A Conceptual Framework for Simulating Autonomic Cloud Markets

Simon Caton[1], Ivan Breskovic[2], and Ivona Brandic[2]

[1] Karlsruhe Institute of Technology, Germany
simon.caton@kit.edu
[2] Vienna University of Technology, Austria
{breskovic,ivona}@infosys.tuwien.ac.at

Abstract. One of the major challenges facing the Cloud paradigm is the emergence of suitable economic platforms for the trading of Cloud services. Today, many researchers investigate how specific Cloud market platforms can be conceived and in some cases implemented. However, such endeavours consider only specific types of actors, business models, or Cloud abstractions. We argue that market platforms for the Cloud paradigm cannot (yet) be rigidly defined, and require the ability to progress and evolve with the paradigm. In this paper, we discuss an alternative approach: autonomic markets. Autonomic markets automatically adapt to changed environmental conditions based upon a given concept of "performance". We describe the autonomic MAPE loop in the context of electronic markets and consider the types of a knowledge produced and required for decision making. Finally, we present a conceptual framework for a market simulator, a critical tool for autonomic markets, based upon experiences using the GridSim simulation tool.

Keywords: Cloud Computing, Electronic Markets, Autonomic Computing, Computational Economics, Market Simulation.

1 Introduction

Today electronic marketplaces are challenged by a highly dynamic context: high product variability, unpredictable participant behaviour, and the emergence of new actors as well as actor types. Consequently, market situations that have previously been unimaginable will arise and novel theories and paradigms are needed to facilitate and control them. Examples of new market contexts are in the domains of Smart Grids, the Internet of Services and Cloud Computing as well as Social and Collaborative Environments. Many of these domains are already or will become inherently reliant upon the economic systems represented by electronic markets that can address their allocation problems.

However, a key challenge is that we do not know the most fitting anatomy of an appropriate market platform. Even assuming the existence of an "adequate" platform, a subtle or disruptive change within the domain can mean that the platform no longer satisfies its domain requirements. Today, electronic market

M. Yousif, L. Schubert (Eds.): Cloudcomp 2012, LNICST 112, pp. 92–102, 2013.

platforms are static and not conceived to handle changes in their domain or elements of uncertainty in their architecture. Therefore, we argue that the market engineering process cannot simply extend traditional approaches, but requires a new methodology. In [6], we introduced an alternative: Autonomic Markets; a goal orientated approach for market engineering to enable autonomic adaption.

To evaluate the vision of an autonomic market, we need an experimental platform and it is not possible to simply map existing markets. Therefore, an appropriate research methodology for their study is simulation, as it enables the creation of what if scenarios and the ability to observe how autonomic adaption evolves a market over time. Through simulation, we can implement economic and management models of autonomic markets to access their performance (with respect to goal fulfilment), tractability and feasibility for different adaption strategies. Although real-life markets cannot be mapped directly for an autonomic market, their traces as well as event and trading catalogues can act as input data as a means to drive specific what-if scenarios. In this paper, we reflect upon the lessons learned in [6], in order to create a set of requirements and conceptual architecture for an autonomic market simulation tool.

This paper is structured as follows: Section 2, presents an overview of the autonomic market vision; Section 3 discusses related work; Section 4 presents a case study of GridSim in simulating markets; Section 5 draws upon this case study articulate a conceptual framework for a autonomic market simulation tool; finally Section 6 summarises the paper and discusses future work.

2 Autonomic Markets: An Overview

Our vision of an autonomic market platform is that *institutional forms* and underlying *infrastructures* can be *adapted* at runtime with the goal of improving a given concept of *"market performance"*. Infrastructure adaption refers to modifying the computational infrastucture of the market platform that enables its core functionality. This, for example, includes computational resources, delivery mechanisms, communication channels, security procedures, etc. Institutional adaption relates to modifying market components such as rules of participation, allocation and pricing mechanisms, and tradable artefacts. Market performance is characterised through specific goals that can include market liquidity, immediacy, stability, security, participant welfare, energy efficiency, allocation efficiency, etc. An institutional market form, i.e. an instantiated parameterisation of market components, is what we refer to as a market configuration. Therefore, infrastructure adaption in our context is what we commonly understand as elastic infrastructures in the Cloud paradigm and institutional adaption is a change in one or more parameter settings and hence a change in market configuration.

By making market platforms autonomic, we hope to enable evolution beyond initial design principles by "learning" or adapting towards ideal configurations, possibly with certain levels of oscillation. Through this ability we can begin to explore, analyse and evaluate autonomic market platforms as well as the impact of different market configurations and goals. Autonomic adaption will enable different capabilities in economics such as autonomic (economic) mechanisms, self-regulation, fault tolerance, as well as autonomic market (re)engineering. To make a market autonomic, we propose applying the extended autonomic control loop, i.e. the MAPE-K cycle, to a complex array of parameterisable (hence adaptable) economic components, where each component can be imagined as a traditional managed element within the Autonomic Computing paradigm. We specifically focus on market platforms for the domain of Cloud computing, as it is well defined domain with respect to its requirements on a market platform. A successful implementation of the Cloud computing methodology (i.e. fulfilling its promise of computing as a commodity) is only possible with appropriate methodologies and techniques for the definition and management of Cloud market platforms. We believe that the application of our autonomic market concept can tackle the challenges of the paradigm. In the remainder of this section, we first provide a motivating example for autonomic markets, before briefly describing the application of the MAPE-K loop to an economic system.

2.1 Motivating Example

Consider that a market provider decides that a market goal is the completion of a certain number of trading transactions per unit of time. Observing the market's adherence to this goal is trivial. However, many different events can cause deviations from this goal, some of which may be exogenous (e.g. external outages) and others observable within the platform (e.g. infrastructure bottlenecks, a reduction in the number of active participants, etc.). To remedy deviations with respect to this goal, several different options can be considered depending on the cause of the deviation. Examples are: (1) scaling computing nodes up or the infrastructure as a whole out to increase the number of concurrent trades per unit time, or reduce the time needed to process individual trades; (2) tuning the matching algorithm to reduce the compute time (e.g. applying a heuristic instead of an optimal algorithm); (3) purging the order book(s) to remove redundant data; and (4) tuning allocation mechanism properties (e.g. the maximum number of entries in the order book, the clearing or pricing functions). Moreover, combinations of these options are also valid, as well as more aggressive adaptions such as changing the allocation mechanism for another. Although in this example a simple market goal has been chosen, there are many more complicated goals as well as goal combinations that can have large impacts upon the stability of a market (e.g. goals concerning market liquidity, revenue, efficiency, etc.).

2.2 Applying the Autonomic Loop to a Market Platform

The MAPE-K loop contains five elements: monitoring, adaptation, planning, execution, and a knowledge management components, which we explain below.

Monitoring data is critical for the instrumentation of any form of adaption. In [6], we defined a monitoring methodology for an autonomic marketplace and demonstrated how the performance of a market platform can be measured with respect to a specific set of market performance indicators. This task is performed by monitoring sensors, which gather low-level monitoring data from the market middleware and implementation of the market model. Using the predefined mappings, the measured monitoring data is mapped to the higher level performance indicators, which is then used to assess the market performance.

The main purpose of a **knowledge** component is to store, manage and analyse real-time monitoring data and experiences from previous adaptations. The knowledge gathered in this process can be (1) empirical, i.e. derived from the observations on the market (e.g. infrastructure status, payoffs from previous adaptations, etc.), (2) contextual, i.e. instance-specific (e.g. initial/desired configurations and business models), and (3) institutional, i.e. concerning the economic anatomy of the marketplace (e.g. valid alternative configurations and market rules, constraints and regulations). While empirical knowledge is gathered through monitoring and logging techniques, contextual knowledge is (initially) set by the market administrator. Institutional knowledge is defined partially by the market administrator and partially established based upon contextual knowledge and changes (i.e. evolutions) within the marketplace.

The **analysis** phase is used to analyse mapped data from the monitoring sensors to derive possible actions for market adaptation in order to improve market performance with respect to a set of goals. As already mentioned, there are two main adaption options: the market's infrastructure and its configuration. Finding which of these options is the most fitting is, however, not trivial. Autonomic adaptation of infrastructure properties has already been discussed in a large body of literature (e.g. [1,8,10]). This, however, is not the case for institutional adaption. To facilitate institutional adaption, we need to understand what different market configurations mean for the fulfilment of a given set of goals, which can be achieved through simulation to enable the analysis of what-if scenarios to determine and assess adaption options.

The **planning** phase of the autonomic adaptation cycle includes two important steps. Firstly, it identifies the most suitable adaptation path(s) for the execution of the infrastructure and/or institutional changes by leveraging contextual knowledge. Secondly, as an adaptation path may include more than one market component or steps, it determines the order and timing of the adaptations to be instrumented. This may result in multiple rounds of the adaptation cycle with the goal of observing how single changes impact the market performance and ultimately lead to an iterative adaption process.

The **execution** phase is the execution of an adaption path. In the case of an infrastructure adaption, this relates to an interaction with the resource fabrics through the platform middleware. For institutional adaption, it refers to a check point of the current market status, a new parameterisation of the market configuration, and redeployment (if necessary) of effected market components.

3 Related Work

For positioning our work within the state-of-the-art, we briefly describe existing research on electronic markets and classify it into two categories: (1) applying foundations of autonomic computing to the implementation of electronic markets, and (2) simulating electronic markets for Grid and Cloud environments.

3.1 Autonomic Markets

To enable the flexibility promised by the Grid and Cloud paradigms, market platforms have to be adaptive and sustainable. We argue that this can be achieved with autonomic (self-* [13]) capabilities. Several early works served as groundwork for prototypical implementation of autonomic aspects in complex systems. For example, [16] discuss the need for autonomic capabilities of distributed service systems and briefly outline the application of the self-* capabilities in this context. Today, autonomic computing is used primarily to address technical issues to make systems autonomic, e.g. negotiation protocols to make Grid or Cloud services self-adaptive [5] or consider autonomic service management frameworks without explicitly considering economic methods (e.g. [12,15]).

The idea of applying economic methods and considerations to autonomic systems was initially proposed by [11]. However, current research focuses on specific economic issues and only partially considers the aspects needed to make marketplaces autonomic. For example, [19] proposed a self-organising resource allocation mechanism in dynamic Application Layer Networks (ALNs). They do not, however, consider issues such as the adaptation of the market itself, but rather the optimisation of a small piece: the allocation mechanism. Similarly, [17] propose mechanisms that can adaptively adjust parameters based on past participant behaviour. An example of economically-inspired market infrastructures is provided by [9] who present a self-optimising infrastructure platform for service delivery using economic (congestion-based) pricing. Yet they, consider only the infrastructure, and not the market itself. [4] study the mapping of high-level business objectives to lower level objectives to enable autonomic access optimisation for databases via an economic scheduler.

3.2 Simulating Electronic Marketplaces

Simulation of electronic markets for Grid and Cloud computing has been discussed in several large research projects, including SORMA [19], GridEcon [18] and 4CaaSt [14]. [19] developed a market simulator to compare centralised and decentralised service allocation mechanisms in market scenarios according to a defined set of metrics. In their work, they considered complex interdependencies that are broken down into two interrelated markets, namely a service market, which involves trading of application services, and a resource market, which involves trading of computational and infrastructure resources such as processors, memory, etc. [18] present the GridEcon platform - a testbed for designing and evaluating economics-aware services in a commercial Cloud computing setting. The authors assume the difficulties in predicting the context of a service market and motivate development of an environment for evaluating its behaviour in an emulated market platform. The platform is composed of the Marketplace, which allows trading goods using different market mechanisms, and the Workflow Engine, which enables a simple composition of a market environment by describing the service interactions between economics-aware services. [14] discuss a mechanism for the resolution of the customers' requirements that enhances the process of selecting Cloud services from the business point of view. The work is related to the 4CaaSt project and aims to create a PaaS Cloud platform that supports the optimised and elastic hosting of Internet-scale multi-tier applications.

[2] discuss a framework for modelling and simulating service-oriented applications and autonomic policies for service provisioning and resource orchestration for Application Layer Networks in utility computing environments. The approach is evaluated within CATNETS project and investigates the use of an economic model (Catallaxy) in distributed environments like Grids and P2P networks. [20] discuss the design of a simulator with a set of features for simulation of Grid testbeds as an extension to GridSim. They model heterogeneous computational resources of variable performance, scheduling of jobs based on various policies, differentiated network service, and workload trace-based simulation.

Although many of these market simulators successfully address some of the main challenges of electronic markets in distributed environments, they are fairly static and do not have any autonomic capabilities. Therefore to orchestrate and evaluate autonomic markets, a more flexible simulation approach is necessary.

4 A Case Study for Market Simulation Using GridSim

In [6], we used GridSim as a means to explore how a market could be monitored as first steps towards adding autonomic capabilities to an electronic market. We selected GridSim for a variety of reasons: (1) it implements numerous mechanisms for resource allocations [3] as well as interfaces for implementing additional mechanisms; (2) it is designed as an extensible multi-layer architecture which allows new (technical) components or layers to be added and integrated [7]; (3) it allows different classes of heterogeneous resources; and (4) as an open-source toolkit it has already been used widely. Although GridSim simulates Grid resource and networks and not the Cloud computing paradigm directly, it is important to note that these two contexts do not differ significantly, as the core techniques for matching buyers to sellers are equal and independent of technical paradigm. Using the layered architecture of GridSim, we implemented three monitoring sensors for: market mechanisms, the market in general, and computational infrastructure of the platform as extensions to the existing GridSim layers to monitor the infrastructure and institutional performance of the market.

The **Mechanism sensor** monitors the performance of a market mechanism, which includes revenue, the number of resource allocations, and average price for a single unit of resource. The actual allocation is handled by GridSim. The mechanism sensor uses the GridSim interfaces to receive a notification of an allocation, i.e. a match of a bid to an ask. Once a resource is allocated, the sensor receives and stores information about the allocation in the knowledge component. Using the same GridSim interfaces, the sensor also gathers mechanism-specific information like number of bids and asks awaiting allocation.

The **Market sensor** gathers market information, for example data concerning the past and current number of sellers and buyers on the market and information concerning the resources traded. This is achieved by using GridSim interfaces of the architecture layers responsible for resource and job management.

The **Infrastructure sensor** monitors the usage of computational resources. In particular, it monitors the utilisation and performance of the underlying operating system and hardware infrastructure. For example, processor utilisation and speed, number of threads, memory usage, hard-disk usage, etc. The infrastructure monitoring is based on the interfaces defined by the *java.lang.Management* package, which is a management interface for monitoring and management of the Java virtual machine as well as the host operating system.

Despite the large flexibility of GridSim, its numerous interfaces and multi-layered architecture, creating a simulation scenario is not a trivial task, as many market actions and the creation of communication objects between the layers of GridSim are left to the user. However, GridSim does provide a small set of examples that illustrate the implementation of simple trading scenarios. In our feasibility study, we applied one of the example scenarios. This example allowed us to control basic trading properties, i.e. the number of buyers and sellers in the market and the number of requests per buyer, etc., which for our purposes was adequate. It also enabled the construction of a market, establishment of participants and resources, and provided an easy platform upon which to implement a monitoring framework. It was also straightforward to implement a basic benchmark scenario to test the monitoring framework.

However, we encountered difficulties when we created more realistic and elaborate scenarios, for example: different participant types (e.g. malicious users, market makers, speculators, monopolists, and other strategic behaviours); more complicated trades, i.e. multiple resource entities; dynamic context: adding or removing participants or resource types at runtime; and engineering aspects like market growth or contraction. When trying to create such scenarios, we encountered runtime exceptions for the following reasons: (1) GridSim expects the number of users to remain fixed; (2) it is not possible to change the quantity of resources that sellers offer and buyers request, i.e. total supply and demand is predefined; (3) new resource types cannot be added at runtime; (4) it is not possible to manage the timings of the bid/ask submissions, this is controlled by GridSim's event handlers, which makes it difficult to implement users with specific participation strategies. Through our efforts to counter these as well as other challenges, we were moving away from GridSim's initial use case, eventually making it impossible to control and extend further. Consequently, we were no longer confident that changes in the market were engineered by us as opposed to errors in the GridSim runtime. It would be easy to say that this is a failing of GridSim, but our scenarios were straying outside of GridSim's scope.

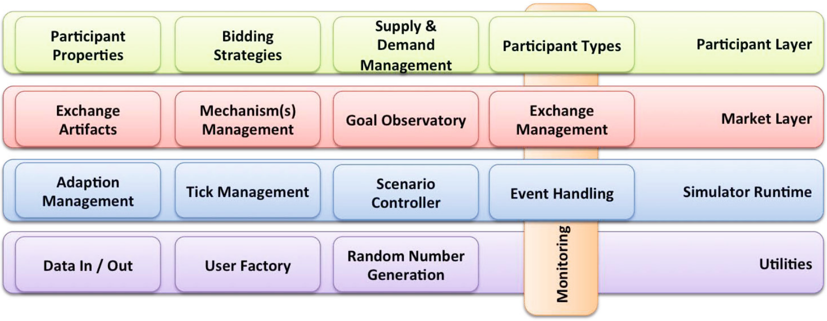

Fig. 1. Conceptual Framework of a Simulation Environment for Autonomic Markets

5 Conceptualisation of a Simulator for Autonomic Markets

Based upon our experiences with GridSim, we realised that trying to simulate different aspects for the study of autonomic markets needs a more flexible simulation approach. In Section 3, we discussed some alternatives to GridSim, but failed to see the ability to capture all aspects that we feel are necessary without significant effort in the extension of an approach. In this section, we propose a conceptual architecture for an autonomic market simulator that will act as a testing environment for the future studies. Fig. 1 illustrates the layers and components of our proposed simulation architecture, which are as follows:

The **Monitoring Framework** captures key information on the market platform through links to the Participant, Market and Simulator layers, and makes this information available to the components that require it (e.g. the Goal Observatory). Monitoring information here captures the state of: mechanisms, the market in general and the computational infrastructure, as described in [6].

The **Participant Layer** captures the aspects necessary to represent market participants as well as their various nuances and differentiating factors. The key component is *participant type*, which identifies whether a participant is a consumer, provider, prosumer, or broker. It also enables different participant flavours like market makers, speculators, monopolists, aggressive and passive participants. In accordance to the typical market simulators, we define *bidding strategies*, as well as the management of *supply and demand*. We use the word "management" to illustrate that this is not a statically defined process, but entails stochastic and dynamic behaviours such as participants joining or leaving the market, as well changes in their individual properties and requirements over time. *Participant properties* capture additional information needed for each participant type, e.g. range of wealth, resource types offered/desired, etc.

The **Market Layer** defines the components to implement an electronic market. This includes: the *artefacts* to be traded, including their type, quantity and period of availability or desirability; different allocation *mechanisms* like the English or Continuous Double Auction, but also the means to create custom mechanisms and have multiple active mechanisms. Mechanism management here refers to the programming constructs to transparently include any arbitrary mechanism by exposing a standard interface. A mechanism manager controls how bids and asks are passed to a mechanism and when instances are created and destroyed; a *Goal Observatory* for defining goals and keeping track of their adherence via the monitoring framework; a *exchange management* that keeps track of all incoming asks and bids, matches, as well as one or more active mechanisms; and finally, *adaption management* as an instantiation of a market adaption component.

The **Simulator Layer** is the basis for the simulator. It includes a singleton *event handler*, as this enables a simple programming model without the need for complex thread or concurrency management, and a *tick manager* to control "time" in the simulator as a sequence of discrete periods. In each tick, we invoke participants in a renewed random order, and give them the option to "act", i.e. do something in the market. We also define a *scenario controller*, which through the event handler can instigate new scenarios for observation, based upon the current time. The scenario controller permits us to create issues of instability or change specific settings in the market to study how the market changes, and later how adaption actions have improved or worsened the situation. We can also layer (simple) scenarios to create more complex compound scenarios.

We also define key **utilities** to assist in market simulation. These include: readers for trace data from existing markets to "stimulate" market events or scenarios as well as writers to store monitoring data; a *participant factory* to facilitate the generation of multiple participant types based on a set of input parameters; and as a key premise for all simulators, a *random number generator* which can simply be the inclusion of the Colt Library[1] or similar.

6 Summary

In this paper, we have argued that existing electronic market platforms are insufficient for immature domains like Cloud computing. Therefore, we proposed the concept of an autonomic marketplace platform: the automatic adaption of the economic models of the platform and its underpinning infrastructure based upon a given concept of "market performance". We described how the autonomic MAPE-K loop can be applied to an electronic market platform. Finally, we presented our experiences in trying to build a simulation tool as a premise for the study and evaluation of an autonomic market using GridSim as a case study. However, we encountered too many scenario specific obstacles that merited a bespoke simulation framework. Building on top of the lessons learned from GridSim, we defined a conceptual framework for a market simulator that can facilitate different aspects of study for an autonomic market. These include the definition of destructive scenarios, stochastic events and extended user types. Our future work is the continued investigation of our simulation tool, its on-going development as well as the development of scenarios for its calibration.

[1] `http://acs.lbl.gov/software/colt/`

References

1. Abdelwahed, S., Bai, J., Su, R., Kandasamy, N.: On the application of predictive control techniques for adaptive performance management of computing systems. IEEE Transactions on Network and Service Management 6(4), 212–225 (2009)
2. de Assunção, M., Streitberger, W., Eymann, T., Buyya, R.: Enabling the simulation of service-oriented computing and provisioning policies for autonomic utility grids. In: Veit, D.J., Altmann, J. (eds.) GECON 2007. LNCS, vol. 4685, pp. 136–149. Springer, Heidelberg (2007)
3. Assunçao, M.D.D., Buyya, R.: An evaluation of communication demand of auction protocols in grid environments. Tech. rep. (2006)
4. Boughton, H., Martin, P., Powley, W., Horman, R.: Workload Class Importance Policy in Autonomic Database Management Systems. In: Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks, pp. 13–22. IEEE Computer Society (2006)
5. Brandic, I., Music, D., Dustdar, S.: Service mediation and negotiation bootstrapping as first achievements towards self-adaptable grid and cloud services. In: Grids meet Autonomic Computing Workshop, in Conjunction with the 6th International Conference on Autonomic Computing and Communications, pp. 1–8. ACM (2009)
6. Breskovic, I., Haas, C., Caton, S., Brandic, I.: Towards Self-Awareness in Cloud Markets: A Monitoring Methodology. In: Proceedings of the 9th IEEE International Conference on Dependable, Autonomic and Secure Computing (2011)
7. Buyya, R., Sulistio, A.: Service and utility oriented computing systems: Challenges and opportunities for modeling and simulation communities. In: Annual Simulation Symposium, pp. 68–81 (2008)
8. Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F.D., Buyya, R.: Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software - Practice and Experience (2010)
9. Callaway, R., Devetsikiotis, M., Viniotis, Y., Rodriguez, A.: An autonomic service delivery platform for service-oriented network environments. IEEE Transactions on Services Computing 3(2), 327–331 (2010)
10. Caton, S., Rana, O.: Towards Autonomic Management for Cloud Services based upon Volunteered Resources. Concurrency and Computation: Practice and Experience 23 (2011)
11. Cheliotis, G., Kenyon, C.: Autonomic economics. In: IEEE International Conference on Electronic Commerce, pp. 120–127 (June 2003)
12. Cheng, Y., Leon-Garcia, A., Foster, I.: Toward an autonomic service management framework: A holistic vision of SOA, AON, and autonomic computing. IEEE Communications 46(5), 138–146 (2008)
13. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer 36, 41–50 (2003)
14. Menychtas, A., Gatzioura, A., Varvarigou, T.: A business resolution engine for cloud marketplaces. In: Proceedings of the Third International Conference on Cloud Computing Technology and Science, pp. 462–469 (2011)
15. Oyenan, W.H., Deloach, S.A.: Towards a systematic approach for designing autonomic systems. Web Intelligence and Agent Systems 8, 79–97 (2010)
16. Papazoglou, M., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: State of the art and research challenges. Computer 40(11), 38–45 (2007)

17. Pardoe, D., Stone, P., Saar-Tsechansky, M., Tomak, K.: Adaptive mechanism design: a metalearning approach. In: 8th International Conference on Electronic Commerce, pp. 92–102. ACM (2006)
18. Risch, M., Altmann, J., Guo, L., Fleming, A., Courcoubetis, C.: The gridecon platform: A business scenario testbed for commercial cloud services. In: Altmann, J., Buyya, R., Rana, O.F. (eds.) GECON 2009. LNCS, vol. 5745, pp. 46–59. Springer, Heidelberg (2009)
19. Streitberger, W., Eymann, T.: A simulation of an economic, self-organising resource allocation approach for application layer networks. Computer Networks 53, 1760–1770 (2009)
20. Sulistio, A., Cibej, U., Venugopal, S., Robic, B., Buyya, R.: A toolkit for modelling and simulating data grids: an extension to gridsim. Concurr. Comput.: Pract. Exper. 20, 1591–1609 (2008)