cache. The data cache stores the most accessed information in memory, so that all requests trying to read the information from a database are routed directly to the information stored by the cache system. Some of the agents which handle a large amount of information are developed with an of the shelf caching system to improve system performance.

When data gathered from the devices becomes obsolete it is transferred to the data archive database. The data stored in this database is aggregated and compressed to save storage space and to facilitate rapid information retrieval of the archived data. After a long period of time the data archive is subjected to a purge operation that aims to remove the existing information which has not been used in a long time.

The multi-agent system is developed for general purpose when it comes to data processing, but it can be used to various domains like ambient intelligence, smart city management, ambient assisted living and for supervising the industrial processes.

IV. Infrastructure management

The IoT architecture needs three types of virtual machines, which have to accomplish a specific task in the architecture:

- Compute VM - compute optimized instances which offer the highest performing processors. These types of instances are used for the data collection end-points. The multi-agent container runs on these instances, where agents specialized for data processing are instantiated.
- Database VM - storage optimized instances that are able to offer support for memory-intensive and random I/O operations. Several instances of this type can be grouped into clusters depending on the technical specifications provided by the company which delivers the database software solution.
- GPU instances - instances that have attached a graphical and general purpose GPU. This type of instance is used for data mining [18], data-matching [19], intrusion detection [20][21] and learning[22].

1) Managing compute instances: In [23] we have performed a suite of benchmark tests on a prototypical implementation of the IoT architecture, the results define the maximum number of connections that can be handled by different types of virtual machines rented from Amazon EC2. The maximum threshold can be used by the BrokerAgent, as a trigger, to launch a new instance in order to distribute the system load.

The standard images, that are built with the purpose of supporting the architecture, are bundled with necessary software packages to start the architecture. When a non

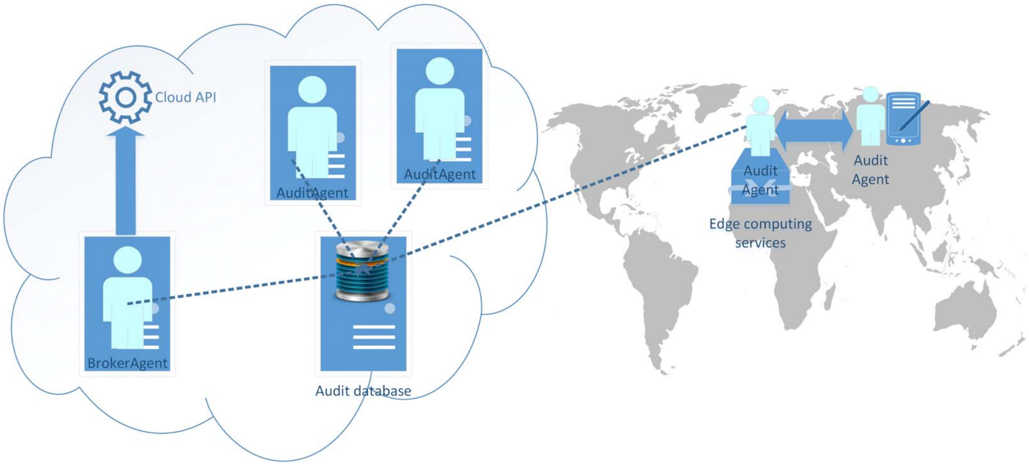

Figure 3. The IoT infrastructure audit.

standard image is launched, the BrokerAgent handles the deployment phase as well the registering phase, where the image is marked as *ready* to process the data collected from the devices.

Event though the proposed architecture targets the Scala programming language and Akka toolkit, to benefit from the highly concurrent and distributed nature of this system, the out of the box scheduling mechanism [24] implemented by this software stack can be greatly improved with the addition of load balancers [25]. Therefore, the BrokerAgent is designed to handle this scenario by grouping the targeted virtual machines, automatically, requesting a load balancer from the cloud provider using the provided API.

Another benefit resulted from the usage of a multi-agent system on top of the Akka toolkit is the communication protocol which is supported on default by every agent. This way the agents are able to exchange information regardless of their location, hence, the infrastructure might be exposed to a high network traffic when the actors are deployed on a virtual machine that has been launched in a different geographical region.

The audit operations performed on the entire infrastructure are important, because they offer valuable information about the resources utilization, system loading, the number of newly added/removed devices and the geographical distribution. In order to offer edge computing services the system should be aware of the resources' positions which are useful when a local system failure occurs. By knowing the exact location of the resources, the edge services can be dynamically managed and deployed offering a flexible alternative when a local resource becomes unavailable.

2) Managing database instances: Database instances represent an important instance type for the architecture, because they handle data persistence. The database instances need to be configured according to the database software which is launched on these instances. When database instances are grouped together in a cluster or ring, they require additional instructions to join the group after a restart. Therefore, the BrokerAgent has to mitigate the interaction between the cluster and the new database instance. For applications targeting the Java platform, the JSch [1] library can be used to send commands over SSH to a virtual machine.

The operations executed via the command line interpreter, like joining a cluster, gathering audit data or a force data replication, are executed by any of the agents running on the local agents container or running on a remote virtual machine, thus allowing the multi-agent system to have total control over the launched instances. Therefore, for security reasons, the interactions with the underlying operating system should be executed only by the agents which have a strict role in the infrastructure management like AuditAgent or BrokerAgent.

Some NoSQL databases, like RIAK, recommend the usage of a load balancer [26] as a best practice, because every node is able to handle the request, hence the incoming requests can be routed to any available instance.

3) Managing GPU instances: The GPU instances are usually very expensive to run, hence only a handful of cloud providers offer such instances, some of them are big players in cloud computing domain like Amazon, Nimbix, Peer1, Penguin computing and Softlayer. Because currently there is no technology available for GPU virtualization, the GPU instances require a physical GPU board available for every virtual machine instance that is launched.

The data processing using a GPU is very fast, hence the data traffic between the GPU instances and the data store can simply overcome the infrastructure network capability. To solve this problem, the GPU instances need to be started in the same data center where the data which is subject

[1] http://www.jcraft.com/jsch/

to processing is available. Also, the network connection between instances should offers support for a higher packet per second to speed up the data transfer. A better solution for networking is a cluster network where the instances launched in the same cluster group are started on the same physical machine, so that the cluster network provides high network bandwidth and low latency for data transfer between instances.

V. Infrastructure audit

The system audit is important in an environment where hundreds of virtual machines are the backbone of a multi-agent architecture (Figure 3). The audit is useful for automatic maintenance as well as human operators, so that the data gathered by the audit operation can be used to maximize the infrastructure performance and to reduce costs. The audit operations offer important data about the network latency, CPU usage, GPU usage, RAM memory loading, and it can also verify if the resources rented from the cloud provider respect the service level agreement (SLA) [27]. The multi-agent system relies on the cloud infrastructure to operate at best performance parameters, so that any change in the infrastructure components can affect the system's overall performance, thus the audit operations must be scheduled after every start/restart of a virtual machine to validate the changes [28].

Because the proposed architecture was developed to target a wide-spread geographical area, the audit information is useful to determine which area offers a high QoS/QoE for end-users. Thus, the edge computing services offered for certain geographical areas can be configured for best performance results.

The AuditAgent uses a database to store different audit results based on the execution time. Storing audit results for a long period of time is useful to identify the parts of the infrastructure that are extensively used so that further actions may be taken to improve performances by moving some of the resources situated in areas with low traffic to the area where the infrastructure is experiencing peaks of traffic.

Considering the vast applications of the Internet of Things in domains like health care, smart city, ambient intelligence and industrial, a lot of private information is transferred between agents which are situated either on the client side or in the cloud. To protect the private information of the end-users it is imperative to execute security audit operations[29] over different components of the system. The heterogeneous character of the devices that interact in the IoT framework offer a wide range of possibilities for a cyber-attack, therefore periodic audit operations are compulsory. The security audit needs to target the client-side applications, edge computing services and the cloud infrastructure periodically, so that any attempt or unauthorized usage of the system has to be identified and reported.

The detected intrusion attempts can be automatically handled by an agent which has a set of rules implemented to deal with such situations. Real time intrusion detection in the Internet of things has been implemented in SVELTE [30] project with an impressive detection rate of almost 100%, thus the task of detecting potential attackers can be assigned to an agent which has full access to the data exchange inside of the multi-agent system.

VI. Conclusion and future work

This paper presents a method for a cloud infrastructure management based on a multi-agent solution. The solution aims to overcome the issues encountered when managing an on-demand cloud infrastructure. Because the entire logic for infrastructure management is detached to specialized agents, that operate on data collected from the system logs or from user defined constraints, it can be used as a standalone component for future projects, which are designed to operate using the infrastructure as a service (IaaS) paradigm.

Another important aspect underlined in this paper is related to infrastructure audit and the positive impact on the system performance when the data collected during the audit operation is used to dynamically reconfigure the system. The audit information is also useful when dealing with location aware services, because it offers valuable information about the current state of the machines which are hosting the edge network services. Hence, the regions which are characterized by heavy network traffic can benefit from a new set of computing resources during traffic peak .

As future work, we are aiming to add a feature that analyses the instances usage patterns, so that the agents will be able to prepare a pool of instances ready to be configured for balancing the incoming requests, thus reducing the unnecessary boot time which delays the newly started instance response time with up to several minutes.

Acknowledgment

The work of the first author was supported by the strategic grant POSDRU/159/1.5/S/137750, "Project Doctoral and Postdoctoral programs support for increased competitiveness in Exact Sciences research" cofinanced by the European Social Fund within the Sectoral Operational Programme Human Resources Development 2007–2013.

References

[1] Bhathiya Wickremasinghe, Rodrigo N Calheiros, and Rajkumar Buyya. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 446–452. IEEE, 2010.

[2] Dominique Guinard, Christian Floerkemeier, and Sanjay Sarma. Cloud computing, rest and mashups to simplify rfid application development and deployment. In *Proceedings of the Second International Workshop on Web of Things*, page 9. ACM, 2011.

[3] Matthias Kovatsch, Simon Mayer, and Benedikt Ostermaier. Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 751–756. IEEE, 2012.

[4] Nathalie Mitton, Symeon Papavassiliou, Antonio Puliafito, and Kishor S Trivedi. Combining cloud and sensors in a smart city environment. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):1–10, 2012.

[5] John Soldatos, Martin Serrano, and Manfred Hauswirth. Convergence of utility computing with the internet-of-things. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 874–879. IEEE, 2012.

[6] Nathan Farrington and Alexey Andreyev. Facebooks data center network architecture. In *IEEE Opt. Interconnects Conf*, pages 5–7. Citeseer, 2013.

[7] Todd Hoff. Scaling twitter: Making twitter 10000 percent faster. *High Scalability*, 2009.

[8] Josep M Pujol, Georgos Siganos, Vijay Erramilli, and Pablo Rodriguez. Scaling online social networks without pains. In *Proc of NETDB*, 2009.

[9] Meredith Farkas. Going where patrons are: Outreach in MySpace and Facebook. *American Libraries*, 38(4):27, 2007.

[10] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[11] Richard Manning. Dynamic and distributed managed edge computing (mec) framework, May 20 2004. US Patent App. 10/850,291.

[12] Yi Lin, Bettina Kemme, Marta Patino-Martinez, and Ricardo Jimenez-Peris. Enhancing edge computing with database replication. In *Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on*, pages 45–54. IEEE, 2007.

[13] Reza Farrahi Moghaddam and Mohamed Cheriet. A note on quality of experience (qoe) beyond quality of service (qos) as the baseline. *arXiv preprint arXiv:1407.5527*, 2014.

[14] Beniamino Di Martino, Dana Petcu, Roberto Cossu, Pedro Goncalves, Tamás Máhr, and Miguel Loichate. Building a mosaic of clouds. In *Euro-Par 2010 Parallel Processing Workshops*, pages 571–578. Springer, 2011.

[15] Salvatore Venticinque, Rocco Aversa, Beniamino Di Martino, and Dana Petcu. Agent based cloud provisioning and management-design and prototypal implementation. In *CLOSER*, pages 184–191, 2011.

[16] Christian Vecchiola, Xingchen Chu, and Rajkumar Buyya. Aneka: a software platform for .net-based cloud computing. *High Speed and Large Scale Scientific Computing*, pages 267–295, 2009.

[17] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N Calheiros. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *Algorithms and architectures for parallel processing*, pages 13–31. Springer, 2010.

[18] Wenjing Ma and Gagan Agrawal. A translation system for enabling data mining applications on gpus. In *Proceedings of the 23rd international conference on Supercomputing*, pages 400–409. ACM, 2009.

[19] Ciprian-Petrişor Pungila, Mario Reja, and Viorel Negru. Efficient parallel automata construction for hybrid resource-impelled data-matching. *Future Generation Computer Systems*, 36:31–41, 2014.

[20] Giorgos Vasiliadis, Spiros Antonatos, Michalis Polychronakis, Evangelos P Markatos, and Sotiris Ioannidis. Gnort: High performance network intrusion detection using graphics processors. In *Recent Advances in Intrusion Detection*, pages 116–134. Springer, 2008.

[21] Sun-il Kim, William Edmonds, and Nnamdi Nwanze. On gpu accelerated tuning for a payload anomaly-based network intrusion detection scheme. In *Proceedings of the 9th Annual Cyber and Information Security Research Conference*, pages 1–4. ACM, 2014.

[22] Lutz F Gruber and Mike West. Gpu-accelerated bayesian learning and forecasting in simultaneous graphical dynamic linear models. Technical report, Technical report, Department of Statistical Science, Duke University, 2014.

[23] Bogdan Manaţe, Victor Ion Munteanu, and Teodor-Florin Fortiş. Towards a Scalable Multi-agent Architecture for Managing IoT Data. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on*, pages 270–275, 2013.

[24] Michael Bevilacqua-Linn, Maulan Byron, Peter Cline, Jon Moore, and Steve Muir. Sirius: distributing and coordinating application reference data. In *Proceedings of the 2014 USENIX conference on USENIX Annual Technical Conference*, pages 293–304. USENIX Association, 2014.

[25] Meenakshi Sharma, Y Anitha, and Pankaj Sharma. An optimistic approach for load balancing in cloud computing. 2014.

[26] Riak load balancing. http://docs.basho.com/riak/1.3.1/cookbooks/Load-Balancing-and-Proxy-Configuration/. Accessed: 2014-09-08.

[27] Djamila Ouelhadj, J Garibaldi, Jon MacLaren, Rizos Sakellariou, and K Krishnakumar. A multi-agent infrastructure and a service level agreement negotiation protocol for robust scheduling in grid computing. In *Advances in Grid Computing-EGC 2005*, pages 651–660. Springer, 2005.

[28] Frank Doelitzscher, Christian Fischer, Denis Moskal, Christoph Reich, Martin Knahl, and Nathan Clarke. Validating cloud infrastructure changes by cloud audits. In *Services (SERVICES), 2012 IEEE Eighth World Congress on*, pages 377–384. IEEE, 2012.

[29] Xiangyu Sun and Changguang Wang. The research of security technology in the internet of things. In *Advances in Computer Science, Intelligent System and Environment*, pages 113–119. Springer, 2011.

[30] Shahid Raza, Linus Wallgren, and Thiemo Voigt. Svelte: Real-time intrusion detection in the internet of things. *Ad hoc networks*, 11(8):2661–2674, 2013.