

Towards Self-Awareness in Cloud Markets: A Monitoring Methodology

Ivan Breskovic,^{*} Christian Haas,[†] Simon Caton[†] and Ivona Brandic^{*}

^{*}Distributed Systems Group, Institute of Information Systems, Vienna University of Technology, Austria
{breskovic, ivona}@infosys.tuwien.ac.at

[†]Karlsruhe Service Research Institute, Karlsruhe Institute of Technology, Germany
{ch.haas, simon.caton}@kit.edu

Abstract—Currently, the Cloud landscape is a fragmented, static and shapeless market that hinders the paradigm’s ability to fulfil its promise of ubiquitous computing on tap and as a commodity. In this paper, we present our vision of an autonomic self-aware Cloud market platform, and argue that autonomic market platforms for Clouds can step up to the challenge of today’s status quo. As our first steps towards achieving this vision, we present a market monitoring methodology, which includes a series of realistic market goals, sets of extractable metrics from a market platform and how to map (i.e. combine and transform) metrics to access goal performance such that autonomic adaption of the market could be undertaken. We have extended a known market simulator for distributed infrastructures (GridSim) with relevant sensors. To demonstrate the usefulness of our approach, we simulate a sudden cease in demand for goods in our market platform.

Index Terms—Cloud Markets, Market Monitoring, Self-awareness, Autonomic Computing

I. INTRODUCTION

Cloud computing represents a novel paradigm for the implementation of highly scalable computing resources provided on demand and for the broad customer base [8], [10]. Currently, three cloud types exist, namely *private Clouds*, which are client dedicated and which can easily comply with specific data governance and regulation, *public Clouds*, which are provider owned and managed, and *hybrid Clouds*. From the economic perspective, private Clouds reduce capital expenditure (CAPEX) and operating expense (OPEX) [9]. Public Clouds are accessed based upon established Service Level Agreements (SLAs) stating objectives of service usage and penalties in case of violation of these objectives. Their economic benefit is the reduction in IT service delivery costs through reduced hardware, software and application maintenance costs. Hybrid Clouds define deployment models where private Clouds can be extended to consume public Cloud infrastructures to manage load peaks. Their economic benefits are cost sharing through horizontal and vertical scaling as well as the reduction of maintenance costs as in public Clouds.

A large body of research into the Cloud paradigm has yielded the technological development of Cloud infrastructures, such as development of the appropriate resource management models [10], [13], solutions for the energy efficient management of Clouds [1] as well as security and privacy solutions [7]. Yet, very little research exists on the development of appropriate market platforms in a similar way

to other commodities like energy, stocks, and assets. In the case of public and hybrid Clouds, flexible, dynamic, and low entry-barrier markets with sufficient stability are crucial if computation (in general) is to become a commodity.

Today the static nature of the Cloud “market” means it cannot adapt adequately to dynamic changes in user requirements or new services. Moreover, due to the large variability in resource types and still low number of traders, markets suffer from low liquidity (the ability to easily and quickly sell or purchase a service at a certain price), repelling potential consumers and disadvantaging new providers. Thus, appropriate methodologies and techniques for the definition and management of Cloud market platforms are crucial and will affect whether Cloud computing can emerge as a new state of the art technology for utility computing. We argue that autonomic capabilities (self-optimisation, self-configuration, self-healing and self-protection [20]) are essential for the creation of such market platforms. To create a market platform with each of these capabilities harmoniously working together is, however, a very challenging task.

In this paper, we present our first steps towards the vision of an autonomic marketplace: a novel methodology for the monitoring of cloud markets. Our goal in this paper is to define a methodology that could enable a market platform to be self-aware, i.e. knowledgeable about its state at multiple levels. With this capability, we could then begin to implement autonomic market platforms with the self-* capabilities mentioned above. The main contribution of this paper is therefore: a methodology for monitoring Cloud market platforms, such that they can be made autonomic and self-aware. A large part of this methodology is the identification of monitoring data that is available and useful for self-aware markets, and the necessary mappings to transform these metrics into indicators for the defined market goals. To demonstrate our approach, we utilize GridSim [11], [12], as a widely used tool for the simulation of Grid and Cloud market behaviour. We have extended GridSim with appropriate market and mechanism sensors as well as simple infrastructure sensors. Based upon the monitoring metrics of the market (which are translated from the low level infrastructure measurement) our monitoring model can sense dynamic changes in market behavior, which is the first step towards establishing self-aware and self-manageable market platforms.

This paper is structured as follows. In Section II, we briefly introduce our vision of a self-aware autonomic market as a placeholder for this work. In Section III, we critically evaluate related work to illustrate the gap in research. In Section IV, we introduce our monitoring methodology, which captures realistic market goals that can be analysed, i.e. they do not consist of abstract economic indicators, but rather observable metrics, as well as the mappings that translate metrics into individual goal performance scores. In Section VI, we evaluate our approach by means of a case study with our GridSim implementation. Finally, in Section VII, we summarise our investigation and discuss key future work.

II. VISION: A SELF-AWARE AUTONOMIC CLOUD MARKET PLATFORM

Many large IT companies, such as eBay, Amazon and Yahoo, attract millions of customers worldwide to buy and sell a broad variety of goods and services via *electronic* marketplaces. This trading model has many advantages over traditional marketplaces. For example, it allows users to place their bids at any time and from any geographical location over the Internet. Furthermore, electronic markets often offer a broad scope of products and services with easy of access, resulting in a large number of participants, and consequentially, a potential for low prices as well as operational costs.

Although the current Cloud realisation offers a simple, fast and inexpensive way to bring consumers and providers together, it also suffers from many challenging situations. These include low market liquidity (caused by broad resource variability and a low number of market participants) [23], a fragmented array of independent providers, and few standard mechanisms for unilateral provider adoption and use. In order to address these challenges and create a Cloud marketplace, a Cloud market should be dynamic and adapt to the current needs of its participants (consumers and providers) as well as address the impact(s) these requirements have upon the market itself.

In our idea of a self-aware Cloud market, a market has the ability to change, adapt or even redesign its anatomy and/or the underpinning infrastructure during runtime in order to improve its performance. This can be done through autonomically scaling up or down the applied market mechanism or underlying computing infrastructure in response to the available resources. Similarly, it could change the market mechanisms or its components in use during the trading process. In our vision, Cloud services (e.g., software and hardware infrastructures) regardless of their provider are traded via electronic Cloud markets. In such environments, four key independent components are needed. First, users (often represented by their agents) authenticate themselves and place bids for a certain service (as consumers) or define offers (*asks*) for services (as providers). Second, an *allocation mechanism* matches placed bids and asks, while a *pricing mechanism* determines price and quantity of a product to be traded between a consumer and a provider. Third, a market front-end, which may be similar to existing multi-provider dashboards (e.g.

Rightscale¹ and IBM's Altocumulus²), or an application in a commonly used (social network) platform (SNP) such as Facebook (as in Fig. 1) for user familiarity and to ease issues like authentication and front-end management. Finally, an autonomic adaption component that enables the market platform to modify itself.

We note that all components shown in Fig. 1, except the *Market Adaption* component, have to various degrees been investigated in past and on-going projects such as SORMA,³ GridEcon,⁴ CloudBus⁵ and SocialCloud [16]. The novel aspect in our vision is autonomic market adaptation. It is realized as a MAPE-K process with five components: **M**onitoring, **A**nalysis, **P**lanning, **E**xecution, and **K**nowledge.

The *Monitoring component* observes the performance of market goals. This task is performed by monitoring sensors, which gather and store low-level monitoring data retrieved from the market middleware and the market allocation mechanism. To assess performance, low-level data (metrics) is mapped into performance scores for the market goals. We elaborate upon this component in Section IV.

The *Analysis component* analyses mapped data from the monitoring component to derive possible actions for market adaptation in order to improve market performance with respect to a set of goals. We define three actions types that can be executed (in ascending order of complexity): i) scaling in/out the market infrastructure (e.g., adding/removing new VMs in order to improve computational performance); ii) modifying market participants (e.g., attracting additional participants through advertising when the market is inefficient due to a decreasing number of active participants) and/or the goods/services offered on the market; iii) modifying market "rules" where a change of market mechanism or its properties (e.g., change to an auction-based allocation mechanism instead of the posted price due to a large number of market participants, or the algorithm to determine allocation(s)) and the market's transaction charges are just two of many possible examples.

The *Planning component* gathers a list of actions from the analysis component and plans the execution steps of the action set. It then sends these steps to the *Execution component*. During the whole adaptation process, a *Knowledge component* stores analysed monitoring data, goal/metric mappings and experiences from previous adaptations.

III. RELATED WORK

Due to the growing importance of distributed systems (Grids and Clouds) in recent years, the scientific community focused on the theoretic foundations and research on how to make such systems adaptive and sustainable, often referring to the original self-* principles of autonomic computing [20]. While research was mainly visionary in the first years after the seminal

¹<http://www.rightscale.com> – last accessed Aug. 2011

²<http://almaden.ibm.com/asr/projects/cloud/> – last accessed Aug. 2011

³<http://www.sorma-project.eu> – last accessed Aug. 2011

⁴<http://www.gridecon.eu> – last accessed Aug. 2011

⁵<http://www.buyya.com/CloudBus> – last accessed Aug. 2011

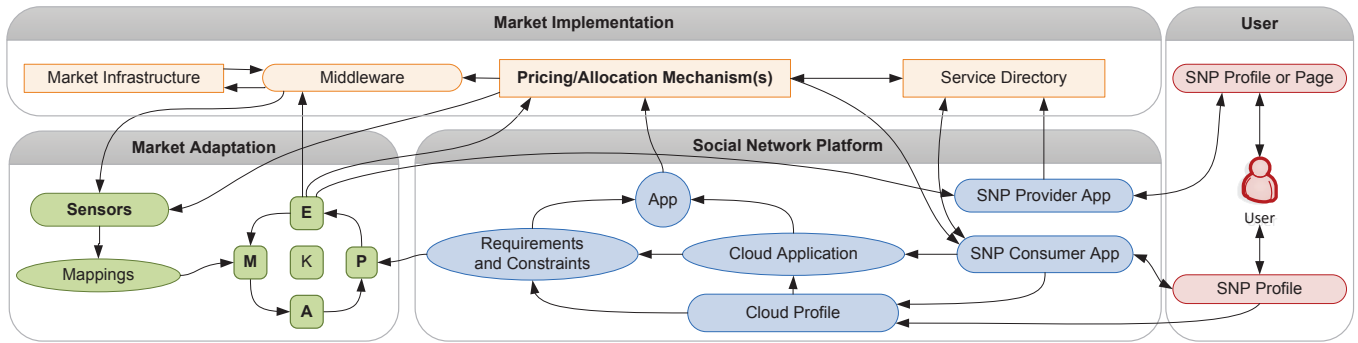


Fig. 1. Self-aware Cloud market

Autonomic Computing article, these early works served as groundwork for more and more prototypical implementation of autonomic aspects in various systems. For example, [25] discuss the need for autonomic capabilities of distributed service systems and briefly outline the application of the self-* capabilities in this context. Today, most scientific work addresses technical issues to make systems autonomic, such as the development of negotiation protocols to make Grid or Cloud services self-adaptive [8], or considers autonomic service management frameworks without explicitly taking economic methodologies into account (e.g., [18], [22], [24]).

In contrast, research on autonomic systems that apply economic methods and considerations, as first proposed by [17], is still in its infancy. In particular, current research in this area often focuses on narrow and specific issues and therefore only partially considers the aspects needed for autonomic marketplaces. For example, a self-organising resource allocation mechanism in dynamic Application Layer Networks is proposed by [28]. They do not, however, consider issues such as the adaptation of the market and the used mechanism itself depending on the available resources, which is a crucial element for potential autonomic marketplaces. [26] propose mechanisms that are able to adaptively adjust their parameters based on the past behaviour of participants. An example of economically-inspired market infrastructures is provided by [14] who present a self-optimising infrastructure platform for service delivery using economic (congestion-based) pricing, but only consider the infrastructure level, and not the market level of the platform. [6] study the mapping of high-level business objectives to lower level objectives in order to enable autonomic optimisation. However, they specifically study an autonomic DBMS and do not consider Grid or Cloud environments.

In contrast to the original self-* principles as described above, self-awareness is a more abstract concept that can be considered a building block of other principles. In order to facilitate self-configuration or self-optimisation, a platform has to be able to gather information about its current state and performance, and be able to draw the right conclusions from the available data. One key aspect in this process is the ability to monitor crucial attributes and other performance measures that provide information about the current state of the market or the platform. Performance monitoring is a widely used

process that is applied in economics in areas such as principal-agent theory (monitoring the performance of agents in order to prevent hidden action, [21] pp. 121), public (government) institutions (i.e. institutions that are not directly involved in a competitive market [30]), and market performance with respect to market power and abuse in decentralised energy markets [27].

Another area that closely resembles the concept of self-aware markets is the monitoring of performance metrics in (Web) services or business processes. Monitoring the performance of Web Services is crucial due to their inherent dynamics and the complexities and dependencies that arise with their invocation in service composition. Run-time monitoring, in this context has been addressed by several authors, e.g. [4], and tries to provide methods and frameworks that address how certain attributes and metrics of Web Services can be continuously monitored. In an enterprise context, Key-Performance-Indicators (KPIs) are defined as attributes linked to the performance of the enterprise, and for each KPI target values are set according to management goals. KPIs, however, tend to be management oriented, as they represent high-level goals, and for this reason, need to be matched to lower-level metrics of the underlying involved business processes. [29] presents a framework that is able to derive dependencies between KPIs and the underlying metrics, in order to be able to identify the causes of KPI violation.

Although these monitoring models target run-time monitoring, they consider (single) service instances or enterprises rather than taking a market perspective. From a conceptual perspective, the approach that is most similar to the intended model of Cloud **market** monitoring is the previously mentioned mechanism by [26]. However, it lacks the needed detail concerning how such a monitoring infrastructure can be implemented. In this paper, we address this point by presenting and evaluating a monitoring model as a precursor to autonomic self-aware Cloud marketplaces.

IV. MONITORING MODEL

In the previous sections, the dependence of self-aware markets on the availability of monitoring data was identified, as only through the continuous monitoring of market characteristics potential inefficiencies can be identified and acted upon. As monitoring market performance bears resemblance

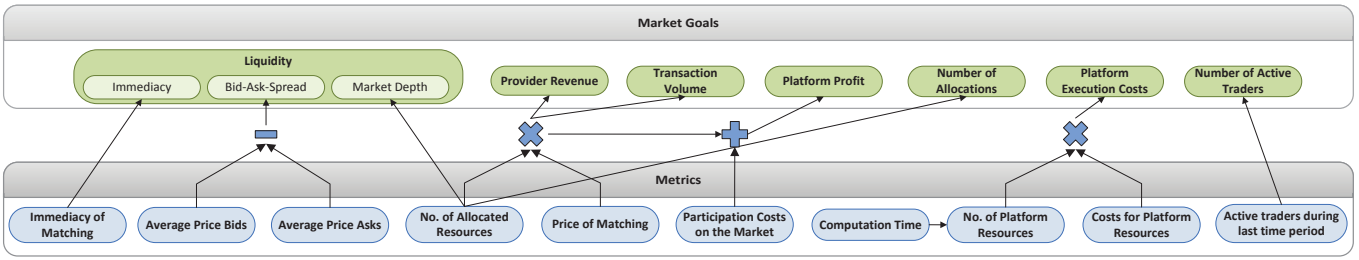


Fig. 2. Mappings between Metrics and Goals

to monitoring KPIs, in the sense that high-level (business or market) goals have to be evaluated against available low-level monitoring data, it is necessary to define goals for market performance. In this section, we therefore focus on three aspects. Firstly, we identify a set of common market goals that are relevant to our scenario. Secondly, we identify what data is available that can be monitored to determine the performance of these goals. Thirdly, as market goals are generally quite abstract, much more than the lower-level monitoring data, we present mappings (appropriate aggregation processes) from the low-level metrics to the market goals.

A. Market Goals

A market can have several goals that it aims to achieve, e.g. concerning the market environment (i.e. what type of goods are traded, who owns the market, etc.) and the target group (system, consumer or provider). From an economic point of view, market goals are often abstract, system-wide goals that depend on the market mechanism used. For example, maximising *welfare* (the sum of consumer and provider surplus) is one of the most desirable and applied goals for markets. However, the downside of this particular goal is its dependence on the concept of provider and consumer utility, which is difficult to universally capture. This means that not all possible market goals, can be monitored in a real environment. In Table I, we define an initial set of market goals that can be assessed using derivable metrics that do not rely on abstract (economic) concepts.

TABLE I
LIST OF MARKET GOALS

Goal	Description
Revenue	Maximize provider revenue
Platform profit	Maximize platform profit
No. of allocations	Maximize the number of successful allocations
Transaction volume	Maximize the volume (price*quantity) of transactions
Platform execution costs	Minimize execution costs for a given quality level
Liquidity	Increase/maximize the easiness to sell different resources on the market
No. of active traders	Maximize the number of actively participating users

Revenue and *platform profit* directly measure the attractiveness of the market to the specific group (providers, platform owners). For providers, revenue instead of profit is chosen

because internal costs are not known and cannot be monitored, hence only the revenue generated with trades on the market can be measured. The *number of allocations*, *active traders* and *transaction volume* consider the impact of the market with respect to trades, and can be important if, e.g., fees are charged per transaction (volume). *Platform execution costs* are directly relevant to the market platform owner, as they want to minimise the resources used to run the market. As some market mechanisms involve extensive computation (e.g. some auction types involve NP-hard calculations), this is an important consideration and represents a trade-off between efficiency and costs. Finally, *liquidity* is a more abstract concept that looks at how easy it is to sell different types of resources on the market.

Most of the goals are self-explanatory from their description, but the concept of liquidity requires a more concrete definition. Liquidity, an important measure of market quality and a concept which is commonly used in financial markets, describes how easy it is to trade a certain volume of the considered good. Its rather abstract definition means that there is no single aggregate value for liquidity. Instead, there exist several standard measures that serve as a proxy for the assessment of liquidity. The most common measures include bid-ask spreads [2], market depth [5] and immediacy [15]. The bid-ask spread measures the difference between bid prices and ask prices of a good; smaller differences (spreads) indicate a more liquid market. Market depth considers the volume of a good that can be traded at a certain point in time. Immediacy determines the time needed for an order to be (successfully) matched. Although these measures are common in financial literature, they cannot be applied directly in our setting, as they are used to measure liquidity for specific assets or goods. In our case, we are interested in the liquidity of the market itself, where potentially many heterogeneous goods are traded. Hence, we need to reconsider these measures for market liquidity defining them as follows:

- 1) Bid-Ask Spread:

$$BAS = \frac{\text{average price bids} - \text{average price asks}}{\text{time interval}}$$

- 2) Market Depth:

$$V = \frac{\text{amount of goods traded}}{\text{time interval}}$$

- 3) Immediacy:

$$I = \text{time}_{\text{trade executed}} - \text{time}_{\text{ask submitted}}$$

While some of the goals can potentially be achieved simultaneously, others compete and can create trade-offs between the goals (e.g. minimisation of platform costs is contradicted by the maximisation of active participants, as more participants create the need for more resources).

B. Monitoring Metrics

To assess the performance of a goal or set of goals, they need to be decomposed into measurable metrics. Table II, defines an initial set of metrics that can be observed in a marketplace, as well as for which goal(s) the metric is relevant. The table also states where in the marketplace the metric is observed: within the *mechanism* used in the market, the *market* platform itself or the *infrastructure* used to run the market.

TABLE II
LIST OF MONITORING METRICS

Metric	Level	Goal Relevance
No. of allocated resources	Market	No. of allocations, transaction volume, liquidity
Average price of matching	Mechanism	Revenue, transaction volume, platform profit
Active traders during last time period	Market	No. of active traders
Average price of bids and offers	Mechanism	Liquidity, platform profit
Time to compute the allocation and pricing	Infrastructure	Platform execution costs, liquidity
Participation costs on the market	Mechanism	Platform profit, revenue
Immediacy of order matching	Market	Liquidity
Number and costs of platform resources	Infrastructure	Platform execution costs

Naturally, other market data can be monitored. Therefore, Table II is not exhaustive, but an initial set of metrics that are the most relevant for steering self-aware autonomic markets.

C. Mapping Metrics to Market Goals

As some market goals defined at the beginning of this section are rather abstract concepts, they cannot be directly derived from monitored data (metrics). Therefore, we need to define not only metrics that can be monitored during the execution of the market, but also mappings that combine and transform metrics into indicators that determine each goal's performance. Some metrics map directly to the goals described previously (such as platform profit), but for all other goals multiple metrics have to be considered. Fig. 2 presents the mappings implemented as a part of this paper to demonstrate the practicability of our monitoring model.

V. CASE STUDY: A MONITORING MODEL AS AN EXTENSION TO GRIDSIM

To assess the strength of our monitoring model, we have implemented monitoring sensors as extensions to GridSim, an open-source toolkit for conducting simulations in Grid environments. In this case study, we monitor the performance of a Continuous Double Auction (CDA) and the underlying

infrastructure to demonstrate how the model can detect market inefficiencies and peculiarities.

GridSim is an adequate platform for implementing monitoring model for several reasons. First, it implements numerous reservation-based and auction mechanisms for resource allocations, including the double, English, Dutch, first-price sealed-bid, and continuous double auction mechanisms [3]. It also provides well-defined interfaces for implementing additional mechanisms and algorithms. Second, it is designed as an extensible multi-layer architecture which allows new components or layers to be added and integrated into GridSim easily [10]. It also provides a comprehensive facility for creating different classes of heterogeneous resources that can be aggregated using application schedulers, also called the resource brokers, for solving compute and data intensive applications. Finally, as an open-source toolkit it has already been used for several research projects [11]. However, GridSim simulates Grid resource and networks and does not consider the Cloud computing paradigm directly. Nevertheless, it is important to note that resource markets for trading Cloud and Grid services do not significantly differ, as although definitions of resources may not be the same, the techniques for matching buyers to sellers are equal. Therefore, although GridSim simulates Grid infrastructures, it is still suitable for the implementation of the simulation of a Cloud market. Moreover, due to the short history of Clouds, most of the popular Cloud simulation toolkits, such as CloudSim, do not yet capture market mechanisms [13], making GridSim a better choice for simulating Cloud markets at this point in time.

A. Implementation of Monitoring Sensors

The monitoring tool measures market and infrastructure performance of GridSim components placed in one or more architecture layers. Fig. 3 presents the layered architecture of GridSim, its components and our extensions. As depicted, GridSim is placed on top of an operating system and a distributed hardware infrastructure. Communication between the infrastructure and other GridSim components is managed by the bottom GridSim layer. This layer additionally handles the interaction or events among GridSim components themselves. It is based on SimJava [19], a general purpose discrete-event simulation package implemented in Java that defines message passing operations, for communication between GridSim components [10]. The second layer models the core elements of the distributed infrastructure, such as resources (e.g., storage) that are essential to create simulations in GridSim. The third layer is concerned with modeling and simulation of services providing various functions such as managing job submission and providing information about available resources. The fourth layer contains components that aid users in implementing schedulers and resource brokers in order to test algorithms and strategies. Finally, the top layer helps users to define scenarios and configurations for validating the algorithms.

Our monitoring tool consists of three sensors: a mechanism sensor, a market sensor, and an infrastructure sensor. Following

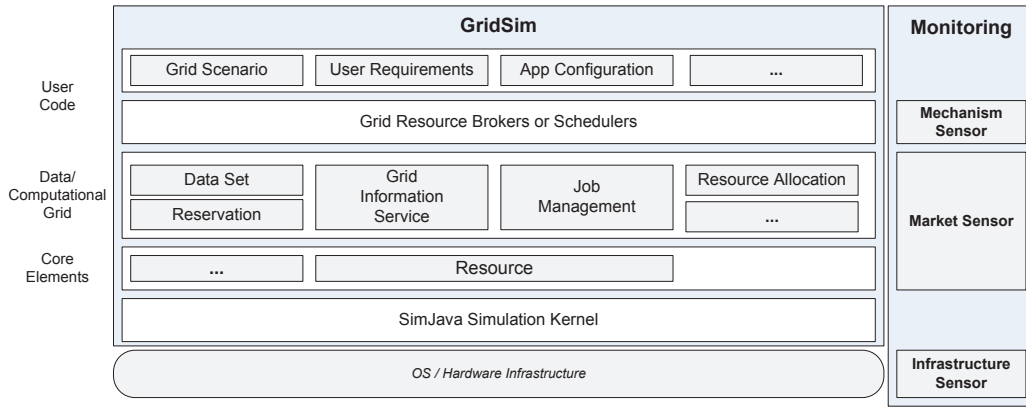


Fig. 3. The layered architecture of GridSim and its components

the layered architecture of GridSim, each sensor extends one or more GridSim layers to monitor market performance, as depicted in Fig. 3.

The **Mechanism sensor** monitors performance of a market mechanism. For example, it monitors revenue, the number of resource allocations, and average price for a single unit of resource. The actual allocation is handled by GridSim. Therefore, as depicted in Fig. 3, the mechanism sensor is built on top of the fourth GridSim layer. The mechanism sensor uses the GridSim interfaces to receive a notification of an allocation, i.e., a match of a buyer’s bid to a seller’s ask. Once a resource is allocated, the sensor receives and stores information about the allocation in the knowledge component. Additionally, using the same GridSim interfaces, the sensor gathers mechanism-specific information, such as number of bids and asks awaiting allocation.

The **Market sensor** gathers market information which is not mechanism-related, but is important for assessing market performance with respect to the market goals, as presented in Table II. For example, the market sensor stores and analyses data concerning the past and current number of sellers and buyers on the market, as well as the information about the resources traded. This is achieved by using GridSim interfaces of the architecture layers responsible for resource and job management (i.e., layers 2 and 3 in Fig. 3).

The **Infrastructure sensor** monitors the performance of a market mechanism with respect to the usage of computational resources. In particular, the infrastructure sensor monitors the utilization and performance of the underlying operating system and hardware infrastructure. For example, this sensor observes processor utilization and speed, number of threads, memory usage, hard-disk usage, etc. The infrastructure layer 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.

B. Assessing Performance of Continuous Double Auction

In our case study, we adopt the Continuous Double Auction (CDA) as the mechanism for matching sellers and buyers of a particular good and for determining the prices at which

trades are executed. In CDA, bids and asks may be placed at any point in time. As implemented in GridSim, a seller’s ask specifies the good to be sold and its ask price. Similarly, a buyer’s bid states the good to be purchased and a bidding price. The orders are maintained in an order book in bid and ask priority queues, which are ordered by price: ascending for bids, descending for asks. Bids and asks with equal prices are ordered by the time of submission.

When a new bid is received, it is compared with the first ask of the order list. Similarly, if the new-coming order is an ask, it is compared with the first bid of the bid queue. A trade is executed if the price in the ask is lower than or equal to the bid’s value. Otherwise, the order is added to the order book. Upon the execution of a trade, participants are informed of the result and the orders deleted from the order book. The trading price is calculated as $(ask + bid) / 2$. Transactions continue in this manner until no more matches can be found.

Note that although the implementation of the monitoring sensors is in this paper described only for CDA, the model can be used similarly to observe any other resource allocation algorithm. CDA has been chosen due to its continuity, which facilitates periodical monitoring and comparison of the market mechanism in time frames. CDAs are also prevalent in many existing markets, stock exchanges being the most prominent example, which illustrates their significance.

VI. EVALUATION

A. Simulation Testbed

The auction process is conducted as follows. Initially, a user submits jobs to a broker. The broker is responsible for submitting and monitoring jobs on the user’s behalf, as well as managing the CDA instance and setting additional parameters, such as job length, quantity of auction rounds, reserve price, and policy (i.e., market mechanism) to be used. Since the broker is also the auctioneer, they inform the bidders that a CDA is about to start, create a call for proposals (CFP), set its initial price, and broadcast the CFP to all bidders. Resource providers set their bids for selling a service to the user to execute a job. Once bids are placed, the auctioneer iteratively clears the auction according to the CDA definition and broadcasts matches to the user and bidders (providers).

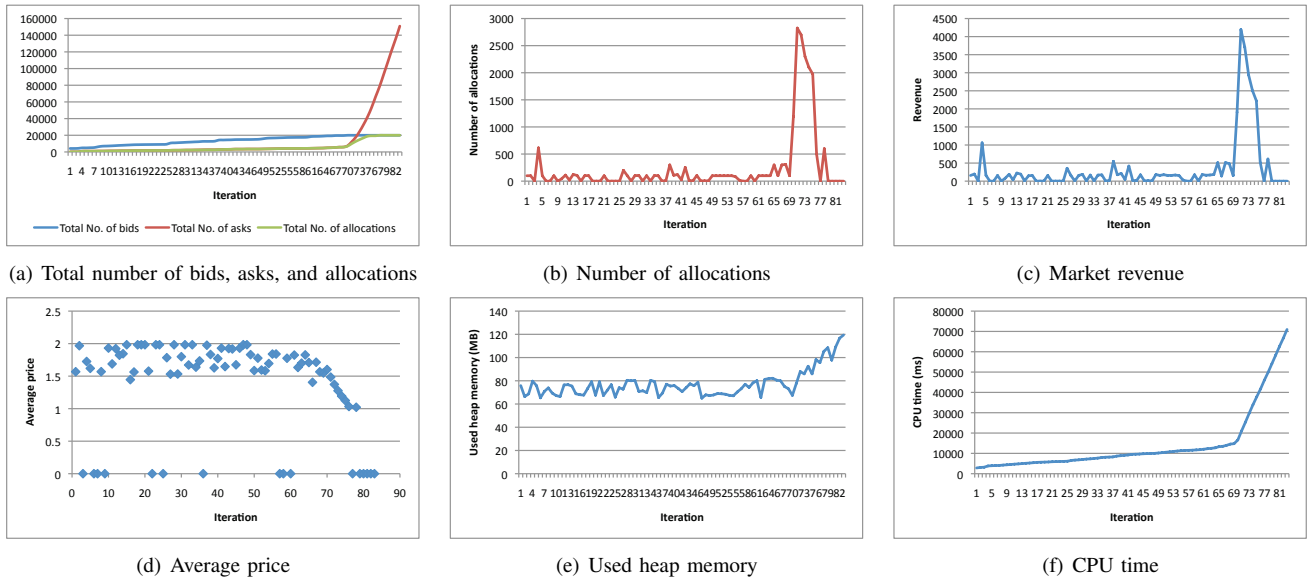


Fig. 4. Simulation results

For the evaluation purposes, we simulated one resource type (for simplicity) traded between 200 buyers and 100 sellers. Each buyer generates 100 jobs uniformly distributed over the simulation time. Our sensors monitor the market infrastructure and market mechanism performance in time frames of 5 seconds. Table III summarizes simulation settings. Our experimental scenario will create a demand higher than the supply, which somewhat contradicts the infinite elasticity assumption of the Cloud paradigm. However, such a scenario serves as a fitting context to highlight inefficiencies and peculiarities within a Cloud market.

TABLE III
SIMULATION SETTINGS

Parameter	Setting
Policy	Continuous Double Auction
Number of buyers	200
Number of sellers	100
Number of jobs per buyer	100
Number of resource types	1
Time interval for monitoring intervals	5 seconds

B. Evaluation Results

The consequence of a high demand and low supply for a single resource type can be seen in Fig. 4(a), which depicts the total (cumulative) number of buyers' bids and sellers' asks, as well as the total number of allocations between bids and asks over the whole simulation. Since supply is much lower than demand, the number of allocations is in line with the number of providers' asks. However, towards the end of the simulation process, when all users have sent all their bids (i.e., when the total number of bids does no longer change), the number of asks dramatically rises. Up until the point where bids stop arriving, all providers received an allocation. At the point when the bids stop coming in, providers start acting more

aggressively by increasing their number of asks, as because competition increases they no longer receive matches.

Several other monitoring metrics can point to the same problem from a different perspective. For example, an increase in supply causes a sudden and short-term increase in number of resource allocations and market revenue. Namely, due to the large number of providers' asks in the order book, users have a higher probability to find a resource for a lower price. Therefore, at the point when users' bids stop coming in, the number of sellers' asks suddenly increases, since all non-allocated bids get quickly allocated, which can be seen in Fig. 4(b). This, of course, affects the market revenue (Fig. 4(c)), which increases proportionally to the number of allocated resources on the market.

Fig. 4(d) depicts the average price of the resource in a given time frame. Note that the price is relatively constant during the whole simulation period, which is due to the persistent number of new bids and asks submitted to the market. However, at the point when the bids stop coming in and the number of new asks dramatically rises, the average price falls by almost 50%. This is due to the low number of bids and large number of asks in the order book. The relation between the number of asks and the average price is caused by the nature of the continuous double auction, which allocates highest bids with the lowest asks. Since high supply results in a large number of low-priced asks on the market, the allocations with low prices are more numerous and the average price sinks.

From the infrastructure perspective, the market instability causes high utilization of hardware resources. Sudden growth of supply causes an increase in CPU and heap memory usage, due to the high number of sellers' asks in the order book and their exponential growth relating to constant reordering of the priority queue (Fig. 4(e) and 4(f)).

VII. CONCLUSION AND FUTURE WORK

In this paper, we have presented our vision of an autonomic self-aware Cloud market platform, and our first steps towards achieving this vision: a market monitoring methodology. Our methodology included a series of realistic market goals, the sets of extractable metrics from a market platform and how sets of metrics are combined and transformed to access goal performance. Our simple evaluation scenario (a sudden cease in demand) illustrated that a sudden change in demand for resources can lead to market instability, and ultimately crashes, as was painfully demonstrated in the recent financial crisis. This, of course, temporarily affected the performance of market goals (both positively and negatively), and in a real deployment would have resulted in excessive and costly utilization of unneeded hardware infrastructure. We have shown that such phenomena can be detected by our monitoring model, which may in the future help to identify and react to sudden changes in the performance of Cloud markets such that we can begin to give these platforms autonomic capabilities and enable them to steer away from and avoid negative market outcomes.

As future work, we intend to investigate similar phenomena and tune our monitoring model accordingly. We also plan to include additional allocation mechanisms for future studies. Ultimately, our goal is to achieve a monitoring basis that allows us to explore and investigate how the remaining components of the Autonomic Market Adaption Cycle illustrated in Fig. 1 can be implemented.

ACKNOWLEDGMENTS

The work described in this paper is supported by the Vienna Science and Technology Fund under the grant agreement ICT08-018 Foundations of Self-Governing ICT Infrastructures (FoSII), and by Awareness, a Future and Emerging Technologies Proactive Initiative funded by the European Commission under FP7 focusing on self-awareness in autonomic systems.

REFERENCES

- [1] O. A and L. Lefevre, "When clouds become green: the green open cloud architecture," in *Parco 2009*, 2009.
- [2] Y. Amihud and H. Mendelson, "Asset pricing and the bid-ask spread," *Journal of Financial Economics*, vol. 17, no. 2, pp. 223 – 249, 1986.
- [3] M. D. D. Assunção and R. Buyya, "An evaluation of communication demand of auction protocols in grid environments," Tech. Rep., 2006.
- [4] F. Barbon, P. Traverso, M. Pistore, and M. Trainotti, "Run-time monitoring of instances and classes of web service compositions," in *IEEE International Conference on Web Services (ICWS'06)*, 2006, pp. 63 – 71.
- [5] M. J. Barclay, T. Hendershott, and D. T. McCormick, "Competition among trading venues: Information and trading on electronic communications networks," *The Journal of Finance*, vol. 58, no. 6, pp. 2637–2665, 2003.
- [6] H. Boughton, P. Martin, W. Powley, and R. Horman, "Workload class importance policy in autonomic database management systems," in *Proceedings of the 7th IEEE International Workshop on Policies for Distributed Systems and Networks*, 2006, pp. 13–22.
- [7] I. Brandic, T. Anstett, D. Schumm, F. Leymann, S. Dustdar, and R. Konrad, "Compliant cloud computing (c3): Architecture and language support for user-driven compliance management in clouds," in *IEEE Cloud*, June 2010.
- [8] I. Brandic, D. Music, and S. Dustdar, "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*. ACM, 2009, pp. 1–8.
- [9] E. Brynjolfsson, P. Hofmann, and J. Jordan, "Cloud computing and electricity: Beyond the utility model," *Communications of the ACM*, vol. 53, no. 5, pp. 32–34, 2010.
- [10] R. Buyya and A. Sulistio, "Service and utility oriented computing systems: Challenges and opportunities for modeling and simulation communities," in *Annual Simulation Symposium*, 2008, pp. 68 – 81.
- [11] R. Buyya, "Gridsim: A grid simulation toolkit for resource modelling and application scheduling for parallel and distributed computing," <http://www.buyya.com/gridsim/>, accessed Aug 4, 2011.
- [12] R. Buyya and M. Murshed, "Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," *Concurrency and Computation: Practice and Experience (CCPE)*, vol. 14, no. 13, pp. 1175–1220, 2002.
- [13] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software - Practice and Experience*, 2010.
- [14] R. Callaway, M. Devetsikiotis, Y. Viniotis, and A. Rodriguez, "An autonomic service delivery platform for service-oriented network environments," *IEEE Transactions on Services Computing*, vol. 3, pp. 327 – 331, 2010.
- [15] G. C. Chako, J. W. Jurek, and E. Stafford, "The price of immediacy," *The Journal of Finance*, vol. 63, no. 3, pp. 1253–1290, 2008.
- [16] K. Chard, K. Bubendorfer, S. Caton, and O. Rana, "Social cloud computing: A vision for socially motivated resource sharing," *IEEE Transactions on Services Computing*, vol. 99, no. PrePrints, 2011.
- [17] G. Cheliotis and C. Kenyon, "Autonomic economics," in *IEEE International Conference on Electronic Commerce*, June 2003, pp. 120 – 127.
- [18] Y. Cheng, A. Leon-Garcia, and I. Foster, "Toward an autonomic service management framework: A holistic vision of SOA, AON, and autonomic computing," *IEEE Communications*, vol. 46, no. 5, pp. 138 – 146, 2008.
- [19] F. Howell and R. McNab, "A discrete event simulation package for java with applications in computer systems modelling," in *1st International Conference on Web-based Modelling and Simulation*, January 1998.
- [20] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, pp. 41–50, January 2003.
- [21] J. Laffont and D. Martimort, *The theory of incentives: the principal-agent model*, ser. Princeton paperbacks. Princeton Univ. Press, 2002.
- [22] C. Lee and J. Suzuki, "An autonomic adaptation mechanism for decentralized grid applications," in *Consumer Communications and Networking Conference (CCNC 2006)*, vol. 1, Jan. 2006, pp. 583 – 589.
- [23] M. Maurer, V. C. Emeakaroha, M. Risch, I. Brandic, and J. Altmann, "Cost and benefit of the SLA mapping approach for defining standardized goods in cloud computing markets," in *International Conference on Utility and Cloud Computing (UCC 2010) in conjunction with the International Conference on Advanced Computing (ICoAC 2010)*, 2010.
- [24] W. H. Oyen and S. A. Deloach, "Towards a systematic approach for designing autonomic systems," *Web Intelligence and Agent Systems*, vol. 8, pp. 79–97, January 2010.
- [25] M. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: State of the art and research challenges," *Computer*, vol. 40, no. 11, pp. 38 – 45, Nov. 2007.
- [26] D. Pardoe, P. Stone, M. Saar-Tsechansky, and K. Tomak, "Adaptive mechanism design: a metalearning approach," in *8th international conference on Electronic commerce*. ACM, 2006, pp. 92–102.
- [27] A. Rahimi and A. Sheffrin, "Effective market monitoring in deregulated electricity markets," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 486 – 493, May 2003.
- [28] W. Streitberger and T. Eymann, "A simulation of an economic, self-organising resource allocation approach for application layer networks," *Computer Networks*, vol. 53, pp. 1760–1770, July 2009.
- [29] B. Wetzstein, P. Leitner, F. Rosenberg, I. Brandic, S. Dustdar, and F. Leymann, "Monitoring and analyzing influential factors of business process performance," in *Enterprise Distributed Object Computing Conference, 2009. EDOC '09. IEEE International*, Sept. 2009, pp. 141 – 150.
- [30] J. S. Wholey and H. P. Hatry, "The case for performance monitoring," *Public Administration Review*, vol. 52, no. 6, pp. 604–610, 1992.