Adaptation in cloud resource configuration

DOI:
10.1186/s13677-016-0057-9

Document Version
Final published version

Hummaida *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2016) 5:7
DOI 10.1186/s13677-016-0057-9

Journal of Cloud Computing:
Advances, Systems and Applications

REVIEW Open Access

Adaptation in cloud resource configuration: a survey

Abdul R. Hummaida*, Norman W. Paton and Rizos Sakellariou

Abstract

With increased demand for computing resources at a lower cost by end-users, cloud infrastructure providers need to find ways to protect their revenue. To achieve this, infrastructure providers aim to increase revenue and lower operational costs. A promising approach to addressing these challenges is to modify the assignment of resources to workloads. This can be used, for example, to consolidate existing workloads; the new capability can be used to serve new requests or alternatively unused resources may be turned off to reduce power consumption. The goal of this paper is to highlight features, approaches and findings in the literature, in order to identify open challenges and facilitate future developments. We present a definition of cloud systems adaptation, a classification of the key features and a survey of adapting compute and storage configuration. Based on our analysis, we identify three open research challenges: characterising the workload type, accurate online profiling of workloads, and building highly scalable adaptation mechanisms.

Keywords: Autonomic cloud, Cloud adaptation, Resource management, Elasticity

Introduction

Cloud computing is an established paradigm for providing on demand computing services to a wide range of users, including enterprises, software developers and researchers. Infrastructure Providers (IPs) manage the base infrastructure, including servers, storage and network connectivity, and typically present this infrastructure as Virtual Machines (VMs). Other providers rent these resources and resell value-added services (VARs) as Platform as a Service (PaaS) or Software as a Service (SaaS).

VARs utilise clouds to lower operating costs by only paying for computing resources they use. The ability to expand to additional resources means they do not have to build capacity upfront. In return for these benefits, VARs typically pay a higher per hour cost for resources used, compared to managing infrastructure directly. IPs, on the other hand, have the challenge of providing these benefits to VARs. IPs build the capacity to cope with increasing demands for computing resources, which requires significant investment in infrastructure, skilled personnel and incurs power costs. Furthermore, with increased competition and commoditisation of cloud services, IPs are under pressure to reduce their prices. Amazon reduced its prices on 41 different occasions in the last few years [1]. The adoption of cloud computing does, however, open up a new market for IPs, where they can run a wide variety of computing requests that previously were housed in private infrastructure.

IPs generate revenue by meeting Service Level Agreements (SLAs). To achieve this, one approach is to periodically *Adapt* the infrastructure configuration. Adaptation typically entails a decision to increase or reduce cloud resource allocation to a workload. For example, the CPU share allocated to a VM running a web server can be reconfigured to a lower share, if SLAs can remain unaffected. The gained capacity can be used to accept new workloads or to reduce power consumption, resulting in an increase in IP profit.

This paper surveys resource reconfiguration, covering 40+ publications that focus on adaptation of computing resources in a cloud context. The chosen publications appeared in cloud focused journals and conferences. Our contributions are a definition for cloud adaptation and a classification that we use to survey the literature. To

*Correspondence: abdul.hummaida@postgrad.manchester.ac.uk
University of Manchester, School of Computer Science, M13 9PL, UK
Manchester, UK

focus the scope of this work, we chose to cover adaptation of compute and storage resources. However, we recognise the potential impact adaptation of network resource can play. For example, multiple under-utilised network routers can be powered down by reconfiguring the network infrastructure, thus lowering power consumption.

Several surveys pull together results of different features of cloud resource management. In [2] the authors surveyed elastic approaches in cloud computing, providing a high level overview of the approaches. Our survey is different as it investigates adaptation and, as we demonstrate later, adaptation is a superset of elasticity. In [3], the authors comprehensively discuss approaches to efficient data centres, choosing to focus on power consumption. Our work covers power as an adaptation objective and also covers SLA and revenue. In [4], the authors surveyed autoscaling, and classified the literature based on the adaptation techniques used. Their work focused on the Infrastructure as a Service (IaaS) client's perspective, while we focus on the IaaS provider, thus their work excluded VM migration and server consolidation. In [5], the authors provide an overview of the mechanisms and techniques employed to manage elasticity from the perspective of a SaaS provider, while we focus on the IaaS provider. In [6], the authors investigate cloud resource management and in [7], the authors present common aspects used in cloud computing environments, such as metrics, tools and strategies. In [8], the authors surveyed the VM allocation problem and models and algorithmic approaches. In [9], the authors present analysis of autonomic resource management in general, and specifically Quality of Service aware autonomic resource management. In [10], the authors surveyed SLA-based cloud research including the techniques used for adaptive resource allocation. In [11], the authors surveyed cloud computing elasticity using a classic systematic review covering metrics and tools. In [12], the authors summarised different method and theory used in cloud resource allocation and monitoring. In [13], the authors depict a broad literature analysis of resource management in the cloud. While there is some overlap from these surveys with our work, they chose a different classification scheme to our work, which focuses on adaptation of resource configuration, enabling us to analyse the factors that influence the adaptation process. Additionally we investigate factors affecting scalability of the various proposals in the literature. To the best of our knowledge there is no other work that uses our chosen dimensions.

As PaaS can be built on top of IaaS, there can be similarities between how resources are adapted in both environments. However, as IaaS is typically presented at the VM abstraction level, IPs have less visibility into the nature of workloads and their configurations. This presents additional challenges for Autonomic [14] approaches to adapting resources on IaaS. "Cloud systems setup" Section introduces cloud infrastructure and lays the foundation for a discussion on how this can be adapted in "Cloud systems adaptation" Section; we also define the dimensions used in the survey. "Adaptation in cloud resource configuration" Section surveys the literature based on the adapted cloud resource, identifying the techniques and approaches used. "Open research challenges" Section presents open challenges in adapting resource configuration and "Conclusion" Section presents our conclusions.

Cloud systems setup

Cloud computing is defined as *"a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"* [15]. In this section we introduce the constituents of cloud systems.

Compute Resource: The core processing capabilities that are used to execute software instructions. We define this as comprising of a CPU, typically in multicore configuration, CPU cache and primary storage memory. Data centres typically house many thousands of servers containing these compute resources.

Storage Resource: Non-volatile secondary storage memory houses the data used by compute resources. As this resource is typically cheaper than primary memory, many operating systems are able to use it as an extension of main memory, to temporarily swap out unused memory state. Many data centres will have servers with access to internal storage as well as to a Storage Area Network that consolidate and abstracts the complexity of accessing storage throughout the data centre.

Network Resource: includes the network cards that connect into servers as well as infrastructure components that include repeaters, load balancers, switches and firewalls. Networks can use different topologies and protocols, which influence the level of security, resilience and Quality of Service.

Virtual Resource: is an abstraction added onto compute, storage and network resources. It enables slicing of these resources into smaller chunks that can be scaled vertically or horizontally. Typically virtualisation is used in a data centre to slice data centre compute resource into Virtual machines, and potentially to present several logical processors by mapping these onto a single physical processor. Network cards and storage are also virtualised and presented as individual devices to VMs.

Service Management Resource (SMR): is a knowledge library where IPs store management objectives, policies, pricing and orchestration information.

Management Tools: are used by IPs to provision, monitor, reconfigure, back up and restore the infrastructure.

IPs typically build the infrastructure and offer access to virtual resources, with a VM being the main component. VMs reside on physical nodes of heterogeneous capabilities where the performance characteristics of compute, storage and network vary. Demand for recourses varies over time as users consume and release these resources. As more resources are used, power consumption in the data centre increases and IPs may choose to optimise the allocation of VMs to physical nodes. In the next section, we will cover IPs objectives and approaches used to optimise this allocation.

Cloud systems adaptation

In this section we introduce the IPs objectives and approaches to adapting the cloud infrastructure.

To meet workload demands, IPs can use Elasticity [16] to reconfigure resources in an autonomic manner. The limitation of this view is that it assumes the IP's objective is to satisfy precisely all workload demands. While this may be true, it may not always be the case, as the IP has finite resources and may apply differentiation on requests. Additionally, the IP may decide it is more cost effective to pay a penalty for an SLA violation instead of scheduling the request. The current view on Elasticity abstracts several complex activities. We refine this view by separating the decision making process from how the cloud environment is reconfigured, by defining elasticity as the *on demand ability, to scale vertically or horizontally segmented resources in discrete units*. To achieve a specific business goal, IPs go through a decision making process that changes the infrastructure, a process we name Cloud Systems Adaptation. We define this as a *change to provider revenue, data centre power consumption, capacity or end-user experience where decision making resulted in a reconfiguration of compute, network or storage resources*. Reconfiguration is the process of increasing or reducing resource allocation to a workload, through elasticity.

Core to cloud systems adaptation is a decision making process that decides the resources to reconfigure and how. Figure 1 shows the inputs into the decision making process, including:

1. The desired management objective in each adaptation cycle from the SMR.
2. The adaptation techniques and infrastructure metrics.

When decision making is complete, Elasticity is used to scale the infrastructure resources.

We define the dimensions of cloud systems adaptation as: 1) *Adapted cloud resource*, which categorises what resources are modified and how; 2) *Adaptation objective* is a desired business outcome; 3) *Adaptation techniques* are a set of analytical and modelling techniques used to achieve the adaptation objective; 4) *Adaptation engagement* categorises when the adaptation process is invoked; 5) *Decision engine architecture* categorises the different architectures used by the decision making engines within the literature; 6) *Managed infrastructure type* categorises whether node capabilities and properties are used in the decision making. These dimensions are presented in Table 1 and discussed in the following subsections.

Adapted resource

We extend the definition of possible resource adaptation from [17] in Table 1, which describes our classification of the literature and the dimensions used. *VM* level adaptation are typically applied to improve/reduce workload performance due to an increased/reduced demand by adjusting *CPU, memory, disk bandwidth* and/or *storage*. For example, a web server running on a VM may need a bigger share of CPU due to an increased number of requests.

Node level adaptation could be applied to add capacity by *powering on* a node. Power consumption could be reduced by using Dynamic Voltage and Frequency Scaling (*DVFS*) [18], before the node is *powered off* when not needed. Node configuration can also be adapted when a VM's requirements extend beyond the capacity of its hosting node, so that it needs to be *migrated* to another node that has the required capacity. Migration can also be used to reduce power consumption, by consolidating VMs into fewer nodes and enabling some nodes to be switched off.

Cluster level adaptation is applied to facilitate node adaptation and to adhere to any reliability policies used by IPs by *adding* and/or *removing* nodes.

Adaptation objective

All of the proposals surveyed drive adaptation to minimise *SLA* violations and some trade this off with a secondary objective. Examples include reducing *power* consumption, maximising IP *revenue* and combined where multiple objectives are sought. A small number of proposals focus on reducing the *customer cost* of using the infrastructure.

Adaptation technique

Several adaptation techniques have been applied to cloud infrastructure in the literature, including *Heuristic*, *Control theory* or *Machine learning* [19].

Heuristic based adaptation techniques use problem specific knowledge to provide a quick solution and trade preciseness of the outcome with lower time complexity, which makes them good candidates for dynamic resource allocation on the cloud. Control theory can provide QoS guarantees by using a feedback controller, that dynamically adjusts the behaviour of the system based on the measured outputs. Machine Learning techniques are

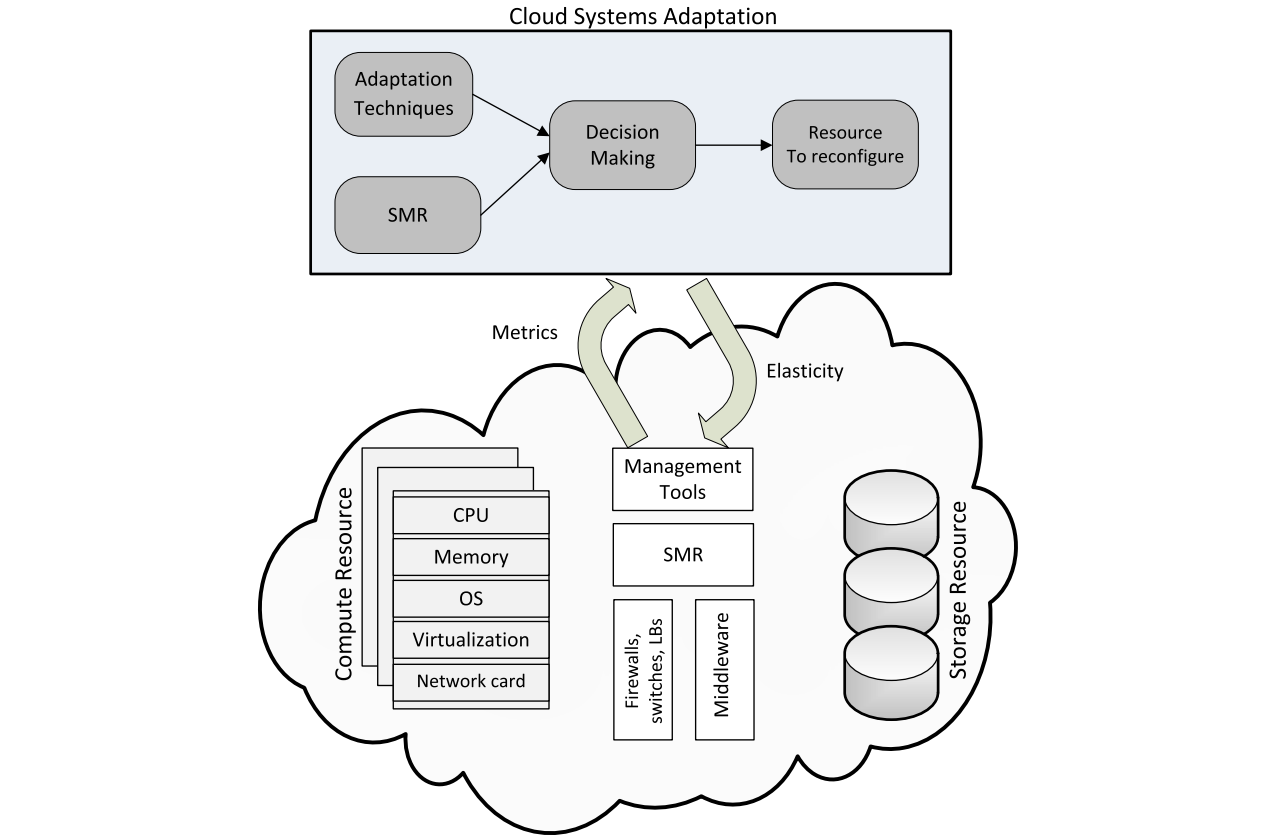

Fig. 1 Cloud Systems Adaptation sequence

grouped into two categories, supervised and unsupervised learning.

Adaptation engagement

Cloud systems adaptation needs to be invoked in order to evaluate the infrastructure and determine whether resource reconfiguration is required. The approaches used in the literature fall onto *Reactive*, *Proactive* and *Hybrid* engagement.

Reactive approaches invoke adaptation when a monitored metric, e.g. CPU utilisation, reaches a specific threshold.

Proactive approaches predict what demands will be placed on the infrastructure and invoke adaptation ahead of the predicted resource contention point.

Hybrid approaches utilise proactive approaches and combine these with reactive approaches, as way to engage adaptation for long and short term time scales.

Table 1 Dimensions for Cloud Systems Adaptations

Dimension	Definition
Adpated Resource	VM $\subset$ {Adjust CPU, Memory, Storage, Disk Bandwidth}
	Node $\subset$ {Power on/off, Adjust DVFS, Migrate VM}
	Cluster $\subset$ {Add/remove nodes}
Adaptation Objective	SLA, Power, Revenue, Customer Cost
Adaptation Technique	Heuristics, Control theory, Queing theory
	Machine learning
Adaptation Engagement	Reactive, Proactive, Reactive/Proactive
Decision Engine Architecture	Central, Hierarchical, Distributed
Managed Infrastructure	Heterogeneous, Homogeneous

Decision engine architecture

The architecture of the decision engine governs where the engine is placed and how it operates. *Centralised* architectures use an engine with a global view of the managed infrastructure and can adapt resource across the entire infrastructure.

Hierarchical architectures typically divide the infrastructure into multiple clusters, placing an engine (Level 1) in each cluster. A global, Level 2, engine coordinates each of the Level 1 engines.

Distributed architectures typically use a Peer-to-Peer protocol [20] that enables nodes to communicate directly without a centralised controller.

Managed infrastructure

Cloud systems are typically diverse and made of a *heterogeneous* set of compute and storage resources [21]. Some of the proposals incorporate the type of the managed infrastructure in the decision making process, while other proposals assume a *homogeneous* infrastructure, where every node has the same capability and power consumption.

Adaptation in cloud resource configuration

In this section we survey the literature that adapt cloud resource configuration, focusing on compute and storage resources. We chose to focus on the reconfigured resource and classify the reviewed literature on this dimension, in order to analyse patterns that are specific to a cloud resource. The reconfigured resources are:

1. CPU and Memory
2. VM Migration
3. Node Power Usage
4. Storage

In general, proposals apply cloud systems adaptation to minimise SLA violations and some trade this off with a secondary objective, by recognising that meeting SLAs is not the only business objective for IPs. To achieve this, proposals use different techniques and engage the adaptation at different points. Additionally, the execution complexity of the proposals impact their ability to scale their approach on data centres with thousands of nodes. Therefore, the *secondary objectives*, *adaptation techniques*, *adaptation engagement* and *decision engine architecture* distinguish the various proposals in their adaptation of cloud resource configuration. The remainder of this section will be structured principally according to the adapted resource, and then within each resource following the remaining dimensions in Table 1. Table 2 provides a summary comparison of the literature, in an IaaS context.

VM adaptation - CPU and memory

As core computing resources, CPU and memory adaptation have been widely researched. Many of the proposals scale the infrastructure horizontally by adding new VMs, typically via predefined VM classes [22–32]. While this is simpler to apply, compared to fine grain CPU and memory configuration, it may lead to wastage by over-allocating resources to workloads as well consume more power. In [33] the authors further argued that fine grain CPU and memory configuration reduces the provisioning overhead and mitigates SLA violations. Other proposals, particularly those focusing on maximising revenue, apply fine grain management of VM resources with CPU and memory configurations modified in discrete values using the Xen [34] hypervisor API. In [35–37], the authors utilise Xen's credit-based CPU scheduler to set the CPU share for workloads and in [37, 38], the authors additionally utilise Xen's ability to define the amount of memory assigned to each VM. The life cycle management of workloads can be categorised into two overlapping phases. Admission control [62], which is the decision to accept a new workload if it contributes to the current management objectives and resource adaptation [10], which reconfigures the infrastructure after a state change. Several proposals treat admission control as distinct phase and assume availability of free resources. While this simplifies the approach, it may unnecessarily power on a new node. Alternatively admission control should be used as an opportunity to apply cloud system adaptation and redistribute existing workloads.

Secondary objectives

Some of the proposals focus on reducing power consumption in the data centre [39–43]. While this has a direct impact on an IPs profits, some of the proposals aim to maximise revenue by increasing capacity to service workloads [19, 36, 44, 45]. In contrast, the authors in [23] aim to reduce the cost of using cloud infrastructure to customers on Amazon EC2 [46], by automatically allocating resources based on the current demand. The authors in [24, 47] aim to reduce the complexity in resource provisioning of the Apache Hadoop framework [48], by enabling automated allocation of resources and configuration parameters, and minimise the incurred infrastructure cost. Both approaches attempt to predict the workload behaviour to optimise run time performance, however they differ in their methodology. The authors in [24] used offline training, while the authors in [47] used historical data from past jobs. The latter approach may initially produce lower optimal allocations as it builds job performance history. However, over time this could enable the approach to build better clusters of workload signatures that enable it to make more optimal allocations. Therefore there is a tradeoff between initial performance and time taken to build workload knowledge. While offline approaches can be used to improve the online decision making process by constructing a model of the system behaviour, this has an upfront overhead and is not practical to apply for every application deployed on IaaS.

To reduce power consumption of a node before turning it off, proposals [3, 40] use power management features in modern nodes (DVFS) to scale down both the frequency of the CPU and the voltage used. An alternative approach to DVFS was used in [42], where the authors incorporate the power cost and priority of a VM in the decision of where to add the VM, thus reducing the number of active nodes. Figure 2 shows the components that may get adapted on compute resources.

Table 2 Summary of literature that adapt cloud resources, ordered by the Decision Engine Architecture

Project	Objective					Resource							Tech	Adapt trigger	Arch	Infra	Workload	Setup [#nodes]
	P	SLA	Rev	Cust cost	Whole VM/node	*CPU*	*Mem*	*Migrate*	Disk I/O	DVFS	Node off	ST						
Zheng [31]	x	x			x			x					GA	P	Central	Hom	Generic	Simulation[200]
Zhang [32]		x			x								QT	P	Central	Hom	Multi tier	Simulation
Zuo [71]	x	x			x			x			x		Heuristic	R	Central	Het	Generic	Simulation
Tchana [66]	x		x		x			x					CSP	R	Central	Het	Generic	Private + AWS
Beloglazov [39, 69]	x	x			x			x			x		Heuristic	R	Central	Het	Generic	Simulation [100] [800]
Wesam [33]		x				x	x						Heuristic	R	Central	Het	Multi tier	Xen test bed
Gmach [57]		x						x			x		CT	R	Central	Hom	Generic	Simulation
Fargo [37]	x	x				x	x			x	x		Heuristic	P	Central	Hom	Web App	Xen test bed
Won Choi [70]		x						x					Heuristic	R	Central	Hom	Generic	Linux test bed
Iqbal [62]		x						x			x		Heuristic	R + P	Central	Hom	Generic	Eucalyptus
Roy [28]		x	x		x								CT	P	Central	Hom	Multi tier	NA
Xiangping Bu [38]		x				x	x						RL	R	Central	Hom	Multi tier	Xen test bed
Padala [35]		x				x			x				CT	P	Layered	Hom	Multi tier	Xen test bed
Xu [51]		x				x							CT	P	Central	Hom	Web App	ESX test bed
Jamshidi [52]		x		x	x								CT	R + P	Central	Hom	Web App	Azure
Bodik [23]		x		x	x								CT	P	Central	Hom	Multi tier	Simulation
Lama [24]				x	x								SML + Heuristic	P	Central	Het	Hadoop	ESX test bed
Koehler [47]				x	x							x	Utility	P	Central	Hom	Hadoop	KVM test bed
Kusic [41]	x	x			x	x					x		CT+ Utility+ TS	P	Central	Het	Multi tier	ESX test bed
Zhu [50]		x				x					x		CT + Utility	R	Central	Hom	Web App	HP-UX
Hasan [55]		x				x							Heuristic	R	Central	Hom	Generic	Test bed
Cardosa [42]	x				x								Utility + Heuristic	R	Central	Hom	Generic	ESX test bed
Shen [40]	x	x				x	x	x		x			TS	P	Central	Het	Web App	Xen test bed
Nathuji [49]		x				x							CT	P	Central	Het	Generic	Hyper-V test bed
Malkowski [25]		x			x								CT + Heuristic	P	Central	Hom	Multi tier	Xen test bed
Lim [74]		x										x	CT	R	Central	Hom	Hadoop	Xen test bed
Ali-Eldin [26]		x			x								CT	R + P	Central	Hom	Generic	Simulation
Zhani [27]	x		x		x			x					Heuristic	R	Central	Hom	Generic	Simulation [400]
Han [45]		x	x			x	x		x				Heuristic	R	Central	Hom	Generic	IC Cloud
Han [54]		x		x	x								QT	R	Central	Hom	Generic	Simulation
Gulati [65]		x				x	x	x	x				Greedy Heuristic	R	Central	Het	Generic	ESX test bed

Table 2 Summary of literature that adapt cloud resources, ordered by the Decision Engine Architecture (*Continued*)

Project	Objective						Resource					Tech	Adapt trigger	Arch	Infra	Workload	Setup [#nodes]
	P	SLA	Rev	Cust cost	Whole VM/node	*CPU*	*Mem*	*Migrate*	Disk I/O	DVFS	Node off	ST						
Berral [56]	x	x						x			x		SML	P	Central	Hom	Generic	Simulation [400]
Addis [19]		x	x			x		x		x	x		Utility + Heuristic	R	Central	Het	Multi tier	IBM test bed
Urgaonkar [29]		x			x								QT	R + P	Central	Hom	Multi tier	Xen test bed
Tolia [73]	x	x						x		x			Heuristic	R	Central	Hom	Generic	Xen test bed
Casalicchio [68]		x	x					x					Heuristic	N/A	Central	Hom	Generic	Workstation
Celaya [30]	x	x			x						x		Heuristic	P	Central	Hom	Parellel	Simulation
Addis [53]	x	x	x			x		x		x	x		Utility + Heuristic	P	Hierarch	Het	Multi tier	IBM test bed [7200]
Zhu [67]		x				x		x			x		CT + Heuristic + TS	P	Hierarch	Hom	Web App	ESX/Simulation
Jung [44]	x	x				x		x			x		Heuristic + Utility+TS	P	Central + Hierarch	Het	Multi tier	Xen test bed
Almeida [36]		x	x			x					x		Utility	P	Hierarch	Hom	Multi tier	Simulation
Nguyen Van [22]		x	x		x			x					Utility + CSP	R	Hierarch	Het	Generic	Simulation
Sedaghat [64]	x							x					Heuristic+ P2P	R	Distrib	Het	Generic	Simulation [100,000]
Wuhib [43]	x	x			x			x					Heuristic + P2P + TS	P	Disrib	Hom	Generic	Simulation [160,000]

Legend: CT=Control Theory; RL= Reinforcement learning; CSP= Constrained satisfaction problem; SML= Supervised machine learning; P2P= Peer-to-Peer; QT= Queuing Theory; GA= Genetic Algorithm; TS= Time series; R= Reactive; P= Proactive; Hom=Homogenous; Het= Heterogeneous

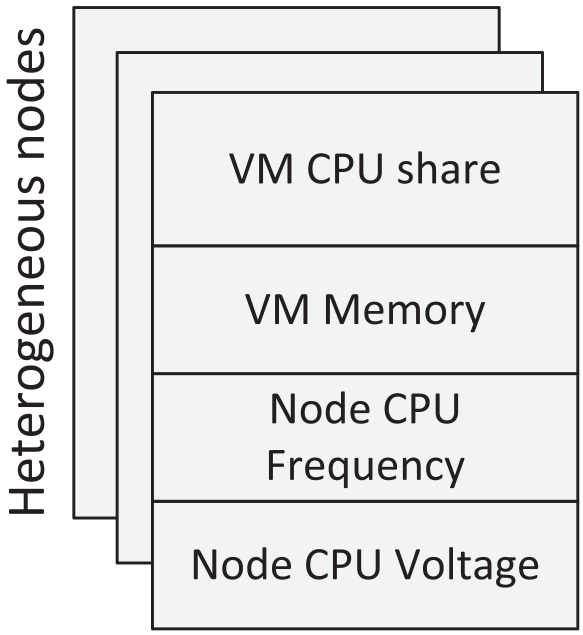

Fig. 2 Compute Resource components

Adaptation technique

A common adaptation technique is Control theory [23, 25, 26, 35, 41, 44, 49, 50], which aims to guarantee system stability by adapting resource configurations at defined intervals. Some of the control theory proposals react to monitored metrics such as CPU utilisation thresholds and workload throughput [50], but most of the proposals surveyed use a proactive mechanism to forecast the future workloads, typically using time series. In contrast, [51] proposed a control theory based approach that utilised Fuzzy Logic to predict short term CPU utilisation. The authors in [52] also used a reactive Fuzzy controller, which can handle conflicting rules. The authors approach attempts to simplify the complexity of setting thresholds by using imprecise thresholds such as *high* and *low* for specifying elasticity rules. However this requires human experts to set multiple values for the approximate thresholds.

Following a heuristic approach, the authors in [53] assign application tiers to nodes, preserving CPU utilisation in each node below a 60 % threshold. A local search optimises the initial allocation, guided by availability guarantees. This architecture results in the heuristic being invoked at three different time-scales, evaluating different adaptation decisions on each time period. In [45], the authors propose a lightweight heuristic based on workload response time, which is defined by the customer. Their approach attempts to satisfy workload response time by incrementally adding CPU and memory resources to the workload. The authors approach requires a deployment portal, but does uniquely account for customer specified constraints, such as budget, when making adaptation decisions. In later work, the authors [54] use an open queueing network, a queuing theory technique, to reduce utilisation cost to customers. Their approach identifies and scales a bottlenecked tier in a multi-tier cloud application. The authors in [32] simplify the cloud application to a typical request queuing model and combine this with binary search.

The heuristic based approach in [55] allows control and adaptation of multiple resources simultaneously, by building groups of resources and performance metrics, which can be adapted based on customer defined events. While the approach uses multiple objective optimisation, the authors did not show empirical evidence of their approach or its ability to scale.

Machine Learning based proposals fall into two categories, supervised learning [24, 56] and unsupervised learning [38]. Given the variability of workloads deployed on cloud infrastructure, Reinforcement Learning (RL) seems promising as it does not rely on pre-constructed models of the controlled infrastructure, by discovering system behaviour online without prior training. The main disadvantage of RL is the online training time, which can be exponential to the size of the explored space, potentially resulting in poor decisions during the learning phase. To combat this challenge, [38] proposed combining RL with a Simplex method to reduce the search space to a smaller valuable set, and then used online CPU and memory utilisation to guide decision making.

Utility based approaches are used to define a measure of usefulness towards a management objective, typically utilising a customer metric like response time as objective to the utility function. Proposals typically build utility frameworks by constructing a performance model of multi-tier applications embedded in an optimisation problem [19, 22, 25, 36, 42], where a utility function expresses satisfaction of each workload towards assigned resources. Utility has also been combined with control theory in [41, 44] to apply a fine grain configuration of CPU and memory, by estimating the benefits of potential adaptations and incorporating a notion of risk. In contrast, the authors in [49] argue that defining multiple levels of QoS, Q-states, beyond the traditional minimum level, is easier for customers to define than utility functions. The challenge with utility based approaches is humans could find it difficult to define the utility functions needed in a complex system.

Adaptation engagement

Proposals adapt CPU and memory configuration either by reacting to a breached metric or by forecasting a metric change. Reactive approaches typically set and monitor a utilisation threshold to CPU and memory [38]. However, setting the optimal threshold is not simple and

typically requires workload knowledge. Some proposals used experimentation [39, 57] to set threshold values. In [45], the authors proposed an alternative approach, where the customer defines the SLA and the IPs set the resource utilisation thresholds. The proposals in [19, 50] react to workload response time, instead of CPU/Memory thresholds, and trigger adaptation to preserve response times to the requested levels. In addition to the challenge of setting the threshold level, reactive approaches risk oscillating system state by reacting too frequently to varying node utilisation. In [55], the authors combat this by using four thresholds and two duration periods to track for how long the threshold has been reached.

Proactive mechanisms are typically time series based, where a sequence of events at defined intervals are analysed to find patterns that can be used to forecast future values. Time series estimators include Auto Regressive Moving Average (ARMA) [58], Smoothing Spline [59], Kalman Filter [60] and Fast Fourier Transform [61]. Proposals that forecast workload arrival rate have an additional challenge to map this to a utilisation forecast. Several proposals tackle this by using an offline phase [24, 37, 44, 49] to build performance models of workloads, which are then used to make adaptation decisions based on online CPU and memory measurements. A disadvantage of the offline approach is it may have a significant overhead and may not cope with dynamic behaviour of some workloads.

A few proposals combine both proactive and reactive approaches in a hybrid approach to engage the adaptation process. In [29], the authors propose a proactive controller that provisions based on peak load seen in the last hour, with a reactive controller for sudden bursts, but it had no ability to scale down CPU/memory resources. In [26, 62], the authors extend this approach and use a reactive controller for scaling up and a proactive controller for scaling down, by removing whole VMs. The authors claim this hybrid approach is able to cope with sudden bursts as well as being able to conserve energy by proactively switching nodes off. The authors did not experiment with gradual scaling of CPU frequency and voltage using techniques such as DVFS, which typically reduces power consumption. The authors in [52] combine both proactive time series analysis and a reactive fuzzy controller. The authors approach attempts to simplify the complexity of setting thresholds by using imprecise thresholds such as *high* and *low* for engaging adaptation. However this requires human experts to set multiple values for the approximate thresholds.

Decision engine architecture

The scalability of a proposal is primarily affected by the execution complexity of the adaptation decision making process. Most proposals are centralised, and memory and CPU adaptation are scheduled across the entire infrastructure. While this gives opportunities for global optimisation, it presents a significant challenge when managing thousands of resources. In [38], the authors used a centralised reinforcement learning engine and the time taken to stabilise performance increased with the size of the managed cluster. The central controller in [41] took significant time to execute the scheduling of 15 nodes, which had 10^9 control options, just to adapt CPU resource - memory configuration was not covered. The centralised engine in [19] was only able to manage 400 nodes with 1000 VMs, when adapting CPU and VM configurations. In later work [53], the authors changed their centralised approach to a hierarchical architecture, resulting in the ability to support 7200 servers with up to 60,000 VMs. In [35], the authors propose an alternative layered approach where each node has a decision engine, with no global controller. While this enabled each node to perform its own allocation, it lost out on the opportunity to redistribute workloads across the data centre infrastructure.

To improve on scalability of the centralised approaches, researchers investigated decentralised approaches such as hierarchical and distributed frameworks. In [44], the authors proposed hierarchical controllers and divided the infrastructure into multiple clusters, where each cluster is managed by a local controller. The hierarchical controllers run at different intervals, with a local cluster controller running more frequently than a global controller. In [36], the authors chose to slice the hierarchy along the operations of the controllers. A Level 1 controller handles VM placement and load balancing, and runs every 30 minutes. A Level 2 controller handles the resources of a node, and runs every few minutes. The challenges with hierarchical approaches include choosing the run time interval of the global controller and the lack of an escalation path between the local and global controllers. Therefore in a sudden burst scenario, a workload may exhibit SLA violations before the global level controller is engaged. An additional challenge is limiting the size of each cluster so it does not become too large for the controller to manage, thus encountering the same challenge as centralised approaches.

Distributed approaches typically focus on VM consolidation and will be covered as part of our analysis of *VM migration*, in the next subsection.

Node adaptation - VM migration

Nodes maybe adapted when a VM's requirements extends beyond the capacity of its hosting node, and it needs to be moved to another node that has the required capacity. Proposals opting to simplify their approach assume the entire infrastructure is homogeneous and has the same computing capability and power consumption, which may lead to suboptimal VM migration decisions. Proposals

that do take the infrastructure capability into account, usually focus on power consumption of nodes. Some proposals assume the ability to capture the relationships between cooperating VMs, and many proposals abstract how workload KPIs, like response time, can be captured. Such proposals are better suited to PaaS, where a deeper integration between the workload and infrastructure is available, and workload metrics and configurations can be made available to the decision-making engine.

Secondary objectives

Proposals apply VM migration primarily to minimise SLA violations, and some proposals aim to reduce power consumption as a secondary objective, by consolidating workloads and switching nodes off, as evidenced by the summary in Table 2.

VM migration adds an overhead and can impact the SLA of the migrated VM and other VMs on the cooperating nodes, yet this is considered acceptable [63] given the opportunities migration can present. In [39], the authors argue that CPU power consumption is the largest contributor to a node's power consumption, thus VM migration can be used to lower power consumption.

Adaptation technique

Beloglazov et al. [39] used a heuristic based adaptation technique and explored three policies, minimisation of migrations (MM), highest potential growth and Random choice, and concluded the MM policy can achieve significant energy savings, compared to non-energy aware policies. The authors argue there is a minor SLA violation trade off, to achieve these energy savings. The MM policy selects VMs with the highest CPU utilisation to migrate to another node. A disadvantage of this approach is it migrates VMs that are already at risk of SLA violation, due to the CPU utilisation, and further increases the risk by adding the cost of live migration. In [43, 64], the authors use a heuristic implemented as a peer-to-peer protocol, enabling nodes to communicate directly without a centralised controller. Two cooperating nodes determine whether to migrate a VM based on the defined objectives. While [43] did not take into account the cost or duration of the conflict before applying the migration, [64] incorporated migration cost into the decision making. In contrast to other proposals, the authors in [44] incorporate the power consumption of the decision engine. Other proposals include VMware's Distributed Resource Scheduler (DRS) [65], which uses greedy hill-climbing to reduce cluster imbalance. DRS incorporates migration cost and benefit, based on workload demands observed in the last hour. Similarly, a greedy heuristic that incorporates migration cost was proposed in [27]. The authors in [31] aimed to reduce the number of nodes used migration as well as reduce VM migration times at the same time, by using a multi-objective Genetic Algorithm based on hybrid group encoding.

In contrast to heuristic based proposals, the authors in [22, 66] uniquely formulated VM migration as a Constrained Satisfaction Problem, taking into account the migration overhead. Tchana et al. [66] combine VM migration with Software migration, by collocating several software applications on the same VM to reduce the number of VMs used. The authors claim significant reduction in power consumption can be achieved by using this approach. However, a limitation of this approach is it requires explicit knowledge of the software being migrated, compared to VM migration, which typically abstracts the software within a VM.

Similar to [44], the authors in [22] used utility as measure the satisfaction of each managed workload and a global decision module prioritises decisions that maximise a global utility.

A less common adaptation technique for VM migration is time series analysis, proposed in [40], to predict contention for resources through a Fast Fourier Transform algorithm. The authors engaged the migration before it is needed, and minimise cost by only migrating when the resource contention is predicted to last beyond a defined period of time. In a multi-adaptation technique, Zhu et al. [67] experimented with integrating a fuzzy logic controller with a trace-based controller, arguing the integration resulted in better resource allocation compared to the non-integrated approach.

Proposals typically do not cover cloud system adaptation during admission control phase, assuming availability, however the authors in [68] migrate VMs during the admission control phase, by using a heuristic solution based on hill climbing search techniques.

Adaptation engagement

To engage VM migration, the authors in [39] used a two-threshold reactive approach. The low threshold aims to lower power consumption and triggers VMs to be migrated off a node, which is then set to sleep mode. The high threshold aims to meet SLA and triggers migration of a VM with the highest utilisation to another node. The double threshold approach takes a snapshot in time of the current CPU utilisation and thus can suffer from false positives caused by workload utilisation peaks and troughs. In later work, Beloglazov et al. [69] proposed an adaptive auto-adjustment of the upper threshold, based on statistical analysis of historical data collected during the lifetime of VMs, combating statistical outliers in their earlier approach. Similarly, the authors in [70] proposed a dynamic threshold approach that finds and adjusts thresholds at runtime. Zuo et al. [71] also use an adaptive threshold. The authors monitor 3 metrics: number of resource requests, resource service capacity and resource

service strength, and propose a dynamic weighted evaluation, dividing the resource load into three states including Overload, Normal and Idle.

Proactive approaches [40, 44] start the VM migration before the conflict occurs, to avoid sustained service degradation from the cost of the migration. In [44], the authors proposed performing a cost and benefit analysis before applying migration, and only invoked a migration if the benefit outweighed the cost of the migration.

Decision engine architecture

Most proposals are centralised and VM migration is scheduled across the entire managed infrastructure. While this gives opportunities for global optimisation, it presents a significant challenge when managing thousands of resources. Despite its name, VMware's Distributed Resource Scheduler [65] uses a centralised load balancing approach to engaging VM migration, so it suffers the same scalability challenges of centralised approaches proposed in academia. Zheng et al. [31] aim to reduce the number of nodes used in migration as well as reduce VM migration times, by using a multi-objective Genetic Algorithm based on hybrid group encoding. The approach used a centralised controller and limited simulation to only 200 nodes. Additionally, the authors did not explore the time complexity of their Genetic algorithm.

To improve the scalability of a centralised approach, researchers investigated hierarchical and distributed frameworks.

Hierarchical approaches tackle the scalability challenge by reducing the frequency of engaging the global controller. The hierarchical approach in [22] used a local decision module for each application and a global decision module. Application satisfaction is regularly measured using a utility function and communicated to the global module, which prioritises requests to satisfy a global utility. An alternative approach was proposed in [67], where an additional Level 3 (L3) controller was used to manage multiple clusters operating at seconds (L1), minutes (L2) and days (L3) intervals. However the authors did not explore the scalability of their approach.

For a distributed and decentralised approach to managing the data centre, the authors in [43, 64] proposed a peer-to-peer protocol that enables nodes to communicate directly without a centralised controller. A periodic node discovery service enables nodes to find new neighbouring nodes to communicate with. On each round of the protocol, two cooperating nodes determine to migrate a VM based on defined objectives. The distributed approaches in [43, 64] are used to redistribute the load across the cluster as well consolidate VMs. Using simulation, the authors claim their approaches can manage more than 100,000 nodes. A challenge with distributed approaches is the lack of a global view of the infrastructure, which impact the ability to reach a globally optimal solution. Additionally, gossip approaches consume considerable bandwidth to implement propagation of node state across the entire data centre infrastructure.

Node adaptation - power

Proposals adapt a node's power configuration to reduce operational costs for IPs. Proposals may use a policy in the VM placement phase to use the most energy-efficient nodes first, apply power management features on a node and eventually migrate VMs and switch the node to a sleep state. To reduce the power consumption of a node before turning it off, some proposals use Dynamic Voltage and Frequency Scaling (DVFS), which is a framework to change the frequency and/or operating voltage of nodes based on system performance requirements. To utilises DVFS, it needs to be supported by both the node and OS. Modern processors typically support multiple levels of frequency/voltage, which can be selected through the OS. Proposals typically select a frequency/voltage level that reduces the node capability and minimises impact to workloads, applying a trade-off between workload performance and power consumption.

An alternative approach to DVFS was used in [42], where the authors incorporate the power cost and priority of a VM in the decision of where to add the VM, thus reducing the number of active nodes. While DVFS has been widely deployed and proven to reduce power consumption, the authors in [72] argue that DVFS can have an impact on mult-tier application performance. They propose a solution to minimise the impact, by increasing the DVFS adjustment frequency and predicting the workload burst cycle.

Adaptation techniques

Proposals differ in their approach to reducing node power consumption, with some researchers opting to migrate VMs and set the node to a sleep state [39, 44, 56, 67], compared to incrementally reduce power consumption by using DVFS.

Beloglazov et al. [39] propose a heuristic to consolidate workloads and switch nodes into a sleep state, arguing that an idle node can consume 70 % of the power consumed by a node running at the full CPU speed. Their approach was able to switch a node to sleep mode within 20 sec. However, the authors did not discuss how nodes can be be woken up from sleep mode if more nodes are required to service requests. Other proposals that do not utilise DVFS include the gossip based protocol in [43], which places new VM requests on the highest loaded node capable of hosting it. VMs are redistributed by moving a VM from a lower loaded to a higher loaded node if it can be hosted. In [64], the authors take into account power consumption in the decision making. The authors in [31] aim to

reduce the number of nodes used in migration as well as reduce VM migration times at the same time, by using a multi-objective Genetic Algorithm based on hybrid group encoding.

In contrast, [19, 40, 73] utilise DFVS to gradually reduce power consumption and switch nodes to a sleep state. The authors in [40] use time series, while [19, 73] use a heuristic to adjust DVFS.

Adaptation engagement

To adapt power configuration, reactive approaches [56, 69] use a low threshold for CPU utilisation to switch nodes to sleep state. In contrast, proactive approaches predict workload utilisation and switch nodes to sleep state at the predicted time intervals. In [41], the authors used a Kalman filter to predict the number of requests. VM capability and power consumption were captured offline, by measuring the average response times achieved when different CPU shares were assigned to the VM. The authors modelled risk in the decision making to cater for the cost of switching nodes on and off, arguing this reduces SLA violations considerably compared to a non risk aware controller. Core to this argument is SLA violations, or opportunity cost, in having to power on a node. However, with commoditisation of Solid state storage (SSD), which offers significant boot performance compared to Hard disk drives, many servers use SSD to boot the operating system. The authors previous conclusions may need to be revisited to re-evaluate whether more nodes using SSD can be left in switched off mode and switched on nearer to the time they are needed. Similarly, [40, 53] proactively adjust the node frequency and eventually switch the node to sleep state.

Decision engine architecture

To consolidate VMs, proposals migrate VMs between nodes by searching for suitable nodes that can take additional VMs without violating another management objective. As the scalability of migrating VMs was covered in the Node Adaptation subsection, here we focus on the approaches to managing power reduction at large scale.

In the gossip based protocol in [43], the authors experimentally assessed the power consumption of the proposal, by measuring the number of active servers. However they do not incorporate an explicit notion of power cost in their policy. In [64], when two nodes communicate they attempt to consolidate all VMs onto one peer and the released peer is set into the power saving mode. If the VMs cannot be entirely consolidated onto one node, the protocol attempts to redistribute the load across the two nodes, taking into account power consumption and migration cost.

Proposals utilising DVFS to lower power consumption typically use a centralised decision engine [19, 40, 73], although Addis et al. proposed a hierarchical architecture in later work [53].

Storage adaptation

Cloud storage adaptation can be applied to both I/O access and the storage itself, although this area is less covered compared to other cloud resources.

Adaptation technique

Control theory is used by researchers to adapt different levels of the storage stack. Padala et al. [35] used an application controller to determine disk I/O resources needed at the node level. While the approach can apply service differentiation, it over-allocates disk I/O bandwidth when these are available, which potentially increases power consumption. In [74], the authors used control theory to adapt the central storage tier, focusing on the Hadoop Distributed File System, from a customer perspective. Offline profiling data was used to build the transfer function into the constructed system model, combining this with online CPU metrics from the storage node.

Another technique used to adapt I/O access is supervised machine learning, proposed in [24], focusing on automated provisioning of Hadoop jobs.

Adaptation engagement

To engage storage adaptation, the approach in [74] reacts to the CPU utilisation of the storage node. The first controller adds and removes storage nodes and a second controller rebalances data across the new set of storage nodes. To ascertain some of the thresholds, the authors used offline experimentation with Cloudstone benchmark. In contrast, the proactive approach in [24] used a two phase approach, where phase one is offline and builds a prediction model using past job information and a k-medoid clustering and support vector machine. Phase two is online and uses a staging area to obtain a resource utilisation signature for newly submitted jobs. These signatures are then matched to the offline constructed data for the decision making process. The authors assume availability of job history information, and the staging area imposes additional costs that have to be met by either the IPs or end users. In contrast, the authors in [35] used a second order ARMA model, taking into account two previous control intervals to predict workload performance, by using response time as the performance metric.

Decision engine architecture

The scalability of centralised approaches is typically problematic [43], however proposals in [24, 35] do not migrate VMs to resolve contention, therefore do not require a global view of the infrastructure. This places less emphasis on the scalability of their approaches.

The centralised proposal in [74] needs to rebalance data when nodes join and leave a storage cluster. During the rebalancing phase, no additional adaptation can be carried out. The impact of this limitation will increase as the number of nodes in the cluster increase, thus limiting the applicability of the approach.

Open research challenges

While there has been considerable research in adaptation of resource configuration, there are several open challenges. Based on our analysis, the following are open challenges in cloud systems adaptation, in an IaaS context:

1. Many of the proposals in the literature focus on managing web/multi tired applications, as can be seen on Table 2, and use application metrics as input into the decision making process. Other proposals attempt to manage generic workload types and typically utilise threshold based approaches to trigger adaptation. A potentially better approach is to characterise the *workload type* and engage adaptation that takes into account the workload type. Several projects attmept to analyse and characterise cloud workloads. Analysis of public Google traces [21, 75–77] has shown variance in the resources utilised and the duration of cloud tasks, making popular simplifications such as being able to slot workloads on resources unsuitable [21]. Additionally, users typically overestimate resources reservations, leading to significant wastage [76]. Some existing approaches aim to predict future workloads using classical prediction models such as ARMA [28], a linear regression model [78] and a hybrid model tuned to bursty web traffic [32, 79]. Other characterisation approaches aim to predict workload resource utilisation, by identifying a feature of the workload. The authors in [80] match applications with appropriate VM types by defining application profiles, which are manually extracted from workflow logs. The authors in [77] classify tasks based on resource utilisation and the authors in [81, 82] extract utilisation usage signatures. The authors in [31] use a load predictor that clusters historical resource utilisation, and select the cluster set with the highest similarity as a training sample into a Neural Network. However, these approaches simplify the impact of colocating VMs, which can lead to significant performance overhead [83, 84]. The authors in [85] tackle colocation interference and perform four parallel classifications on each application to evaluate the impact of vertical and horizontal scale, server configuration, and the impact of colocating applications. However this approach needs specific knowledge of the application in order to profile and classify. Based on the current state of art, there is no generic non application aware online classification of workload types, which are typically deployed on IaaS. A generic mechanism to predict whether the workload is a user desktop, web server, file server or batch job, can enable the decision engine to adapt resource configuration in an optimal way for the workload type. This can potentially allow the workload to complete quicker or conserve resource otherwise not utilised by the workload, and enable colocation of VMs in a way that does not introduce interference.
2. *Offline profiling* and staging area approaches are typically used to experimentally derive workload resource requirements. However this has an upfront overhead and is not practical to apply for every application that will be deployed on a IaaS. Several proposals have attempted online profiling and/or monitoring of workloads, however these typically require explicit knowledge of the application [85], or an output from the VM such as latency or response time [82, 86, 87], which is typically not available to IPs. More research is need into application agnostic mechanisms that can extract workload resource requirements, and impact of adaptation, dynamically at run time.
3. *Scalability* of computing systems is an understood challenge in traditional enterprise infrastructure. However cloud environments magnify this challenge due to the larger size and heterogeneity of infrastructure used in cloud data centres. Table 2 shows a summary of the proposals in the literature, including the number of nodes each proposal attempted to manage. This shows many of the proposals do not explore the scalability of their approach and typically implement a centralised decision engine. Some of the proposals explore scalability of managing several thousand nodes, which is still significantly below many modern data centres, which can house more than 100,000 nodes [88]. Proposals that explored scalability capable of managing modern data centres tend to implement a distributed decision engine. However these approaches trade off ability to manage a large infrastructure with a reduction in optimal resource allocation. Additionally, these approaches consume considerable bandwidth for the nodes to communicate directly across the entire infrastructure. More research is required to demonstrate robust and practical application of distributed approaches, which can achieve similar level of optimal allocation as centralised approaches.

Conclusion

This paper presented a definition of cloud systems adaptation and a classification of the key features. We analysed the literature and highlighted approaches and techniques used to enable adaptation of cloud resource configuration.

Workload management on IaaS entails controlling the admission of new workloads and periodically adapting resource configuration to achieve a management objective. Proposals in the literature aim to minimise SLA violations and some trade this off with a secondary objective, such as reducing power consumption or maximising IP revenue. To achieve these objectives, several adaptation techniques have been used. The architecture of the decision engine has a significant impact on the scalability of a proposal, with centralised approaches not being able to scale on large data centres. While there has been considerable research, we have highlighted several open challenges that are worthy of further investigation.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

AH carried out the survey of the literature, drafted the manuscript and identified open research challenges. NP and RS provided insight and guidance in developing the structure and dimensions for the literature classification, critically reviewed the paper and suggested additional papers to investigate. All authors read and approve the final manuscript.

Authors' information

Abdul Hummaida is a PhD candidate in the School of Computer Science, University of Manchester, UK. He completed his bachelor's degree on Software Engineering from the School of Computer Science, University of Manchester, UK. He is currently working on scalability of cloud management systems. His research interests include cloud computing, autonomic computing and workload management. He is also a Director of Software Engineering at Appsense.

Norman Paton is a Professor of Computer Science at the University of Manchester, where he co-leads the Information Management Group. He works principally on databases and distributed information management. Current research interests include pay-as-you-go data integration, sensor query processing and infrastructures for adaptive systems development. He also works on genome data management, in particular exploring the use of data integration techniques for making better use of experimental and derived data in systems biology. He has been an investigator on over 40 research grants from the UK research councils, the EU and industry, and has published around 200 refereed articles.

Rizos Sakellariou is with the School of Computer Science at the University of Manchester, UK, where he carries out research in the broad area of parallel and distributed systems while at the same time he enjoys teaching and never stops to be amazed by university politics. He has published over 100 research papers in the area.

Received: 4 November 2015 Accepted: 20 May 2016
Published online: 24 May 2016

References

1. Jassy A Amazon Web Services Summit. https://aws.amazon.com/summits/san-francisco/. Accessed May 2016
2. Galante G, Bona LCEd (2012) A survey on cloud computing elasticity. In: Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing, UCC '12. IEEE Computer Society, Washington, DC, USA. pp 263–270
3. Beloglazov A, Buyya R, Lee YC, Zomaya A (2011) A taxonomy and survey of energy-efficient data centers and cloud computing systems. Adv Comput 82:47–111
4. Botran TL, Miguel-Alonso J, Lozano JA (2014) Auto-scaling techniques for elastic applications in cloud environments. J Grid Comput 12(4):559–592
5. Najjar A, Serpaggi X, Gravier C, Boissier O (2014) Survey of Elasticity Management Solutions in Cloud Computing. In: Computer Communications and Networks. Springer, 236 Gray's Inn Road, Floor 6, London WC1X 8HB, UK. pp 235–263
6. Jennings B, Stadler R (2015) Resource management in clouds: Survey and research challenges. J Netw Syst Manag 23(3):567–619
7. Coutinho EF, Carvalho Sousa FR, Rego PAL, Gomes DG, Souza JN (2014) Elasticity in cloud computing: a survey. Ann Telecommun - annales des télécommunications 70(7):289–309. doi:10.1007/s12243-014-0450-7
8. Mann ZA (2015) Allocation of virtual machines in cloud data centers—a survey of problem models and optimization algorithms. ACM Comput Surv 48(1):11–11134. doi:10.1145/2797211
9. Singh S, Chana I (2015) Qos-aware autonomic resource management in cloud computing: A systematic review. ACM Comput Surv 48(3):42–14246. doi:10.1145/2843889
10. Faniyi F, Bahsoon R (2015) A systematic review of service level management in the cloud. ACM Comput Surv 48(3):43–14327. doi:10.1145/2843890
11. Naskos A, Gounaris A, Sioutas S (2016) Cloud Elasticity: A Survey. In: Karydis I, Sioutas S, Triantafillou P, Tsoumakos D (eds). Algorithmic Aspects of Cloud Computing: First International Workshop, ALGOCLOUD 2015, Patras, Greece, September 14-15, 2015. Revised Selected Papers. Springer, Cham. pp 151–167
12. Mohamaddiah MH, Abdullah A, Subramaniam S, Hussin M (2014) A survey on resource allocation and monitoring in cloud computing. Int J Mach Learn Comput 4(1):31–38
13. Singh S, Chana I (2016) A survey on resource scheduling in cloud computing: Issues and challenges. J Grid Comput 14(2):1–48
14. Murch R (2004) Autonomic Computing. IBM Press, 1 New Orchard Rd, Armonk, NY 10504, US
15. NIST Sp 800-145: Definition of cloud computing. Technical report, NIST, 100 Bureau Drive, Gaithersburg, USA (Sep 2011). NIST. http://csrc.nist.gov/publications/PubsSPs.html. Accessed May 2016
16. Herbst NR, Kounev S, Reussner R (2013) Elasticity in cloud computing: What it is, and what it is not. In: 10th International Conference on Autonomic Computing. pp 23–27
17. Maurer M, Brandic I, Sakellariou R (2013) Adaptive resource configuration for cloud infrastructure management. Futur Gener Comput Syst 29(2):472–487
18. Magklis G, Semeraro G, Albonesi DH, Dropsho SG, Dwarkadas S, Scott ML (2003) Dynamic frequency and voltage scaling for a multiple-clock-domain microprocessor. IEEE Micro 23:62–68
19. Addis B, Ardagna D, Panicucci B, Zhang L (2010) Autonomic management of cloud service centers with availability guarantees. In: 2010 IEEE 3rd International Conference on Cloud Computing. IEEE, Washington, DC, USA. pp 220–227
20. Sedaghat M, Hernández-Rodriguez F, Elmroth E (2014) Autonomic resource allocation for cloud data centers: A peer to peer approach. In: IEEE International Conference on Cloud and Autonomic Computing. IEEE, Washington, DC, USA. pp 131–140
21. Reiss C, Tumanov A, Ganger GR, Katz RH, Kozuch MA (2012) Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In: Proceedings of the Third ACM Symposium on Cloud Computing, SoCC '12. ACM, New York, NY, USA. pp 7–1713. doi:10.1145/2391229.2391236 http://doi.acm.org/10.1145/2391229.2391236
22. Van HN, Tran FD, Menaud J-M (2009) Sla-aware virtual resource management for cloud infrastructures. In: IEEE International Conference on Computer and Information Technology. IEEE, Washington, DC, USA 2:357-362
23. Bodík P, Griffith R, Sutton C, Fox A, Jordan M, Patterson D (2009) Statistical machine learning makes automatic control practical for internet datacenters. In: Proceedings of the 2009 Conference on Hot Topics in Cloud Computing, HotCloud'09. USENIX Association, Berkeley, CA, USA
24. Lama P, Zhou X (2012) Aroma: Automated resource allocation and configuration of mapreduce environment in the cloud. In: Proceedings of

the 9th International Conference on Autonomic Computing, ICAC '12. ACM, New York, NY, USA. pp 63–72
25. Malkowski SJ, Hedwig M, Li J, Pu C, Neumann D (2011) Automated control for elastic n-tier workloads based on empirical modeling. In: Proceedings of the 8th ACM International Conference on Autonomic Computing, ICAC '11. ACM, New York, NY, USA. pp 131–140
26. Ali-Eldin A, Tordsson J, Elmroth E (2012) An adaptive hybrid elasticity controller for cloud infrastructures. In: 2012 IEEE Network Operations and Management Symposium. IEEE, Washington, DC, USA. pp 204–212
27. Zhani MF, Cheriton DR, Zhang Q, Simon G, Boutaba R (2013) Vdc planner: Dynamic migration-aware virtual data center embedding for clouds. In: IEEE International Symposium on Integrated Network Management. IEEE, Washington, DC, USA. pp 18–25
28. Roy N, Dubey A, Gokhale A (2011) Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting. In: IEEE International Conference on Cloud Computing. pp 500–507
29. Urgaonkar B, Shenoy P, Chandra A, Goyal P, Wood T (2008) Agile dynamic provisioning of multi-tier internet applications. ACM Transactions on Autonomous and Adaptive Systems 3(1)
30. Celaya J, Sakellariou R (2014) An adaptive policy to minimize energy and sla violations of parallel jobs on the cloud. In: IEEE/ACM 7th International Conference on Utility and Cloud Computing. IEEE, Washington, DC, USA. pp 507–508
31. Zheng S, Zhu G, Zhang J, Feng W (2015) Towards an adaptive human-centric computing resource management framework based on resource prediction and multi-objective genetic algorithm. Multimedia Tools and Applications:1–18
32. Zhang Q, Chen H, Shen Y, Ma S, Lu H (2016) Optimization of virtual resource management for cloud applications to cope with traffic burst. Futur Gener Comput Syst 58:42–55. doi:10.1016/j.future.2015.12.011
33. Dawoud W, Takouna I, Meinel C (2011) Elastic virtual machine for fine-grained cloud resource provisioning. Glob Trends Comput Commun Syst 269:11–25
34. Citrix:Xen. http://www.xenserver.org. Accessed May 2016
35. Padala P, Hou K-Y, Shin KG, Zhu X, Uysal M, Wang Z, Singhal S, Merchant A (2009) Automated control of multiple virtualized resources. In: Proceedings of the 4th ACM European Conference on Computer Systems, EuroSys '09. ACM, New York, NY, USA. pp 13–26
36. Almeida J, Almeida V, Ardagna D, Cunha Í, Francalanci C, Trubian M (2010) Joint admission control and resource allocation in virtualized servers. J Parallel Distrib Comput 70:344–362
37. Fargo F, Tunc C, Al-Nashif Y, Akoglu A, Hariri S (2014) Autonomic workload and resource management of cloud computing services. In: IEEE International Conference on Cloud and Autonomic Computing. IEEE, Washington, DC, USA. pp 101–110
38. Bu X, Rao J, Xu C-Z (2011) Model-free learning approach for coordinated configuration of virtual machines and appliances. In: 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems. IEEE, Washington, DC, USA. pp 12–21
39. Beloglazov A, Abawajyb J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Futur Gener Comput Syst 28:755–768
40. 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. ACM, New York, NY, USA. pp 5–1514
41. Kusic D, Kephart JO, Hanson JE, Kandasamy N, Jiang G (2008) Power and performance management of virtualized computing environments via lookahead control. In: Autonomic Computing ICAC. IEEE, Washington, DC, USA. pp 3–23
42. Cardosa M, Korupolu MR, Singh A (2009) Shares and utilities based power consolidation in virtualized server environments. In: 11th IFIP/IEEE International Conference on Symposium on Integrated Network Management. pp 327–334
43. Wuhib F, Stadler R, Spreitzer M (2012) Dynamic resource allocation with management objectives: implementation for an openstack cloud. IEEE Trans Netw Serv Manag 9(2):213–225
44. Jung G, Hiltunen MA, Joshi KR, Schlichting RD, Pu C (2010) Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures. In: International Conference on Distributed Computing Systems. IEEE, Washington, DC, USA. pp 62–73
45. Han R, Guo L, Ghanem MM, Guo Y (2012) Lightweight resource scaling for cloud applications. In: International Symposium on Cluster, Cloud and Grid Computing. IEEE, Washington, DC, USA. pp 644–651
46. Amazon:AWS. http://aws.amazon.com/ec2/. Accessed May 2016
47. Koehler M (2014) An adaptive framework for utility-based optimization of scientific applications in the cloud. J Cloud Comput Adv Syst App 3:4
48. Apache:Hadoop. http://hadoop.apache.org. Accessed May 2016
49. 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. ACM, New York, NY, USA. pp 237–250
50. Zhu X, Wang Z, Singhal S (2006) Utility-Driven Workload Management Using Nested Control Design. In: American Control Conference. IEEE, Washington, DC, USA
51. Xu J, Zhao M, Fortes J, Carpenter R, Yousif M (2008) Autonomic resource management in virtualized data centers using fuzzy logic-based approaches. Clust Comput 11:213–227
52. Jamshidi P, Ahmad A, Pahl C (2014) Autonomic resource provisioning for cloud-based software. In: Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2014. ACM, New York, NY, USA. pp 95–104
53. Addis B, Ardagna D, Panicucci B, Squillante MS, Zhang L (2013) A hierarchical approach for the resource management of very large cloud platforms. IEEE Trans Dependable Secure Comput 10:253–272
54. Han R, Ghanem MM, Guo L, Guo Y, Osmond M (2014) Enabling cost-aware and adaptive elasticity of multi-tier cloud applications. Futur Gener Comput Syst 32:82–98
55. Hasan MZ, Magana E, Clemm A, Tucker L, Gudreddi SLD (2012) Integrated and autonomic cloud resource scaling. In: Network Operations and Management Symposium. IEEE, Washington, DC, USA. pp 1327–1334
56. Berral JL, Goiri In, Nou R, Julià F, Guitart J, Gavaldà R, Torres J (2010) Towards energy-aware scheduling in data centers using machine learning. In: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10. ACM, New York, NY, USA. pp 215–224
57. Gmach D, Rolia J, Cherkasova L, Kemper A (2009) Resource pool management: Reactive versus proactive or lets be friends. Computer Networks: The International Journal of Computer and Telecommunications Networking 53:2905–2922
58. Box GEP, Jenkins GM, Reinsel GC (2008) Time Series Analysis: Forecasting and Control. 4th ed.. John Wiley & Sons Inc, 111 River Street Hoboken, NJ 07030-5774
59. Boor CD (2001) A Practical Guide to Splines. 1st ed.. Springer, 233 Spring Street, New York, NY 10013-1578, USA
60. Kalman RE (1960) A new approach to linear filtering and prediction problems. J Fluids Eng 82:35–45
61. Loan CV (1987) Computational Frameworks for the Fast Fourier Transform. Society for Industrial and Applied Mathematics, 3600 Market Street, 6th Floor, Philadelphia, PA
62. Iqbal W, Dailey MN, Carrera D, Janecek P (2011) Adaptive resource provisioning for read intensive multi-tier applications in the cloud. Futur Gener Comput Syst 26:871–879
63. Voorsluys W, Broberg J, Venugopal S, Buyya R (2009) Cost of virtual machine live migration in clouds: A performance evaluation. In: Proceedings of the 1st International Conference on Cloud Computing, CloudCom '09. Springer, Berlin, Heidelberg. pp 254–265
64. Sedaghat M, Hernández-Rodriguez F, Elmroth E, Girdzijauskas S (2014) Divide the task, multiply the outcome: Cooperative vm consolidation. In: IEEE International Conference on Cloud Computing Technology and Science. IEEE, Washington, DC, USA. pp 300–305
65. Gulati A, Shanmuganathan G, Holler A, Ahmad I (2011) Cloud-scale resource management: Challenges and techniques. In: Proceedings of the 3rd USENIX Conference on Hot Topics in Cloud Computing, HotCloud'11. USENIX Association, Berkeley, CA, USA. pp 3–3
66. Tchana A, Palma ND, Safieddine I, Hagimont D, Diot B, Vuillerme N (2015) Euro-par 2015: Parallel processing: 21st international conference on parallel and distributed computing, Vienna, Austria, August 24-28, 2015, proceedings: 305–316

67. Zhu X, Young D, Watson BJ, Wang Z, Rolia J, Singhal S, McKee B, Hyser C, Gmach D, Gardner R, Christian T, Cherkasova L (2008) 1000 Islands: Integrated Capacity and Workload Management for the Next Generation Data Center. In: International Conference on Autonomic Computing. IEEE, Washington, DC, USA. pp 172–181
68. Casalicchio E, Menascé DA, Aldhalaan A (2013) Autonomic resource provisioning in cloud systems with availability goals. In: Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference, CAC '13. ACM, New York, NY, USA. pp 11–110
69. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr Comput Pract Experience 24:1397–1420
70. Choi HW, Kwak H, Sohn A, Chung K (2008) Autonomous learning for efficient resource utilization of dynamic vm migration. In: Proceedings of the 22Nd Annual International Conference on Supercomputing, ICS '08. ACM, New York, NY, USA. pp 185–194
71. Zuo L, Shu L, Dong S, Zhu C, Zhou Z (2016) Dynamically weigh0ted load evaluation method based on self-adaptive threshold in cloud computing. Mob Networks Appl :1–15. doi:10.1007/s11036-016-0679-7
72. Wang Q, Kanemasa Y, Li J, Lai CA, Matsubara M, Pu C (2013) Impact of dvfs on n-tier application performance. In: Proceedings of the First ACM SIGOPS Conference on Timely Results in Operating Systems, TRIOS '13. ACM, New York, NY, USA. pp 51–516
73. Tolia N, Wang Z, Marwah M, Bash C, Ranganathan P, Zhu X (2008) Delivering energy proportionality with non energy-proportional systems: Optimizing the ensemble. In: Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08. USENIX Association, Berkeley, CA, USA. pp 2–2
74. Lim HC, Babu S, Chase JS (2010) Automated control for elastic storage. In: Proceedings of the 7th International Conference on Autonomic Computing, ICAC '10. ACM, New York, NY, USA. pp 1–10
75. Di S, Kondo D, Cappello F (2013) Characterizing cloud applications on a google data center. In: Parallel Processing (ICPP), 2013 42nd International Conference On. IEEE, Washington, DC, USA. pp 468–473
76. Moreno IS, Garraghan P, Townend P, Xu J (2013) An approach for characterizing workloads in google cloud to derive realistic resource utilization models. In: Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium On. IEEE, Washington, DC, USA. pp 49–60
77. Zhang Q, Zhani MF, Boutaba R, Hellerstein JL (2014) Dynamic heterogeneity-aware resource provisioning in the cloud. IEEE Transactions on Cloud Computing 2(1):14–28. doi:10.1109/TCC.2014.2306427
78. Yang J, Liu C, Shang Y, Cheng B, Mao Z, Liu C, Niu L, Chen J (2013) A cost-aware auto-scaling approach using the workload prediction in service clouds. Inf Syst Front 16(1):7–18. doi:10.1007/s10796-013-9459-0
79. Liu C, Shang Y, Duan L, Chen S, Liu C, Chen J (2015) Optimizing Workload Category for Adaptive Workload Prediction in Service Clouds. In: Barros A, Grigori D, Narendra CN, Dam KH (eds). Service-Oriented Computing: 13th International Conference, ICSOC 2015, Goa, India, November 16-19, 2015, Proceedings. Springer, Berlin, Heidelberg. pp 87–104
80. Chard R, Chard K, Bubendorfer K, Lacinski L, Madduri R, Foster I (2015) Cost-aware elastic cloud provisioning for scientific workloads. In: Cloud Computing (CLOUD), 2015 IEEE 8th International Conference On. IEEE, Washington, DC, USA. pp 971–974
81. Gong Z, Gu X, Wilkes J (2010) Press: Predictive elastic resource scaling for cloud systems. In: Network and Service Management (CNSM), 2010 International Conference On. IEEE, Washington, DC, USA. pp 9–16
82. Zhang L, Zhang Y, Jamshidi P, Xu L, Pahl C (2015) Service workload patterns for qos-driven cloud resource management. J Cloud Comput 4(1):1–21. doi:10.1186/s13677-015-0048-2
83. Xu F, Liu F, Jin H, Vasilakos AV (2014) Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions. Proc IEEE 102(1):11–31. doi:10.1109/JPROC.2013.2287711
84. Feller E, Ramakrishnan L, Morin C (2015) Performance and energy efficiency of big data applications in cloud environments: A hadoop case study. J Parallel Distrib Comput 79–80:80–89. doi:10.1016/j.jpdc.2015.01.001. Special Issue on Scalable Systems for Big Data Management and Analytics
85. Delimitrou C, Kozyrakis C (2014) Quasar: Resource-efficient and qos-aware cluster management. In: Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '14. ACM, New York, NY, USA. pp 127–144. doi:10.1145/2541940.2541941 http://doi.acm.org/10.1145/2541940.2541941
86. Tsoumakos D, Konstantinou I, Boumpouka C, Sioutas S, Koziris N (2013) Automated, elastic resource provisioning for NoSQL clusters using TIRAMOLA. In: Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium On. IEEE, Washington, DC, USA. pp 34–41
87. Naskos A, Stachtiari E, Gounaris A, Katsaros P, Tsoumakos D, Konstantinou I, Sioutas S (2015) Dependable horizontal scaling based on probabilistic model checking. In: Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium On. IEEE, Washington, DC, USA. pp 31–40
88. Miller R Data Center Knowledge. http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-%web-servers/. Accessed May 2016