Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 318 (2015) 179–196

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

A Case Study in Inspecting the Cost of Security in Cloud Computing

Said Naser, Said Kamil and Nigel Thomas

School of Computing Science, Newcastle University, UK

Abstract

The growing demands of business and the competition in the provision of services has led to many enterprises outsourcing IT provision using cloud computing to handle business processes and data management. Ideally, the benefits that are offered by cloud computing technology can accommodate the rapidly increased demands of the organizations and individual customers. However, the usage of cloud computing has potential security concerns. The proposed research will put forward an approach for investigating the cost of security in cloud computing. The proposed method is based on a multi-level security model, which uses the distribution of partitioned workflows upon hybrid clouds. Furthermore, the PEPA Eclipse plug-in tool will be used to create a cost model that uses the valid deployment choices that generated by the multi-level security model. The outcomes that can be obtained by means of implementing this research will used to describe the evaluation of the performance. Thus, predictions of the performance can be derived as well as the potential cost of deployment under different options and scenarios.

Keywords: Energy efficiency, discrete event simulation, performance evaluation.

1 Introduction

Over the last few years cloud computing has become a valuable option for a considerable number of organizations. The reasons behind this move are the growing capability of outsourced solutions and the high cost of buying and maintaining infrastructure. This allows organizations to exploit the advantages offered by cloud computing such as: high performance, availability, scalability and the low cost (i.e. pay on demand).

According to NIST[2] [16], the definition of cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. These features are attracting organizations

[1] Corresponding authosr: { said.kamil | nigel.thomas }@ncl.ac.uk

[2] National Institute of Standards and Technology

http://dx.doi.org/10.1016/j.entcs.2015.10.026

1571-0661/© 2015 The Authors. Published by Elsevier B.V.

to migrate toward cloud computing to cope with their rapidly increased processes and related data. Services offered by an organization are often business processes presented as workflows. The deployment of workflows on internal resources (i.e. private cloud) can affect the performance of services where the resources are limited [14]. On the other hand, cloud computing (i.e. public cloud) can overcome the limited resources problem that is facing enterprises and can offer high performance with cost-saving.

Although the challenges facing large scale computing have been understood for some years [12], cloud computing has highlighted several security concerns, for instance where the organizations data will be stored and how ensure the confidentiality and privacy of information. Correspondingly, the security aspects are one of the main concerns for many organizations [5,4]. Moreover, according to surveys that conducted by IDC [7] in 2008 and 2009, security is still the top challenge for cloud services. Thereby, some companies tend to use a combination of public and private clouds based on the sensitivity of data; private cloud as they perceive more secure and public clouds to gain the benefit of high performance, scalability and availability.

Corporations can obtain many significant improvements in security by using cloud computing [17]. Based on the multi-level security model presented by Watson [27], this research will investigate the cost of security for the deployment of the partitioned workflow in cloud competing. Moreover, the PEPA Eclipse plug-in tool will be used to create a cost model that relies on the valid selections generated by means of Watsons approach. Furthermore, a new cost model will be developed to evaluate the variety of distribution possibilities, where each option has its characteristic to deploy partitioned workflow on federated clouds.

The expected results will help to answer the following questions: how many resources are needed for specific deployment?; which better to deploy on private cloud or public clouds based on the sensitivity of data and the resources that organizations have? This work is taking into account the security requirements that should be met. This paper is structured as follows. The next section presents a motivating example of a cloud workflow drawn from the literature. In Section 3 we will discuss some other approaches to analysis. Section 4 will briefly introduce the Markovian process algebra PEPA, followed by models representing different cloud deployment options in Section 5. In Section 6 we will present some results and finally draw some conclusions and outline some further work.

2 Motivating example

A considerable amount of research has addressed cloud computing and its associated concerns of security in addition to the trade-off of the cost of deployment on private cloud and public cloud and the partitioning of workflow onto clouds in terms of security concerns. Our research has adopted the multi-level security model of Watson [27] that originally extends the Bell-LaPadula security model [2]. The example in this paper is a Healthcare Data Analysis example that introduced by

Watson. As can be seen in Figure 1. patient data that would be analyzed is shown as a workflow, which originally is a medical research application. In the first place, it reads data as input which includes the patient name and his/her measurements of heart rates. Then, the service anonymize takes off the name producing anonymized data. Consequently service named analyze starts to analyze the data. As a result, a summary of analyzed heart rates measurements will be created as output.

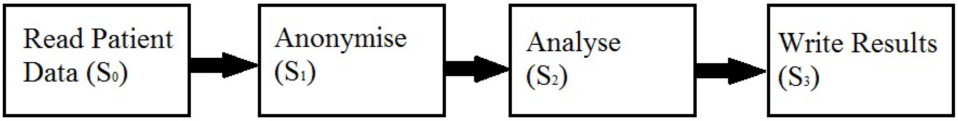

Fig. 1. Workflow of health care data analysis [27]

The primary workflow that is shown in Figure 1 and introduced by Watson [27]; has been modelled as a directed graph, where data and services are represented as nodes and the data dependencies denoted as edges. The Bell-LaPadula security conditions (no read-up and no write-down) are applied on the workflows and extended to include the cloud properties. Furthermore, a security level is assigned for each service as well as data which are consumed and produced by services. The model is extended to include clouds that will accommodate service and data of workflow, where a security level is assigned to a cloud. Additionally, the security scheme of transferring data and services between clouds has been addressed, through insuring the security level assigned to the source besides the destination. As well as to this, a tool has been developed by Watson [27] to generate valid distribution choices automatically (see Figure 2) that complies with the security constraints mentioned earlier. Finally, a very simple cost model has been created to rank the deployment selections that produced by the tool based on cost. While the calculation of the cost in [27] is depending on three factories: Data storage (size and time), CPU and Data transfer in and out.

3 Approaches to the evaluation of cloud computing

Pearson and Sander [18] introduced an approach that assisted customers to select the more appropriate cloud service provider through the decision support system that has been developed. An overview of this approach can be seen in Figure 3. Additionally, the outcomes of this system can give an assessment to threats that may possibly arise through the distribution of private information. As a consequence, the costs of the selection of service provider will be lowered. The approach of Pearson and Sander is concerned with the potential cloud security threats and to reduce the risk. However, our research will investigate the cost of the distribution of workflows over clouds by way of a different methodology with the aim of mitigate the cost.

In a similar manner to the Pearson and Sander scheme, Wenge *et al* [29] proposed a method for assessing a cloud provider, particularly with respect security aspects and the collaboration with other cloud providers. This technique can be used to

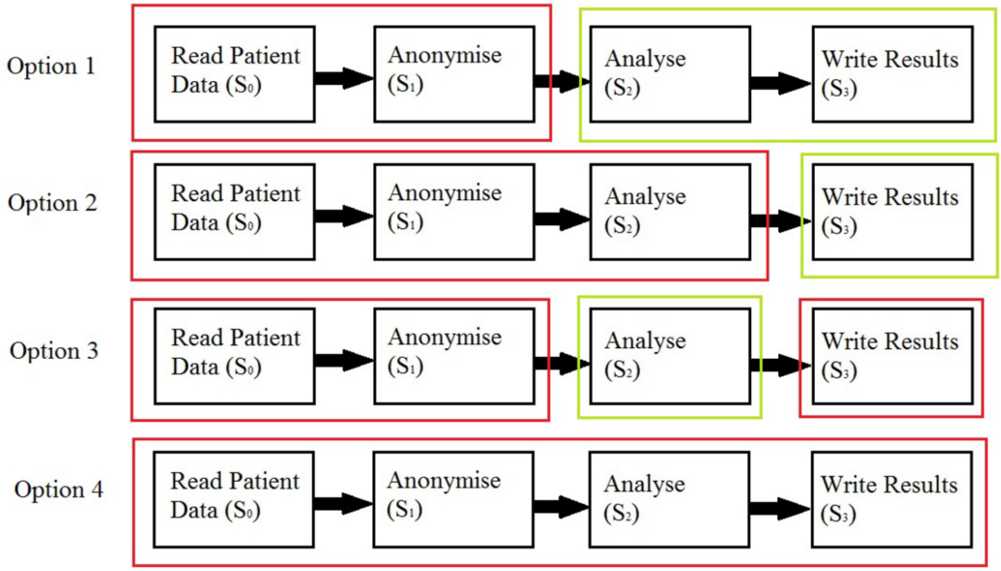

Fig. 2. Valid deployment options, where boxes red and green referring to private and public clouds respectively [27]

help customers to determine the most appropriate services providers as well as to the provision of a solution for the security risks associated.

Goettelmann *et al* [8] developed a methodology for partitioning a business process and deploying it on a set of clouds that had been taken into account security conditions. As can be seen in Figure 4, this method utilized a decentralization algorithm that proposed a mechanism to insure security constraints and quality of service. One limitation of this approach is that, the method encountered some problems in synchronization of messages.

It can be noticed that, there are some similarities between the Watson scheme and the Goettelmann *et al* method, through the manner of partitioning of workflows while meeting security requirements and distribution process over clouds. Nevertheless, the difference is that Watson considers only sequential workflows tasks. Furthermore, [29] and [8] do not investigate the cost of deployment choices over clouds and [8] claims that the calculation of the cost will be time and resource consuming.

Mach and Schikuta [15] introduced a new cost model, where they stated that their economic cost model can provide a beneficial information for cloud provider and consumer based on the calculation of the servers energy consumption. They assume that a cloud cost model can be considered as a traditional production process that has production factors and produced goods. As well, the benchmark SPECpower ssj2008 that developed by SPEC is used to test performance and power consumption. The produced information can be used to make right decisions for the business strategies that might be implemented and its impact. In spite of some similarities with Watson method as well as to the work presented by Pearson and Sander, Mach and Schikutas work lies in operational cost that are related to power

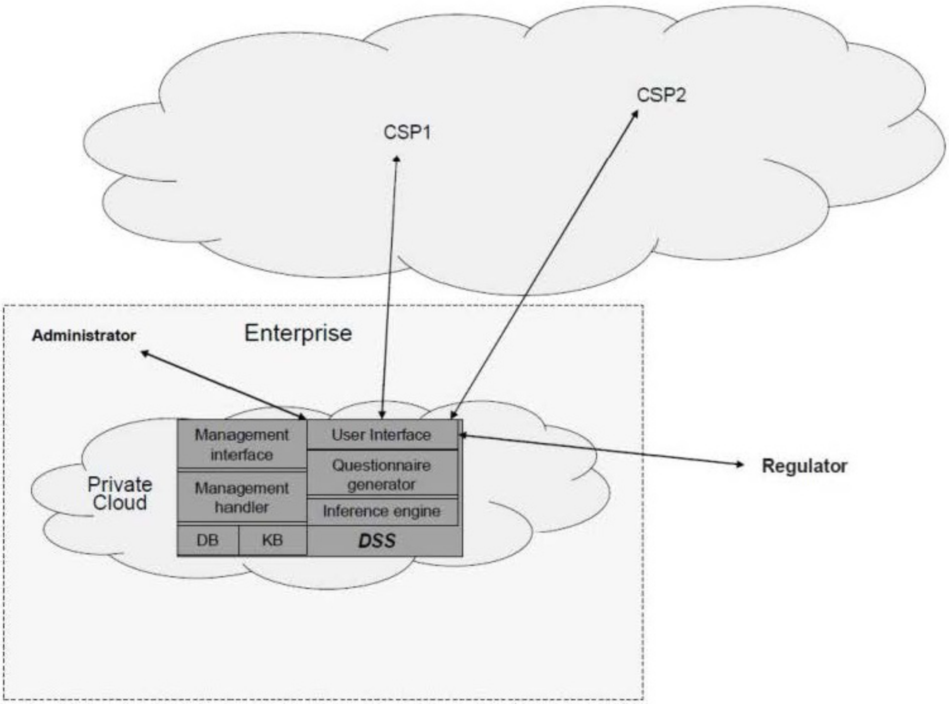

Fig. 3. Overview of the approach of Pearson and Sander [18]

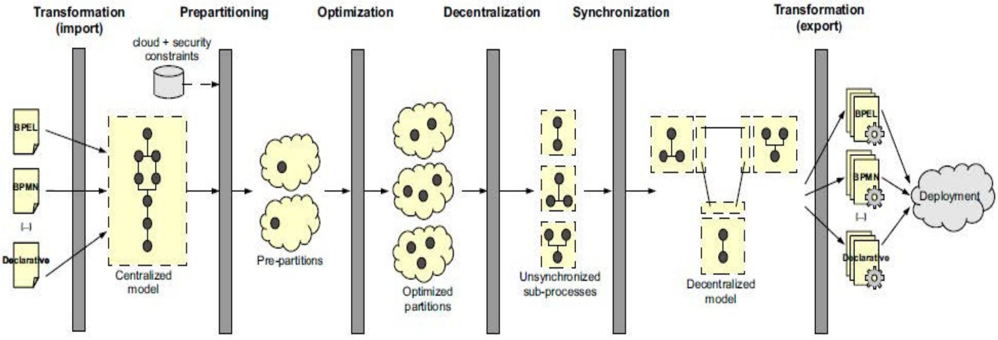

Fig. 4. Decentralization approach [8]

consumption, while our research is concerned with deployment cost in cloud environments.

A novel tool has been developed by [28] for dynamic exception handling, which extends the multi-level security model of [27]. The authors indicate that the tool can discover alternative partitions with low-cost paths to deal with exceptions that may occur during run time. The work uses dynamic exception handling for partitioned workflow on federated clouds and has adopted Watsons simple cost model of multi-level security model. This means that, they have not made changes to the method of calculation of deployment cost, but they consider exceptions are handled to achieve fault tolerance and to find another low cost paths for the successfully completion of a deployed process.

From the side of security, Capgemeni [17] described the cloud computing with associated risks, which are already associated with other types of using outsourcing. Capgemeni concentrates on multi-tenancy and required compliance, which is assumed to be more relevant to cloud computing. Capgemeni argued that, significant benefits in security can be gained on adoption of cloud computing over traditional outsourcing.

A further approach has been presented by Nada *et al* [6] for partitioning BPEL, a workflow description language. A program technique is used in order to partition automatically. This approach states that the distribution of data brings several improvements on performance for example reducing network traffic. In addition, the method of Nada *et al* has attempted to reduce the cost of communication between partitions.

From the consumers perspective Dillon *et al* [5] argue that, numerous models of cost can be raised specially with the use of hybrid cloud distribution model where, enterprise data needs to be transferred from its source (private cloud) to its destination (public cloud) and vice versa. The integration of consumer data with the cloud providers is shown to add additional cost. Likewise, Leinwand [13] also discusses the cost of using cloud computing. The charge of an application that generates a lot of bandwidth using Windows Azure Platform has been presented as an example. Additionally, it has been suggested that, if size of data in excess of 50 gigabytes the cloud consumer should buy his own bandwidth.

A new methodology is presented by [20], where it evaluates the performance of clouds in order to ensure that a specific service get its proper level. Also, their methodology is an extension to ASTAR method, which defined as a method for evaluation of configurations in the context of service-oriented architectures [21]. An approach that has been developed to consist of five steps: identify benchmark, identify configuration, run tests, analyze and recommend. As a result of the analysis, the recommendations is given for a specific configuration, where the calculation of cost for each service is used for optimization.

An experimental methodology is introduced by Cerotti *et al* [3], where the performance of multi-core systems in the cloud have been evaluated. Several types of benchmarks such as: DaCapo and IOzone have been implemented on VirtualBox and Amazon EC2 platforms. Hence, the obtained outcomes are used to acquire estimations of the response time. Although numerous of experiments are implemented on real platforms, the cost only mentioned from the provider side, whereas their findings show that the provision of many single-core machines is more cost-effective than single machines with many cores.

3.1 Workflow modelling

The importance of modelling business processes is stated by Adriansyah *et al* [1], through an approach that provides an analysis of the performance. The techniques that have been used in their work starting with create a process model via YAWL language and then replay the developed event logs framework against the model. Open source tool have been used such as: Open Source framework process mining ProM and extensible event stream XES. Some similarities to our proposed approach can be observed in modelling the business process and performance analysis. However, the aim is different, as the proposed work will concentrate on the cost in cloud taking into account the importance of evaluation of performance of the modelled systems.

Workflows modelling languages acting essential role in abstracting business processes, modelling and analyzing. Therefore, several workflow modelling languages including YAWL and BPEL will be investigated in order to create a new cost model for the deployment of partitioned workflow over hybrid clouds. In addition to this, extensive experiments will be implemented to simulate the performance behaviour of completion of workflow activities. Also, a comparison between the results those will be obtained from the afore mentioned modelling languages will take place.

YAWL is a workflow language that was developed to overcome the limitations of other workflow management systems. Petri Nets have been chosen as a starting point, due to their support to the most of the workflow patterns. The Petri Nets structure is then extended to add higher abstraction level patterns [24,25]. One advantage of YAWL is that it supports human actions. According to [24] the following are some of features offered via YAWL: it can discover the dependencies of control-flow; it can use XML, XQuery and XPath in order to handle data; it can support dynamic workflow, thus can deal with new changes; it provides an environment that can straightforwardly make changes to particular requirements. On the other hand, YAWL is not standard language and lacks industry support.

BPEL is a workflow modelling language designed to support the representation of business process specifically with web services aspects and it is based on XML language. BPEL is an orchestration language (i.e. identifies an executable process, which involves exchange message with other systems). BPEL is standardized by the Organization for the Advancement of Structured Information Standards (OASIS) and supported by numerous organizations such as: IBM, Microsoft and Oracle. According to [26] BPEL can be described in two ways: firstly, abstract business process, secondly executable business process. In addition, different versions of BPEL are available for instance BPEL4WS and WS-BPEL. However, the BPEL language does not offer enough support to patterns for example: Synchronization and Advanced Branching [26].

4 PEPA

A formal presentation of PEPA is given in [9], in this section a brief informal summary is presented. PEPA, being a Markovian Process Algebra, only supports actions that occur with rates that are negative exponentially distributed. Specifications written in PEPA represent Markov processes and can be mapped to a continuous time Markov chain (CTMC). Systems are specified in PEPA in terms of *activities* and *components*. An activity (α, r) is described by the type of the activity, α, and the rate of the associated negative exponential distribution, r. This rate may be any positive real number, or given as unspecified using the symbol $\top$.

The syntax for describing components is given as:

$$P ::= (\alpha, r).P | P + Q | P/L | P \underset{\mathcal{L}}{\bowtie} Q | A$$

The component $(\alpha, r).P$ performs the activity of type α at rate r and then behaves like P. The component $P + Q$ behaves either like P or like Q, the resultant behaviour being given by the first activity to complete.

The component P/L behaves exactly like P except that the activities in the set L are concealed, their type is not visible and instead appears as the unknown type τ.

Concurrent components can be synchronised, $P \underset{\mathcal{L}}{\bowtie} Q$, such that activities in the cooperation set L involve the participation of both components. In PEPA the shared activity occurs at the slowest of the rates of the participants and if a rate is unspecified in a component, the component is passive with respect to the activities of that type. $A \stackrel{\mathit{def}}{=} P$ gives the constant A the behaviour of the component P. The shorthand $P||Q$ is used to denote synchronisation over no actions, i.e. $P \underset{\emptyset}{\bowtie} Q$. We employ some further shorthand that has been commonly used in the study of large parallel systems. We denote $A[N]$ to mean that there are N instances of A in parallel, i.e. $A|| \ldots ||A$.

5 The model

The approach used to model the motivating scenario above is to directly associate a PEPA action with a task in the workflow and the share these action with either a public or private cloud component. The approach is similar to that previously used by the authors to model security protocols [30]. Four simple PEPA models have been derived based on the four main distribution choices that adopted from [27] and illustrated in Figure 2. In each model there is a component which represents the workflow and components that represent the public and private cloud services. The difference between each model is only in the services offered by the public and private clouds. By considering each option and investigating different capacities of public cloud servers, it is possible to explore which configuration offers the best overall performance.

In Options 1-3 the system consists of three types of component, $Service_i$, $Public$ and $Private$. Option 4 is constructed just from $Service_i$ and $Private$ components. The $Service_i$ component is a sequential component where each task of the workflow is performed in turn. Multiple instances of this component are specified and its actions are shared with multiple instances of $Public$ and $Private$ components according to the option being modelled. The number of $Service_i$ instances N_1 depicts the number of concurrent executions of the workflow, whereas the number of

Public and *Private* instances, N_2 and N_3 respectively, respresent the number of servers available in the public and private clouds. In Option 1 the activities *read* and *anonymize* will be distributed on a private cloud whereas *analyze* and *write* will be deployed on public clouds. Option 2 differs in that only the *write* activity will be deployed on the public cloud and in option 3 only the *analyse* activity will be deployed onto the public clouds. In option 4 all the actions activities are deployed only on the private cloud and so no *Public* component is defined in this case.

The workflow component, which is identical in each model, is specified as a simple sequential flow, as follows:

$$Service_0 \overset{def}{=} (readData, r).Service_1$$
$$Service_1 \overset{def}{=} (anonymize, s).Service_2$$
$$Service_2 \overset{def}{=} (analyze, t).Service_3$$
$$Service_3 \overset{def}{=} (writeResult, r).Service_0$$

The private and public cloud components are then specified for each model as follows.

- Option 1

$$Private \overset{def}{=} (readData, r).Private + (anonymize, s).Private$$
$$Public \overset{def}{=} (analyze, t).Public + (writeResult, r).Public$$

- Option 2

$$\begin{aligned}Private &\overset{def}{=} (readData, r).Private + (anonymize, s).Private\\ &\quad +(analyze, t).Private\end{aligned}$$
$$Public \overset{def}{=} (writeResult, r).Public$$

- Option 3

$$\begin{aligned}Private &\overset{def}{=} (readData, r).Private + (anonymize, s).Private\\ &\quad +(writeResult, r).Private\end{aligned}$$
$$Public \overset{def}{=} (analyze, t).Public$$

- Option 4

$$\begin{aligned}Private &\overset{def}{=} (readData, r).Private + (anonymize, s).Private\\ &\quad +(analyze, t).Private + (writeResult, r).Private\end{aligned}$$

The system equation for options 1 to 3 is given as,

$$System \overset{def}{=} Service0[N_1] \underset{L}{\bowtie} (Private[N_2]||Public[N_3])$$

Where the cooperation set $L = \{readData, anonymize, analyze, writeResult\}$. For option 4, where the entire workflow is deployed only on the private cloud, the system equations is given by,

$$System \overset{def}{=} Service0[N_1] \underset{L}{\bowtie} Private[N_2]$$

It is important to note here that the rates of each action are specified in both the workflow component ($Service_i$) and the corresponding cloud component (*Private* or

$Public$). In the terminology of PEPA, this is referred to as active-active cooperation. The reason for specifying the rates in this way is because we wish to exploit the definition of the apparent rate in such a way to investigate the scalability of the cloud provision. In this way the apparent rate of the action *anonymize*, for example, will be given by the product of s multiplied by the minimum of the number of $Service_1$ and $Private$ components currently enabled. That is, an individual instance of the anonymize action will never be be served at a rate greater than s, regardless of how many instances there are or how many servers are available. This condition might not always be preserved if we were to use passive actions in our model.

6 Experimental results

The models introduced above were analysed using the PEPA Eclipse Plug-in [23]. We assume that the *analyze* action will be the most computationally expensive (hence the rate t is relative small) and that *readData* and *writeResults* are relatively fast actions. For ease of presentation we also assume that the rates of *readData* and *writeResults* actions are the same (specified as r in the models above). Thus we have focused our experiments in a number of scenarios whose rates for *readData*, *anonymize*, *analyze* and *writeResults*, are given in Table 1. In each of the models representing options 1, 2 and 3 were have further considered different capacities within the public cloud by varying N_2, which represents the number of servers. For ease of comparison, the number of servers in the private cloud, N_3, has been fixed as 1 and the number of workflow instances, N_1, has been fixed at 20.

In the first set of experiments we compute throughput based on the direct continuous time Markov chain steady state solution in the PEPA Eclipse Plug-in. This gives rise to a set of models each with 1771 states whose results are shown in Figures 5-9.

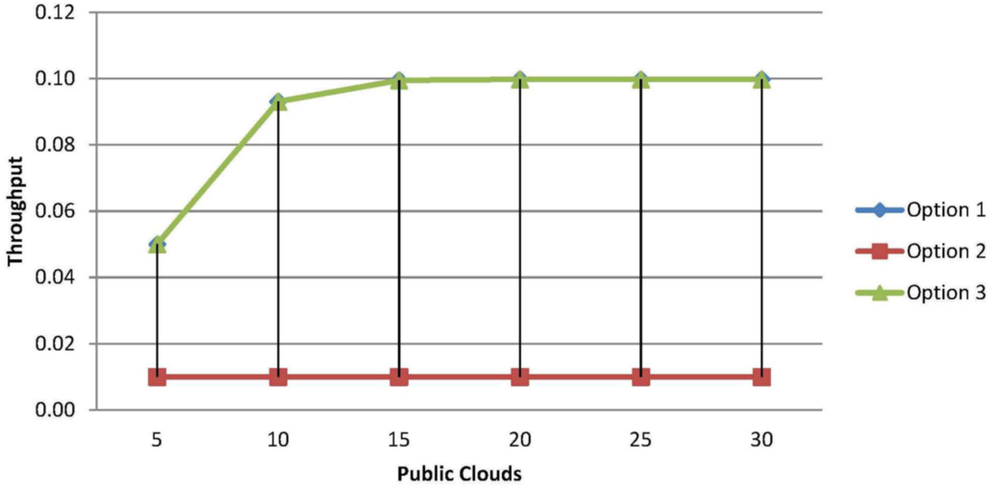

Fig. 5. Throughput of options (1, 2 and 3) by using rates of assumption 1 from Table 1.

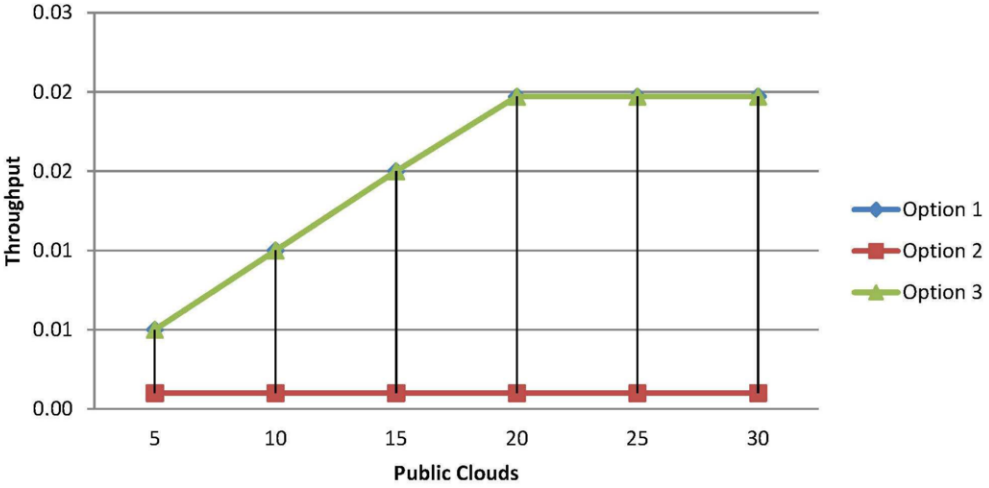

Fig. 6. Throughput of options (1, 2 and 3) by using rates of assumption 2 from Table 1.

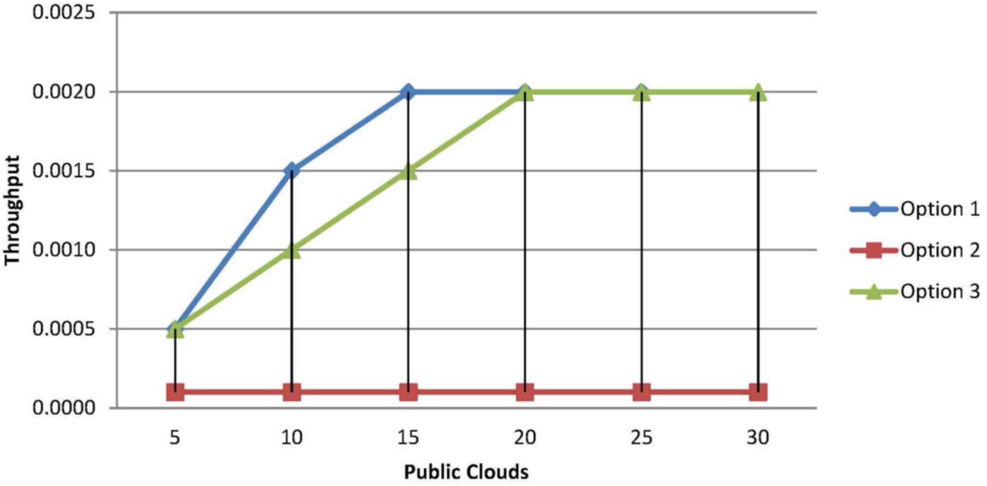

Fig. 7. Throughput of options (1, 2 and 3) by using rates of assumption 3 from Table 1.

The first observation to make from this set of experiments is that option 1 and option 3 perform almost identically in each case. This is not surprising given that these options differ only in whether the relatively fast *writeResults* action is performed in the public or private cloud. A further observation is that option 2 has a consistently lower throughput than either of the other options. This is unsurprising given that the relatively slow analyze action is being performed in the (non-scaled) private cloud in option 2, whereas it is performed in the (scalable) public cloud in options 1 and 3. In each case there is no increase in throughput beyond 20 servers, because there are only 20 workflow instances. In some cases the maximum throughput is reached with fewer servers as the low capacity of the private cloud means that actions performed there become a bottleneck in the system.

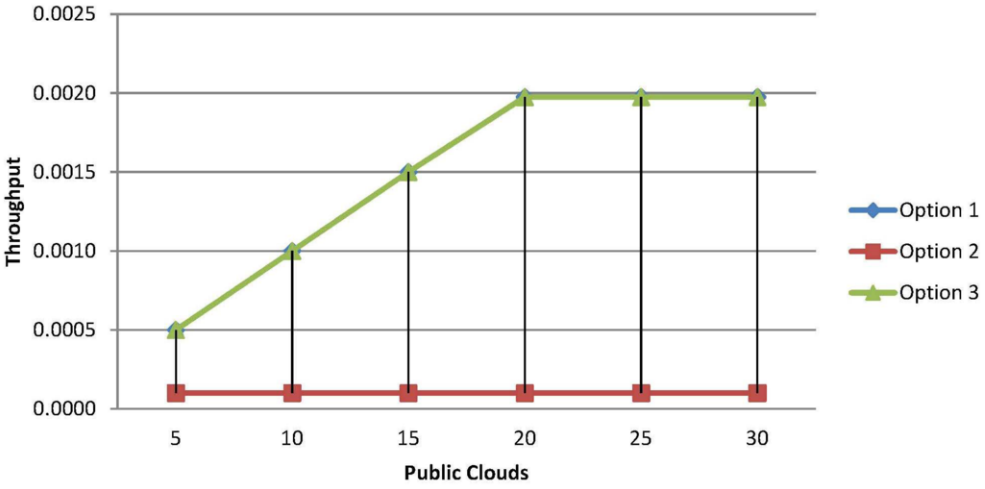

Fig. 8. Throughput of options (1, 2 and 3) by using rates of assumption 4 from Table 1.

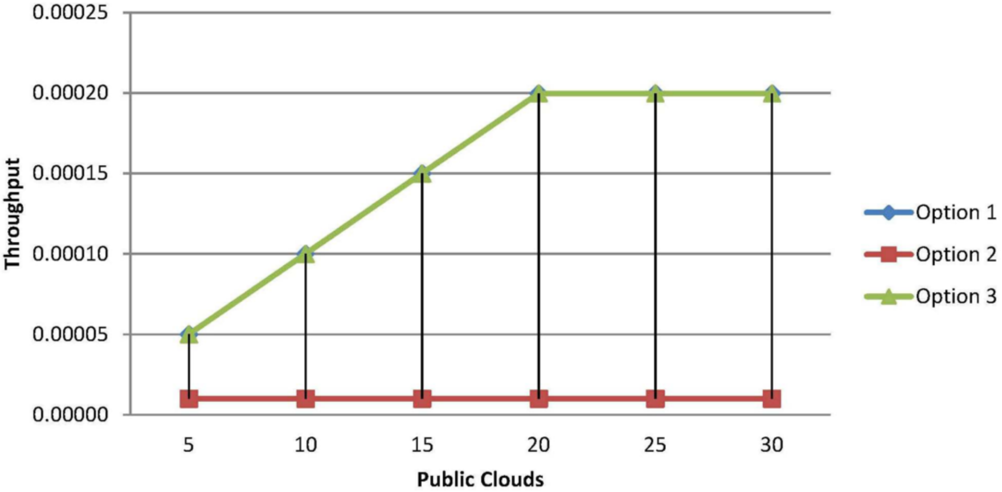

Fig. 9. Throughput of options (1, 2 and 3) by using rates of assumption 5 from Table 1.

Figure 10 shows the corresponding results for option 4, where all actions are deployed on the single instance of the private cloud. Here it is obvious that the throughput is the same as the rate of the analyze action, as this becomes the bottleneck.

In Figure 11 we further investigate the behaviour of options 1 and 3 by introducing a different rate for the*writeResults* action, which has been given the label w. We investigated a number of different combinations of rates and in many situations there was very little difference between the performance of each option. However, under certain circumstances, such as that shown in Figure 11, the overhead of *writeResults* in the private cloud is sufficient to limit the throughput of option 3.

Clearly the experiments presented here are on a limited scale and in many prac-

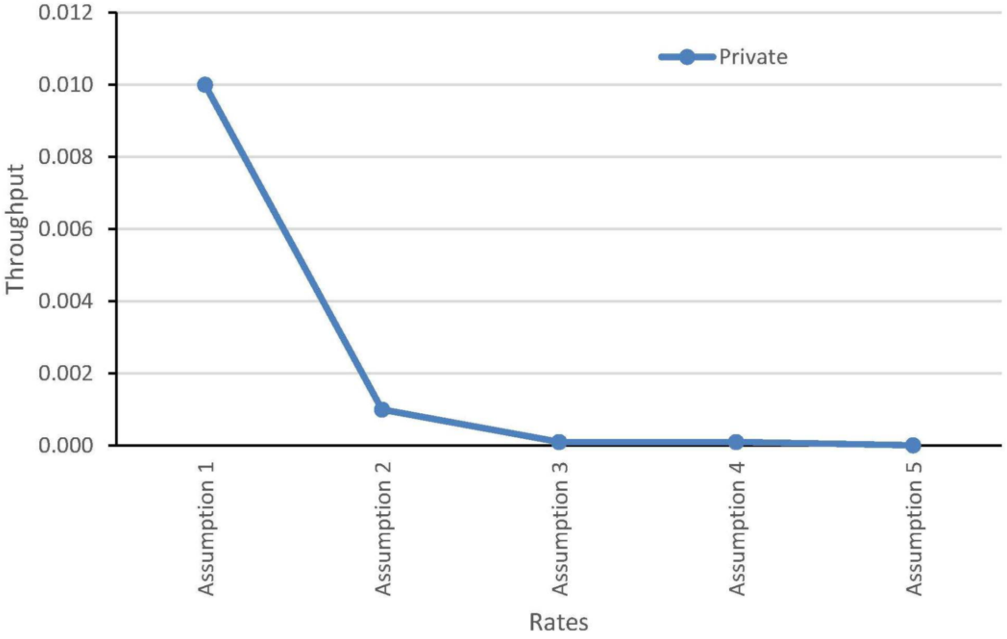

Fig. 10. Throughput of option 4 on a private cloud.

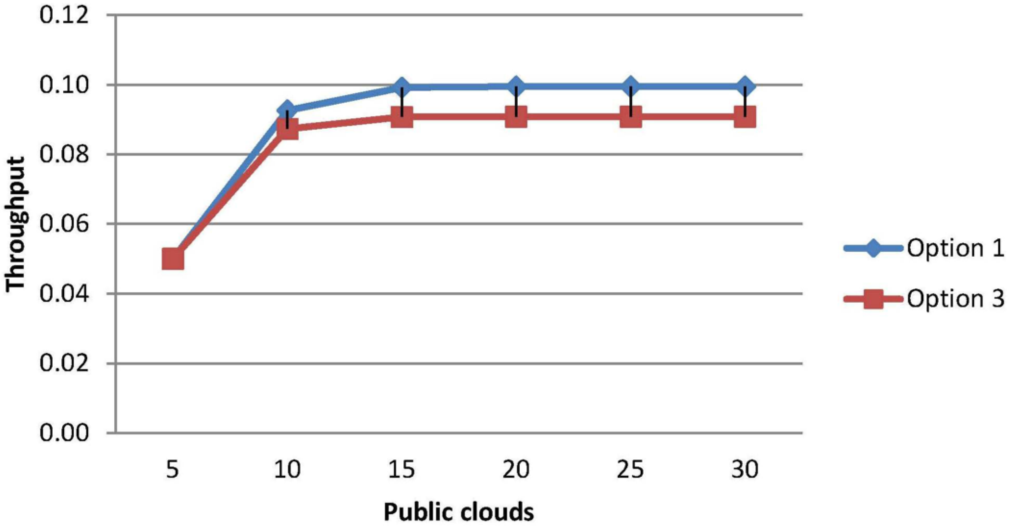

Fig. 11. Throughput of options 1 and 3, where r=1, s=0.1, t=0.01 and w=0.1.

tical circumstances we might wish to consider greater volumes of both workload and service capacity. In general this becomes problematic when using a direct solution of a continuous time Markov chain as the underlying state space rapidly becomes prohibitively large. A state space in the order of 1 million states can be solved directly with standard methods given sufficient memory to store the large sparse matrices required, although such a capacity is well beyond most users of the PEPA

Eclipse Plug-in. Hence if we wish to investigate larger systems than the one explored above, we need to utilise other methods.

The PEPA Eclipse Plug-in is equipped with two scalable analysis approaches, stochastic simulation (via Gillespies method) and fluid flow approximation by means of ordinary differential equations [10]. These techniques not only allow much larger systems to be considered, but they also facilitate transient analysis which provides further insight into the system behaviour.

Figures 12 and 13 show the transient evolution of the system under option 1 before steady state is reached. In both instances the number of workflow instances, N_1, is 20 and the rates employed are those given by assumption 1 in Table 1. In Figure 12 the number of public instances, N_3, is 5 and in Figure 13 the number of public instances is 10. The graphs show the populations of each service type, i.e. the number of workflows that are doing each action at any given time. The two graphs look fairly similar and certainly the populations of $Service_0$ and $Service_3$ appear indistinguishable. However, the populations of $Service_1$ and $Service_2$ are subtly altered by the increase in service capacity. In Figure 12 the population of $Service_1$ is decreasing steadily over time as anonymize actions are completed, with the population of $Service_2$ increasing correspondingly. Recall from Figure 5 that the throughput here is fairly low, so very few workflow instances are completing. However, when the service capacity is doubled in Figure 13 the situation changes due to the significant increase in throughput. Now the $Service_3$ population is levelling off as more workflow instances complete. Thus the system is approaching steady state much more rapidly than in Figure 12.

A more dramatic effect can be observed if we increase the number of workflow instances to 2000, as in Figure 14 and 15. Here we observe the evolution of the system under option 3 over a much longer period. Even with a long period of observation the small capacity system in Figure 14 ($N_3 = 5$) fails to reach steady state. However, when we double the service capacity in Figure 15 ($N_3 = 10$), steady state is reached relatively quickly. The importance of these observations in a practical setting is to illustrate that increasing the service capacity not only has a significant effect on throughput, but also means that the deployment is more stable and the predicted (steady-state) performance is more likely to be observed at any given instant of time.

7 Conclusions and further work

In this paper we have presented some initial work in modelling workflow deployment using PEPA. The aim of this work is to explore the costs associated with different security decisions. This paper has been motivated through an example of a healthcare application. We have shown that a simple form of model with standard analysis tools can provide insight into the behaviour of the system in question.

The system we have explored in this paper is clearly very simple. The application we have studied has a linear flow, whereby there is a fixed sequence of actions with no choice or deviation. It will therefore be interesting to explore more com-

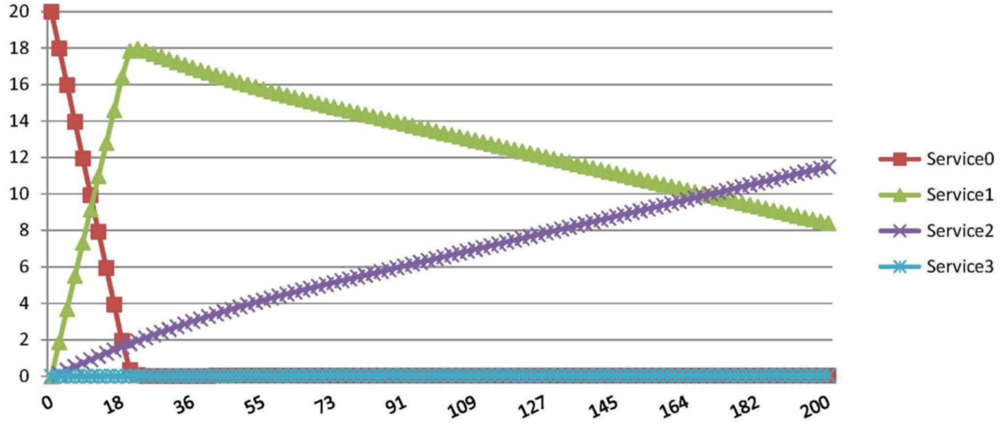

Fig. 12. ODE transient analysis for 20 workflows instances in option 1 model using assumption 1 from Table 1 with 5 public instances.

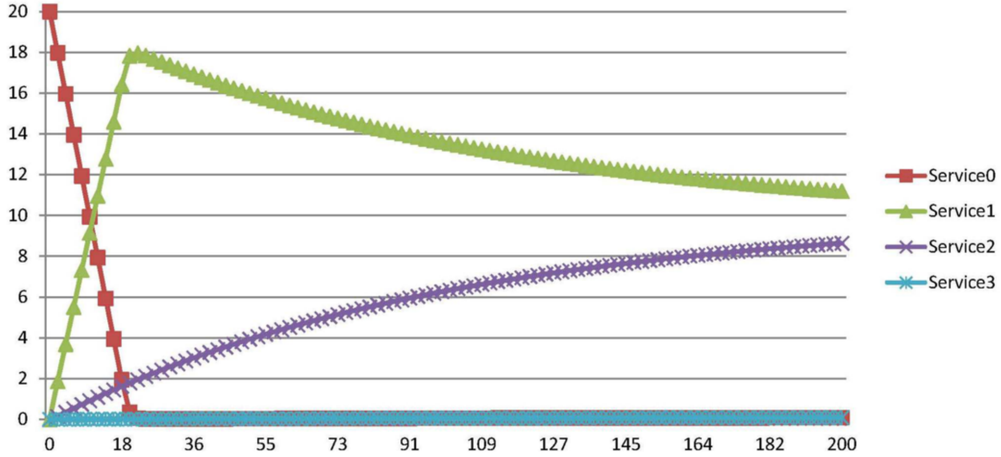

Fig. 13. ODE transient analysis for 20 workflows instances in option 1 model using assumption 1 from Table 1 with 10 public instances.

plex workflows, for example those involving choice and loops, to investigate model behaviours that may arise in such scenarios. In addition it would be interesting to consider different sets of public cloud servers deployed for different actions.

The model we have developed to study this application also has some limitations. The most significant of these is that we have not modelled any data transfer costs. Clearly it would be a simple matter to add some network delays between actions being undertaken in different locations. This would enable a clearer comparison between the performance of different deployment options in public and private clouds where different transfer costs will be evident.

We have considered servers to belong to a homogeneous set. A consequence of this assumption is that an action will occur at the same rate wherever it is executed (in the private cloud or on any of the public cloud servers). In practice there are a wide range of service levels which can be purchased within clouds and within those

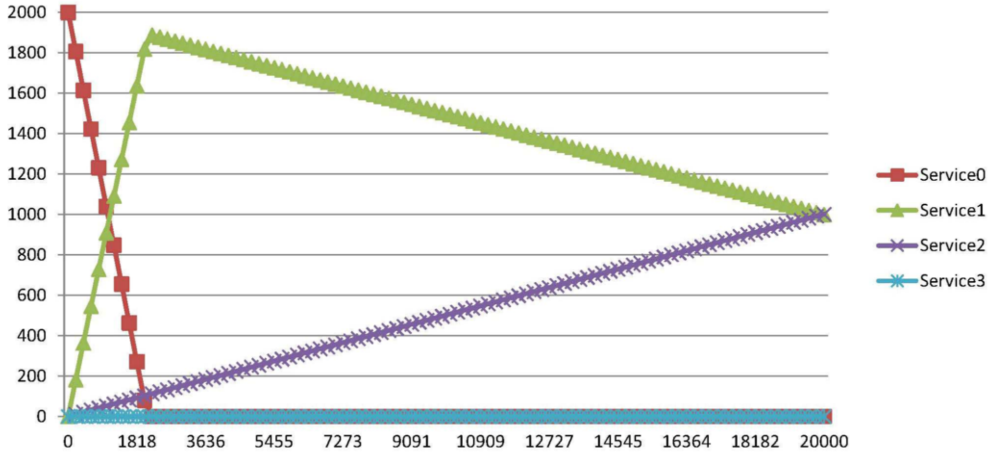

Fig. 14. ODE analysis for 2000 workflows instances in option 3 model using assumption 1 from table 1 with 5 public instances

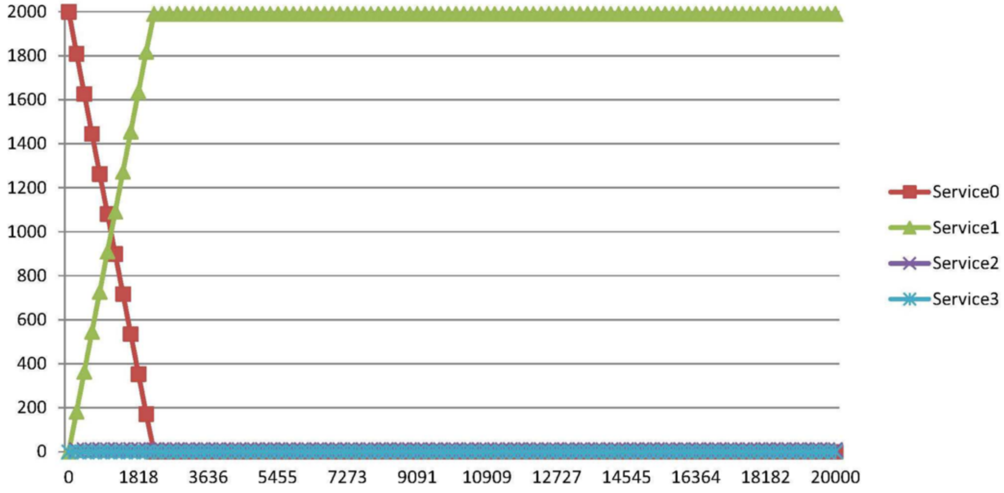

Fig. 15. ODE analysis for 2000 workflows instances in option 3 model using assumption 1 from table 1 with 10 public instances

options there is wide range of performance that might be experienced. Predicting exactly what a provider will offer on any specification is not generally possible in the current evolution of cloud systems. However it would be relatively straight forward to give different performance characteristics for different systems (at least public and private) and compute the average performance at different service levels.

Clearly to have practical value approaches such as the one described here need to be validated against real implementations. Performing such experiments in a rigorous way is a difficult and time-consuming business. Our aim therefore is to further develop the modelling approach to show the potential of this line of investigation before undertaking work to validate the results.

Acknowledgements

The authors would like to take this opportunity to thank Prof Paul Watson of Newcastle University for taking time to explain his approach and the example used in this paper, as well as providing the authors with a copy of his analysis tool.

References

[1] Adriansyah, A., B. van Dongen, D. Piessens, M. Wynn and M. Adams, *Robust Performance Analysis on YAWL Process Models with Advanced Constructs*, Journal of Information Technology Theory and Application, **12**, 2012.

[2] Bell, D.E. and L.J. La Padula, *Secure computer systems: mathematical foundations*, No. MTR-2547-VOL-1, Mitre Corp Bedford, MA, 1973.

[3] Cerotti, D., M. Gribaudo, P. Piazzolla, G. Serazzi, End-to-End Performance of Multi-core Systems in Cloud Environments, in: 'Computer Performance Engineering', LNCS 8168, Springer, pp. 221-235, 2013.

[4] Chen, Y., V. Paxson and R. Katz, *Whats New About Cloud Computing Security?*, University of California, Berkeley, Rep. UCB/EECS-2010-5, 2010.

[5] Dillon, T., C. Wu and E. Chang, *Cloud Computing: Issues and Challenges*, in: 'Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications', pp. 27-33, 2010.

[6] Fox, A., R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin and I. Stoica, *Above the Clouds: A Berkeley View of Cloud Computing*, University of California, Berkeley, Rep. UCB/EECS 28, 2009.

[7] Gens, F., *New IDC IT Cloud Services Survey: Top Benefits and Challenges*, 2009. `http://blogs.idc.com/ie/?p=730`

[8] Goettelmann, E., W. Fdhila and C. Godart, *Partitioning and Cloud Deployment of Composite Web Services under Security Constraints*, in: 'Proceedings of the IEEE International Conference on Cloud Engineering', pp. 193-200, 2013.

[9] Hillston, J., 'A compositional approach to performance modelling', Cambridge University Press, 1996.

[10] Hillston, J., *Fluid flow approximation of PEPA models*, in: 'Proceedings of the Second International Conference on the Quantitative Evaluation of Systems', pp. 33-42, 2005.

[11] Hillston J. and L. Kloul, *A Function-Equivalent Components Based Simplification Technique for PEPA Models*, in: 'Formal Methods and Stochastic Models for Performance Evaluation', LNCS 4054, pp. 16-30, Springer, 2006.

[12] Jarvis, A., N. Thomas and A. van Moorsel, *Open issues in grid performability*, International Journal of Simulation, **5**(5), pp. 312, 2004.

[13] Leinwand, A., *The Hidden Cost of the Cloud: Bandwidth Charges*, 2009. `http://gigaom.com/2009/07/17/the-hidden-cost-of-the-cloud-bandwidth-charges`

[14] Mace, J., A. van Moorsel and P. Watson, *The case for dynamic security solutions in public cloud workflow deployments*, in: 'Proceedings of the IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops', pp. 111-116, 2011.

[15] Mach, W. and E. Schikuta, *A Consumer-Provider Cloud Cost Model Considering Variable Cost*, in: 'Proceedings of the 9th IEEE International Conference on Dependable, Autonomic and Secure Computing', pp. 628-635, 2011.

[16] Mell P. and T. Grance, *The NIST Definition of Cloud Computing, National Institute of Standards and Technology*, U.S. Department of Commerce, Special Publication 800-145, 2011.

[17] Newcombe, L., *Putting Cloud security in perspective*, Capgemini, 2011. `http://www.capgemini.com/resources/putting-cloud-security-in-perspective`

[18] Pearson, S. and T. Sander, *A mechanism for policy-driven selection of service providers in SOA and cloud environments*, in: 'Proceedings of the 10th Annual International Conference on New Technologies of Distributed Systems', pp. 333-338, 2010.

[19] Radev, D., V. Denchev and E. Rashkova, *Steady-state solutions of Markov chains*, in: 'Proceedings of the 7th Balkan Conference on Operational Research', 2005.

[20] Stantchev, V., *Performance Evaluation of Cloud Computing Offerings*, in: 'Proceedings of the 3rd International Conference on Advanced Engineering Computing and Applications in Sciences', pp. 187-192, 2009.

[21] Stantchev, V., 'Architectural Translucency', vol. 8, GITO mbH Verlag, 2008.

[22] Ter Hofstede, A.H.M., W.P. van der Aalst, M. Adams and N. Russell, 'Modern Business Process Automation: YAWL and its Support Environment', Springer, pp. 241-242, 2009.

[23] Tribastone, M., A. Duguid and S. Gilmore, The PEPA eclipse plugin, ACM SIGMETRICS Performance Evaluation Review, **36**(4), pp. 28-33, 2009.

[24] Van der Aalst, W.P. and A.H.M. ter Hofstede, *YAWL: yet another workflow language*, Information Systems, **30**(4), pp. 245-275, 2005.

[25] Van der Aalst, W.P., L. Aldred, M. Dumas and A.H.M. ter Hofstede, *Design and Implementation of the YAWL System*, in: 'Advanced Information Systems Engineering', LNCS 3084, pp. 142-159, Springer, 2004.

[26] Vasko M. and S. Dustdar, *A view based analysis of workflow modeling languages*, in: 'Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing', pp. 293-300, 2006.

[27] Watson, P., *A multi-level security model for partitioning workflows over federated clouds*, Journal of Cloud Computing, **1**(1), pp. 1-15, 2012.

[28] Wen, Z. and P. Watson, *Dynamic Exception Handling for Partitioned Workflow on Federated Clouds*, in: 'Proceedings of the 5th IEEE International Conference on Cloud Computing Technology and Science', pp. 198-205, 2013.

[29] Wenge, O., M. Siebenhaar, U. Lampe, D. Schuller and R. Steinmetz, *Much Ado about Security Appeal: Cloud Provider Collaborations and Their Risks*, in: 'Proceedings of the 1st Service-Oriented and Cloud Computing', LNCS 7592, pp. 80-90, Springer, 2012.

[30] Zhao, Y. and N. Thomas, *Efficient solutions of a PEPA model of a key distribution centre*, Performance Evaluation, **67**(8), pp. 740-756, 2010.