IEEE Xplore: http://ieeexplore.ieee.org/document/7575860/

DOI: 10.1109/FiCloud.2016.32

Towards Efficient Resource Management in Cloud Computing: A Survey

Markus Ullrich and Jörg Lässig
University of Applied Sciences Zittau/Görlitz
Department of Computer Science
{*mullrich,jlaessig*}*@hszg.de*

Martin Gaedke
Technische Universität Chemnitz
Department of Computer Science
martin.gaedke@informatik.tu-chemnitz.de

***Abstract*—In the area of cloud computing virtualization has become an indispensable means to enable the efficient utilization of existing compute infrastructure. Selecting the right amount of virtualized resources for an application in such an environment is not an easy task and requires the utilization of three strongly interconnected resource management areas: resource modeling, resource estimation and resource discovery & selection. Most solutions enable an accurate selection of the most appropriate virtual resource package for specific application types already. Support for arbitrary applications, however, is rarely considered which means approaches in this area are usually not applicable for general use cases and, more importantly, difficult to compare with each other. We analyze the most promising existing research in resource management and examine monitored values, supported application classes and the most important criteria for evaluating the effectiveness of the approach. We identify key similarities and differences as well as open research challenges. The discussion about possible solutions includes application classification and the introduction of a general application model to support the selection of the most appropriate resource management approaches for arbitrary applications.**

1. Introduction

Many cloud computing definitions, such as the most popular by the NIST [1], highlight several features such as flexibility, scalability, worldwide and mobile access and the *pay-as-you-go* principle. Hence, the cloud allows for a more efficient utilization of compute resources resulting in cost reduction for the consumer. Naturally, the aim of the consumer is to reduce the overall cost spent on a particular service in the cloud like *Infrastructure as a Service (IaaS)* whilst still being able to fulfill *quality of service (QoS)* requirements.

One of the most important benefits, besides the improved mobile access, is the *increased flexibility* for developing, deploying and managing new services and applications [2], [3]. This is of particular interest for small and medium sized enterprises (SMEs) and researchers who are able to focus on running and testing software or benchmarks without managing their own data center or applying for specialized hardware [4].

A major challenge in this area that we are concerned with is maximizing the efficiency of utilizing virtualized resources in the cloud. Obvious benefits include increased cost savings for cloud service consumers but also reduced energy consumption and thus greenhouse gas emissions. The broad topic we are referring to is resource management for IaaS clouds [5]. In this area, the resource demand of an application that is deployed in a cloud environment is of significant importance. We define the resource demand as the minimum amount of hardware resources (available memory, number of CPU cores, CPU core speed, etc.) that are required for an application to be executed such that QoS requirements can be fulfilled. A broad term for methods that aim to predict the amount of resources an application requires is *resource demand estimation*. Resource modeling, i.e. modeling the resource consumption of an application, is of great importance for predicting the demand and will also be covered in this work as well as resource discovery & selection which is concerned with selecting the most appropriate set of resources for an application based on the predicted demand and/or QoS criteria defined by the user.

Another research area which is of particular interest for this work is *application classification* which is usually concerned with identifying networking applications based on network traffic. Other application classification methods have not been widely studied yet but methods based on *resource consumption* have been successfully utilized before, e.g., for improved scheduling of virtualized resources in a data center [6]. In this work, we are going to identify key classes of cloud applications and highlight the potential benefits of applying current application classification methodologies to resource management for IaaS clouds.

The rest of this paper is organized as follows. In Section 2 we provide a more detailed formulation of the problem including current obstacles and challenges and we identify the most important application classes for this research area. In Section 3 we survey and categorize existing approaches in resource modeling, estimation and discovery & selection for IaaS clouds. In Section 4 we explain how application classification can be used to support resource management and elaborate further on existing challenges and possible solutions in this area. Finally, Section 5 concludes this work and we give final remarks on this topic regarding the future development of solutions.

2. Problem Formulation

Resource management is a broad research field comprised of many highly interconnected areas. Thus, it is difficult to study one specific area alone without regard to the others. Manvi and Shyam provide a good overview of different resource management areas with a brief explanation for each of them [5]. Furthermore, they survey the state-of-the-art in resource provisioning, allocation, mapping and adaption in their work. Other areas mentioned by the authors are resource modeling, resource estimation, resource discovery & selection, resource brokering and resource scheduling. A short explanation about each of those areas is given in their work. In this section, we briefly want to discuss the relations between the most relevant areas for our research.

Resource *provisioning* is concerned with the allocation of a cloud providers resources to a customer. However, not only the size and the number of virtual machines (VMs) required by an application needs to be considered, but also the software installed on them, as well as the amount of time those resources have to be available [7]. The goal of resource *allocation* is economic resource provisioning in a multi-tenant environment. TAn important related area is resource mapping, which is the correspondence of required resources by the user and available resources in the cloud. It enables identifying existing resources and matching them to a specific purpose [5]. Identifying that purpose and selecting an optimal or close-to-optimal resource configuration for an application are tasks that can be supported by resource estimation.

Resource estimation involves the prediction of (a) the resource demand of an application given the expected workload and optional configuration parameters for the application and (b) the expected execution time of a job or part of an application on a certain resource configuration. In the literature, mainly three different approaches to this problem can be found:

1) an application model is created locally to emulate the application behavior in the cloud,
2) the application is executed in a simulated cloud environment,
3) the resource demand of similar applications with the same estimated workload is used as a reference.

Further, the combination of any of the above methods is possible. What we can observe is that modeling is important for each of these methods in some way. The first approach utilizes application modeling, the second approach resource modeling and for the third approach modeling is not necessarily required, but it can support the process of selecting similar applications based on similar model parameters. Therefore, resource modeling also supports most areas of resource management indirectly by aiding resource estimation.

Another major area of resource management is resource discovery & selection which is important in all other areas of resource management. Especially pre-selecting VM instances or cloud provider can significantly reduce the benchmarking effort for resource estimation. This selection process can in turn also involve the application of resource modeling and estimation techniques but usually in a less compute intensive manner, e.g. using heuristics or linear regression with the aid of historical data.

The last important area in resource management is resource monitoring. To enable efficient, automated resource management, it is important to monitor critical resources and values, e.g. physical or virtual hardware utilization like CPU-usage, memory consumption or network traffic, but also QoS-related values like average response time or requests per minute. For resource estimation methods it needs to be carefully considered which values are necessary in a specific application case and which are not. Monitoring too many values can cause a significant overhead and thus lead to reduced application performance. If not enough or even the wrong values are monitored, the collected data might lead to less accurate prediction results. A good survey about resource monitoring has been provided by Aceto et al., where the authors further elaborate on particular challenges in this area [8].

Depending on the amount of resources provided by an instance, and the number of virtual instances the execution time and the cost for the execution of an application can vary by a reasonable amount [9]. Hence, for efficient resource management, it is valuable to know in advance which resource configuration is most suited for an application. Even if the resource configuration consists of just a single VM there are many factors to consider already. The resource demand of an application might vary due to application specific parameters. Further, the impact of virtualization is a major contributing factor to this decision as well as it has a non-neglectable impact on application performance [10], especially caused by *performance instabilities* in virtual networks [11], [12]. Lastly, the cost for each vendor needs to be considered. The impact of virtualization is even more significant on compute heavy applications with a variable resource demand that are able to scale *vertically*, i. e. with the amount of available resources on one VM, and/or *horizontally*, i.e. with the number of VMs the application is deployed on, as it is the case for many scientific applications. Another important branch of applications are software as a service (SaaS) applications where it is key to fulfill QoS requirements like low response times or a high number of requests per second whilst keeping cost at a minimum. Since the startup time of a VM causes a small delay between the request for additional resources and the resources being available [13], it is important to know when new resources should be provisioned in advance to be able to keep *over- and under-provisioning* at a minimum. While many studies, which we will discuss later in more detail, show the successful application of resource demand or load-balancing algorithms for different applications, comparing these studies is difficult due to most of them being application specific and, in many cases, methodologies that are difficult to compare are used to evaluate the effectiveness of an approach.

While surveying the related work, we kept track of the most frequently used application classes in resource

TABLE 1. APPLICATION CLASSES IN RESOURCE MANAGEMENT FOR IAAS CLOUDS BY DIFFERENT CATEGORIES

Type of Workload							
Web Server and SaaS			**Scientific Applications**		**Benchmark Applications**		
Communication	File Storage	Processing	Map-Reduce	Learning Algorithms	Micro-Benchmarks	System-Benchmarks	Application Benchmarks
Online Shops	Interactive	DBMS					
Scalability							
Multi-Tenancy			Vertical Scalability		Horizontal Scalability		
Favored Resource							
CPU Intensive		Memory Intensive		IO Intensive		Network Intensive	
Deployment Setup							
Single Layer			Multiple Layers				

management for IaaS clouds. The type of an application can range from CPU-, memory-, IO-, or network-intensive applications to loosely coupled and web-server like online shops or DBMS. Also the resource configuration that is required for an application is important, ranging from a single machine over multiple machines, supporting horizontal scaling, to complex multi-tier applications with different loosely coupled services interacting with each other. Table 1 shows an overview of the identified classes. We will explain the importance of each of these categories in more detail in section 4.

3. State-of-the-Art in Resource Management

We compare the most promising and current work in resource modeling, demand estimation as well as discovery & selection based on monitored or considered resource values, which application classes can be supported and which evaluation criteria have been used to validate the effectiveness of the approach. The comparison is summarized in Table 2.

Resource Modeling. The first two approaches show the interconnection between resource modeling, estimation and selection since all three areas are covered in these papers. We believe they give a great example on how well the different areas support each other.

For instance, the approach called *EMUSIM* which has been developed by Calheiros *et al.* [17] shows that modeling can reduce the benchmarking effort for performance estimation of cloud computing applications. The authors use an automated emulation framework [28] to emulate application behavior in the cloud. Based on that, a model is created, which is validated with further benchmarks. After it has been successfully validated, the model can be used to *simulate* the behavior of the application to test larger numbers of application execution requests. For the simulation the authors use *CloudSim* [29], which supports modeling and the simulation of data centers, users accessing the services hosted in those data centers and modeling of resource and VM provisioning algorithms. Since simulation requires generally *less hardware resources* than emulation, the benchmarking effort for resource estimation tasks is reduced in this environment in the long run. The proposed solution is particularly useful to measure the ability of an application to handle *concurrency* and to scale *horizontally*. Using simulation to determine the runtime of more complex tasks in a distributed environment is far more accurate than estimates using regression techniques. Regarding the authors, it is also *cost effective*. The authors further evaluated the outcomes of the simulation stage of their application in a public cloud and discovered that the *performance differences* between the simulated and the real system, which are represented by the normalized service time for the executed application, increase with the *concurrency level*. This means for applications that support a high level of concurrency, the correct estimation of the performance in the cloud is still a difficult task. In its current version, their approach only supports loosely coupled CPU-intensive applications but it is planned to extend this list with other application types such as Web-servers, DBMS and parallel applications.

Predicting the resource usage and response time for a generic workload without emulating the application behavior in the cloud is also possible. Rak et al. developed a solution, specifically for *mOSAIC* cloud applications that is solely based on simulation [16]. Therefore, it allows the prediction of the performance on various VM configurations before selecting the most appropriate. Further, a detailed performance/cost trade-off analysis is possible. However, to build the simulation model a set of benchmarks, which are application specific and must be generated in advance, have to be executed first which introduces a huge initial overhead. The historical data gathered from these benchmarks is required to create the simulation model. The approach by Rak et al. has been validated on a private cloud setup with a simple XML Analyzer *mOSAIC* application which consists of multiple components. The authors mentioned in their work that the prediction accuracy for the response time and corresponding workload cost is roughly 15%. However, since this approach is only suited for *mOSAIC* applications we found it very limited at its current state. Nonetheless, it offers a lot of potential for similar environments supporting the development and deployment of applications in IaaS clouds.

If application elements are not loosely coupled it is necessary to model how each of the components affect each other for effective resource management. Hajjat et al. [14] approached this problem with a method to calculate the correlation coefficients across application elements. This approach is useful to determine if elements of an application can be geographically distributed, in case no correlation is

TABLE 2. OVERVIEW OF MONITORED RESOURCES AND EVALUATION CRITERIA AS WELL AS APPLICATION CLASSES USED FOR THE VERIFICATION OF THE APPLIED METHODS FROM THE LITERATURE ABOUT RESOURCE MANAGEMENT

Paper	Monitored / Considered Resources										Evaluated With			Application Class					
														Workload			Scalability		
	CPU	Memory	Storage	Workstations	OS	Data Size / Workload	Network Bandwidth	Network Loads / Delays	Thread Dependencies	VM Launching Time	Time / QoS / Performance	Cost	Energy	Web Server	Scientific	Benchmark	Multi-Tenant	Vertical	Horizontal
Resource Modelling																			
Hajjat et al. (2015) [14]							X				X			X			X		X
Wu et al. (2015) [15]	X									X	X					X			
Rak et al. (2013) [16]						X					X	X		X			X		
Calheiros et al. (2013) [17]	X			X		X					X	X			X		X	X	X
Resource Demand Estimation																			
Loff and Garcia (2014) [18]	X			X		X				X	X				X	X			X
Tak et al. (2013) [19]	X	X				X	X		X		X			X		X	X		
Bankole and Ajila (2013) [20]	X	X	X			X		X			X			X		X	X		
Li et al. (2011) [21]	X	X				X			X		X			X		X	X		
Resource Discovery & Selection																			
Zhang et al. (2015) [22]	X	X	X	X	X		X	X			X	X						X	
Sadooghi et al. (2015) [23]	X	X	X	X		X	X	X			X	X			X	X		X	X
Borhani et al. (2014) [24]	X			X		X					X	X		X				X	
Kaisler et al. (2012) [25]											X	X							
Garg et al. (2011) [26]	X	X	X	X			X	X		X	X	X						X	X
Li et al. (2010) [27]	X	X	X			X		X		X	X	X		X	X	X	X	X	

evident, or not and to estimate the performance in such cases. The authors further characterized bad performance periods, which occur due to instabilities of the virtualized hardware in cloud computing. According to their work, such periods are usually short-lived, frequent and involve only a small subset of application components. It was also concluded that requests that involve multiple application components should be handled using the most suitable element for each subtask whilst not considering the data center locations, i.e. the geo distribution of the individual elements. This generally leads to a better performance compared to selecting a single data center. The approach has been validated with a variety of web-server based applications, a social application (Twissandra), a data-intensive web application (Thumbnail) and two enterprise applications (StockTrader and DayTrader). However, currently the approach has only been tested on the AWS cloud (Twissandra and DayTrader) and Microsoft Azure (Thumbnail and StockTrader) using one and not various VMs from each provider.

A different approach to model performance fluctuations but on a resource level, has been presented by Wu et al. [15]. The authors measured large variations in computing power not only on public but also on private cloud systems, on AWS and FermiCloud respectively, during VM operation. The deployment of a new VM on the same physical machine has been identified as one of the biggest influencing factors for VM performance. Furthermore, the VM performance degrades over time even if the workload of an application does not increase. Although the authors have not created a full reference model for VM performance variations, they additionally identified influencing factors for the VM startup time, which are (a) the process of transferring the image to the physical machine and (b) booting the VM, and created a VM launching CPU utilization overhead reference model for each of these factors. In their ongoing work, the authors plan to utilize these models to develop a VM performance variation reference model caused by resource contentions after VM creations.

Resource Estimation

In most cases, resources in IaaS are delivered as *packaged VMs* and the resource demand of an application is estimated as the number and the capacity of VMs used for the application execution. One problem with packaged resources is, that no standards for their size exists. Accordingly, every cloud provider creates his own types of packages instead. This makes accurate performance comparisons between different cloud providers difficult. In our previous work [30], we classified resource estimation algorithms into two major categories, based on whether they consider multiple VM types, that includes different cloud providers, or not. One important idea behind using this classification is, that it corresponds with some of the application classes we established earlier. According to Table 1 an application can have three different scalability classes. Considering different sized VMs in a resource estimation approach means that effectively the ability of an application to scale vertically, e.g with more RAM or a faster CPU, is researched. We categorized these approaches as part of the resource selection area instead of resource estimation. In approaches that consider just a single VM type, usually the number of how many of them are deployed varies. Thus, the ability of an application to scale horizontally is researched, often in combination with a load-balancing algorithm. The following papers belong to the latter category.

Li *et al.* presented *CloudProphet*, an approach in which a so called shadow is created by benchmarking an application locally to emulate its behavior in the cloud afterwards [21]. This enables a very accurate prediction of the actual application performance for when it is deployed under the same circumstances. Unfortunately, it is not possible to accurately predict the performance on a different VM without emulating the application a second time. The authors stated, that regression methods could be used to predict the performance instead, based on the amount of provided resources, but this would add another layer of complexity and thus significantly *decrease the prediction performance*. The application behavior on multiple VMs and thus the ability to scale horizontally is also not considered in this solution. Furthermore, applications that are computationally expensive are not suited well for this approach since it relies heavily on *benchmarking*.

A similar approach by Tak et al. [19] creates a PseudoApp that mimics the resource consumption of an application. PseudoApp also works with multi-layered applications and has been evaluated with the TPC-W E-commerce benchmark. Since the PseudoApp has no specific requirements like operating system or installed software, it can easily be deployed and thus used for comparing the performance of an application on different VMs which is usually a very time consuming process. The determined prediction error in the throughput for the application for different workloads is 2-8% on an in-house private cloud and on the AWS cloud. Although this approach seems to work well, it still has several limitations. It does, for example, not accurately capture the real memory access pattern of the application and the impact of L1/L2/L3 caches. We also believe PseudoApp is still heavily reliant on cost intensive benchmarking which is a major problem for most prediction tasks.

Besides emulation, also traditional forecasting methods are utilized. That is generally the case for load-balancing algorithms where scaling decisions have to be made either in real-time or only a few minutes in advance to compensate VM startup latencies. Bankole and Ajila compare three selected machine learning techniques to predict the future CPU utilization, response time and throughput of an application based on historical data [20]. The results show that for predicting these values for the TPC-W benchmark, support vector regression is superior to linear regression and neural networks. However, although it is planned in future work, the approach has not been tested on a public cloud infrastructure yet.

Instead of comparing different algorithms with each other, Loff and Garcia presented an approach that uses a weighted k neural network that combines the results for an ensemble of forecasting methods (Holt-Winters, ARIMA and StructTS) [18]. The authors observe under- and over-provisioning and calculate the impact of these values separately. To avoid under-provisioning errors, a padding value is added to each forecast based on a probable resource usage burst. The individual forecasting methods are then compared with the combined estimation using the mean absolute percentage error (MAPE). The results show that the predictions by the ensemble of the three forecasting methods reduces the MAPE by half.

Resource Discovery & Selection. The first step for selecting a VM is the selection of the *cloud provider*. Li et al. state in an early comparison of cloud providers that some cloud instances are better at handling a high resource demand of a specific resource, e.g. CPU or IO, than others [27]. The presented approach, called *CloudCmp*, is useful to roughly estimate which provider should be used for a certain application.

Similarly Garg et al. [26] proposed a framework for comparing and ranking IaaS provider with 13 different key performance indicators (KPIs) based on service measurement index (SMI) attributes provided by the Cloud Service Measurement Initiative Consortium (CSMIC)[1]. For the ranking of the cloud services a mechanism based on the Analytic Hierarchy Process (AHP) has been proposed. The approach is validated in a case study with data collected from AWS EC2, Windows Azure and Rackspace. Currently, the ranking algorithm does only cover quantifiable QoS attributes like elasticity or service response time. In future work, also qualitative attributes will be considered as well as variations in QoS attributes like performance variations of VMs.

A similar approach for selecting resources in IaaS clouds by Zhang et al. also relies on AHP but adds significant new contributions [22]. First, besides several weighted quantitative features like storage, RAM, CPU speed and cost, multiple qualitative features like location, CPU architecture,

1. https://slate.adobe.com/a/PN39b - last visited: 08.04.2016

operating system and QoS criteria are considered as well. The authors further provide a clear formulation of the research problem for IaaS service selection and implement a generic service for collecting QoS values from different sources. In this approach, the cloud provider or VM with the lowest cost/benefit ratio is selected. This ratio is calculated based on all available attributes for a specific provider and the requirements of the cloud user.

Alternatively, a cloud provider selection approach that is based on specific application requirements like business objectives, QoS attributes and *architectural decisions* has been developed by Kaisler *et al.* [25]. Although it is not based on the resource demand, this decision framework is useful to select the cloud provider for an application first, which narrows down the options for the selection of an appropriate VM as well. Furthermore, *pre-paid* instances, which cost less than usual but have to be ordered and paid for in advance, are also considered in this framework.

Borhani et al. use CPU micro-benchmarks to detect CPU sharing on VMs of different cloud providers during the execution of a blogging application [24]. Further a workload generator that generates different kinds of increasing workload has been used to measure performance, cost and performance variations. The results can be used to recommend a provider and an appropriate instance based on the type of expected workload - which can either be read intensive, write intensive or both - for the blogging application.

An approach that relies on micro-benchmarks for performance comparison of compute and storage services in IaaS clouds has been presented by Sadooghi et al. [23]. Their work is concerned with the applicability of cloud computing for scientific applications. The micro-benchmarks are used to capture the raw performance of VMs, specifically in the AWS cloud, which is compared to the performance of a typical non-virtualized system. The results show, that private clouds are generally more cost efficient. In terms of performance the authors conclude that the virtualization effect on memory and CPU performance is relatively low in their application case. However, the variations in network latency are very high for standard instances compared to HPC instances which show a more stable network performance and less overhead of the network virtualization. Regarding storage performance the authors further investigated different AWS services like EBS, S3 and DynamoDB. Since EBS volumes are remotely accessed over the network the performance compared to local storage is very poor and the throughput even in a RAID setup cannot exceed 120 MB/s. The S3 storage service is also outperformed by the commonly used PVFS for scientific applications. However, in terms of scaling S3 starts to perform better if more than 96 instances are used for an application. Another observation made in this paper is that the latency of the DynamoDB service does not change much with scales but compared to ZHT, an open source consistent NoSql database, the performance is significantly lower. In conclusion this work can help researchers in deciding if the cloud and which specific services should be utilized for their particular scientific computing workload.

4. Discussion

Beside performance differences between cloud providers that impede the creation of general solutions for efficient resource management, especially network instabilities in public clouds make the comparison between local resources and VMs in the cloud rather difficult. A possible solution for this problem is the assignment of a *performance rating* for the different resources that are utilized in a VM. This can be used to pre-select instances using *resource thresholds* for certain application classes similar to the CloudCmp approach by Li *et al.* [27]. The performance ratings of preselected VMs can further be utilized to enhance emulation and simulation frameworks like *EMUSIM* [17] for a more accurate simulation of the application behavior in a public cloud. One remaining challenge in this regard are the performance fluctuations of cloud VMs. To incorporate these in the performance rating, more detailed models about VM behavior that include the cause of these fluctuations are necessary. The work by Wu et al. is a good step in this direction [15]. However, comparing the VM performance of different cloud provider is just one challenge towards efficient resource management in the cloud. Being able to compare resource management techniques to select the most appropriate based on a specific application case is another major issue.

Whilst analyzing the existing resource management solutions so far we noticed that many approaches are very application specific. Resource discovery & selection approaches are more flexible in this regard but they usually require more input about the application from the user, rather than relying on benchmarking or monitoring. Furthermore, the required input significantly differs between solutions. This issue and the problem that most solutions rely on just a single application type for the verification of their solution is worsened by the different and difficult to compare methodologies that are used for the evaluation of the effectiveness of an approach, making more difficult to compare in general. Usually time and QoS constraints are the most often regarded evaluation criteria. Cost is more important in the resource selection process. Surprisingly, energy demands have not been widely considered in the research on public IaaS clouds. For verifying the effectiveness of a solution the most popular application that is used for that purpose is the TPC-W benchmark, which is an exemplary web server application. However, it has been argued by Binnig et al. that the TPC-W, as most other commonly used benchmarks in the literature, is not sufficient for cloud computing applications as they do not consider scalability, peak load and fault tolerance in an appropriate way [31]. As pointed out by Aceto et al., standard test beds are necessary to compare models more efficiently [8]. In addition, Weingärtner et al. stated that larger workloads are required to simulate the dynamic nature of the cloud since usually, models for profiling and forecasting are just evaluated on specific and well-known workloads [32]. We propose, given the availability of standard test beds, that application classification can be applied

to resource estimation for arbitrary applications to select the most appropriate prediction and selection algorithms based on the identified application types. Many applications have shown a high efficiency of classification methods to detect different networking applications already [33]–[35]. Even *clustering methods* have been considered to identify new or frequently changing protocols [36], [37]. Similarly, clustering, potentially combined with resource modelling, could prove to be useful in identifying applications that are similar to each other and thus benefit from the same resource estimation algorithm. Benchmarking applications indirectly by using only representative applications from the same class has been successfully applied by Chhetri et al. already [38]. We believe, that the development of a general application model for cloud applications that can be used to represent multiple different application classes will be useful to support the detection of similar applications in the context of resource management. Challenges that remain are identifying the most common classes and creating an ideal test bed for each of them. During our literature review we made an attempt to extract the most commonly used application classes in the research, which are summarized in table 1. In the following, we would like to highlight further advantages of this classification.

In cloud computing, only certain types of applications benefit from properties like high elasticity and pay-per-use. Those are applications that either (a) do not run full time, i.e. the infrastructure for the execution has to be provisioned for a certain amount of time only, or (b) if they do run full time they are able to scale up or down according to the current workload of the application. The first category includes benchmarking and also scientific applications. Benchmarking applications can further be categorized, e.g. into different types of micro benchmarks for measuring specific attributes like IO performance or network latency. Further, those benchmarks can be either application or system related meaning they either measure application performance values like response time or requests per second, or system performance values like CPU-load or memory consumption. Scientific applications include learning algorithms and algorithms for big data analysis, e.g. map-reduce. Examples for full time applications are web or application server which includes SaaS applications as well. Those can also be further categorized based on their main infrastructure requirements. Communication servers like Twitter require good network capabilities, services for storing files like Dropbox rely mostly on storage, and the performance of processing, e.g. image processing applications like Thumbnail or stream processing for big data, is usually directly related to the performance of the CPU and the network. Another important property of each category are specific workload patterns that are related to it, e.g. the increased likelihood of online shops having more customers in December due to the holiday season. That enables a more detailed prediction of future resource demands based on historical data.

Besides the type of workload, applications can also be classified based on their scalability properties. Obvious classes are vertical or horizontal scalability. Further, multi-tenancy is also related to the scalability of an application. An application might also be categorized into all three classes or none of them if it does not scale at all.

A type of classification that has been successfully utilized for improved scheduling of virtualized resources in a data center before [6] are favored resources by an application. Those could be either CPU, Memory, Network or IO intensive applications. Again, it is possible for an application to belong to multiple of these classes. Knowing the favored resource of an application can be used for scaling decisions like the selection of an appropriate load-balancer, e.g. if an application is CPU intensive it is useful to choose an approach that involves monitoring CPU utilization. This can reduce the monitoring overhead and also the benchmarking effort for resource demand estimation.

Another important class is the deployment setup, more specifically if an application consists of just a single layer or multiple layers that interact with each other. In case of multiple layers, each part of the application should be considered as an independent application itself and categorized accordingly into one or multiple of the above classes.

5. Conclusion and Future Work

Resource management for IaaS clouds offers many promising solutions for various application cases already. However, most of these solutions have only been tested in specific scenarios and are not comparable to other solutions due to the lack of standardized test environments. Having multiple test environments is a necessity due to the fact that many different application classes with different requirements, i.e. on the infrastructure, exist. We presented the most promising approaches in resource estimation, modeling and discovery & selection and identified key classes of applications that have to be considered. We mentioned general challenges for resource management in cloud computing and specific challenges regarding the comparability of different approaches. We discussed possible solutions like the creation of automated test beds which will enable a better comparison of existing and new resource management approaches. Finally, we highlighted the potential benefits of applying application classification in combination with a general application model in this area. We believe that this combination can have a significant impact, especially regarding the development and automation of general resource management solutions. Our future research involves the evaluation of these possibilities.

References

[1] P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, NIST Std., Jan 2011.

[2] B. Narasimhan and R. Nichols, "State of cloud applications and platforms: The cloud adoptors' view," *Computer*, vol. 44, no. 3, pp. 24–28, Mar 2011.

[3] P. Gupta, A. Seetharaman, and J. R. Raj, "The usage and adoption of cloud computing by small and medium businesses," *International Journal of Information Management*, vol. 33, no. 5, pp. 861–874, Oct 2013.

[4] D. Lifka, I. Foster, S. Mehringer, M. Parashar, P. Redfern, C. Stewart, and S. Tuecke, "Xsede cloud survey report," XSEDE (Cornell Center for Advanced Computing and ANL and The University of Chicago and Rutgers University and Indiana University), Tech. Rep., 9 2013.

[5] S. S. Manvi and G. K. Shyam, "Resource management for infrastructure as a service (iaas) in cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 41, pp. 424–440, 5 2014.

[6] J. Zhang and R. Figueiredo, "Application classification through monitoring and learning of resource consumption patterns," in *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, April 2006, pp. 10–19.

[7] B. Sotomayor, R. S. Montero, I. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *Internet Computing, IEEE*, vol. 13, no. 5, pp. 14–22, Sept 2009.

[8] G. Aceto, A. Botta, W. de Donato, and A. Pescap, "Cloud monitoring: A survey," *Computer Networks*, vol. 57, no. 9, pp. 2093 – 2115, 2013.

[9] M. Ullrich, K. ten Hagen, and J. Lässig, "Public cloud extension for desktop applications–case study of a data mining solution," in *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on*. IEEE, 2012, pp. 53–64.

[10] J. Sahoo, S. Mohapatra, and R. Lath, "Virtualization: A survey on concepts, taxonomy and associated security issues," in *Second International Conference on Computer and Network Technology (ICCNT)*. IEEE, Apr 2010, pp. 222–226.

[11] G. Wang and T. Ng, "The impact of virtualization on network performance of amazon ec2 data center," in *IEEE INFOCOM*, Mar 2012, pp. 1–9.

[12] J. Schad, J. Dittrich, and J.-A. Quian-Ruiz, "Runtime measurements in the cloud: observing, analyzing, and reducing variance," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 460–471, Sep 2010.

[13] M. Mao and M. Humphrey, "A performance study on the vm startup time in the cloud," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, June 2012, pp. 423–430.

[14] M. Hajjat, S. Pn, A. Sivakumar, and S. Rao, "Measuring and characterizing the performance of interactive multi-tier cloud applications," in *Local and Metropolitan Area Networks (LANMAN), 2015 IEEE International Workshop on*, April 2015, pp. 1–6.

[15] H. Wu, S. Ren, T. Steven, and G. Garzoglio, "A step toward deploying real-time applications on cloud - modeling cloud performance fluctuation," in *21st IEEE Real-Time and Embedded Technology and Applications Symposium*, 2015.

[16] M. Rak, A. Cuomo, and U. Villano, "Cost/performance evaluation for cloud applications using simulation," in *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013 IEEE 22nd International Workshop on*, June 2013, pp. 152–157.

[17] R. N. Calheiros, M. A. Netto, C. A. De Rose, and R. Buyya, "Emusim: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications," *Software: Practice and Experience*, vol. 43, no. 5, pp. 595–612, 2013.

[18] J. Loff and J. Garcia, "Vadara: Predictive elasticity for cloud applications," in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, Dec 2014, pp. 541–546.

[19] B. C. Tak, C. Tang, H. Huang, and L. Wang, "Pseudoapp: Performance prediction for application migration to cloud," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, May 2013, pp. 303–310.

[20] A. Bankole and S. Ajila, "Cloud client prediction models for cloud resource provisioning in a multitier web application environment," in *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*, March 2013, pp. 156–161.

[21] A. Li, X. Zong, S. Kandula, X. Yang, and M. Zhang, "Cloudprophet: towards application performance prediction in cloud," in *ACM SIGCOMM*, vol. 41, Aug 2011, pp. 426–427.

[22] M. Zhang, R. Ranjan, M. Menzel, S. Nepal, P. Strazdins, and L. Wang, "A cloud infrastructure service recommendation system for optimizing real-time qos provisioning constraints," *IEEE Systems Journal*, vol. X, p. X, 2015.

[23] I. Sadooghi, J. Hernandez Martin, T. Li, K. Brandstatter, Y. Zhao, K. Maheshwari, T. Pais Pitta de Lacerda Ruivo, S. Timm, G. Garzoglio, and I. Raicu, "Understanding the performance and potential of cloud computing for scientific applications," *Cloud Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[24] A. Borhani, P. Leitner, B.-S. Lee, X. Li, and T. Hung, "Wpress: An application-driven performance benchmark for cloud-based virtual machines," in *Enterprise Distributed Object Computing Conference (EDOC), 2014 IEEE 18th International*, Sept 2014, pp. 101–109.

[25] S. Kaisler, W. Money, and S. Cohen, "A decision framework for cloud computing," in *System Science (HICSS), 2012 45th Hawaii International Conference on*, Jan 2012, pp. 1553–1562.

[26] S. Garg, S. Versteeg, and R. Buyya, "Smicloud: A framework for comparing and ranking cloud services," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, Dec 2011, pp. 210–218.

[27] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: comparing public cloud providers," in *ACM SIGCOMM*, vol. 10, 2010, pp. 1–14.

[28] R. N. Calheiros, R. Buyya, and C. A. F. De Rose, "Building an automated and self-configurable emulation testbed for grid applications," *Software: Practice and Experience*, vol. 40, no. 5, pp. 405–429, 2010.

[29] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[30] M. Ullrich and J. Lässig, "Current challenges and approaches for resource demand estimation in the cloud," in *IEEE International Conference on Cloud Computing and Big Data (IEEE CloudCom-Asia 2013)*, Dec 2013, pp. 387–394.

[31] C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, "How is the weather tomorrow?: Towards a benchmark for the cloud," *Proceedings of the Second International Workshop on Testing Database Systems*, pp. 1–6, 2009.

[32] R. Weingrtner, G. B. Brscher, and C. B. Westphall, "Cloud resource management: A survey on forecasting and profiling models," *Journal of Network and Computer Applications*, vol. 47, pp. 99–106, 1 2015.

[33] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 50–60, Jun. 2005.

[34] H. Jiang, A. W. Moore, Z. Ge, S. Jin, and J. Wang, "Lightweight application classification for network management," in *Proceedings of the 2007 SIGCOMM Workshop on Internet Network Management*, 2007, pp. 299–304.

[35] Y.-D. Lin, C.-N. Lu, Y.-C. Lai, W.-H. Peng, and P.-C. Lin, "Application classification using packet size distribution and port association," *Journal of Network and Computer Applications*, vol. 32, no. 5, pp. 1023 – 1030, 2009.

[36] B. Kurt, A. Cemgil, M. Mungan, N. Polat, A. Ozdogan, and E. Saygun, "Network management without payload inspection: Application classification via statistical analysis of bulk flow data," in *Future Network & Mobile Summit (FutureNetw), 2012*, Jul 2012, pp. 1–8.

[37] R. Goss and N. G.S., "Automated network application classification: A competitive learning approach," in *IEEE Symposium on Computational Intelligence for Communication Systems and Networks (CIComms)*, Apr 2013, pp. 45–52.

[38] M. Chhetri, S. Chichin, Q. B. Vo, and R. Kowalczyk, "Smart cloudmonitor - providing visibility into performance of black-box clouds," in *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on*, June 2014, pp. 777–784.