Journal of Network and Computer Applications 82 (2017) 93–113

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Survey on prediction models of applications for resources provisioning in cloud

Maryam Amiri*, Leyli Mohammad-Khanli

Faculty of Electrical and Computer Engineering, University of Tabriz, 29 Bahman Blvd, Tabriz, East Azerbaijan, Iran

ARTICLE INFO

Keywords:
Cloud Computing
Prediction
Application
Workload
Resources Provisioning

ABSTRACT

According to the dynamic nature of cloud and the rapid growth of the resources demand in it, the resource provisioning is one of the challenging problems in the cloud environment. The resources should be allocated dynamically according to the demand changes of the application. Over-provisioning increases energy wasting and costs. On the other hand, under-provisioning causes Service Level Agreements (SLA) violation and Quality of Service (QoS) dropping. Therefore the allocated resources should be close to the current demand of applications as much as possible. Furthermore, the speed of response to the workload changes to achieve the desired performance level is a critical issue for cloud elasticity. For this purpose, the future demand of applications should be determined. Thus, the prediction of the application in different aspects (workload, performance) is an essential step before the resource provisioning. According to the prediction results, the sufficient resources are allocated to the applications in the appropriate time in a way that QoS is ensured and SLA violation is avoided. This paper reviews the state of the art application prediction methods in different aspects. Through a meticulous literature review of the state of the art application prediction schemes, a taxonomy for the application prediction models is presented that investigates main characteristics and challenges of the different models. Finally, open research issues and future trends of the application prediction are discussed.

1. Introduction

Cloud computing is a computing paradigm that provides services such as infrastructure, platform and software based on a pay-as-you-go model (Coutinho et al., 2015; Kulkarni and Agrawal, 2014). Elasticity is one of the prominent features of cloud computing (Petcu and Vzquez-Poletti, 2012). It is the degree of the system adaptability to the workload changes by provisioning and deprovisioning the resources automatically in a way that the allocated resources match the current demand (Herbst et al., 2013). So the elastic application allocates or releases the resources according to its requirements. To comply with the obligations, cloud needs to allocate a suitable amount of resources according to the current demand of applications. Under-provisioning causes Service Level Agreements (SLA) violation, Quality of Service (QoS) dropping and the customer dissatisfaction. This may lead to the loss of customers and a decrease in revenue. On the other hand, Over-provisioning wastes energy and resources and it even increases costs like network, cooling and maintenance. So the resources management is a complicated process in cloud and an efficient resource management technique is required (Singh and Chana, 2016c). As Fig. 1 shows, the efficient resources management plan impacts on three different aspects of cloud. It fulfils SLA and satisfies cloud customers. It guarantees the cloud obligations to its users. So customers will adhere to cloud in the future. It also prevents the resources wasting. So the energy consumption and the operational cost decrease. The reduction of energy consumption leads to decrease carbon emissions, which could facilitate green cloud computing. Both of the cost reduction and the revenue increase improve the profit of cloud providers (Kumar and Buyya, 2012). Therefore, the efficient resources management allocates the minimum amount of required resources for SLA fulfillment (Manvi and Krishna Shyam, 2014) and leaves the surplus resources free to deploy more Virtual Machines (VMs) (Garg et al., 2014). For this purpose, the resources allocated to each application should be close to the application demand in a way that SLA is satisfied and resources wasting is minimized.

Furthermore, the speed of response to the workload changes to achieve the desired performance level is a critical issue for elasticity (Coutinho et al., 2015). Although the important advantage of elasticity is to match the amount of resources allocated to the application with the amount of resources it requires, the time that resources take to be

* Corresponding author.
E-mail addresses: maryam.amiri@tabrizu.ac.ir (M. Amiri), l-khanli@tabrizu.ac.ir (L. Mohammad-Khanli).

http://dx.doi.org/10.1016/j.jnca.2017.01.016
Received 9 October 2016; Received in revised form 16 December 2016; Accepted 15 January 2017
Available online 18 January 2017
1084-8045/

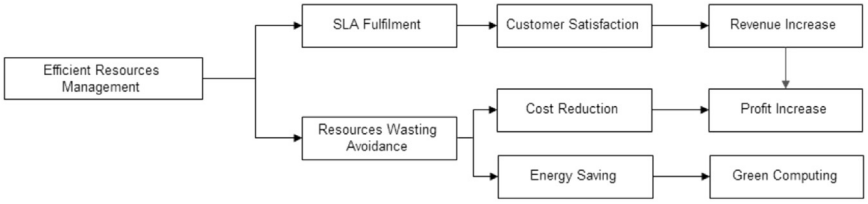

Fig. 1. The Influence of Efficient Resources Management on Different Aspects of Cloud: Revenue, Green Computing and Profit.

ready to use is a potential problem (Galante and Bona, 2012). Cloud elasticity and dynamic resources allocation are based on the virtualization techniques (Hwang et al., 2016). The VM provisioning technologies take several minutes (Jiang et al., 2013). This delay is intolerable for the tasks that need the resources scaling during the computation. It might lead to SLA violation, QoS dropping and finally a reputation loss of cloud. To reduce the delay, there are three approaches. The first approach, VM provisioning technologies, assists to ready new VMs in seconds for the requests (Jiang et al., 2013). The state of the art VM provisioning technologies, such as streaming VM technology (Labonte et al., 2004) and VM cloning (Lagar-Cavilla et al., 2009) cannot decrease time wasting of VM creation (Jiang et al., 2013). The second approach is about to ask all customers to provide a plan of the future resources demand. It is not possible according to the cloud obligations and the lack of customers' knowledge (Jiang et al., 2013). Due to VM technologies and the limitations of the customers' knowledge, the future demand prediction is the only practical and effective solution for the fast resources provisioning. A proactive prediction method predicts the future demand fluctuations in a way that the resource manager has enough time to provide the appropriate resources before occurring the workload burstiness.

If the sudden increase of the future demand is predicted, the resource manager scales up the infrastructure and prepares VMs according to the predicted future demand before the surge of demand occurs. In the same way, according to the demand reduction, the allocated resources are released. The released resources can be used to create new VMs or to allocate them to VMs that need more resources. Indeed, allocated resources are quickly matched with the demand and the rapid elasticity (Mell and Grance, 2011) is accomplished. Thus, SLA is satisfied, energy wasting is avoided and on demand provisioning is fulfilled for systems implemented by using cloud services.

However, providing cloud services that guarantee dynamic QoS requirements of users and avoid SLA violation is a big challenge. Currently, the services are provisioned and scheduled according to the resources availability, without the guarantee of the expected performance. Singh and Chana (2015a). Therefore, the future demand prediction is an indispensable step for the rapid elasticity implementation and the effective resource provisioning in the dynamic cloud environment.

Although many literatures such as Galante and Bona (2012), Huang et al. (2014), Manvi and Krishna Shyam (2014), Weingartner et al. (2015), Aceto et al. (2013), Singh and Chana (2016a), Singh and Chana (2015a), Singh and Chana (2016b), Huebscher and McCann (2008), Coutinho et al. (2015) survey cloud computing in different aspects, there is a lack of a detailed investigation of the application prediction in cloud. This paper presents a survey on the prediction of the application in different aspects such as the performance and the workload. The contributions of this paper are as follows:

- To the best of our knowledge, this paper is the first survey on the prediction of cloud applications. It presents a comprehensive review of the newest and the most prominent prediction models.
- A general taxonomy for proposed models, techniques and frameworks of the application prediction is presented. Literatures are grouped based on their proposed methods and explained briefly.
- Open research issues, challenges and the future trends of the application prediction in cloud are presented.

This paper is structured as follows: Section 2 describes different aspects of the application prediction such as main challenges, characteristics, needs and evaluation metrics for the prediction in cloud. Section 3 presents the modeling approaches for the application prediction. Section 4 investigates the proposed prediction methods and describes their advantages and disadvantages. In Section 5, different techniques are compared and challenges and directions are explained. Finally, the paper is concluded in Section 6.

2. Application prediction

The application prediction is to forecast the future behaviour of the application in different dimensions such as the workload and the performance. So the workload and performance prediction are branches of the application prediction. The application prediction is an essential step for the efficient resources management in cloud. According to the future demand of the application, the efficient resources provisioning should detect the minimum amount of resources to fulfill QoS parameters such as CPU utilization, response time, availability, reliability and security (Singh and Chana, 2016a). Table 1 shows the QoS requirements of different applications (Singh and Chana, 2016c, 2015b). We recommend that readers interested to the resources management refer to Singh and Chana (2016a), Singh and Chana (2016b). In this section, the application prediction is considered in different aspects. At first, the different dimensions of

Table 1
Cloud Applications and their QoS requirements (Singh and Chana, 2016c).

Applications	QoS requirements
Web sites	Reliable storage, high network bandwidth, high availability
Technological computing	Computing capacity, reliable storage
Endeavour software	Security, high availability, customer confidence level, correctness
Performance testing	Execution time, energy consumption and execution cost
Online transaction processing	Security, high availability, internet accessibility, usability
Central financial services	Security, high availability, changeability, integrity
Storage and backup services	Reliability, persistence
Productivity applications	Network bandwidth, latency, data backup, security
Software/project development and testing	User self-service rate, flexibility, creative group of infrastructure services, testing time
Graphics oriented	Network bandwidth, latency, data backup, visibility
Critical internet applications	High availability, serviceability, usability
Mobile computing services	High availability, reliability, portability

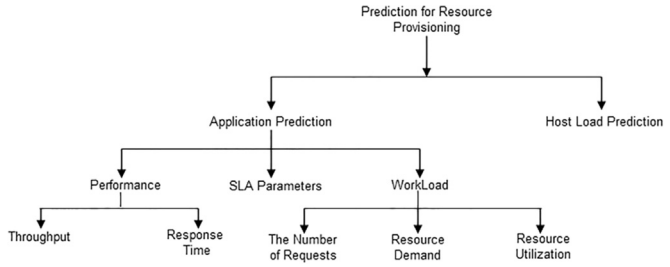

Fig. 2. Different Levels of Prediction for Resource Provisioning.

the application that are predicted in different literatures are studied. Then main characteristics, challenges, needs and metrics for the application prediction are investigated in more detail.

2.1. Different dimensions of prediction

In general, the goal of the application prediction is to describe the future behaviour of the application on a specific aspect based on the collected information. As it is shown in Fig. 2, the application prediction can be performed in different dimensions of the application. In the highest level, the prediction can be employed on VMs or Physical Machines (PMs). In the VM level, most methods and techniques investigated in this paper focus on the prediction of workload, performance and SLA parameters of cloud applications/services. According to Fig. 2, the concept "workload" is interpreted in different ways in various literatures:

- In some papers such as Liang et al. (2014); Yang et al. (2014a); Kupferman et al. (2009) the application workload is equivalent to the number of requests of the application. In this case, the future number of requests of the application is the output of prediction methods.
- The workload is interpreted as the future demand of VMs in some methods such as Jiang et al. (2013), Weijia et al. (2013). In this case, the future demand of resources of VMs are predicted.
- In Garg et al. (2014); Jheng et al. (2014); Yin et al. (2014), the resources utilization of VMs are considered as a workload. In two last cases, it is assumed that each application is capsulated inside one VM.

In the performance prediction, the application performance (throughput and response time) is predicted based on resources allocated to it (Manvi and Krishna Shyam, 2014). Indeed, the main goal is to determine the best resource allocation to achieve the desirable performance level.

Some methods such as Akindele and Samuel (2013), Leitner et al. (2009) predict SLA parameters of applications. They usually predict SLA violation of the application according to its workload or the resources allocated to it. Other methods such as Yang et al. (2014b); Di et al. (2014) predict the resources utilization in the host level. In this case, the goal is to predict the appropriate number of PMs. The main focus of this paper is on the workload and the performance prediction of applications.

In this paper, we survey more than 100 papers in the field of the application prediction. Table 2 shows different dimensions of applications that are predicted in different literatures. As it is shown in Table 2, the focus of many methods proposed in literatures is on the resources utilization. These methods predict the utilization, the load or the usage of different resources. The resources considered in most literatures are CPU and memory. The performance of applications is predicted in several literatures. The main goal of suggested methods is to determine the best resource allocation according to the application performance. The resources demand is another field of the prediction. Note that the resources demand usually determines the size of VMs. Therefore, the resources usage is much lower than the resources demand. Thus, most of the proposed methods are based on the resources utilization (Zhang et al., 2012). The number of requests of data objects such as videos are predicted in some papers. A few numbers of methods predict the execution time of jobs, the number of PMs, VMs or users, the number of requests and the power consumption of applications. Each method is described in Section 4 briefly.

2.2. Characteristics, challenges, needs and evaluation metrics

Fig. 3 shows the main characteristics, challenges, needs and evaluation metrics for the application prediction. The characteristics determine the most important attributes that each prediction method should possess. The challenges express the prominent obstacles that each prediction algorithm encounters. The needs explain the reasons that encourage the resources managers to employ the prediction methods. The evaluation metrics explain the performance of the prediction method. In the rest of this section, each one is explained in detail.

2.2.1. The needs for application prediction

As it is mentioned in the previous section, the future demand prediction is crucial for the effective resources provisioning. The main reasons for the application prediction can be summarized as follows:

- Application Management: in the cloud environment, applications share their resources. Therefore, their resources requirements should be predicted properly. It means that resources allocated to each application should be proportional to its workloads. For this purpose, an accurate prediction method is essential to predict the resources requirements of applications. So QoS dropping caused by contention is avoided.
- Resource/Cost Management: to optimize the resources utilization, the prediction of the resources demand is essential. The free resources could be allocated to the applications with more requirements, or could be used to create new VMs. It decreases the extra costs like network, labor and maintenance in addition to the cost of energy wasting (Jiang et al., 2013). Thus, not only resources wasting is avoided, but also the extra costs are reduced.

In summary, the prediction of the future resources requirements of the applications makes the resources manager allocate an appropriate share of resources to each application. Thus, SLA is satisfied and both costs related to the maintenance and the energy consumption are minimized.

2.2.2. Prediction characteristics

The most important characteristics of the application prediction are as follows:

- Accurate: The prediction models are evaluated by the accuracy of the predicted results. The models whose outputs are closer to the actual values are more reliable. The prediction models analyze the historical data of the system and learn the behaviour of the system. Therefore, the accuracy of the prediction model is dependent on the traced behaviour of the application in the past and the nature of the model.
- Adaptability (online learning): The cloud environment is dynamic and is changing continuously. Therefore, the prediction model should be able to adapt to the changes. For this purpose, the model should learn the behavioural changes of the application. The time goes by, the model should be able to improve the learned knowledge of the application behaviour and decrease its prediction error. However, many methods proposed until now could not adapt to the behavioural changes of the application very well.
- Proactive: The VM creation and migration are time consuming

Table 2
The prediction dimensions of applications in different literatures.

The prediction dimension of the application	References
The Number of Requests/Jobs	Liang et al. (2014), Yang et al. (2014a), Kupferman et al. (2009), Chang et al. (2014)
Resources Utilization	Xu et al. (2013), Yang et al. (2014b), Jheng et al. (2014), Akindele and Samuel (2013), Saripalli et al. (2011), Mishra et al. (2010), Zhenhuan et al. (2010), Caron et al. (2010), Islam et al. (2012), Kousiouris et al. (2014), Khan et al. (2012), Bobroff et al. (2007), kioka and Muraoka (2004), Duy et al. (2011), Wu et al. (2007), Dingyu et al. (2012), Bey et al. (2009), Di et al. (2014), Garg et al. (2014), Lu et al. (2015) ,Govindan et al. (2009), Antonescu et al. (2013), Shen et al. (2011), Sheng et al. (2014), Hu et al. (2016), Matsunaga and Fortes (2010), da Silva et al. (2013)
Execution Time of Jobs	Ganapathi et al. (2009), Dinda (2002), Li et al. (2004), Li et al. (2005), Smith et al. (1998), Miu and Missier (2012), Pietri et al. (2014), da Silva et al. (2013), Duan et al. (2009)
SLA Parameters	Akindele and Samuel (2013), Kousiouris et al. (2014), Leitner et al. (2009)
Future Demand of Resources	Jiang et al. (2013), Weijia et al. (2013), Shi et al. (2012), Alasaad et al. (2015), Zhang et al. (2007), Chaisiri et al. (2012), Fang et al. (2012), Prevost et al. (2011), Tang et al. (2014), Meng et al., (), Chen et al. (2015), Amiri et al. (2016)
The Number of Users	Roy et al. (2011)
The Number of Requests of Data Objects	Zhang et al. (2014), Niu et al. (2011), Gursun et al. (2011)
Performance Prediction	Wu et al. (2013), Xiong et al. (2011), Kundu et al. (2012), Toffetti et al. (2010), Li et al. (2009), Calheiros et al. (2011b), Li et al. (2011), Bennani and Menasce (2005), Doyle et al. (2003), Liu et al. (2005), Nathuji et al. (2010), Padala et al. (2008), Rao et al. (2009), Tesauro et al. (2006), Xu et al. (2012), Cuomo et al. (2015), Lee et al. (2015)
Power Consumption	Govindan et al. (2009), McCullough et al. (2011)
The Number of VMs	Liu et al. (2015)
The Number of PMs	Zhang et al. (2012), Urgaonkar et al. (2008), Amiri et al. (2016)

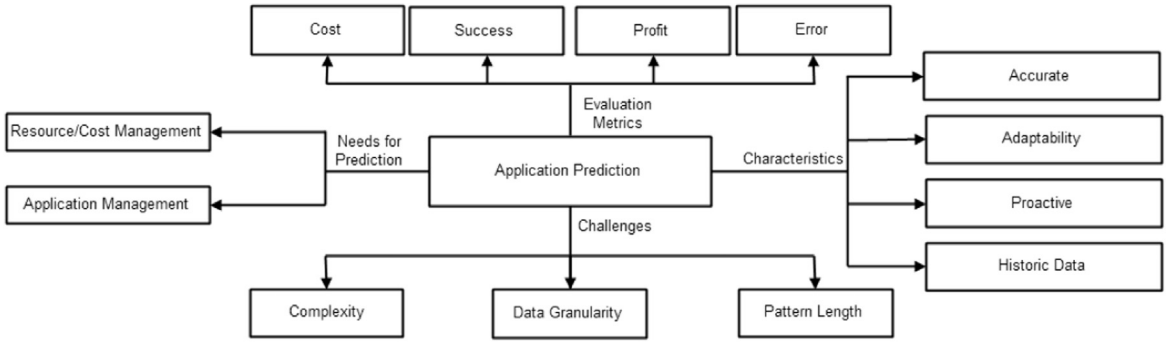

Fig. 3. Main characteristics, challenges, needs and metrics for the application prediction.

processes. Therefore, the prediction should be proactive. It means that before the workload burstiness occurs, the model should be able to predict the future demand sooner in a way that the resource manager has enough time to provide the appropriate resources. In reactive models, according to the system changes, the resources are provisioned. Therefore, users suffer from wasting time for the resources provisioning. It decreases QoS and cloud might lose its customers in the long term (Sun et al., 2015).

- Historic Data: Different types of resources are allocated to cloud services: hardware resources like processor cores, storage and network, and software resources like database connections and thread pools (Singh and Chana, 2016b; Altevogt et al., 2016). The application behaviour is affected by different resources. As it is shown in Fig. 4, the usage traces of different resources should be monitored, collected and recorded. According to the prediction dimensions of the application in Table 2, Table 3 shows the resources, SLA parameters, performance metrics and data objects investigated in the different literatures. As Table 3 shows most of the literatures focus on one or two resources (CPU and memory), SLA parameters and performance metrics. An effective prediction model should investigate all resources and parameters that make an impact on the application behaviour. It should also consider the correlation between resources. Thus, patterns extracted from historic data could show the application behaviour in different dimensions and estimate the future behaviour more accurately.

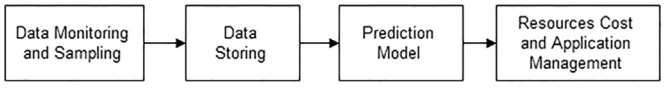

Fig. 4. Phases of the Application Prediction.

2.2.3. Prediction challenges

The challenges of prediction models are introduced as follows that should be handled correctly:

- Complexity: Each prediction model needs the computation resources to estimate the future behaviour of the application. The computation resources consumption of the prediction model should not be significant in comparison with other applications. So, the time and the space complexities of the prediction model should be reasonable in a way that its deployment is affordable.
- Data Granularity: The initial phase for designing the prediction model is to determine which resources should be monitored. In the next step, the length of sampling intervals should be defined. The long term sampling, coarse grained, causes the model to lose the dynamism of the system. The model cannot capture the system behaviour in different situations. On the other hand, the short term

Table 3
Historic data investigated in different literatures for the application prediction.

Historic data		References
Resources	CPU	Liu et al. (2005), McCullough et al. (2011), Garg et al. (2014), Bey et al. (2009), kioka and Muraoka (2004), Jheng et al. (2014), Dingyu et al. (2012), Lazowska et al. (1984), Akindele and Samuel (2013), Wu et al. (2007), Lee et al. (2015), Matsunaga and Fortes (2010), Liu et al. (2005),Zhang et al. (2007), Yin et al. (2014), Dinda (2002), Dinda (2000), Li et al. (2004), Rao et al. (2009), Xu et al. (2012), Xu et al. (2013), Yang et al. (2014b), Yang et al. (2014b), Mishra et al. (2010), Zhenhuan et al. (2010), Caron et al. (2010), Islam et al. (2012), Kousiouris et al. (2014) , Khan et al. (2012), Bobroff et al. (2007), Akindele and Samuel (2013), Di et al. (2014), Wu et al. (2007) Lu et al. (2015), Antonescu et al. (2013), Hu et al. (2016), Shen et al. (2011), Weijia et al. (2013), Fang et al. (2012), Meng et al., (), Liu et al. (2005), Zhang et al. (2012)
	Memory	Jheng et al. (2014), Tang et al. (2014), da Silva et al. (2013), Matsunaga and Fortes (2010), Rao et al. (2009), Xu et al. (2012), Mishra et al. (2010), Shen et al. (2011), Weijia et al. (2013), Meng et al., (), Zhang et al. (2012), Kousiouris et al. (2014), Di et al. (2014),Antonescu et al. (2013)
	Disk	da Silva et al. (2013), Matsunaga and Fortes (2010)
	Network	kioka and Muraoka (2004), Prevost et al. (2011), Chen et al. (2015), Niu et al. (2011)
SLA Parameters	Response Time	Kousiouris et al. (2014), Akindele and Samuel (2013), Leitner et al. (2009)
	Execution Time	Kousiouris et al. (2014)
	Throughput	Akindele and Samuel (2013)
Performance Metrics	SLA Penalty Cost/Total SLA Revenue	Tesauro et al. (2006), Xiong et al. (2011)
	$\frac{Requests(or\ Operations)}{Second}$	Kundu et al. (2012), Padala et al. (2008)
	Response Time	Garg et al. (2014), Padala et al. (2008), Bennani and Menasce (2005), Doyle et al. (2003), Liu et al. (2005), Rao et al. (2009), Cuomo et al. (2015), Xu et al. (2012), Toffetti et al. (2010)
	Throughput	Bennani and Menasce (2005), Rao et al. (2009), Xu et al. (2012), Toffetti et al. (2010)
	Execution Time	Nathuji et al. (2010), Lee et al. (2015), Li et al. (2011)
Data Objects	Video	Niu et al. (2011), Zhang et al. (2014),Gursun et al. (2011)

sampling, fine grained, increases the cost of data collection and processing. It may also include the details that are not useful and the model complexity increases to capture them.

- Pattern Length: In most of the prediction models, the pattern length is fixed. In these models, using a sliding window, the extracted patterns have a predefined length. The constraint of the pattern length restricts the model to the specific patterns and prevents the model from learning the other useful patterns. However, choosing the pattern length is a challenge. The pattern length should be selected in a way that the most popular patterns can be extracted and the application behaviour can be estimated accurately.

2.3. Evaluation metrics

Evaluation metrics describe the performance of the prediction methods. As Fig. 3 shows, evaluation metrics could be classified into four groups:

- Cost: The prediction error leads to SLA violation or resources wasting. The cost metrics are used to measure the cost resulted from the prediction error. The Cloud Prediction Cost (CPC) (Jiang et al., 2013) combines the cost of SLA violation and the cost of idled resources linearly. The importance of costs could be adjusted by the resources manager. The Sacrificed Ratio of the Economy Profit (SREB) (Shi et al., 2012) measures the cost of SLA violation. The Potentially Sacrificed Ratio of the Economy Profit (PSREB) (Shi et al., 2012) investigates both of the costs of resources wasting and SLA violation.
- Success: Success metrics determine how much the prediction method is able to forecast the future behaviour of the application accurately. *SuccessRate* in Di et al. (2014) is defined as the ratio of the number of accurate predictions to the total number of predictions. The accurate prediction falls within some *delta* of the actual value. *PRED*(25) is similar to *SuccessRate* that $delta = 25\%$. R^2 is used to measure the goodness-of-fit of the prediction models (Islam et al., 2012).
- Profit: Profit metrics are used to compute the profit-rate of the cloud provider. The profit-rate is calculated based on the revenue obtained from renting out the resources to the application and the costs of SLA violation and resources wasting (Zhenhuan et al., 2010).
- Error: Error metrics measure the difference between the real behaviour and the predicted behaviour of the application in different ways. The deviation metric considers the difference between the actual value and the predicted value directly. In Khan et al. (2012), the deviation is decomposed into the underestimate and overestimate errors. As it is shown in Table 4, MSSE, MSE, MAPE, RMSE, MAE, MRE, PESD, MER and RSE are employed to measure the prediction error in different literatures. Table 4 shows the type, the name and the abbreviation of metrics and literatures that employ them to evaluate the predicted results.

Although obviously accurate predictions are the best outcome, the costs of SLA violation and resources wasting are not equal in different clouds. The total cost of the prediction error should be computed based on the importance of SLA violation and resources wasting in different clouds (Jiang et al., 2013). For example, the cost of the resources overestimate is more tolerable than resources underestimate's to fulfill QoS parameters. So the cost and the profit metrics could provide a more reliable evaluation of predictors according to cloud's goals.

3. Modeling approaches for the application prediction

In this section, a general taxonomy for proposed models, techniques and frameworks of the application prediction is presented. In general, according to our investigation, prediction models proposed until now could be divided into four groups. As it is shown in Fig. 5, this taxonomy includes table driven methods, control theory, queuing theory and machine learning techniques. In the following subsections, each group is introduced.

Table 4
The evaluation metrics employed in different literatures to evaluate the prediction methods.

Type of Metric	Name of metric	Abbreviation of metric	References
Error	Deviation	–	Yang et al. (2014a), Wu et al. (2013), Shen et al. (2011), Khan et al. (2012)
	Mean Squared Error	MSE	Yang et al. (2014b), Wu et al. (2007), Shi et al. (2012), Xu et al. (2013), Prevost et al. (2011)
	Root Mean Squared Error	RMSE	Islam et al. (2012), Shi et al. (2012), Yin et al. (2014), Bey et al. (2009), Akindele and Samuel (2013), Prevost et al. (2011)
	Mean Segment Squared Error	MSSE	Yang et al. (2014b), Di et al. (2014)
	Relative Squared Error	RSE	Zhang et al. (2012)
	Mean Error Rate	MER	kioka and Muraoka (2004)
	Mean Absolute Percentage Error	MAPE	Islam et al. (2012), Shi et al. (2012), Akindele and Samuel (2013), Duy et al. (2011)
	Mean Absolute Error	MAE	Dingyu et al. (2012), Xu et al. (2013), Kousiouris et al. (2014)
	Mean Relative Error	MRE	McCullough et al. (2011), Dingyu et al. (2012)
	Prediction Error Standard Deviation	PESD	Leitner et al. (2009)
Success	Success Rate	–	Di et al. (2014)
	Prediction in Level 25%	PRED(25)	Yang et al. (2014b), Akindele and Samuel (2013)
	Coefficient of Determination	R^2	Islam et al. (2012), Padala et al. (2008), Garg et al. (2014), Ganapathi et al. (2009)
Cost	Cloud Prediction Cost	CPC	Jiang et al. (2013)
	Sacrificed Ratio of the Economy Profit	SREB	Shi et al. (2012)
	Potentially Sacrificed Ratio of the Economy Profit	PSREB	Shi et al. (2012)
Profit	–	–	Khan et al. (2012)

3.1. Table driven methods

In table driven methods, the application behaviour is recorded in a table for different values of the workload intensity and different amounts of resources allocated to it. The interpolation is used to calculate the values that have not been recorded in the table (Bennani and Menasce, 2005). This method has a low scalability due to the number of applications, different states of the resources allocation and different types of workloads. The table building is time consuming. Furthermore, numerous experiments should be performed to fill the table. According to these points, this method is obsolete and new models do not use it.

3.2. Control theory

In control models, the goal is to control resources shared between cloud applications. If the model controls a resource, for example CPU, a Single Input Single Output (SISO) model is used. According to Fig. 6a, the SISO model correlates the output y to the input u. For example in Liu et al. (2005), the SISO model maps the CPU share of the application to the inverse of its response time. Otherwise, if the controller operates on the multiple resources, a Multi Input Multi Output (MIMO) model is used (Fig. 6b). In Nathuji et al. (2010), the resources usage of all VMs hosted on a server is mapped to their

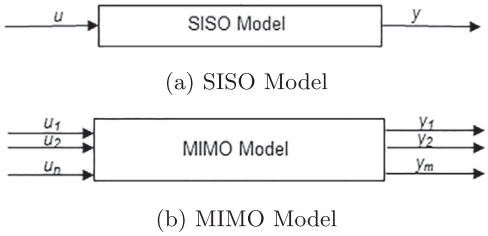

Fig. 6. Models SISO and MIMO of Control Theoty.

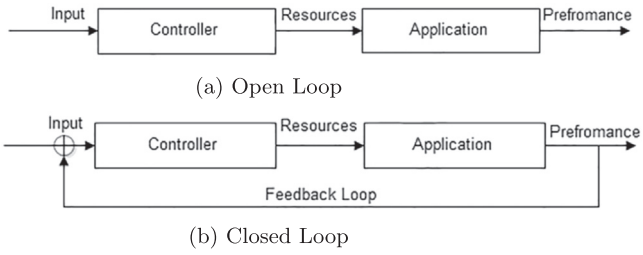

Fig. 7. Open Loop and Closed Loop Control Systems.

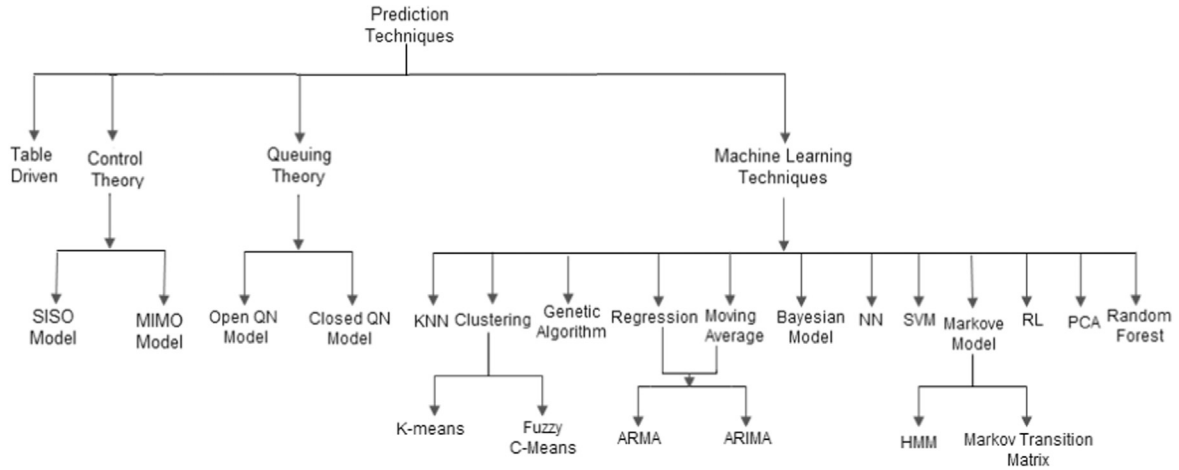

Fig. 5. The taxonomy of prediction methods.

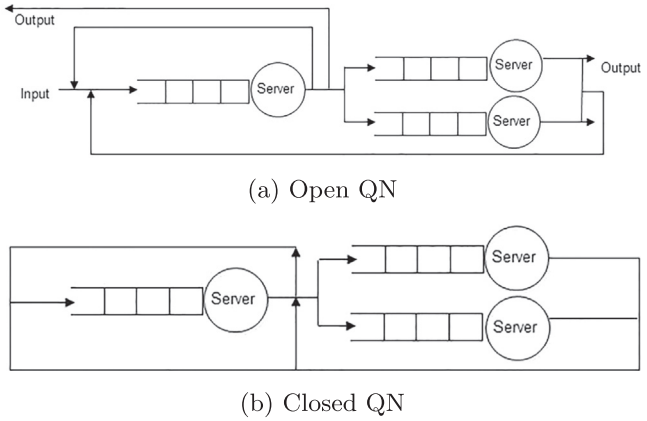

Fig. 8. An example of Open and Closed QN models.

performance by using the MIMO model.

Control systems can be divided into two groups: Open loop control systems and closed loop control systems. In open loop control systems, according to Fig. 7a, the performance of the application (output) depends on allocated resources (the input signal) and the output has no effect on the input to control the resources allocated to the application. In closed loop control systems, Fig. 7b, there is a feedback loop that compares the application performance with the desirable performance. The controller adjusts resources allocated to the application according to its performance goal (Liu et al., 2005).

3.3. Queuing theory

The Queuing Network (QN) model can be used to predict the performance of applications. It models the relationship between the workload and the performance criteria (Wu et al., 2013). In QN, each server allocated to the application is a queuing system (Urgaonkar et al., 2008). The jobs departing from one queue arrive at another queue. These models have parameters such as the requests arrival rate and the average resources requirements of requests that should be specified (Lazowska et al., 1984; Urgaonkar et al., 2008). These parameters can be estimated by solving some equations resulted from the system evaluation.

The open QN and the closed QN are two different types of QN. According to Fig. 8a, in the open QN arrivals and departures could be external. Thus, the number of jobs in the system varies with time (Jain, 2010). As Fig. 8b shows, in the closed QN, there is a constant population in the network and no external sources (virtamo,). In Bennani and Menasce (2005), the response time of the transactional workload and the throughput of batch jobs are modeled by using the open QN and the closed QN respectively.

3.4. Machine learning techniques

The newest proposed approaches are based on machine learning techniques. The machine learning based methods predict the application behaviour in different dimensions. Not only they are used to predict the future behaviour of resources (Jiang et al., 2013; Liu et al., 2015; Akindele and Samuel, 2013), but also they are used to predict SLA violation (Leitner et al., 2009), the application performance (Xiong et al., 2011) and the execution time of jobs (Ganapathi et al., 2009; Miu and Missier, 2012; Li et al., 2004). Furthermore, the machine learning techniques are employed for the resources allocation (Amiri et al., 2016; Xu et al., 2012) and the preprocessing steps such as feature selection (Smith et al., 1998; Li et al., 2005). Most of these methods such as Xu et al. (2013), Yang et al. (2014b), Akindele and Samuel (2013), Jiang et al. (2013) need a training phase to learn the application behaviour. As Fig. 9 shows, in the training phase, machine learning techniques explore the history of the application behaviour. Based on the application experiences in the past, they build a model that can predict the application behaviour in the future.

The machine learning techniques usually model the application behaviour as a time series. Most of the methods are based on a sliding window with length m which includes the previous behaviour of the application in the interval of $[t-m \ldots t-1]$ that t is the current time. According to the constructed model and the previous behaviour of the application in the sliding window, the future state in the time t is predicted. In the next step, the sliding window moves rightwards for one position. Selecting the appropriate length of the sliding window is a challenge. Fig. 10 shows the time series and the sliding window with length m.

Reinforcement Learning (RL) is a knowledge-free machine learning technique (Tesauro et al., 2006). It learns the optimal policies in dynamic environments. Thus, it is appropriate to adapt to the workload dynamics. The RL is used to manage the resources in cloud (Rao et al., 2009; Tesauro et al., 2006; Xu et al., 2012; Huang et al., 2014).

The machine learning and statistics are closely related fields (Unpingco, 2016). Therefore, we consider some statistical prediction methods as machine learning methods. Fig. 5 shows the machine learning approaches used for the application prediction in cloud. These approaches are considered in more detail in the next section.

4. Overview of prediction methods

As it has been mentioned in the previous section, the application prediction methods are classified into four main groups. In a more detailed investigation, the role of methods and techniques employed for the application prediction is summarized in Table 5. Table 5 groups the role of techniques and methods employed in different literatures into three groups prediction methods, preprocessing steps and structure improvement. The preprocessing steps are applied to data before the prediction. The prediction results can be classified into two groups one-step ahead and multi-step ahead predictions based on the interval length of prediction. Some methods such as evolutionary algorithms and fuzzy logic are used to improve the structure of the prediction methods. Briefly, it can be summarized that:

- Most of the methods provide the one-step ahead prediction. A few number of methods provide the multi-step ahead prediction. These methods provide a general view of the future trend of the application behaviour. However, in these methods, the prediction accuracy decreases as the interval length of prediction increases.
- Some methods proposed in literatures such as Xu et al. (2013), Khan et al. (2012), Bey et al. (2009), Kundu et al. (2012), Gursun et al. (2011) apply clustering techniques to the load of resources, tasks and VMs. Clustering groups the similar objects in the same clusters. Thus, clustering based methods usually predict the future trend of each cluster instead of each object. These methods extract the correlation between the similar objects and provide more accurate results. In Chen et al. (2015), clustering is also used for structuring the premise part of the fuzzy system. Furthermore, the dimension reduction methods such as Principal Component Analysis (PCA) (Smith, 2002) provide an efficient representation of data objects with less features (Gursun et al., 2011). They are useful to improve the time and the space complexities of the prediction methods.
- A few numbers of methods smooth the time series of the application behaviour using filters (Wu et al., 2007; Antonescu et al., 2013; Niu et al., 2011). Although filtering provides more accurate prediction results, it eliminates useful information about the behaviour dynamics of cloud applications. Thus, the resource manager could not allocate the appropriate resources to the applications according to their needs.
- The Hurst exponent (Qian and Rasheed, 2007) and Markov models are based on statistical and probabilistic analyses. They model the

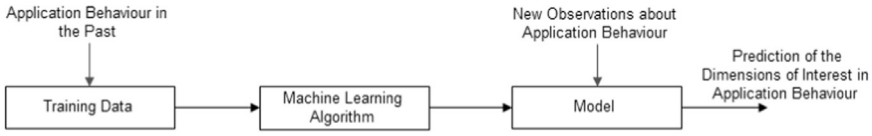

Fig. 9. The phases of machine learning techniques for the application behaviour prediction.

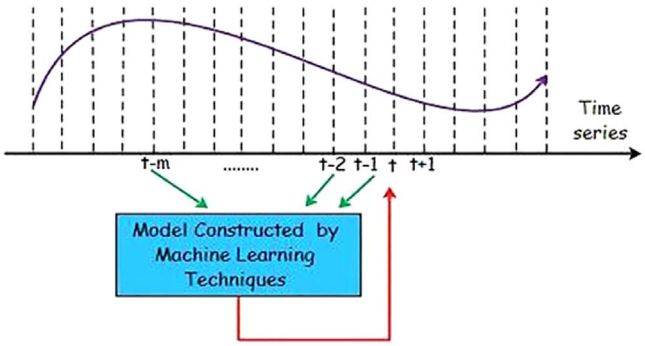

Fig. 10. Employing the sliding window and machine learning techniques on the time series.

future trend of the system based on its past behaviour. Thus, prediction methods can provide more meaningful results.

- Fuzzy logic and evolutionary algorithms improve the structure of prediction methods. Evolutionary algorithms can find the near-optimal structure of prediction methods. Fuzzy logic models uncertainty and ambiguity of the system. Thus, the prediction frameworks provide more interpretable results.

As it is shown in Table 5, some literatures employ several methods. In the following subsections, literatures are grouped based on their proposed methods and explained briefly. Although these groups may have overlaps, it is tried to classify literatures based on their prominent methods.

4.1. Methods based on machine learning and statistical techniques

In this section, methods based on machine learning techniques, statistics and mathematics are investigated.

4.1.1. Regression and moving average

Yang et al. in Yang et al. (2014a), propose a method based on Linear Regression (LR) (Adhikari and Agrawal, 2013) to predict the number of requests for each cloud service. According to the workload fluctuations, the prediction method adjusts itself through the recomputation of parameters of the regression model. The authors assume the workload trend is almost linear in the short term intervals. An auto scale model is also proposed in the VMs level.

The prediction method proposed in Roy et al. (2011) is based on the second order Auto Regressive Moving Average (ARMA) (Adhikari and Agrawal, 2013). The workload is the number of users. The parameters of the ARMA model are fixed. It is clear that fixed values for parameters are not suitable because they cannot be adapted according to the workload dynamism.

Saripalli et al. in Saripalli et al. (2011) propose a two-step approach that includes the Load Tracking (LT) and the Load Prediction (LP). In the first step, LT extracts a representative view of the load trace. The proposed cubic-spline LT can model the high fluctuations of the load better than the other linear LTs based on Moving Average (MA) (Adhikari and Agrawal, 2013). LP predicts the next LT value based on the line connecting the first and the last LT values.

Dinda in Dinda (2002) proposes a Running Time Advisor (RTA) to predict the running time of compute-bound tasks based on their nominal time (the running time of the task on a vacant machine) and the predictions of the host load. The host load is predicted by using Auto Regressive (AR) and MA. RTA estimates the running time of the task and a confidence interval for it. Based on the prediction results, a Real-time Scheduling Advisor (RTSA) (Dinda, 2000) determines the most appropriate host for the task.

Three algorithms, Right Scale, LR and AR are compared in Kupferman et al. (2009). The Right Scale algorithm is a democratic voting process that decides to scale up or scale down the infrastructure according to the majority of machines votes. In the AR model, there are the history and the adaptation windows. The output of the model is the average value of the adaptation window. To fill the adaptation window, the prediction process is repeated based on the values of the history window and the elements of the adaptation window that have been predicted already. Therefore, the prediction error of the first elements of the adaptation window expands to the subsequent elements and grows gradually. LR finds a function that is the closest to the set of points. Experimental results show that Right Scale has the least performance among three approaches. It is also highly dependent on the threshold values. LR is susceptible to the small fluctuations. Although AR is more reactive than LR, it is not effective enough for the resources deprovisioning.

Tang et al. in Tang et al. (2014) propose a bin-packing algorithm that selects a PM for each VM, according to its future memory usage. In addition to allocating enough memory to each VM, the number of PMs should be minimized. The memory usage of VMs is modeled as a random variable. To predict the probability distribution of the future memory demand, the AR model is used. The model parameters are updated for each VM frequently.

In Zhang et al. (2012), a control model is proposed to adjust the number of servers. The resources usage of PMs is predicted by using Auto Regressive Integrated Moving Average (ARIMA) (Adhikari and Agrawal, 2013). The ARIMA model is an ARMA model that has been differenced several times. This method repeats the one-step ahead prediction to provide the multi-step ahead prediction. Finding the optimal number of PMs is modeled as a convex optimization problem.

Miu et al. in Miu and Missier (2012) define “input predictability” of algorithms. According to this property, the accuracy of prediction results of predictors depends on selected features of their inputs. The authors identify a combination of features to learn the regression models that predict the execution time of jobs accurately.

In McCullough et al. (2011), the effect of the complexity and the variability of the hardware is checked based on the accuracy of prediction models of the power consumption. The different types of linear and non-linear regression models are compared. Experimental results show that linear regression models provide reliable results in the single core case. In the multi core case, finding an appropriate prediction model for all benchmarks is not possible. The error of prediction models is also high for the CPU power. Furthermore, as the complexity increases and the internal state changes, the prediction error increases.

Fang et al. in Fang et al. (2012) classify a cloud data center into several groups. The future demand of resources of each group is predicted by using ARIMA based methods. The predicted demand is fed into a group controller that manages the cloud nodes of the group. A dispatcher also dispatchs requests among groups.

Smith et al. in Smith et al. (1998) predict the runtime of parallel

Table 5
The role of methods and techniques used for the application prediction in different literatures.

Methods and techniques	Prediction method	Preprocessing steps	Structure improvement	References
Methods based on Regression and Moving Average	✓			Yang et al. (2014a), Akindele and Samuel (2013), Jiang et al. (2013), Roy et al. (2011), Saripalli et al. (2011), Islam et al. (2012), Kupferman et al. (2009), Wu et al. (2007), Nathuji et al. (2010), Padala et al. (2008) Zhang et al. (2012), Prevost et al. (2011),Wu et al. (2013), Weijia et al. (2013), Tang et al. (2014), Niu et al. (2011), Shi et al. (2012), Gursun et al. (2011), Liu et al. (2005), Zhang et al. (2007) McCullough et al. (2011),Fang et al. (2012), Liu et al. (2015), Hu et al. (2016), Dinda (2002), Matsunaga and Fortes (2010), Li et al. (2004), Li et al. (2005), Smith et al. (1998), Miu and Missier (2012)
Bayesian Theory	✓			Di et al. (2014), Duan et al. (2009)
K Nearest Neighbor (KNN)	✓			Akindele and Samuel (2013), Li et al. (2005)
Random Forest	✓			Cetinski and Juric (2015)
Neural Network (NN)	✓			Xu et al. (2013), Yang et al. (2014b), Akindele and Samuel (2013), Jiang et al. (2013), Islam et al. (2012), Duy et al. (2011), Prevost et al. (2011), Kousiouris et al. (2014), Garg et al. (2014), Kundu et al. (2012), Li et al. (2009), Leitner et al. (2009), Chang et al. (2014), Matsunaga and Fortes (2010), Duan et al. (2009), Chen et al. (2015)
Support Vector Machine (SVM)	✓			Akindele and Samuel (2013), Kundu et al. (2012), Jiang et al. (2013), Liu et al. (2015), Matsunaga and Fortes (2010)
Markov Model		✓		Xu et al. (2013), Zhenhuan et al. (2010), Khan et al. (2012), kioka and Muraoka (2004), Lu et al. (2015)
Clustering/Classification		✓	✓	Xu et al. (2013), Khan et al. (2012), Bey et al. (2009), Meng et al., (), Kundu et al. (2012), Mishra et al. (2010), Gursun et al. (2011), Pietri et al. (2014), da Silva et al. (2013), Chen et al. (2015)
Dimension Reduction		✓		Gursun et al. (2011)
Reinforcement Learning (RL)	✓			Rao et al. (2009), Tesauro et al. (2006), Xu et al. (2012), Huang et al. (2014), Amiri et al. (2016)
String Matching	✓			Caron et al. (2010), Dingyu et al. (2012), Hu et al. (2016)
Evolutionary Algorithm (Genetic Algorithm)			✓	Yang et al. (2014b), Jiang et al. (2013), Antonescu et al. (2013), Li et al. (2005), Smith et al. (1998)
Kernel Canonical Correlation Analysis	✓			Ganapathi et al. (2009)
Fuzzy Logic			✓	Bey et al. (2009), Chen et al. (2015)
Hurst Exponent		✓		Lu et al. (2015), Govindan et al. (2009)
Periodogram	✓			Bobroff et al. (2007)
Workload Generator/Workload Factoring		✓		Yin et al. (2014), Ganapathi et al. (2009), Zhang et al. (2014), Li et al. (2011),Cuomo et al. (2015)
Signal Processing	✓			Zhenhuan et al. (2010), Shi et al. (2012), Shen et al. (2011)
Mathematical Derivation	✓			Jheng et al. (2014), Dingyu et al. (2012)
Smoothing Using Filters		✓		Wu et al. (2007), Antonescu et al. (2013), Niu et al. (2011)
Multi-Step Ahead	–	–	–	Wu et al. (2007), Dingyu et al. (2012), Zhang et al. (2012)
QN Model	✓			Bennani and Menasce (2005), Padala et al. (2008), Wu et al. (2013), Doyle et al. (2003), Zhang et al. (2007), Urgaonkar et al. (2008), Calheiros et al. (2011b)
Control Theory	✓			Liu et al. (2005), Nathuji et al. (2010), Padala et al. (2008)
Optimization Problem	✓			Sheng et al. (2014), Alasaad et al. (2015), Zhang et al. (2012), Chaisiri et al. (2012)
Curve Index Model	✓			Liang et al. (2014)
Histogram/Probability Distribution/ Known Workload	✓			Urgaonkar et al. (2008), Chaisiri et al. (2012), Calheiros et al. (2011b), Toffetti et al. (2010)

applications based on the similar applications that have run in the past. The similar applications are found based on the characteristics. The goal is to find the application characteristics that could identify the most similar applications for the runtime prediction. The genetic algorithm and the greedy search are used to identify the application characteristics. LR and Mean predict the runtime of applications based on the similar applications. According to experiment results, the genetic algorithm finds the best characteristics and Mean could provide more accurate results. In Smith et al. (2004), the authors show the combination of predictors improves the prediction results.

Weijia et al. in Weijia et al. (2013) present an online bin-packing algorithm to optimize the number of servers and allocate resources based on the VMs migration. A load predictor should estimate the resources demand of VMs and the load of PMs. A modified version of Exponentially Weighted Moving Average (EWMA) is proposed. It predicts the future state based on the increase or the decrease of the last observed value. The modified version, Fast Up Fast Down (FUFD), also uses the negative values for its weights.

Li et al. in Li et al. (2004) predict the start time of jobs on clusters. At first, the execution time of running and queued jobs are predicted by using AR and LR. The results of predictors are combined by using a technique whose error and performance are reasonable. Based on the prediction results, a scheduler is simulated to determine how jobs are scheduled. Thus, it is possible to estimate how long it will take before a newly-submitted job starts its execution.

Niu et al. in Niu et al. (2011) propose the regression model Box Jenkins Adhikari and Agrawal (2013) to predict the future population for new videos based on the release time of videos and the access behaviour of users. The logarithmic transformation is applied to equalize fluctuations before applying the method Box Jenkins. It removes the system dynamics and accurate behavioural information. The server bandwidth demanded by channels is also predicted by using seasonal ARIMA and AR. In these models, finding the correct values of parameters is critical.

In Li et al. (2005), the accuracy and the efficiency of two feature selection algorithms, Improved Reduct, the new version of Reduct algorithm (Hu, 1995), and the genetic algorithm are considered to predict the runtime of the application on clusters. Although both of the algorithms provide the same accurate results, Improved Reduct is more computationally efficient than the genetic algorithm. Two prediction methods, Similarity Templates and Instance Based Learning (IBL), are also evaluated. The Similarity Templates method clusters jobs according to the selected features. For each cluster, methods such as MA and LR are used to predict the runtime of jobs. IBL algorithms such as K-Nearest-Neighbor (KNN), Weighted Average (WA) and Linear Locally Weighted Regression (LLWR) find the nearest neighbors of the job according to the selected features and predict its run time based on the nearest jobs. According to experimental results, the Similarity Templates method is more accurate than IBL algorithms.

Advantages and disadvantages: Although regression-based methods are simple, their reliance is based on the oversimplified assumptions of the application workload (the linear relationship). These methods are trained and parameterized based on the past observations of the application behaviour. Therefore they cannot capture the behavioural changes of applications. Indeed, as the prediction error increases, regression-based methods should be re-trained to adapt to the changes of the workload. It takes a lot of time and resources.

4.1.2. Neural network

In Duy et al. (2011), a feed forward Neural Network (NN) is used to predict the host load. The linear prediction methods cannot model the non-linear behaviour of the host load and cannot be adapted to the fluctuations. The proposed method improves the restrictions of linear methods.

Li et al. in Li et al. (2009) model the task performance using machine learning techniques to predict the performance across the program execution. The training data is collected from the execution of functions. For each function, an NN whose outputs are the execution time and the size of the output data is trained.

In Leitner et al. (2009), a method is presented that predicts SLA violation in the runtime of composite services based on check points. The check points are specified by the user and the prediction is performed in them. The check points should be selected in a way that there is enough time to react to SLA violation. An NN is used to predict SLA violation. To adapt to the behavioural changes, the prediction model is trained periodically.

Garg et al. in Garg et al. (2014) consider the resources allocation in a data center that includes the non-interactive and the transactional workloads. The proposed method predicts the CPU utilization of transactional applications by NN. The job scheduler tries to run the non-interactive jobs with transactional applications whose resources demand is highly dynamic. Indeed, during the under-load of transactional applications, the CPU cycles are stolen and allocated to the batch jobs.

In Kousiouris et al. (2014), an approach is presented to correlate low level resources attributes with high level information of applications and users' behaviour. The proposed method has two layers. Each layer is modeled by using NN: 1) The translation layer considers the workload and the resources parameters as input and predicts QoS of the application, 2) The behaviour layer predicts the application workload for the translation layer. The time series of the workload is the input of this layer.

Duan et al. in Duan et al. (2009) suggest a hybrid Bayesian-neural network method to estimate the execution time of workflow activities in Grid. The Bayesian network estimates the performance probability distribution against different factors affecting the performance. In the next step, factors that have a low influence on the execution time are detected and eliminated by using Pearson Product-Moment Correlation Coefficient (PMCC) (Kornbrot, 2005). NN exploits the probability distribution to predict the execution time of the workflow activities.

Yang et al. in Yang et al. (2014b) use Phase Space Reconstruction (PSR) (Xu, 2009) before the host load prediction. The reconstructed multi dimensional time series is fed to the Group Method of Data Handling (Farlow, 1981) based on an Evolutionary Algorithm (EA-GMDH network). The proposed EA-GMDH eliminates the structural constraints of GMDH. At first, the best structure of EA-GMDH is found in the training phase. Then, the trained network is used to predict the future host load.

Chang et al. in Chang et al. (2014) predict the workload of cloud servers using a recurrent neural network (Mell and Grance, 2002). They consider the workload as the number of processes assigned to servers. Their evaluation results show that the recurrent neural network needs a fewer number of training data in compared to the feed forward neural network. It could also predict the rapid changes of the workload better than the regression method.

Chen et al. in Chen et al. (2015) propose a system to predict the resources demand. Due to the workload dynamics in different periods, the base predictors such as the Second Moving Average model (SMA), the Exponential Moving Average method (EMA), the AR model, and the trend seasonality model (TSM) are selected. The output of the base predictors is sent to a Fuzzy Neural Network (FNN) which improves the accuracy of the prediction results. The clustering algorithms are used to optimize the FNN system.

Advantages and disadvantages: NNs do not require restrictive assumptions about the form of the application workload and can model well the nonlinear behaviour of the application. They can also be used to model the correlation among the usage of different resources of the application. The structure of NNs is determined based on historical data. Thus, retraining NNs is essential to adapt to the behavioural changes of the application workload. Furthermore, NN is a "black-box" method and cannot provide any insights into the behavioural patterns

of the workload for the resources manager.

4.1.3. Markov model, clustering, dimension reduction and fuzzy logic

Xu et al. in Xu et al. (2013) cluster the load before the prediction. For each cluster, the best Hidden Markov Model (HMM) (Blunsom, 2004) is constructed by using the Bayesian Information Criterion (BIC) (Watanabe, 2013). Based on the cluster of the current load, an Elman NN is used to predict the future load. The Elman network is optimized by the genetic algorithm. Due to the real-time nature of the cloud environment, the required time to optimize NN by the genetic algorithm is challenging.

In Khan et al. (2012), VMs that show the similar behaviour over time are grouped in a co-cluster. In the next step, the predictable groups are constructed. Each group includes the predictable and the predicting co-clusters. For each predictable group, an HMM is defined. The goal is to predict the next observation based on the current observation. The most probabilistic observation is selected. According to the reported results, this method cannot be applied to all VMs because some VMs cannot be clustered in any groups.

Bey et al. in Bey et al. (2009) propose a method to predict the CPU load. The method is based on fuzzy clustering and an Adaptive Network based Fuzzy Inference System (ANFIS) (Jang, 1993). In the first step, the time series of the CPU load is clustered by using fuzzy C-means. Using ANFIS, the CPU load of each cluster is predicted. The final result is the weighted sum of the predicted values. Experimental results show that the prediction error is correlated with the number of clusters. This method cannot also predict the sudden spikes in both under-load and overload states.

Meng et al. in Meng et al., (2010) present a resources provisioning approach based on the VM multiplexing. According to the point that the peak and the valley of the VMs demand don't usually coincide, the joint-VM resource provisioning leads to more utilization of resources. The required resources for each joint-VM are estimated. According to the estimated joint size, sufficient resources are allocated.

In Kundu et al. (2012), a sub-modeling technique is proposed. This technique divides the space of input parameters into several non-overlapping subregions by using clustering techniques. For each subregion, a model is constructed.

The task scheduling is considered as a bin-packing problem. To reduce the time complexity of the bin-packing, a task classification method is proposed in Mishra et al. (2010). For this purpose, the tasks are characterized in two dimensions: the time execution and the resources consumption. Each dimension is described by using several qualitative coordinates. Finally, after the representation of tasks in the new space, K-means is used to construct the classes of tasks. In this method, choosing the appropriate break points is difficult for each qualitative coordinate. It seems the fuzzy classification is a better approach to select the break points.

Gursun et al. in Gursun et al. (2011) present a method that exploits the inherent structure of data to predict the number of the video access. This method is based on PCA and clustering. According to the access frequency of videos, they are divided into two groups, frequent access and rare access. For the frequent access group, common patterns are extracted by using PCA. The extracted principal components are predicted by using the ARMA method. The combination of predicted principal components estimates the future access number of videos. In the rare access group, the access peak of videos is determined. The areas around the peaks are extracted and normalized. These created time series are clustered by using the bottom-up hierarchical clustering technique. After mapping the video to the appropriate cluster, the future number of the video access is specified by using the scaled mean of the cluster.

Pietri et al. in Pietri et al. (2014) predict the execution time of the scientific workflows based on the structure of workflows and the runtime characteristics of their tasks. Based on data dependency between the workflow tasks, a Top-Down Approach (TDA) and a Bottom-Up Approach (BUA) group the tasks into some levels. The characteristics of each level are determined according to the tasks assigned to it. Finally, a simple analytical level-based model is proposed to predict the execution time of the workflow.

Silva et al. in da Silva et al. (2013) predict the task parameters such as the runtime, the disk space, and the memory consumption based on the size of input data of tasks. At first, tasks are classified by the workflow type and the task type respectively. Based on the collected dataset, the correlation between each parameter and the size of input data is calculated. If a parameter is not correlated with the size of input data, the dataset is split into smaller groups by using a clustering technique. Finally, if the parameter is correlated with the size of input data, it is estimated according to the ratio *parameter/input data size*, otherwise, it is predicted as a mean value.

Akoika et al. in kioka and Muraoka (2004) determine the CPU load as the sum of the mean value of the CPU load, the seasonal variations and the irregular variations. However, the irregular variations are removed. The Markov Model is used to provide the one-step ahead prediction of the CPU load. This model is also extended to provide the one-step ahead prediction of the network load.

Advantages and disadvantages: The Markov model is derived from historical data easily. The transition matrix of the Markov model is understood by the resources manager readily. However, the assumption of the Markov Model is very restrictive and is not valid for many application workloads. The Markov model could not be used to predict the application behaviour for the long periods of time. To adapt to the workload changes, the Markov model should be rebuilt based on the recent observations. The clustering techniques are applied to the load of resources, tasks and VMs. The prediction methods are applied to each cluster instead of single objects. It decreases the time and the cost of the prediction. It also captures the correlation among VMs or resources. Thus, the prediction methods could provide better results. However, selecting an appropriate clustering algorithm and the number of clusters are the principal challenges. Fuzzy logic models the uncertainty and ambiguity of data. It simulates the human reasoning and provides the interpretable results for the resources manager. It also enables the resources manager to incorporate its initial knowledge of the application behaviour in the prediction methods. Thus, the prediction methods can provide more accurate results. However, the fuzzy rules, the membership functions and the inference systems should be chosen cautiously in a way that knowledge of the resources manager and information of the past behaviour of applications can be represented well.

4.1.4. Hurst exponent and bayesian theory

In Lu et al. (2015), two known statistical approaches, the Hurst exponent and the Markov transition matrix, are used to evaluate VMs. At first, the Hurst exponent is used to identify the long term correlations of VMs. The Hurst exponent value shows the predictability of the VM behaviour. If there is no Hurst exponent value for VMs, the Markov transition matrix is used to evaluate VMs. However, this method cannot determine predictability of the VM behaviour. These approaches can be used to allocate resources to VMs dynamically.

Govindan et al. in Govindan et al. (2009) use the profiling to determine the power consumption and the resources usage of applications. The time series of the power consumption of each application is converted to the Probability Density Function (PDF). The proposed method calculates the value of Hurst exponent to specify the degree of self-similarity of the time series of the application power. Based on the Hurst parameter and the determined burstiness, an appropriate strategy is taken to provision the power needs of workloads.

Di et al. in Di et al. (2014) propose a host load prediction method for the long term intervals. Each prediction interval is divided into a sequence of sequential segments whose length increases exponentially. The goal is to predict the mean load of the host in each interval. The prediction method is based on the Bayes classifier. However, the mean

load prediction is challenging because the mean load cannot present a representative view of the load fluctuations in the long term intervals.

Advantages and disadvantages: The Hurst exponent is simple. It determines whether the application behaviour is predictable. It provides information about the future trend of the application behaviour based on the historical data. However, the Hurst exponent is not computable for all applications. Therefore, it can be used along with other prediction methods to improve the prediction results. The Bayesian theory is simple. It is able to incorporate initial information of the resources manager as prior probabilities in the predictor. However, if the resources manger cannot provide prior information, it should be estimated from the past behaviour of the application. The features describing the application behaviour should also be mutually-independent. Note that to adapt to the behavioural changes of the application, the prior probabilities should be recomputed.

4.1.5. Histogram, probability distribution and benchmarking

Urgaonkar et al. in Urgaonkar et al. (2008) predict the peak demand of resources by using the histogram of the historical observations of the workload. According to the predicted demand, the queuing model determines the number of servers that should be allocated to each tier of multi-tier applications.

In Chaisiri et al. (2012), the demand of VMs is modeled as a probability distribution. Based on the optimization formulation of stochastic integer programming, an optimal cloud resource provisioning (OCRP) algorithm is proposed to minimize the cost of the resources provisioning.

Toffeti et al. in Toffetti et al. (2010) present a surrogate model to predict the performance of the application as a function of the tunable configuration parameters. The proposed surrogate model is Kiriging (Bachoc, 2014) that interpolates the space of the application parameters by the Gaussian process. It is claimed that this model provides a more accurate result than the regression analysis and it can be computed quickly.

Advantages and disadvantages: Although methods based on the histogram and the probability distribution are simple and their computational cost is trivial, their predicted results are not reliable. These methods assume that the distribution of workloads are known and fixed over time that it is inconsistent with the dynamic nature of cloud.

4.1.6. Mathematical derivation and string matching

In Jheng et al. (2014), the average utilization of resources CPU, Memory and RAM of PMs is predicted by using the Grey forecasting model. The Grey model does not need much training data. This model is based on the simple mathematical derivations. There are two challenges in this method: 1) The time correlation is assumed, 2) The training data is sampled at a large distance. Experimental results show that this method cannot guarantee the reliable prediction results for workloads with high fluctuations.

In order to avoid the growth of the error for the n-step ahead prediction, Dingyu et al. in Dingyu et al. (2012) propose a derivative-based approach to provide the multi-step ahead prediction of the CPU load. For this purpose, the changes of the CPU load (the first order derivative) are considered. The future amount of the change is predicted by using a local polynomial function. To predict the change trend, the time series of it is constructed by using two values −1(decrease) and +1 (increase). With identification of the trend and the amount of the change, the next value of the CPU load is estimated. Although the method is interesting, choosing the length of the Immediately Preceding Sequence (ISP) and the order of the polynomial function are challenging.

The method proposed in Caron et al. (2010) is based on identifying patterns in the system history that are similar to the current pattern of the system. Therefore, the similar patterns are extracted from the history and are interpolated according to their similarities to the current pattern. The similar patterns identification is modeled as a string matching problem. Reported experimental results show that the pattern length has a significant impact on the prediction results.

Hu et al. in Hu et al. (2016) consider three prediction models, some time series approaches, a Kalman filter based model (Faragher, 2012) and a pattern matching model. They propose a new trigger strategy based on the pattern matching model to reduce the automatic scaling delay of threshold based strategies. The elasticity mechanism is triggered based on the trend increase and the CPU workload at the moment.

Advantages and disadvantages: The methods based on the mathematical derivation are sound in theory. They usually need no training data. They are quick ways to describe the changes of the workload. Thus, they are appropriate for the application workloads that are changing rapidly. Although methods based on the string matching are fast, they need a preprocessing step: the time series of the application workload should be converted to a discrete form. The length of patterns and the number of alphabet symbols for discretization of the time series affect the prediction results significantly. Thus, choosing the length of pattern is challenging.

4.1.7. Combination of support vector machine, neural network and regression

Jiang et al. in Jiang et al. (2013) use a group of prediction methods for the future demand prediction of VMs and the capacity planning. The authors use five prediction methods for the time series prediction. The individual results are combined linearly. The initial weights of predictors are equal. According to the prediction error of the methods, the weights are updated. The group method and the individual methods are compared based on the prediction cost. According to the reported results, the group predictor provides the least cost among the prediction methods.

Liu et al. in Liu et al. (2015) classify the service workloads based on their dynamism and assign a prediction model to each workload category. They model the workload classification as an optimization problem that maximizes the prediction accuracy. LR and SVR are used to predict the slow and the fast changes of the workloads respectively.

In Akindele and Samuel (2013), a set of machine learning techniques (NN, Support Vector Machine (SVM) and LR) is used to predict the CPU utilization of VMs. These methods are also used to predict SLA parameters (response time and throughput). The prediction results show that SVR, composed of LR and SVM, provides better results than LR and NN.

Matsunaga et al. in Matsunaga and Fortes (2010) consider several machine learning algorithms and propose a new version of the algorithm Predicting Query Runtime (PQR) (Gupta et al., 2008), PQR2, to predict the resources utilization of applications. PQR generates a binary tree whose nodes are regression classifiers that are selected according to their accuracy and leaves are the range of the feature that should be predicted. Thus, PQR could combine different classifiers. Instead of outputting the range on leaves, PQR2 adds regression functions at the leaves and selects the best regression model for data. According to the experiment results, no machine learning algorithm could provide the best results for different applications and PQR2 could provide more accurate prediction results than PQR.

Islam et al. in Islam et al. (2012) evaluate two methods NN and LR to predict the resources usage. They investigate the data collected from the TPC-W benchmark with two situations the presence and the absence of the sliding window. Experimental results show that using the sliding window along with both NN and LR improves the prediction results. However, the size of the sliding window has the great effect on the prediction accuracy.

The prediction tools such as Right Scale and Enomaly that are used for the automatic failure management and the resources provisioning, only consider SLA satisfaction. Therefore, the allocated resources are usually overestimated (Prevost et al., 2011). Prevost et al. in Prevost

et al. (2011) suggest a prediction model to predict the future load of the network. The goal is to service the input requests with the minimum amount of the required power. NN and the linear predictor are compared. Evaluation results show that the linear predictor provides more accurate results.

Xiong et al. in Xiong et al. (2011) present a management system of resources that is composed of two components: 1) The system modeling module predicts the system performance for a given resources allocation. This module can be modeled by using machine learning techniques. 2) The resources allocation module considers the resources of hosts shared between VMs and the number of replica for each VM. The system modeling module estimates the system performance for each plan of the resources allocation searched by the resources allocation module. The grid search is used to find the near optimal allocations.

Vazquez et al. in Vazquez et al. (2015) evaluate several time series forecasting models, AR, MA, simple and double exponential smoothing (Hyndman et al., 2008), ETS (error, trend, seasonal) (Hyndman et al., 2008), ARIMA and NN, for their ability to predict the real cloud workloads. According to their experiment results, there is no model that can provide the most accurate results. The prediction accuracy of methods depends on the considered workloads.

Advantages and disadvantages: The methods based on the regression and NN were considered in Sections 4.1.1 and 4.1.2 respectively. The SVM based methods can model the nonlinear behaviour of applications. They are able to provide the multi-step ahead prediction. Due to the black box nature of SVM, any insights into the behavioural patterns of the workload cannot be provided. It is not easy to incorporate the initial knowledge of the resources manager in the structure of SVM. The parameters of SVM should be learned from the past behaviour of applications. Learning SVM takes a long time. Note that to adapt to behavioural changes of applications, the learning process of SVM should be repeated.

4.1.8. Filtering and signal processing

The prediction method proposed in Zhenhuan et al. (2010), PRESS, is composed of signal processing methods and statistical learning algorithms. At first, PRESS computes dominant frequencies of the resources usage variations using the Fast Fourier Transform (FFT). Based on the extracted frequencies, the size of the pattern window is determined and the time series is divided into several pattern windows. If all pattern windows are similar, PRESS detects the repetitive behaviour in the time series and predicts the future state based on the average values of the pattern windows. Otherwise, PRESS reverts to the statistical state-driven method. By using the Markov chain, the most probabilistic state is selected as a future state.

Shen et al. in Shen et al. (2011) suggest Cloud Scale which provides the elastic resources scaling. The Cloud Scale uses PRESS, the hybrid prediction method presented in Zhenhuan et al. (2010). Cloud Scale provides two complement schemes to handle the prediction error: adaptive online padding adds dynamic values to the amounts of predicted resources demand to reduce the risk of the resources underestimate. If the adaptive online padding cannot prevent the error of the underestimate, the reactive error correction detects and corrects it.

Shi et al. in Shi et al. (2012) suggest a prediction method based on the wavelet analysis to forecast the resources demand of cloud services in the Content Delivery Network (CDN). In the first step, the time series of the resource demand is decomposed to the low and the high frequencies. The time series corresponding to the low and the high frequencies are reconstructed. The future value of each reconstructed time series is predicted by using the ARMA method. The final value is the weighted sum of the predicted results.

Wu et al. in Wu et al. (2007) propose a hybrid model composed of AR with the confidence interval to predict the CPU load of the host. Firstly, Kalman filter is used to minimize the error of measurement. Then, the smoothing filter Savitzky-Golay (Schafer, 2011) is applied to smooth the load fluctuations. In the next step, the coefficients of AR are computed and the n-step ahead prediction is provided. It increases the prediction error gradually for further steps. The authors assume that load is stable. It is clear that the cloud environment is dynamic and the filtering eliminates useful information about the dynamic behaviour of the CPU load.

In Antonescu et al. (2013), a model is proposed to find the best mapping VMs to hosts. It is assumed that VMs experience the behaviour with the seasonality patterns. The Holt-Winter algorithm (Kalekar, 2004) is also used to smooth utilization data. This algorithm considers the data trend and the seasonality. It decreases the weights assigned to older data exponentially. The group oriented genetic algorithm is used to find mapping VMs to hosts. The convergence time of the genetic algorithm, the seasonality behaviour of VMs and smoothing data are challenging.

In Bobroff et al. (2007), the periodogram is used to identify the dominant periods or frequencies of the time series of the resource demand. After identifying the periodic components, VMs that show the periodic behaviour are selected for the dynamic resources management. If the prediction error increases, the periodic components should be computed according to the demand variations again.

Advantages and disadvantages: Filters are used to smooth the time series of the workload. Filtering usually provides the accurate results for linear models. Due to the non-linear behaviour and the load dynamics of cloud applications, filtering does not seem appropriate here. It eliminates the useful information about the load dynamics. Thus, the resource manager cannot allocate the appropriate resources to applications. The signal processing techniques, FFT and the wavelet transform, extract the cyclic and the acyclic workload patterns respectively. They need no restrictive assumption about the workload behaviour. In the fourier transform, the detailed time information of the workload patterns is lost in the long-term. The Fourier transform is very sensitive to the workload fluctuations.

4.1.9. Workload generator and workload factoring

Yin et al. in Yin et al. (2014) propose a workload generator based on a 2-state Markovian arrival process. Based on this generator, a performance analysis method is suggested that predicts the PDF of the CPU utilization. The generator produces workloads with different intensities of burstiness and subsequently the PDF of the CPU utilization is predicted in different situations.

Cuomo et al. in Cuomo et al. (2015) propose a methodology for generating the performance model of the application from its descriptor. The application descriptor includes all the application components, the cloud resources (queues and key-value stores) and the details of their interconnections. The benchmark model is made up of a workload generator that focuses on the component that should be benchmarked. A set of the application benchmarks and a simulation model are derived from the application descriptor. The simulation model predicts the performance behaviour of the application based on the performance parameters extracted from the benchmarks execution.

Ganapathi et al. in Ganapathi et al. (2009) suggest a statistical framework using the Kernel Canonical Correlation Analysis (KCCA) (Hardoon et al., 2004) to predict the execution time of the map-reduce jobs. After building the model KCCA, three nearest neighbors of the intended job are determined and their performance vectors are extracted from the subspace β. The final performance vector of the intended job is predicted by using a weighted average of the performance vectors of the nearest neighbors. Having a real workload is a precondition for the implementation and the evaluation of this framework. Therefore, a workload generator is designed that generates the anonymized workloads from the map-reduce traces.

Zhang et al. in Zhang et al. (2014) propose an intelligent workload factoring service. The workload factoring process is modeled as a hyper-graph partitioning problem. The data objects of applications and requests of services are modeled as vertexes and nets of the hyper-

graph respectively. This problem is converted to a knapsack problem and a greedy approach is used to decompose the workload: vertexes of the base workload zone that have a higher popularity, are transmitted to the flash workload zone. The authors suggest a hybrid cloud model. The base workload could be managed in small data centers and the flash crowd load should be handled by using a public cloud.

Li et al. in Li et al. (2011) propose a trace-and-replay tool, CloudProphet. It predicts the performance of the application without needing to deploy it in a target cloud. The proposed method includes two phases: 1) Tracing: in this phase, the workload information is recorded from the local run of the application, 2) Replaying: an agent emulates the traced workload of the application. The real performance of the application is predicted in the target cloud based on the performance of the agent. If the workload replayed in cloud is different from the traced workload, CloudProphet stops replaying the workload and starts a new run of the application. However, it can create overhead if the application has many synchronization events.

Advantages and disadvantages: The workload generator is useful to evaluate different configurations without overhead of reproducing the real workloads. It can produce the workloads with different intensities of the fluctuations and the burstiness. Thus, the prediction methods can be evaluated under different workloads. This improves the adaptability and the accuracy of the prediction methods. The hybrid clouds can be used to manage the dynamic workloads. The workload factoring is appropriate for the hybrid models of cloud computing.

4.1.10. Curve index model and optimization problems

Liang et al. in Liang et al. (2014) propose a prediction method based on the Modified Index Curve Model to predict the number of the application requests. After the prediction, VMs are reconfigured in a way that the number of hosts and VMs is minimized and the resources utilization rate is also maximized. The resources reconfiguration problem is modeled as an optimization problem.

The goal of the algorithm proposed in Sheng et al. (2014) is to minimize the execution time of tasks. It is solved as a convex optimization problem. When the future workload and the host load are determined, the optimal resources allocation is found. In the next step, the lower and the upper bounds of the workload prediction ratio are determined. Based on these bounds, the upper bound of the time execution is derived. It is clear that an accurate prediction method is necessary for this algorithm.

In Alasaad et al. (2015), a prediction based resource allocation algorithm is proposed. It should guarantee the reservation of enough resources and the minimization of the reservation cost. This method assumes that there is an accurate prediction method to predict the demand for the streaming capacity. Based on the prediction results, a hybrid resource provisioning scheme composed of the resources reservation and the resources allocation is suggested.

Advantages and disadvantages: The curve index model can provide the multi-step ahead prediction. It is appropriate for applications whose requests are updating constantly. Although the model is simple, predicting the precise number of requests based on the curve index model is unexpected due to the dynamic nature of cloud. The optimization techniques are one of the most popular methods for the resources allocation. By describing the predicted workload and the available resources as the constraints of the optimization problem, the optimal or the near optimal configurations of resources are found. Thus, the time complexity of the optimization techniques should be reasonable to find the optimal configurations according to the workload dynamics.

4.1.11. Reinforcement learning

Rao et al. in Rao et al. (2009) propose an RL based approach, VCONF, to capture VMs automatically. VCONF uses RL based on the model for the scalability and the adaptability. At first, an appropriate model is trained based on the collected samples. Then, the model predicts the reward of unseen action-state pairs.

In Tesauro et al. (2006), the state and the action of RL are the arrival rate of requests and the number of servers respectively. RL is used to train a non-linear approximator. The non-linear approximator is used as an external policy for the resources management. It avoids suffering from the potentially poor performance during the online learning.

Xu et al. in Xu et al. (2012) propose a resources provisioning approach based on the Unified Reinforcement Learning (URL). It is based on the model. The model extracts the relationship between the current configuration and the observed performance feedback. It predicts the reward of unseen state-action pairs.

Amiri et al. in Amiri et al. (2016) employ RL for the resources provisioning in cloud. Firstly, the future demand of CPU is predicted by using NN. Based on the future demand, the number of PMs is determined by the learned policy. The authors improve the convergence speed of RL using fuzzy approaches.

Advantages and disadvantages: The RL needs no domain knowledge (Tesauro et al., 2006; Xu et al., 2012). It is able to adapt to the behavioural changes of applications using generating new policies. The scalability of RL is poor in the large state space. The initial policies of RL affect the convergence speed to the optimal policy. Thus, the poor initial policy might lead to the poor performance of RL (Huang et al., 2014).

4.2. Methods based on queuing network models

Bennani et al. Bennani and Menasce (2005) present an approach that determines the dynamic switching of applications between servers. They consider two types of the workloads, batch and online transactions. The performance metrics are calculated based on the QN models. The response time of online transactions is computed by using multi class open QN models. The throughput of batch jobs are calculated by using multi class closed QN models. For using the QN models, the arrival rates of each class of applications should be determined.

Doyel et al. in Doyle et al. (2003) suggest an approach that predicts the resources allocation using internal models of the service behaviour. The internal models of the service behaviour are based on the queuing theory. Although these models are inexpensive, they assume the stable average-cost per-request behaviour and predict the average-case performance. Determining the model parameters is also challenging.

Chalheiros et al. in Calheiros et al. (2011b) propose a mechanism for the VM provisioning. In this mechanism, a workload analyzer predicts the arrival ratio of requests based on historical data or known workloads. A performance modeler models the system as a queuing network. It predicts the response time, the reject ratio and the resources utilization. If the estimated parameters are below the QoS metrics, the number of VMs allocated to applications is updated.

In Zhang et al. (2007) the CPU demand of different types of transactions is estimated by using the regression based methods. The estimated values are used to parameterize QN. QN determines the resources requirements of multi-tier applications according to the workload fluctuations.

Advantages and disadvantages: QN can model multi-tier applications with the desirable number of tiers. The flow of requests in each tier are modeled as a queuing network. Although the QN needs no training phase, it is very sensitive to the parameterization. The precise estimation of parameters such as the arrival ratio and the service time of requests is expensive.

4.3. Methods based on the control theory

Li et al. in Liu et al. (2005) present a feed back control approach that allocates the least resources to applications in a way that applications meet the performance goals. The CPU share of each application is mapped to its response time by using the first-order

AR model. For the fixed workload, a Proportional Integral (PI) controller is used due to its simplicity. However, finding a linear model that describes the system behaviour according to the workload dynamics is difficult in real time systems. To adapt to the changes, an adaptive controller is proposed. It adjusts parameters of the first-order AR model according to the workload variations.

Q-Cloud proposed in Nathuji et al. (2010) handles the interference among VMs hosted on a server using the dynamic adjustment of resources allocated to applications. There is a closed loop controller on each server. It maps the resources usage of all VMs to their performance level using an MIMO model. The MIMO model estimates optimal control inputs (resources allocated to VMs) for the desired performance.

Padala et al. in Padala et al. (2008) present a resources control system, AutoControl, that is adapted to system dynamics for SLA satisfaction. AutoControl is composed of two layers. The first layer includes an application controller (one controller for each application) that determines the required resources of the application in the next control interval. The second layer includes controllers of nodes (one controller for each node). The output of application controllers is send to the node controllers. Each node controller considers the required resources of the hosted VMs and allocates the resources among them.

Wu et al. in Wu et al. (2013) present a feed back control algorithm whose goal is to maximize the profit rate. In this algorithm, the cost and the benefit are calculated for different combinations of reconfiguration actions and VMs. A combination that has the most profit and the least cost is selected. For this purpose, the cost and the benefit of each combination should be predicted. The response time of multi tier applications with particular workloads is also predicted by using the queuing model. Due to the cloud complexity, modeling cloud using simple and linear equations is challenging.

Advantages and disadvantages: The control theory can be used for resources allocation to multi-tier applications. For designing a feedback controller, the relationship between the resources utilization and the performance measures should be determined. Therefore, the processing model of applications should be determined correctly (Huang et al., 2014). The approaches based on the control theory can handle the unexpected fluctuations of workloads efficiently. They can also tune their parameters to model the behavioural changes of applications (Patikirikorala and Colman, 2010; Desmarais, 2006).

The disadvantages of reviewed models and techniques cause inefficiency in the resources provisioning. In the next section, the main advantages, challenges and disadvantages of the prediction groups are discussed. The directions are also proposed to improve the prediction methods.

5. The challenges and directions for application prediction

As it has been mentioned in Section 3, the methods of the application prediction can be classified into three groups machine learning based, control theory based and QN model based. In Section 5.1, advantages and disadvantages of each group are discussed. We identify the challenges and open issues involved in existing prediction methods in Section 5.2. Section 5.3 considers some new trends of the application prediction and Section 5.4 proposes some suggestions for the future research.

5.1. The advantages and disadvantages of different approaches

The control systems are used to control resources shared between cloud applications. The advantages and disadvantages of the control theory based methods are summarized as follows (Huang et al., 2014; Patikirikorala and Colman, 2010; Desmarais, 2006):

(A) **Advantages**:
- It is possible to capture the relationship between the workload and the performance metrics.
- It is possible to ensure that performance of applications is above a minimum threshold. Thus, SLA violation is avoided.
- It is effective to handle uncertainty and behavioural changes of applications by using the feedback control.
- It is possible to include the domain knowledge to model the application behaviour.

(B) **Disadvantages**:
- Some controllers assume a restrictive constraint: the linear controllers assume that the application behaviour is linear (Zhu et al., 2009). Thus, there is a potential of instability.
- Although the non-linear controllers model the application behaviour accurately, their mathematical computations are complex.
- Some controllers such as fuzzy controllers (Cao et al., 2012; Lama and Zhou, 2013) are based on the rule based approaches. The rules extraction is not easy for the resource management. The ability of the controllers depends on the defined rules. Furthermore, the rule based approaches do not have the learning capability.

The QN models are usually used to predict the application performance. For the QN models, the items below can be stated (Ghezzi and Tamburrelli, 2009; Urgaonkar et al., 2007):

(A) **Advantages**:
- QN provides an abstract representation of multi-tier applications.
- These models generate accurate predictions.
- They need no training phase.

(B) **Disadvantages**:
- They require the domain knowledge. Determining the model parameters is difficult. Assuming the specified probability distributions for some parameters is not reasonable due to the dynamic nature of cloud.
- To adapt to the behavioural changes of the application, a new model should be constructed.
- Solving the non-linear queuing equations is difficult at runtime.

The advantages and the disadvantages of machine learning techniques are listed in Table 6. In general, machine learning methods need the history of the application behaviour to train. Therefore, the monitoring and the recording of the application behaviour are necessary. The prediction accuracy of these methods is based on the behavioural similarities of the application in the training and the test phases. If the application behaviour in the test phase is not correlated with one in the training phase, the predicted results are not reliable and the training phase should be repeated so that the model can be adapted to the workload dynamics. The different types of the retraining are as follows (Leitner et al., 2009):

- Periodic training retrains the system in the regular intervals.
- Instance-base training retrains the system when a specific amount of new samples is monitored.
- On-demand training retrains the system based on the user's explicit request.
- On-error training retrains the system when the prediction error is more than a threshold value.
- Custom-retraining retrains the system based on the conditions defined by the user.

The on-error training is one of the most popular methods that is used to retrain the prediction models.

Table 6
Strengths and weaknesses/challenges of machine learning and statistical techniques employed for the application prediction.

Machine learning and statistical techniques	Strengths	Weaknesses/Challenges
LR, AR, ARIMA, ARMA	Simplicity Interpretability	Continues retraining to adapt to workload changes Assumption of being Independent data Assumption of the linearity of application behaviour Selecting the order of the model
Bayesian Theory	Simplicity Simple interpretation	Independence of features describing application behaviour Inability to adapt to workload changes
NN	Simple implementation Modeling nonlinear behaviour of application	Picking the correct topology The long time and a lot of data for training Non-interpretable results for resources manager Inability to adapt to workload changes Not being effective for extrapolation
SVM	Modeling the nonlinear behaviour of application Multi-step ahead prediction	Inability to incorporate knowledge of resource manager Non-interpretable results for resources manager High algorithmic complexity Inability to adapt to workload changes Speed of training and testing
Markov Model	Simple Providing a general overview of application behaviour	Invalid assumptions for many application workloads Much train data to provide reliable result Selecting the order Inability to adapt to workload changes
Clustering	Reduction of cost and time of prediction Capturing correlation among VMs or resources Efficient algorithmic complexity	Fixed number of clusters The number of clusters
String Matching	Linear time for matching	Selecting the length of pattern Requiring a way to discrete time series of workload Loss of efficiency by increasing the number of alphabet symbols
Evolutionary Algorithm	Finding the appropriate structure of prediction methods	Time consuming for convergence Local optimum
KCCA	Extracting correlation among resources and performance metrics	Sensitivity to outliers Selecting the Kernel
Fuzzy logic	Modeling uncertainties and ambiguities	Determining fuzzy rules and membership functions
Hurst Exponent	Simple Exhibiting predictability of time series of workload	Not computable for all applications
RL	Adaptability to the behavioural changes of workload No need for domain knowledge	Poor scalability in the large state space Initial policies
Random Forest	Simple Not expecting linear features Handling high dimensional spaces Very fast to train	Slow to provide real-time predictions Not being effective for extrapolation Difficult to interpret the prediction result Size of model
KNN	No training phase No assumptions about data Simple	Slow and Computation cost Selecting the value of parameter K Selecting the type of the distance metric

5.2. The challenges and open issues for the application prediction

There are some important open issues and challenges for the application prediction in cloud. In this section, we identify the following challenges and open issues to be tackled by the future research.

Choosing the pattern length: It is predicted more enterprises will migrate to cloud and more workloads will move to cloud-based platforms (Cloud Standards Customer Council, 2013). According to this migration, cloud will be faced with some big challenges such as performance, automation, provisioning and scaling. So resources management should be automated to deal with the dynamism of workloads (Buyya et al., 2012; Flores et al., 2015). For this purpose, the future demand of applications should be predicted accurately in a way that the resources manager is able to reallocate resources before the workload changes occurs. The accurate prediction models should be able to extract all of the behavioural patterns of the application workloads. Most of the machine learning methods such as NN, SMA and Markov Model are based on the fixed pattern length. They cannot extract all useful patterns whose length is less/more than the fixed length. Choosing the length of the pattern (the length of the sliding window) for different regions of workloads is one of the most important challenges in these methods.

The model transparency and adaptability: The existing prediction models such as NN, SVM and regression based methods are the discriminative learning approaches. The discriminative learning approaches do not present and understand the behavioural patterns of workloads explicitly. They learn the model parameters to optimize an utility function (such as an error criterion) (Tu, 2007). The model parameters are of little interest to the resources manager. So some

methods are needed to extract knowledge from the model and improve the model transparency. The discriminative methods also need much training data. The behavioural changes of the application workload might start after training the prediction model. To adapt to the workload changes, the model should be retrained over new data. Gathering the new data and retraining the models might be very time consuming. Due to the cloud nature, models should be selected whose retraining and continuous updates are not time consuming and do not require the heavy computations. So some mechanism is needed to adapt the discriminative methods and QN models to incorporate the new behavioural patterns in predictors immediately (Sumathi, 2006).

The model robustness: The other key issue is that systematic approaches are usually required to develop the prediction models. These approaches should be able to address the choice of the appropriate architecture, the internal parameters and the stopping criteria for the training phase. These approaches should guarantee the model robustness, that is, the predictive ability of models to generalize on unexperienced workloads (Shahin et al., 2009). The robustness of the controller depends on compatibility of the models used to describe the dynamic behaviour of the workload. More previous works focus on the controller design and ignore the error of the system modeling phase (Wang et al., 2005). In practice, the error introduced by the modeling phase might increase and causes the instability of the controller. Therefore, the modeling error is a important issue that should be addressed to guarantee the performance (Huang et al., 2014).

The rule set design: The fuzzy control based approaches derive rules for the resources allocation. They update the rules according to the changes of the system. The rule design is very challenging. Although the simple rules are simply derived and they do not require the complex computing, they could not manage the resources efficiently. On the other hand, finding the complex rules might be difficult and time consuming. They also are more sensitive to the workload dynamism that it might lead to instability of the system (Huang et al., 2014).

Control inputs selection: The relationship between control inputs (such as memory and CPU usage) and the performance metrics might vary under the dynamic behaviour of workloads (Wang et al., 2005). The performance metrics depend on the different control inputs in different regions of the workload. To allocate resources efficiently, the most relevant control inputs to the performance metrics should be selected due to the dynamic behaviour of workloads.

Standard test bed: The different literatures use different workloads to test their proposed prediction methods. There is no standard test bed to evaluate the prediction methods. This can lead to models that just work on a specific type of workloads. Therefore, the prediction methods are not evaluated with a real cloud environment and might not be able to address the dynamism of the workloads of cloud applications (Weingartner et al., 2015; Aceto et al., 2013).

5.3. New trends of the application prediction:

There are a few new trends that impact the current research of the application prediction in cloud. In the following section, the application prediction trends are briefly discussed.

Adjustment of Sampling Intervals: The prerequisite of the accurate prediction is to understand the behaviour of the application. Therefore, the real time monitoring is requisite for the prediction, anomaly detection and SLA satisfaction (Andreolini et al., 2015, 2013). Anderolini et al. in Andreolini et al. (2015) consider the monitoring of big data in cloud. They propose if the system behaviour is stationary, the length of sampling intervals should increase. If there are significant variations between samples, the length of sampling intervals should be reduced in a way that the dynamic behaviour of the application can be captured. In their method, two parameters (cost and quality) adjust the length of sampling intervals dynamically. Therefore, meanwhile the resources consumption is reduced, the application behaviour is tracked efficiently. Although tuning the sampling intervals decreases the computational cost, it should guarantee the capturing of the workload changes reliably. Thus, without the accuracy loss of the predicted results, the overhead of prediction methods is reduced.

The hybrid cloud computing model: In general, the workload fluctuations can classified into two groups: 1) The predictable workloads occur according to the access patterns in the specific time intervals, 2) The unexpected workloads occur according to the sudden rise in the application's popularity (Urgaonkar et al., 2008; Calheiros et al., 2011a). In Zhang et al. (2014), a hybrid cloud computing model is proposed. A workload factoring service splits the workload into two parts, the base load and the flash crowd load, based on its dynamism. The base load is managed in the small data center, while the flash crowd load is provisioned on demand through cloud services. Due to the reduction of workloads dynamics in the base load zone, the workload prediction and the resources management are simpler and more efficient in this zone. The flash crowd load platform takes the advantage of the elastic nature of the cloud infrastructure. The effectiveness of the hybrid cloud model depends on the strength of the workload factoring service to decompose the incoming workload. Researchers could focus on developing more innovative algorithms to decompose the incoming workload.

Hybrid prediction approaches: Many of the literatures reviewed in Section 4 predict the future workload using one prediction model. According to the evaluation results reported in Vazquez et al. (2015); Matsunaga and Fortes (2010), a single method cannot provide the most accurate results for different types of workloads. The direction of new research efforts has been on the hybrid prediction methods that merge the prediction strength of individual prediction models (Liu et al., 2015; Jiang et al., 2013; Chen et al., 2015). In Jiang et al. (2013), several prediction models are used to predict the future workload. The results predicted by different methods are merged by a weighted linear combination strategy. The initial weights of predictors are equal. According to the prediction error of each method, the weights are updated. In Chen et al. (2015), some simple predictors, that are based on regression models, estimate the future demand of resources. An FNN receives the results of base predictors and predicts the final results. In Liu et al. (2015), according to the workload dynamics, the service workloads are classified into fast time-scale data or slow time-scale data. LR and SVR are used to predict the slow time-scale and the fast time-scale workloads respectively. Cetinski et al. in Cetinski and Juric (2015) present an Advanced Model for Efficient Workload Prediction in the Cloud (AME-WPC) which merges statistical and machine learning methods. Firstly, features are extracted and scored by using a Two-phase Pattern Matching (TPM) method. The additional features are also extracted and used to extend historical data. Finally, the Random Forest method (Breiman, 2001) predicts the future workload and confidence factors are applied to the workload predictions. Although the hybrid approaches could provide more accurate results, the time delay, computation complexity and choosing the intelligent strategies to combine the results of individual methods should be considered in the future research.

Code originated models of performance: Lee et al. in Lee et al. (2015) believe that the ideal performance models are built directly from the source code of the applications. Thus, the models are consistent and independent of the host platforms. They propose a prototype system, COMPASS (Code Originated Models of Performance via Automatic Source Scanning), to generate the parameterized performance model of the application. At first, COMPASS generates the parameterized Aspen performance model from the application source code by using a static analysis framework. Aspen (Abstract Scalable Performance Engineering Notation) is a modeling framework that defines a formal grammar to model the application behaviour and the machine (Spafford and Vetter, 2012). In the next step, an analysis suite predicts the performance characteristics of the application using the Aspen model. This performance model could predict the resources

usage and the runtime of the applications under different architectures and configurations.

5.4. Suggestions to develop new application prediction approaches

New prediction approaches should be developed in the direction of improving the existing methods. We propose the new directions for researchers to develop the new approaches on the application prediction in cloud as follows:

- The new approaches should be able to extract all the behavioural patterns of workloads independent of the fixed pattern length. Thus, the new approaches could improve the prediction accuracy.
- The new approaches should be able to be adapted according to the workload variations. For this purpose, they should have the capabilities of online learning and decreasing the prediction error with time. Thus, they could incorporate the behavioural changes of workloads in predictors.
- Unlike existing methods, the new approaches could focus on unearthing the interesting trends or patterns of the workload variations explicitly. Thus, the behavioural patterns of workloads are more readily interpretable by the resources manager.
- The prediction approaches should not need to make many assumptions about the workload behaviour. So they would be more general and could be used for the different types of workloads.
- Researchers could focus on developing the new approaches whose parameters are independent of the model structure. Indeed, the parameters could depend on the characteristics of cloud data centers. So they could be estimated from domain knowledge easily.
- Researchers could develop the new prediction approaches based on both of the reactive and the proactive methods. The proactive prediction methods should be able to extract all access patterns correctly. Furthermore, the reactive provisioning methods are essential to correct the error of the prediction methods. The reactive provisioning methods react to the surge of fluctuations or the deviation from the expected behaviour. They allocate the additional resources according to the workload increase to prevent SLA violation (Urgaonkar et al., 2008).
- Different types of resources which include physical resources such as compute, memory, storage, servers, processors and networking are allocated to cloud applications (Singh and Chana, 2016a). Most of the existing methods focus on one or two resources and ignore the correlation between resources. Researchers could investigate the correlation between resources and provide more understandable results for the resources manager.

As it is mentioned in Section 5.2, the standard test beds are required to evaluate the prediction approaches. The standard test beds should include the different types of workloads. Researchers should create the standard test beds using scenarios mixing the different types of workloads. The test beds should be able to simulate the dynamic nature of cloud properly (Weingartner et al., 2015).

6. Conclusion

This paper presents a comprehensive survey on the application prediction for effective resources allocation in cloud. The main reasons for the prediction, the main characteristics, challenges and a general taxonomy of prediction methods have been presented. The newest and the most prominent methods have been reviewed briefly. The challenges, advantages and disadvantages of methods have been investigated. Finally, according to dynamic nature of cloud and deficiencies of methods presented until now, proposals and solutions have been suggested to improve the accuracy and the efficiency of the prediction methods.

References

Aceto, G., Botta, A., De Donato, W., Pescape, A., 2013. Cloud monitoring: a survey. Comput. Netw. 57, 2093–2115.

Adhikari, R., Agrawal, R., 2013. An Introductory Study on Time Series Modeling and Forecasting. LAP LAMBERT Academic Publishing, Saarbrucken, Germany.

Akindele, A.B., Samuel, A.A., 2013. Predicting cloud resource provisioning using machine learning techniques. In: 2013 Proceedings of the 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1–4, Vancouver, Canada.

Akioka, S., Muraoka, Y., 2004. Extended forecast of CPU and network load on computational grid. In IEEE International Symposium on Cluster Computing and the Grid, 2004. CCGrid 2004, pp. 765–772, Chicago, Illinois, USA.

Alasaad, A., Shafiee, K., Behairy, H.M., Leung, V.C.M., 2015. Innovative schemes for resource allocation in the cloud for media streaming applications. IEEE Trans. Parallel Distrib. Syst. 26 (4), 1021–1033. http://dx.doi.org/10.1109/TPDS.2014.2316827, (ISSN 1045-9219).

Altevogt, P., Denzel, W., Kiss, T., 2016. Cloud Modeling and Simulation. In: Murugesan, S., Bojanova, I. (Eds.), Encyclopedia of Cloud Computing. John Wiley & Sons, Ltd, Chichester, UK.

Amiri, M., Feizi-Derakhshi, M.R., Mohammad-Khanli, L., 2016. IDS fitted Q improvement using fuzzy approach for resource provisioning in cloud. J. Intell. Fuzzy Syst. Prepr., (Prepr.), 1–12. http://dx.doi.org/10.3233/JIFS-151445, (URL ⟨http://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs151445⟩).

Andreolini, M., Colajanni, M., Pietri, M., Tosi, S., 2015. Adaptive, scalable and reliable monitoring of big data on clouds. J. Parallel Distrib. Comput. 7980, 67–79. http://dx.doi.org/10.1016/j.jpdc.2014.08.007, (ISSN 0743-7315. URL ⟨http://www.sciencedirect.com/science/article/pii/S074373151400149X⟩).

Andreolini, M., Colajanni, M., Pietri, M., Tosi, S., 2013. Real-time adaptive algorithm for resource monitoring. In: Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013), pp. 67–74, Zurich, Switzerland. http://dx.doi.org/10.1109/CNSM.2013.6727811

Antonescu, A.F., Robinson, P., Braun, T., 2013. Dynamic SLA management with forecasting using multi-objective optimization. In 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 457–463, Ghent, Belgium.

Bachoc, F., 2014. Kriging Models with Gaussian Processes - Covariance Function Estimation and Impact of Spatial Sampling. URL ⟨http://www.math.univ-toulouse.fr/fbachoc/Bachoc_Forge_les_eaux.pdf⟩.

Bennani, M.N., Menasce, D.A., 2005. Resource allocation for autonomic data centers using analytic performance models. In: Proceedings of the Second International Conference on Automatic Computing, ICAC '05, pp. 229–240, Washington, DC, USA. IEEE Computer Society. http://dx.doi.org/10.1109/icac.2005.50

Bey, K.B., Benhammadi, F., Mokhtari, A. Guessoum Z., 2009. CPU load prediction model for distributed computing. In: 2009 Proceedings of the Eighth International Symposium on Parallel and Distributed Computing, pp. 39–45, Lisbon, Portugal, 2009. http://dx.doi.org/10.1109/ISPDC.2009.8

Blunsom, P., 2004. Hidden Markov Models. University of Melbourne, 2004. URL ⟨http://digital.cs.usu.edu/cyan/CS7960/hmm-tutorial.pdf⟩.

Bobroff, N., Kochut, A., Beaty, K., 2007. Dynamic placement of virtual machines for managing SLA violations. In: 2007 Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management, pp. 119–128, Munich, Germany.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32. http://dx.doi.org/10.1023/A:1010933404324.

Buyya, R., Calheiros, R.N., Li, X., 2012. Autonomic cloud computing: Open challenges and architectural elements. In: 2012 Proceedings of the Third International Conference on Emerging Applications of Information Technology, pp. 3–10, Kolkata, India, IEEE.

Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F.D., Buyya, R., 2011a. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw.: Pract. Exp. 41 (1), 23–50. http://dx.doi.org/10.1002/spe.995, (ISSN 0038-0644).

Calheiros, R.N., Ranjan, R., Buyya, R., 2011b. Virtual machine provisioning based on analytical performance and QoS in cloud computing environments. In 2011 International Conference on Parallel Processing, pp. 295–304, Taipei, Taiwan, ISBN 0190–3918. http://dx.doi.org/10.1109/ICPP.2011.17

Cao, J., Zhang, W., Tan, W., 2012. Dynamic control of data streaming and processing in a virtualized environment. IEEE Trans. Autom. Sci. Eng. 9 (2), 365–376.

Caron, E., Desprez, F., Muresan, A., 2010. Forecasting for grid and cloud computing on-demand resources based on pattern matching. In: IEEE Proceedings of the Second International Conference on Cloud Computing Technology and Science (CloudCom), pp. 456–463, Indianapolis, Indiana, USA. http://dx.doi.org/10.1109/CloudCom.2010.65

Cetinski, K., Juric, M.B., 2015. AME-WPC: advanced model for efficient workload prediction in the cloud. J. Netw. Comput. Appl. 55, 191–201. http://dx.doi.org/10.1016/j.jnca.2015.06.001.

Chaisiri, S., Lee, B.S., Niyato, D., 2012. Optimization of resource provisioning cost in cloud computing. IEEE Trans. Serv. Comput. 5 (2), 164–177. http://dx.doi.org/10.1109/TSC.2011.7, (ISSN 1939-1374).

Chang, Y.C., Chang, R.S., Chuang, F.W., 2014. A Predictive Method for Workload Forecasting in the Cloud Environment, Springer Netherlands, pp. 577–585. http://dx.doi.org/10.1007/978-94-007-7262-5_65

Chen, Z., Zhu, Y., Di, Y., Feng, S., 2015. Self-adaptive prediction of cloud resource demands using ensemble model and subtractive-fuzzy clustering based fuzzy neural

network. Computational Intelligence and Neuroscience, 2015, 2015. ISSN 1687–5265. URL http://dx.doi.org/10.1155/2015/919805

Cloud Standards Customer Council. Migrating applications to public cloud services: Roadmap for success. Technical report, 2013. URL ⟨http://www.cloud-council.org/deliverables/CSCC-Migrating-Applications-to-Public-Cloud-Services-Roadmap-for-Success.pdf⟩.

Coutinho, E.F., de Carvalho Sousa, F.R., Rego, P.A.L., Gomes, D.G., de Souza, J.N., 2015. Elasticity in cloud computing: a survey. Ann. Telecommun. - Ann. Des. Telecommun. 70 (7), 289–309. http://dx.doi.org/10.1007/s12243-014-0450-7, (ISSN 1958-9395. URL http://dx.doi.org/10.1007/s12243-014-0450-7).

Cuomo, A., Rak, M., Villano, U., 2015. Performance prediction of cloud applications through benchmarking and simulation. Int. J. Comput. Sci. Eng. 11 (1), 46–55. http://dx.doi.org/10.1504/IJCSE.2015.071362.

da Silva, R.F., Juve, G., Deelman, E., Glatard, T., Desprez, F., Thain, D., Tovar, B., Livny, M., 2013. Toward fine-grained online task characteristics estimation in scientific workflows. In: Proceedings of the 8th Workshop on Workflows in Support of Large-Scale Science, WORKS '13, pp. 58–67, Denver, CO, USA. ACM. ISBN 978-1-4503–2502-8. URL http://dx.doi.org/10.1145/2534248.2534254

Desmarais, R.J., 2006. Adaptive Solutions to Resource Provisioning and Task Allocation Problems for Cloud Computing. (Ph.D. thesis), Department of Computer Science, University of Victoria, Victoria, BC, Canada.

Di, S., Kondo, D., Cirne, W., 2014. Google hostload prediction based on bayesian model with optimized feature combination. J. Parallel Distrib. Comput. 74 (1), 1820–1832. http://dx.doi.org/10.1016/j.jpdc.2013.10.001, (ISSN 0743-7315).

Dinda, P.A., 2002. Online prediction of the running time of tasks. Clust. Comput. 5 (3), 225–236. http://dx.doi.org/10.1023/A:1015634802585, (ISSN 1573-7543).

Dinda, P.A., 2000. Resource Signal Prediction and Its Application to Real-time Scheduling Advisors. (Ph.D. thesis), School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.

Dingyu, Y., Jian, C., Cheng, Y., Jing, X., 2012. A multi-step-ahead CPU load prediction approach in distributed system. In: 2012 Proceedings of the Second International Conference on Cloud and Green Computing, pp. 206–213, Xiangtan, Hunan, China. http://dx.doi.org/10.1109/CGC.2012.32

Doyle, R.P., Chase, J.S., Asad, O.M., Jin, W., Vahdat, A.M., 2003. Model-based resource provisioning in a web service utility. In: Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems - Volume 4, USITS'03, pp. 5–5, Seattle, Washington, USA. USENIX Association.

Duan, R., Nadeem, F., Wang, J., Zhang, Y., Prodan, R., Fahringer, T., 2009. In: 2009 Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID '09, Shanghai, China, pp. 339–347. IEEE Computer Society. ISBN 978-0-7695–3622–4. URL http://dx.doi.org/10.1109/CCGRID.2009.58

Duy, T.V.T., Sato, Y., Inoguchi, Y., 2011. Improving accuracy of host load predictions on computational grids by artificial neural networks. Int. J. Distrib. Parallel Syst. 26 (4), 275–290. http://dx.doi.org/10.1080/17445760.2010.481786, (ISSN 1744-5760).

Fang, W., Lu, Z., Wu, J., Cao, Z., 2012. RPPS: A novel resource prediction and provisioning scheme in cloud data center. In: 2012 IEEE Proceedings of the Ninth International Conference on Services Computing, Honolulu, HI, USA, pp. 609–616. http://dx.doi.org/10.1109/SCC.2012.47

Faragher, R., 2012. Understanding the basis of the Kalman filter via a simple and intuitive derivation. IEEE Signal Process. Mag. 29 (5), 128–132.

Farlow, S.J., 1981. The GMDH algorithm of Ivakhnenko. Am. Stat. 35 (4), 210–215.

Flores, A.A., Mendes, R.d.S., Brscher, G.B., Westphall, C.B., Villareal, M.E., 2015. Decision-theoretic model to support autonomic cloud computing. In: Proceedings of the Fourteenth International Conference on Networks (ICN 2015), Spain. IARIA, pp. 218–223.

G. Sun Z. Lu J. Wu X. Wang P. Hung. A Novel Reactive-predictive Hybrid Resource Provision Method in Cloud Datacenter. Springer International Publishing, 2015, 33–47. (ISBN 978-3-319-26979-5, URL ⟨http://dx.doi.org/10.1007/978-3-319-26979-5_3⟩)

Galante, G., Bona, L.C.E.d., 2012. A survey on cloud computing elasticity. In: 2012 IEEE Proceedings of the Fifth International Conference on Utility and Cloud Computing, Chicago, IL, USA, pp. 263–270. http://dx.doi.org/10.1109/UCC.2012.30

Ganapathi, A., Chen, Y., Fox, A., Katz, R., Patterson, D., 2009. Statistics-driven workload modeling for the cloud. In: 2010 IEEE Proceedings of the 26th International Conference on Data Engineering Workshops (ICDEW 2010), Long Beach, California, USA, pp. 87–92.

Garg, S.K., Toosi, A.N., Gopalaiyengar, S.K., Buyya, R., 2014. SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter. J. Netw. Comput. Appl. 45, 108–120. http://dx.doi.org/10.1016/j.jnca.2014.07.030, (ISSN 1084-8045, URL ⟨http://www.sciencedirect.com/science/article/pii/S1084804514001787⟩.).

Ghezzi, C., Tamburrelli, G., 2009. Predicting performance properties for open systems with KAMI. In: Proceedings of the 5th International Conference on the Quality of Software Architectures, QoSA 2009, QoSA '09, pp. 70–85, 1574269, 2009. Springer-Verlag. http://dx.doi.org/10.1007/978-3-642-02351-4_5

Govindan, S., Choi, J., Urgaonkar, B., Sivasubramaniam, A., Baldini, A., 2009. Statistical profiling-based techniques for effective power provisioning in data centers. In: Proceedings of the 4th ACM European conference on Computer systems, EuroSys '09, Nuremberg, Germany. ACM, pp. 317–330. http://dx.doi.org/10.1145/1519065.1519099

Gupta, C., Mehta, A., Dayal, U., 2008. PQR: Predicting query execution times for autonomous workload management. In: Proceedings of the 2008 International Conference on Autonomic Computing, Chicago, IL, USA, pp. 13–22. IEEE Computer Society. ISBN 978-0-7695–3175–5. URL http://dx.doi.org/10.1109/ICAC.2008.12

Gursun, G., Crovella, M., Matta, I., 2011. Describing and forecasting video access patterns. In: Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM 2011), Shanghai, China, pp. 16–20.

Hardoon, D.R., Szedmak, S., Shawe Taylor, J., 2004. Canonical correlation analysis: an overview with application to learning methods. Neural Comput. 16 (12), 2639–2664.

Herbst, N.R., Kounev, S., Reussner, R., 2013. Elasticity in cloud computing: What it is, and what it is not. In: Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013), San Jose, CA, USA.

Hu, X., 1995. Knowledge Discovery in Databases: An Attribute-oriented Rough Set Approach. (Ph.D. thesis), Department of Computer Science, University of Regina, Regina, Sask., Canada. UMI Order No. GAXNN-08457.

Hu, Y., Deng, B., Peng, F., Wang, D., 2016. Workload prediction for cloud computing elasticity mechanism. In 2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 244–249, Chengdu, China. http://dx.doi.org/10.1109/ICCCBDA.2016.7529565

Huang, D., He, B., Miao, C., 2014. A survey of resource management in multi-tier web applications. IEEE Commun. Surv. Tutor. 16 (3), 1574–1590. http://dx.doi.org/10.1109/SURV.2014.010814.00060, (ISSN 1553-877X).

Huebscher, M.C., McCann, J.A., 2008. A survey of autonomic computing–degrees, models, and applications. ACM Comput. Surv. 40 (3). http://dx.doi.org/10.1145/1380584.1380585, (ISSN 0360-0300, URL http://dx.doi.org/10.1145/1380584.1380585).

Hwang, K., Bai, X., Shi, M., Li, Y., Chen, W.G., Wu, Y., 2016. Cloud performance modeling and benchmark evaluation of elastic scaling strategies. IEEE Trans. Parallel Distrib. Syst. 27 (1), 130–143. http://dx.doi.org/10.1109/TPDS.2015.2398438.

Hyndman, R., Koehler, A., Ord, K., Snyder, R., 2008. Forecasting with Exponential Smoothing: The State Space Approach. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-540-71918-2

Islam, S., Keung, J., Lee, K., Liu, A., 2012. Empirical prediction models for adaptive resource provisioning in the cloud. Future Gener. Comput. Syst. 28 (1), 155–162. http://dx.doi.org/10.1016/j.future.2011.05.027, (ISSN 0167-739X).

Jain, R., 2010. Queueing network. URL ⟨http://www.cse.wustl.edu/jain/iucee/ftp/k_32qn.pdf⟩.

Jang, J.S.R., 1993. ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst., Man, Cybern. 23 (3), 665–685. http://dx.doi.org/10.1109/21.256541, (ISSN 0018-9472).

Jheng, J.-J., Tseng, F.-H., Chao, H.-C., Chou, L.-D., 2014. A novel VM workload prediction using Grey Forecasting model in cloud data center. In: 2014 International Conference on Information Networking, pp. 40–45, Phuket, Thailand. http://dx.doi.org/10.1109/icoin.2014.6799662

Jiang, Y., Perng, C.-S., Li, T., Chang, R.N., 2013. Cloud analytics for capacity planning and instant VM provisioning. IEEE Trans. Netw. Serv. Manag. 10 (3), 312–325.

Kalekar, P.S., 2004. Time series forecasting using Holt-Winters exponential smoothing. URL ⟨http://www.jal.xjegi.com/fileup/PDF/75.pdf⟩.

Khan, A., Yan, X., Tao, S., Anerousis, N., 2012. Workload characterization and prediction in the cloud: A multiple time series approach. In 2012 IEEE Network Operations and Management Symposium, pp. 1287 – 1294, Maui, HI, USA.

Kornbrot, D., 2005. Pearson product moment correlation. In: Everitt, B., Howell, D. (Eds.), Encyclopedia of statistics in behavioral science. Wiley, Hoboken.

Kousiouris, G., Menychtas, A., Kyriazis, D., Gogouvitis, S., Varvarigou, T., 2014. Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in cloud platforms. Future Gener. Comput. Syst. 32, 27–40. http://dx.doi.org/10.1016/j.future.2012.05.009, (ISSN 0167-739X. URL ⟨http://www.sciencedirect.com/science/article/pii/S0167739×12001057⟩.).

Kulkarni, S., Agrawal, P., 2014. Analysis of TCP Performance in Data Center Networks. Springer, New York. http://dx.doi.org/10.1007/978-1-4614-7861-4, (ISBN 978-1-4614-7860-7).

Kumar, S., Buyya, R., 2012. Green Cloud Computing and Environmental Sustainability, pp. 315–339. John Wiley & Sons, Ltd. ISBN 9781118305393. URL http://dx.doi.org/10.1002/9781118305393.ch16

Kundu, S., Rangaswami, R., Gulati, A., Zhao, M., Dutta, K., 2012. Modeling virtualized applications using machine learning techniques. In: Proceedings of the 8th ACM SIGPLAN/SIGOPS conference on Virtual Execution Environments, VEE '12, pp. 3–14, 2151028. ACM. http://dx.doi.org/10.1145/2151024.2151028

Kupferman, J., Silverman, J., Jara, P., Browne, J., 2009. Scaling into the cloud. CS270 - ADVANCED OPERATING SYSTEMS.

Labonte, F., Mattson, P., Thies, W., Buck, I., Kozyrakis, C., Horowitz, M., 2004. The stream virtual machine. In: Proceedings of the 13th International Conference on Parallel Architecture and Compilation Techniques, PACT '04, Antibes Juan-les-Pins, France, pp. 267–277. ISBN 1089-795X. http://dx.doi.org/10.1109/PACT.2004.1342560

Lagar-Cavilla, H.A., Whitney, J.A., Scannell, A.M., Patchin, P., Rumble, S.M., de Lara, E., Brudno, M., Satyanarayanan, M., 2009. Snowflock: rapid virtual machine cloning for cloud computing. In: Proceedings of the 4th ACM European conference on Computer systems, EuroSys '09, Nuremberg, Germany. ACM, pp. 1–12. http://dx.doi.org/10.1145/1519065.1519067

Lama, P., Zhou, X., 2013. Autonomic provisioning with self-adaptive neural fuzzy control for end-to-end delay guarantee. In: Proceedings of IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), San Francisco, California, US.

Lazowska, E.D., Zahorjan, J., Graham, G.S., Sevcik, K.C., 1984. Quantitative System Performance: Computer System Analysis Using Queueing Network Models. Prentice-Hall, Inc., ISBN 0-13-746975-6.

Lee, S., Meredith, J.S., Vetter, J.S., 2015. COMPASS: A framework for automated performance modeling and prediction. In: Proceedings of the 29th ACM on International Conference on Supercomputing, ICS '15, Newport Beach/Irvine, CA, USA, 2015. ACM, pp. 405–14. http://dx.doi.org/10.1145/2751205.2751220

Leitner, P., Wetzstein, B., Rosenberg, F., Michlmayr, A., Dustdar, S., Leymann, F., 2009. Runtime prediction of service level agreement violations for composite services. In: Proceedings of the 2009 International Conference on Service-oriented Computing, 1926639. Springer-Verlag, pp. 176–186.

Li, A., Zong, X., Kandula, S., Yang, X., Zhang, M., 2011. CloudProphet: towards application performance prediction in cloud. In: Proceedings of the ACM SIGCOMM 2011 Conference, 2018502. ACM, pp. 426–427. http://dx.doi.org/10.1145/2018436.2018502

Li, H., Groep, D., Templon, J., Wolters, L., 2004. Predicting job start times on clusters. In: Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid. IEEE Computer Society Press, Chicago, Illinois, USA, pp. 301–308. http://dx.doi.org/10.1109/CCGrid.2004.1336581

Li, H., Groep, D., Wolters, L., 2005. An evaluation of learning and heuristic techniques for application run time predictions. In: Proceedings of the 11th Annual Conference of the Advance School for Computing and Imaging (ASCI), Netherlands.

Li, J., Ma, X., Singh, K., Schulz, M., Supinski, B.R.d., McKee, S.A., 2009. Machine learning based online performance prediction for runtime parallelization and task scheduling. In 2009 IEEE International Symposium on Performance Analysis of Systems and Software, Boston, Massachusetts, USA, pp. 89–100.

Liang, Q., Zhang, J., Zhang, Y.H., Liang, J.M., 2014. The placement method of resources and applications based on request prediction in cloud data center. Inf. Sci. 279, 735–745. http://dx.doi.org/10.1016/j.ins.2014.04.026, (ISSN 0020-0255. URL ⟨http://www.sciencedirect.com/science/article/pii/S0020025514004733⟩.).

Liu, C., Shang, Y., Duan, L., Chen, S., Liu, C., Chen, J., 2015. Optimizing workload category for adaptive workload prediction in service clouds. In: Proceedings of the 13th International Conference on Service-Oriented Computing (ICSOC 2015), Goa, India. Springer-Verlag Berlin Heidelberg, pp. 87–104.

Liu, X., Zhu, X., Singhal, S., Arlitt, M., 2005. Adaptive entitlement control of resource containers on shared servers. In: 2005 Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management, 2005. IM 2005, pp. 163–176, Nice, France. http://dx.doi.org/10.1109/INM.2005.1440783

Lu, C.T., Chang, C.W., Chang, J.S., 2015. VM scaling based on Hurst exponent and Markov transition with empirical cloud data. J. Syst. Softw. 99, 199–207, (ISSN 0164-1212).

Manvi, S.S., Krishna Shyam, G., 2014. Resource management for Infrastructure as a Service (IaaS) in cloud computing: a survey. J. Netw. Comput. Appl. 41, 424–440. http://dx.doi.org/10.1016/j.jnca.2013.10.004, (ISSN 1084-8045. URL ⟨http://www.sciencedirect.com/science/article/pii/S1084804513002099⟩.).

Matsunaga, A., Fortes, J.A.B., 2010. On the use of machine learning to predict the time and resources consumed by applications. In: 2010 Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, Melbourne, Victoria, Australia. IEEE Computer Society, pp. 495–504. URL http://dx.doi.org/10.1109/CCGRID.2010.98

McCullough, J.C., Agarwal, Y., Chandrashekar, J., Kuppuswamy, S., Snoeren, A.C., Gupta, R.K., 2011. Evaluating the effectiveness of model-based power characterization. In: Proceedings of the 2011 USENIX Conference on USENIX Annual Technical Conference, Portland, OR, USA. USENIX Association, pp. 12–12.

Mell, P., Grance, T., 2002. The NIST definition of cloud computing. Technical Report GMD Report 159, German National Research Center for Information Technology.

Mell, P., Grance, T., 2011. NIST Special Publication 800–145, 2011. URL ⟨http://www.csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf⟩.

Meng, X., Isci, C., Kephart, J., Zhang, L., Bouillet, E., Pendarakis, D., 2010. Efficient resource provisioning in compute clouds via VM multiplexing. In: Proceedings of the 7th international conference on Autonomic computing, ICAC '10, Washington, DC, USA, ACM, pp. 11–20. http://dx.doi.org/10.1145/1809049.1809052

Mishra, A.K., Hellerstein, J.L., Cirne, W., Das, C.R., 2010. Towards characterizing cloud backend workloads: insights from Google compute clusters. ACM SIGMETRICS Perform. Eval. Rev. 37 (4), 34–41. http://dx.doi.org/10.1145/1773394.1773400, (ISSN 0163-5999).

Miu, T., Missier, P., 2012. Predicting the execution time of workflow activities based on their input features. In: Proceedings of the 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, SCC '12, Salt Lake City, UT, USA. IEEE Computer Society, pp. 64–72. http://dx.doi.org/10.1109/SC.Companion.2012.21

Nathuji, R., Kansal, A., Ghaffarkhah, A., 2010. Q-clouds: managing performance interference effects for qos-aware clouds. In: Proceedings of the 5th European conference on Computer systems, EuroSys '10, Paris, France. ACM, pp. 237–250. http://dx.doi.org/10.1145/1755913.1755938

Niu, D., Liu, Z., Li, B., Zhao, S., 2011. Demand forecast and performance prediction in peer-assisted on-demand streaming systems. In: 2011 Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM), Shanghai, China, pp. 421–425.

Padala, P., Hou, K.Y., Shin, K.G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A., 2008. Automated control of multiple virtualized resources. In: Proceedings of the 4th ACM European conference on Computer systems, EuroSys '09, Nuremberg, Germany. ACM, pp. 13–26. http://dx.doi.org/10.1145/1519065.1519068

Patikirikorala, T., Colman, A., 2010. Feedback controllers in the cloud. In: Proceedings of the 17th Asia Pacific Software Engineering Conference (APSEC 2010), Sydney, Australia.

Petcu, D., Vzquez-Poletti, J.L., 2012. European Research Activities in Cloud Computing. Cambridge Scholars Publishing, United Kingdom, (ISBN 1443835072, 9781443835077).

Pietri, I., Juve, G., Deelman, E., Sakellariou, R., 2014. A performance model to estimate execution time of scientific workflows on the cloud. In: Proceedings of the 9th Workshop on Workflows in Support of Large-Scale Science, WORKS '14, New Orleans, LA, USA. IEEE Press, pp. 11–19. ISBN 978-1-4799–7067-4. URL http://dx.doi.org/10.1109/WORKS.2014.12

Prevost, J.J., Nagothu, K., Kelley, B., Jamshidi, M., 2011. Prediction of cloud data center networks loads using stochastic and neural models. In: 2011 Proceedings of the 6th International Conference on System of Systems Engineering, Albuquerque, New Mexico, USA, pp. 276 – 281.

Qian, B., Rasheed, K., 2007. Hurst exponent and financial market predictability. In: Proceedings of the 2nd IASTED International Conference on Financial Engineering and Applications, Cambridge, MA, USA, pp. 203–209.

Rao, J., Bu, X., Xu, C.Z., Wang, L., Yin, G., 2009. VCONF: A reinforcement learning approach to virtual machines auto-configuration. In: Proceedings of the 6th international conference on Autonomic computing, ICAC '09, Barcelona, Spain. ACM, pp. 137–146. http://dx.doi.org/10.1145/1555228.1555263

Roy, N., Dubey, A., Gokhale, A., 2011. Efficient autoscaling in the cloud using predictive models for workload forecasting. In: 2011 IEEE Proceedings of the 4th International Conference on Cloud Computing, 2055550, IEEE Computer Society, pp. 500–507. http://dx.doi.org/10.1109/cloud.2011.42.

Saripalli, P., Kiran, G.V.R., Shankar, R.R., Narware, H., Bindal, N., 2011. Load prediction and hot spot detection models for autonomic cloud computing. In: 2011 Proceedings of the Fourth IEEE International Conference on Utility and Cloud Computing, Melbourne, Victoria, Australia, IEEE Computer Society, pp. 397–402. http://dx.doi.org/10.1109/ucc.2011.66.

Schafer, R.W., 2011. What is a Savitzky-Golay filter. IEEE Signal Process. Mag. 28 (4), 111–117.

Shahin, M.A., Jaksa, M.B., Maier, H.R., 2009. Recent advances and future challenges for artificial neural systems in geotechnical engineering applications. Advances in Artificial Neural Systems, 2009 http://dx.doi.org/10.1155/2009/308239.

Shen, Z., Subbiah, S., Gu, X., Wilkes, J.. 2011. CloudScale: elastic resource scaling for multi-tenant cloud systems. In: Proceedings of the 2nd ACM Symposium on Cloud Computing, SOCC'11, 2038921, pp. 1–14 http://dx.doi.org/10.1145/2038916.2038921.

Sheng, D., Cho Li, W., Cappello, F., 2014. Adaptive algorithm for minimizing cloud task length with prediction errors. IEEE Trans. Cloud Comput. 2 (2), 194–207. http://dx.doi.org/10.1109/TCC.2013.16, (ISSN 2168-7161).

Shi, P., Wang, H., Yin, G., Lu, F., Wang, T., 2012. Prediction-based federated management of multi-scale resources in cloud. Adv. Inf. Sci. Serv. Sci. 4 (6), 324–334.

Singh, S., Chana, I., 2015a. QoS-aware autonomic resource management in cloud computing: a systematic review. ACM Comput. Surv. 48 (3). http://dx.doi.org/10.1145/2843889.

Singh, S., Chana, I., 2015b. Qrsf: qos-aware resource scheduling framework in cloud computing. J. Supercomput. 71 (1), 241–292. http://dx.doi.org/10.1007/s11227-014-1295-6, (ISSN 0920-8542, URL ⟨http://dx.doi.org/10.1007/s11227-014-1295-6⟩).

Singh, S., Chana, I., 2016a. Cloud resource provisioning: survey, status and future research directions. Knowl. Inf. Syst. 49 (3), 1005–1069. http://dx.doi.org/10.1007/s10115-016-0922-3.

Singh, S., Chana, I., 2016b. A survey on resource scheduling in cloud computing: issues and challenges. J. Grid Comput. 14 (2), 217–264. http://dx.doi.org/10.1007/s10723-015-9359-2.

Singh, S., Chana, I., 2016c. Resource provisioning and scheduling in clouds: QoS perspective. J. Supercomput. 72 (3), 926–960. http://dx.doi.org/10.1007/s11227-016-1626-x.

Smith, W., Foster, I., Taylor, V., 2004. Predicting application run times with historical information. J. Parallel Distrib. Comput. 64 (9), 1007–1016. http://dx.doi.org/10.1016/j.jpdc.2004.06.008, (ISSN 0743-7315).

Smith, L., 2002. A Tutorial on Principal Components Analysis, URL ⟨http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf⟩.

Smith, W., Foster, I.T., Taylor, V.E., 1998. Predicting application run times using historical information. In: Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing, IPPS/SPDP '98, Orlando, Florida, USA, Springer-Verlag, pp. 122–142.

Spafford, K.L., Vetter, J.S., 2012. Aspen: A domain specific language for performance modeling. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC'12. IEEE Computer Society Press.

Sumathi, S.S., S., 2006. Introduction to Data Mining and its Applications. Springer-Verlag Berlin Heidelberg.

Tang, Z., Mo, Y., Li, K., Li, K., 2014. Dynamic forecast scheduling algorithm for virtual machine placement in cloud computing environment. J. Supercomput. 70 (3), 1279–1296. http://dx.doi.org/10.1007/s11227-014-1227-5, (ISSN 0920-8542, URL ⟨http://dx.doi.org/10.1007/s11227-014-1227-5⟩).

Tesauro, G., Jong, N.K., Das, R., Bennani, M.N., 2006. A hybrid reinforcement learning approach to autonomic resource allocation. In 2006 IEEE International Conference on Autonomic Computing (ICAC), Dublin, Ireland, pp. 65–73. http://dx.doi.org/10.1109/ICAC.2006.1662383.

Toffetti, G., Gambi, A., Pezz, M., Pautasso, C., 2010. Engineering autonomic controllers for virtualized web applications. In: Proceedings of the 10th International Conference on Web Engineering (ICWE 2010), Vienna, Austria, Springer-Verlag, pp. 66–80.

Tu, Z., 2007. Learning generative models via discriminative approaches. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, Minnesota, USA, IEEE Computer Society.

Unpingco, J., 2016. Python for Probability, Statistics, and Machine Learning. Springer International Publishing, 1 edition.

Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M., Tantawi, A., 2007. Analytic modeling of multitier internet applications. ACM Trans. Web 1 (1). http://dx.doi.org/10.1145/1232722.1232724, (ISSN 1559-1131).

Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P., Wood, T., 2008. Agile dynamic provisioning of multi-tier internet applications. ACM Trans. Auton. Adapt. Syst. 3 (1), 1–39. http://dx.doi.org/10.1145/1342171.1342172, (ISSN 1556-4665).

Vazquez, C., Krishnan, R., John, E., 2015. Time series forecasting of cloud data center workloads for dynamic resource provisioning. J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Ap plications (JoWUA) 6 (3), 87–110.

Virtamo. Survey on queueing network URL ⟨https://www.netlab.tkk.fi/opetus/s383143/kalvot/E_qnets.pdf⟩.

Wang, Z., Zhu, X., Singhal, S., 2005. Utilization vs. SLO-based control for dynamic sizing of resource partitions. In: Proceedings of the 16th IFIP/IEEE Ambient Networks International Conference on Distributed Systems: Operations and Management, DSOM'05, Barcelona, Spain.

Watanabe, S., 2013. A widely applicable Bayesian information criterion. J. Mach. Learn. Res. 14, 867–897.

Weijia, S., Zhen, X., Qi, C., Haipeng, L., 2013. Adaptive resource provisioning for the cloud using online bin packing. IEEE Trans. Comput. 99. http://dx.doi.org/10.1109/TC.2013.148, (PrePrints)(, ISSN 0018-9340, URL ⟨http://doi.ieeecomputersociety.org/10.1109/TC.2013.148⟩).

Weingartner, R., Brascher, G.B., Westphall, C.B., 2015. Cloud resource management: a survey on forecasting and profiling models. J. Netw. Comput. Appl. 47, 99–106.

Wu, H., Zhang, W., Zhang, J., Wei, J., Huang, T., 2013. A benefit-aware on-demand provisioning approach for multi-tier applications in cloud computing. Front. Comput. Sci. 7 (4), 459–474. http://dx.doi.org/10.1007/s11704-013-2201-8, (ISSN 2095-2228, URL ⟨http://dx.doi.org/10.1007/s11704-013-2201-8⟩).

Wu, Y., Yuan, Y., Yang, G., Zheng, W., 2007. Load prediction using hybrid model for computational grid. In: Proceedings of the 8th IEEE/ACM International Conference on Grid Computing, GRID '07, 1513472, IEEE Computer Society, pp. 235–242. http://dx.doi.org/10.1109/grid.2007.4354138.

Xiong, P., Chi, Y., Zhu, S., Moon, H.J., Pu, C., Hacigumus, H., 2011. Intelligent management of virtualized resources for database systems in cloud environment. In: 2011 IEEE Proceedings of the 27th International Conference on Data Engineering, pp. 87–98, 2005598, IEEE Computer Society http://dx.doi.org/10.1109/icde.2011.5767928.

Xu, C.-Z., Rao, J., Bu, X., 2012. URL: a unified reinforcement learning approach for autonomic cloud management. J. Parallel Distrib. Comput. 72 (2), 95–105. http://dx.doi.org/10.1016/j.jpdc.2011.10.003, (ISSN 0743-7315, URL ⟨http://www.sciencedirect.com/science/article/pii/S0743731511001924⟩..

Xu, D.Y., Yang, S.L., Liu, R.P., 2013. A mixture of HMM, GA, and Elman network for load prediction in cloud-oriented data centers. J. Zhejiang Univ. SCIENCE C. 14 (11), 845–858. http://dx.doi.org/10.1631/jzus.C1300109, (ISSN 1869-1951, URL ⟨http://dx.doi.org/10.1631/jzus.C1300109⟩).

Xu, P., 2009. Differential phase space reconstructed for chaotic time series. Appl. Math. Model. 33 (2), 999–1013. http://dx.doi.org/10.1016/j.apm.2007.12.021, (ISSN 0307-904X, URL ⟨http://www.sciencedirect.com/science/article/pii/S0307904×07003526⟩).

Yang, J., Liu, C., Shang, Y., Chen, B., Mao, Z., Liu, C., Niu, L., Chen, J., 2014a. A cost-aware auto-scaling approach using the workload prediction in service clouds. Inf. Syst. Front. 16 (1), 7–18. http://dx.doi.org/10.1007/s10796-013-9459-0, (ISSN 1387-3326, URL ⟨http://dx.doi.org/10.1007/s10796-013-9459-0⟩).

Yang, Q., Peng, C., Zhao, H., Yu, Y., Zhou, Y., Wang, Z., Du, S., 2014b. A new method based on PSR and EA-GMDH for host load prediction in cloud computing system. J. Supercomput. 68 (3), 1402–1417. http://dx.doi.org/10.1007/s11227-014-1097-x, (ISSN 0920-8542, URL ⟨http://dx.doi.org/10.1007/s11227-014-1097-x⟩).

Yin, J., Lu, X., Chen, H., Zhao, X., Xiong, N.N., 2014. System resource utilization analysis and prediction for cloud based applications under bursty workloads. Inf. Sci. 279. http://dx.doi.org/10.1016/j.ins.2014.03.123.

Zhang, H., Jiang, G., Yoshihira, K., Chen, H., 2014. Proactive workload management in hybrid cloud computing. IEEE Trans. Netw. Serv. Manag. 11 (1), 99–100.

Zhang, Q., Cherkasova, L., Smirni, E., 2007. A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In: Proceedings of the Fourth International Conference on Autonomic Computing, ICAC '07, Jacksonville, Florida, USA, pp. 27–27. http://dx.doi.org/10.1109/ICAC.2007.1.

Zhang, Q., Zhani, M.F., Zhang, S., Zhu, Q., Boutaba, R., Hellerstein, J.L., 2012. Dynamic energy-aware capacity provisioning for cloud computing environments. In: Proceedings of the 9th International Conference on Autonomic computing, ICAC '12, , 2371562, pp. 145–154ACM. http://dx.doi.org/10.1145/2371536.2371562.

Zhenhuan, G., Xiaohui, G., Wilkes, J., 2010. PRESS: PRedictive Elastic Resource scaling for cloud systems. In: Proceedings of the 6th International Conference on Network and Service Management, CNSM 2010, , Niagara Falls, Canada, pp. 9–16 http://dx.doi.org/10.1109/CNSM.2010.5691343.

Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A., Padala, P., Shin, K., 2009. What does control theory bring to systems research? SIGOPS - Oper. Syst. Rev. 43 (1), 62–69. http://dx.doi.org/10.1145/1496909.1496922, (ISSN 0163-5980).

Maryam Amiri received the BS degree in computer engineering from the University of Arak, Arak, Iran, in 2009; the MS degree in computer engineering from the Bu-Ali Sina University, Hamedan, Iran, in 2012; She is currently a Ph.D. Student in the Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran. Her research interests include cloud computing, distributed systems, data mining and prediction.

Leili Mohammad-Khanli received the BS degree in computer engineering from the Shahid Beheshti University, Tehran, Iran, in 1995; the MS in computer engineering from the Iran University of Science and Technology, Tehran, Iran, in 1989; and the Ph.D. degree in computer engineering from the Iran University of Science and Technology, Tehran, Iran, in 2007. She is currently an associate professor in the Department of Computer Engineering, the Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran. Her research interests include grid computing, cloud computing, computer networks, quality assurance services and distributed systems.