J Internet Serv Appl (2012) 3:67–75
DOI 10.1007/s13174-011-0053-8

SI: FOME - THE FUTURE OF MIDDLEWARE

Cloud management

Nigel Cook · Dejan Milojicic · Vanish Talwar

Received: 17 November 2011 / Accepted: 23 November 2011 / Published online: 8 December 2011

Abstract Cloud computing offers a number of benefits, such as elasticity with the perception of unlimited resources, self-service, on-demand, automation, etc. However, these benefits create new requirements for management of cloud computing. On the back-end, economic limitations dictate careful consolidation of servers with clear sustainability analysis; managed levels of abstractions are higher (from hardware, to VMs, to services); and reliability, availability, and supportability are built into higher levels of systems and services. On the client-side, cloud services have to be easy to use/manage, perform well, and be reliable. On both sides, geographical distribution and its implications on business continuity is a rule rather than exception; scalability is built-in by design; and QoS is still being defined. In this paper, we discuss new requirements and approaches to cloud management. We present a few examples of cloud management for private, public, and HPC clouds. Based on these, we derive conclusions about manageability of current platforms and then make predictions about the research challenges of future cloud management. We expect these findings to help designers of next generation hardware and software platforms to develop more manageable systems and solutions.

Keywords Cloud services · Service management · Middleware · Heterogeneity · Integration · Scalability · Service level agreements

N. Cook
Hewlett Packard, Littleton, CO, USA
e-mail: nigel.cook@hp.com

D. Milojicic (✉) · V. Talwar
Hewlett Packard, Laboratories, Palo Alto, CA, USA
e-mail: dejan.milojicic@hp.com

V. Talwar
e-mail: vanish.talwar@hp.com

Abbreviations

QoS:	Quality of Service;
SLA:	service level agreements;
IT:	Information Technology;
DevOps:	Development Operations;
NVRAM:	Nonvolatile Random Access Memory;
AWS:	Amazon Web Services;
VM:	virtual machines;
CAPEX/OPEX:	Capital/Operational Expenditure;
SSD:	Solid State Disks;
WBEM:	Web-Based Enterprise Management

1 Introduction

Cloud computing is an emerging paradigm, with growing popularity and adoption [1]. Cloud providers host shared servers, and deliver computing, storage, and software to end-consumers as a service. Both Gartner and IDC have estimated healthy growth of cloud computing adoption [2, 3]. Cloud services include compute-on-demand, online storage, online/shared office applications, key value store, and email, among many others services. Examples of public cloud providers are Amazon AWS [4], GoGrid [5], and RackSpace [6]. Several other companies have cloud offerings, such as HP [7], Google [8], IBM [9], and Microsoft [10].

Traditional Web companies, such as Google and Yahoo, have proprietary cloud management stacks. Amazon was among the first to publish their interfaces for cloud, including management. *Eucalyptus* is an open source implementation of Amazon interfaces [11]. RightScale [12] focuses primarily on cloud management aspects of clouds. Most recently, OpenStack [13] is an effort to develop a cloud stack

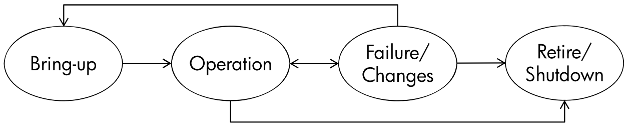

Fig. 1 Life-cycle of a managed object

by a number of companies (over 130 at the time of writing this paper and growing). In addition, there are other open source cloud stack efforts under way, such as OpenNebula [14] and Tashi [15]. Research efforts and testbeds include RESERVOIR [16], Open Cirrus [17], and Open Cloud Consortium [18]. Other examples of cloud management among many include CloudWatch, Nimsoft, MMC, Mesos [19], Monalytics [20, 21], vManage [22], and multiple managers [23].

Traditional standardization organizations, such as *DMTF*, *NIST*, and *IEEE*, have independent efforts in standardizing different aspects of clouds and cloud management. They are still early in the process to understand the impact of these efforts. Amazon Web Services interfaces appear to be a de facto standard interface, while OpenStack is getting momentum as an open source implementation thereof.

Cloud computing is enabled by advances in virtualization, service-oriented computing, and utility computing. There are several requirements for cloud computing to be successful. These include low-cost, SLA compliance, security guarantees, high availability, energy efficiency, and accurate accounting. The key to meeting these requirements is effective management of cloud resources and services. This covers all aspects of the data center life-cycle from bring-up, provisioning, scheduling, monitoring, failure management, and shutdown.

As IT becomes increasingly automated, so does the importance of IT manageability. This is especially true in cloud where automation is essential for driving down the cost. Manageability is defined as the collective processes of deployment, configuration, optimization, and administration during the life-cycle of IT systems and services. Recent examples of Amazon and VMware outages, which impacted the business continuity of a number of hosted companies, are key indicators of the importance of manageability.

Manageability has multiple dimensions. *Resource management* concerns scheduling and resource assignment, performance and availability, virtual machines, workload, and OS functions. *Automation* addresses deployment, provisioning, monitoring, configuration, changes, and problems.

Manageability targets managed objects, which can be hardware or software (object, service, data, etc.). The life-cycle of a managed object is presented in Fig. 1, from bring-up, through operation, over failures/changes, till retire/shutdown. A managed object can have different granularity and composition. The life-cycle of a managed object

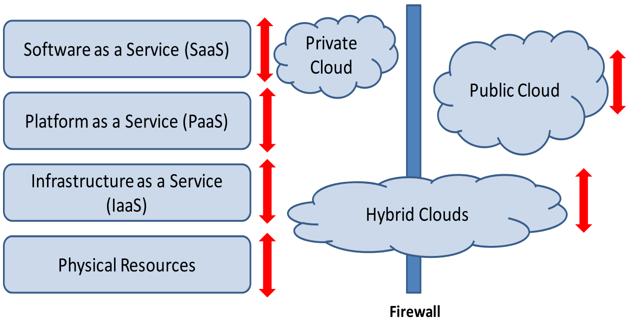
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)
Physical Resources
Private Cloud
Public Cloud
Hybrid Clouds
Firewall

Fig. 2 Managing clouds and cloud services

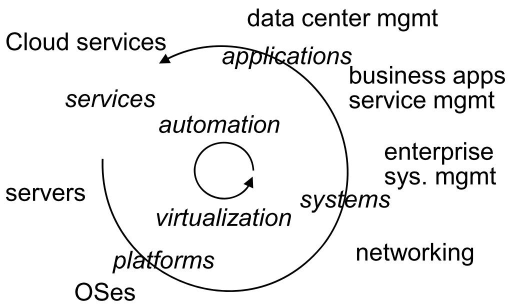

Fig. 3 Levels of management

can also be of different duration; in clouds, it is typically shorter compared to non-clouds.

While the above figure is true in cases when the full system is owned and managed by the service provider, in the case of clouds this is not true. Different parts of the system can be managed by different owners and in different domains, behind different firewalls (see Fig. 2, red arrows indicating independent management domains).

Figure 3 shows complexity of different phases and levels of management and how these phases and levels interact. Cloud services are managed at the top of this spectrum, but their management depends on managing objects lower in the dependency chain. Since different objects are managed independently, there is a need for integration of individual managers to avoid inconsistency or undesired behavior.

A distinct feature of cloud service management is "self-service," typically accomplished through a portal (see Fig. 4). An important interplay exists between development and delivery of services. The cloud management environment sits on top of the stack of different layers of cloud delivery engines, automation engines, and deployment templates and best practices.

Many of the insights in this paper we based on our prior work in management of clouds [24], scalable monitoring and analysis [25, 26], distributed systems [27], service compatibility [28], SLA management [29], adaptation [30], service deployment [31], federation [32], policy management [33], model-based management [34], change management

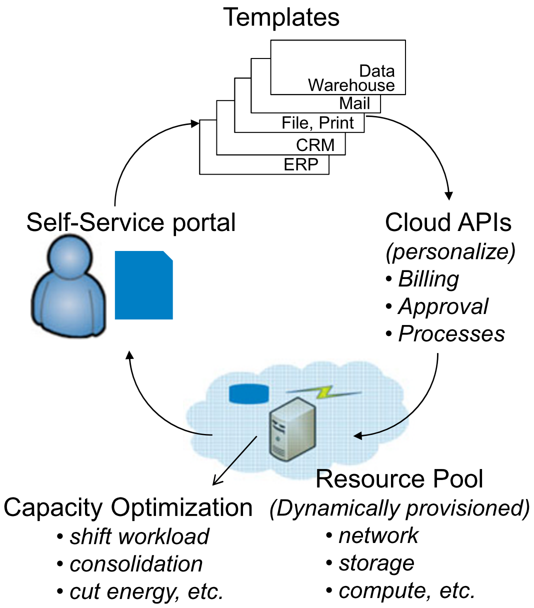

Fig. 4 Self-service at the top of service management

[35], sustainability [36] and supportability [37]. We have also derived a lot of insights from similar "Future of Software Engineering" workshops, as well as from the specific paper on the future of middleware [38].

The rest of the paper is organized in the following manner. In Sect. 2, we present three examples of contemporary cloud management. Section 3 summarizes some of the IT industry trends. In Sect. 4, we discuss requirements and research challenges. Finally, we summarize the paper in Sect. 5.

2 State of the art cloud management examples

2.1 Managing private clouds: CloudSystem matrix

HP CloudSystem is an example of a layered management stack for private but also public or hybrid cloud environments. The environment is constructed as a layering of abstraction as follows:

Virtualization management. The lowest layer provides a life-cycle management of a set of virtualized resources that are drawn from a pool of capacity in the data center. Examples of the virtualized resources and the corresponding management include virtualized servers, storage, and networking, which can be managed by VMware vCenter, Microsoft System Center Virtual Machine Manager, or HP Insight Control. This can be applied to private cloud environments and for public cloud environments, such as OpenStack or Amazon EC2. Each environment provides a notion of an underlying resource capacity implied by a combination of the physical resource being virtualized, or by quotas applied to consumption of individuals or groups. These systems provide capabilities to manage the life-cycle of their virtualized resources, as well as provide monitoring information about the resource consumption and availability of their specific components.

Cloud service composition. Built on top of the virtualization management, is the component that manages composition of the virtualization environment to create aggregate cloud service infrastructure. An aggregate service is one that uses a heterogeneous mix of virtualized resources or resource geographies to realize a service offering. Composition of services requires a model of the service components and their relationships, as well as modeling of the capacity and relationships of the underlying virtualized resources. The composition layer uses these two models to schedule use of the virtualized resources to match the infrastructure demand generated by the composite service. Scheduling algorithms take account of service quality considerations, which include both availability considerations and isolation or compliance requirements between different services. This layer monitors the state of the infrastructure elements, alerting on failures, and monitors resource consumption with a goal of providing optimal utilization of the underlying resources, including energy and network bandwidth.

Application management models the components of a business application and the relationship to the infrastructure provided from the cloud service composition layer. The infrastructure needs of the application can vary by the stage in the life-cycle, or due to varying workload demands placed on the service. As an example, during the development phase of an application, the application may reside on virtualized resources entirely contained within a testbed constructed from public cloud resources, while during production that same application may reside both on an internal private cloud holding the application transaction engine and one or more external clouds providing the Web interface and catalog components. While the application is running, the service responsiveness is monitored, and if it falls outside of set limits, then scaling adjustments are made, both by adjusting the number of running application instances or by requesting adjustment of the infrastructure supplied by the cloud service composition layer.

QoS management. In addition to the DevOps environment (see Sect. 2.2), there is also a layering of delivery management for applications, which includes scaling of the instances of the application to achieve necessary service levels, maintaining operation in the presence of maintenance cycles, and optimization of facilities utilization by removing unneeded capacity from a service automatically. In order to achieve application management, the application needs to conform to patterns supported by the cloud PaaS layer. The result of this conformance is that the PaaS platform manages the scalability and availability aspects of the services, rather than each application development team needing to create and operate a separate strategy for these aspects.

Challenges for enterprise clouds at the composition layer include algorithms for distributed placement and scheduling of virtualized resources into the distributed capacity pools, particularly for requests targeted at times in the future. For the application management and scaling, a key issue is understanding the scaling model of an application, and interpreting the root cause of application service level changes. Other challenges specific to private, public, and hybrid clouds include:

- *Automated elasticity and SLA guarantees, security, and availability* in shared environments are hard to support.
- *Unified and integrated management across compute, storage, and network* does not exist, preventing end-to-end management of applications and cloud services.
- *Federated management across clouds instances* is hard to achieve for independently managed private clouds.

2.2 Managing public clouds: Internet data centers

There has been a recent surge in new Internet companies such as Facebook, Twitter, Google, Amazon, and LinkedIn. These companies provide online services such as search, social computing, and shopping, and they are hosted within large-scale and globally deployed data centers accessed by millions of customers/users worldwide. Systems management in such large-scale infrastructures provides several challenges. Below we highlight three trends that provide specific challenges and opportunities towards next generation systems management in such infrastructures.

Massive scale in terms of users, machines, data. Existing Internet data centers already contain several hundreds of thousands of machines and this number is increasing to meet the growth in the number of users accessing the online services. A simple back of the envelope calculation easily shows that we can expect several millions of managed objects in such future data centers. This poses several challenges for the automated deployment of OS/VM/application images, load balancing to meet demands, fail-over/reliability of machines and software, as well as capacity planning to ensure service demands are met. Constraints to meet CAPEX, OPEX, and sustainability goals along with requirements to meet guaranteed service levels pose challenges for the design of scalable management systems. Furthermore, large-scale systems pose challenges for system logging, monitoring, and analysis for abnormal system behavior to meet high traffic rates. Various frameworks such as Scribe are in use by these companies but they are challenged by increasing scale. The growth of data and its storage poses additional challenges to ensure appropriate dynamic partitioning, migration, and replication to meet service demands, as well as to perform traditional archiving and backup.

Services built by integrating multiple open source frameworks. Internet companies are challenged with the need to reduce the time to bring new services to the market and at the same time ensure scalability. Recent trends include leveraging open source frameworks to quickly bring up the back-end infrastructure in operation at low-cost and leveraging resources to provide better core services. This has resulted in many open source frameworks such as Hadoop, Cassandra, Thrift, Storm, Hive, HBase, MySQL, PHP, Flume, etc. The Internet companies integrate these various open source frameworks on Linux to provide their back-end and processing infrastructure. While this speeds up the time it takes to bring up the infrastructure, it poses several challenges for ongoing operations and management. First, automated configuration management across multiple tools is a challenge. Gluing together multiple pieces written by different developers requires painful and careful integration and setting of the configuration parameters. Given different possible combinations of the integration, the current processes for configuration are ad hoc. Further, there are challenges for tuning the framework, both individual and integrated, end-to-end.

Furthermore, there are also challenges for end-to-end diagnosis of these integrated frameworks, especially in scenarios where they are pipelined together, e.g. for streaming data processing. Each framework supports a self-managing capability that allows it to recover from failures and abnormalities. However, when these frameworks are integrated together, there is a lack of an end-to-end self-managing capability, and allowing individual self-management loops to proceed without coordination leads to unpredictable behavior and inefficiencies. There is a need to develop an end-to-end monitoring and analysis framework that can be deployed on-demand in such multi-stage frameworks.

DevOps. A new DevOps model is emerging, i.e. developer and sys-admin operations are merging: several of today's Internet companies develop in-house services and the operations work is also done by in-house system administrators. This implies a culture where development and operations work together with shared responsibility. This is in contrast to previous models where software used to be packaged and shipped. System administrators, who were completely disconnected from the original developers, would deploy the package. An update or new release would occur about once in a year.

In today's Internet companies, releases happen more frequently and do not require physical packaging. Releases take place sometimes weekly or even daily. Agile development methodologies are in use for this new DevOps model. This changes the way administrators and system management tools are designed for deployment and release. Given the shared responsibility, the gap between the silos of program development and operations/admin tasks is disappearing. This implies there is tighter integration between programs and system admin tasks and greater importance for operational efficiency during development. This poses a new model for system management and a new set of tools for this integrated DevOps model. DevOps focuses on application life-cycle management for developers, not end-users, taking products through life-cycle stages: from *package* (application model) through *publish* (environment-specific deployment models); *provision* and *deploy*; *workload management*; and back to *package* (complete cycle). Specific DevOps functions include:

- Modeling & Configuration Management
- Infrastructure Provisioning
- Application Deployment
- Infrastructure and Application Monitoring
- Embedded Workload Management

Challenges in this use case include the following:

- *Heterogeneity* of deployment environments, e.g. multiple infrastructure choices, databases, or hypervisors, as well as working across private and public clouds.
- *Automated release and testing,* to enable stable products (as the versions of managed objects change and the deployed base grows substantially).
- *Support and documentation,* to resolve issues in a production environment with performance life-cycle management; enough information needs to be captured to enable support to identify problems and provide feedback through DevOps to developers to diagnose and fix issues.
- *Modeling for automated configuration management,* to address complex configurations of service compositions.
- *Maintaining stringent service level guarantees*: to ensure continuous availability of global Internet services with low latency response time even in the presence of flash crowds.

2.3 Managing HPC in the clouds: towards exascale

Today's use of clouds for high performance computing is growing, but it is limited to small scale, testing and development. Amazon has built a top-500 supercomputer in its cloud with 7 k cores and achieved speeds of 41.82 teraflops, making it the 231st fastest supercomputer in the world (at the time). They accomplished it with Linux on Intel Xeon X5570 with a 10 Gig Ethernet interconnect. It was deprovisioned soon after running the test but it demonstrated supercomputer-based processing at the price of $1.60/node hour.

At the high-end of HPC, the US Department of Energy is preparing an Exascale program, and so are governments of other countries, such as in Europe, China and Japan. An excerpt from the current DOE proposal for Exascale computing list of requirements is presented in Table 1.

Table 1 HPC evolution

	2010	2015	2018
Power	6 MW	15 MW	20 MW
Nodes #	18,700	5,000	100,000
Node concurrency	12	∼1,000	∼10,000
Interconnect BW	1.5 GB/s	1 TB/s	2 TB/s
MTTI	Day	∼Day	∼Day

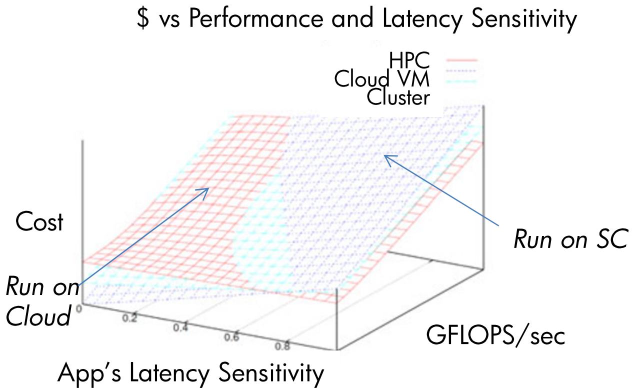

Fig. 5 HPC applications and target platforms

These parameters represent the boundaries of high-end HPC, but in many ways they are evolving in a similar direction as high-end data centers. The major differences are slower interconnects and less powerful computation nodes, similarity is in power, cooling, and packaging.

In the future, clouds will contain improved interconnects, such as photonics, that will enable more HPC applications to be executed in the cloud. The requirements for next generation supercomputers are becoming very similar to cloud requirements even though some of the design choices may be different.

Of particular interest is differentiating which applications are best suited to which platform. Figure 5 shows the types of applications that best suit clouds and supercomputers. Applications that exhibit less latency sensitivity and can be allocated to 'lower cost' resources are best suited for clouds.

A management platform that can perform such matching automatically will benefit HPC cloud adoption.

The following challenges remain for wider adoption of HPC in clouds:

- *Latency*: current interconnects deployed in cloud data centers do not offer sufficient performance for HPC applications. Photonics offers some promise for the future.
- *Cost*: to enable clouds for HPC, managing cost and pricing is essential. Existing pricing models will have to be

expanded, including physical clusters, job submissions, and future reservations.
- *Power*: as HPC grows in performance, power will continue to be one of the main obstacles both for HPC and HPC in the cloud. Carefully managing power consumption is critical for reducing power cost (power capping, server consolidation, migration, etc.).
- *Virtualization*: while overheads are of less concern for cloud applications, they limit virtualization use for HPC applications. For example, in HPC applications I/O virtualization is not used at all.
- *Security*: it will be unacceptable to execute some applications globally due to national security concerns. In addition, privacy and export rules limit the use to specific regions. Automated management of regulatory compliance will be a key differentiator.

3 IT industry trends

New technology development always results in faster, bigger, more reliable devices, such as memory, CPU, interconnect, networks, etc. However, today we are at a point where some new technology transitions will have a lasting impact on management.

NVRAM systems will have persistency and low latency storage access, driving the need for low-latency and lightweight management stacks. This will require new management models (e.g., new WBEM) and new hardware monitoring and other management tools.

Novel memory hierarchies, multi-core, photonics and advances in networking will change systems design and implementation. Management stacks will need to be optimized, lightweight, and decentralized.

Power and cooling dominate OPEX/CAPEX. To limit these costs, interfaces will have to be exposed for system and application power management.

As a result, *operating systems* will get redesigned with built-in management in various components (similarly to SMART in disks). There will be multiple components in the architecture that will contribute to management. Therefore integration and federation of management domains will become important.

Data-intensive computation and continuous production of data (from sensors and many other devices) will require the ability to archive, and manage the data life-cycle. Data elasticity is not the same as computation elasticity (stateful v. stateless; continuously produced and updated). Management will have to be intertwined with functional support; boundaries between functional support and management are disappearing.

New application models such as social networking and big data will require new management architectures and algorithms. This will result in new management models, which will be application-driven.

4 Future of cloud management: requirements and research challenges

In this section we summarize some of the requirements and challenges of future cloud management.

4.1 Future cloud management requirements

- *Global scale* (7–8 B users), mobile access by most users, elasticity at this scale.
- *Ease of use* resulting in *short time-to-manage*, using visual tools, analytics, what-if analysis, predictions, etc.
- *Cost efficiency*, understanding the costs of hosting services (infrastructure, services, and business objectives).
- *Support for SLAs with multiple objectives*, ability to make trade-offs in an easy and predictable way.
- *Availability and business continuity.* Managing replication at the resources level and at the service level; trading off replication cost for the degree of availability.
- *Automated regulatory compliance.* Due to the global nature of cloud computing, export and privacy rules need to be verified automatically.

4.2 Future cloud management research challenges

Meeting the above requirements, will expose new research challenges to cloud management. New challenges are derived from the level of scale, resource limitations (power in particular), reliability at such scale, and complexity of managing data, QoS, and integration. These challenges are discussed in more detail below and also summarized in Table 2 for cloud management today and for research direction.

- *Management at scale.* Global and mobile access will result in unpredictable scale up and down. Elasticity of access also results in elasticity of management. *Federation* will be a way to address scalability and to connect independently managed clouds.
- *Sustainability.* Environmental awareness is becoming increasingly regulated and it will become a requirement, not just a desirable feature. Power limitations will drive cost and scale as data centers continue to grow.
- *Reliability and support.* As scale continues to grow, failure rates will also increase, leaving no choice but to automate support. Support will also move away from reactive towards deferred and proactive. Supportability and reliability will be built into the design across all layers.
- *QoS. SLA management* was always hard and it will only grow in complexity with global access, a wide variety of standard and non-standard interfaces, and different APIs

Table 2 Summary of state of the art and research direction of cloud management

Management functionality	State of the art	Research direction
Management at scale and federation	Hundreds of thousands of nodes in data centers; zones and service-level integration, incremental scalability; simple visualization.	Hierarchies of domains, federations of independently managed data centers and clouds; visualization analytics at full scale.
Sustainability	Tracking power, CO_2 and water usage, minimizing environmental impact; introduction of end-to-end sustainability.	Trading off sustainability for QoS, automated sustainability and SLA management, accounting for sustainability of mobile services delivery.
Support and reliability	Reactive at the high end with field engineers, deferred at the low end with minimal human use; semi-automated.	Preventive, substantially automated, self-healing and rejuvenation of components; field engineers only used at the very high end.
QoS: SLA management	Simple services level objectives. Lack of compliance and enforcing SLAs. No integration with business models.	Multi-objectives, business objectives (pricing, costing). Automated enforcement and compliance. Hierarchical decomposition of SLAs.
Data management	Data center data deduplication, petascale of structured and unstructured data; disks and tapes or backups; regulatory compliance.	Global deduplication, Exascale largely unstructured data; hierarchies of storage around NVRAM with disks at the bottom; global compliance.
Integration of management components	Component integration at a single layer, local feedback loops; rapid deployment, configuration management and patching; orchestration of global services.	Choreographies and closed loops of loosely coupled domains addressing power, performance, availability, etc. individually and with trade-offs (e.g. power-performance for power capping).
Quantifying manageability	Checklist of management functions, documentation, time and steps to manage objects and services.	Measuring Quality of Management (QoM), elasticity of management (matching manageability capabilities to those of functionality supported), ease of management.

for SLA management. Multiple objectives will result in further complexity.

- *Data management.* With continuous generation of new data from sensors, multimedia data formats, and many other sources, the ability to manage this data, and compress, deduplicate, archive, and dispose of it, according to regulatory compliance, will be a huge challenge.
- *Integration* of management components, and run-time *composition*. Increasingly more integrated services will result in even higher complexity of versioning, compatibility, and coordination among multiple management components.
- *Quantifying cloud manageability* is a research challenge. Some of the ways to quantify manageability are listed below, but new models and metrics need to be devised:
  - ◦ Checklist of manageability functions
  - ◦ Number of steps to manage towards desired state
  - ◦ Time to manage (including time to insight)
  - ◦ Documentability (e.g. lines of management code)
  - ◦ Elasticity of management (manage at scale)
  - ◦ Availability and continuity of management
  - ◦ Ease of use (GUIs, visualization, analytics, etc.)

5 Summary

In this paper, we evaluated cloud management today and some of the trends that we see coming in the future. We presented three examples of cloud management: public, private, and HPC. For each, we emphasized challenges for the future of cloud management. We then related cloud management trends to the general trends in the IT industry. Based on these trends, we summarized some of the requirements and research challenges of future cloud management.

Cloud computing has a fundamental role in the future of society, as most IT is migrating towards the cloud. As mobile services find their way into the cloud, it will become even more ubiquitous. The role of cloud management will become essential—particularly in regard to how scale, DevOps, and QoS are addressed. With the tremendous amount of data expected to be generated, data-intensive operations will become dominant compared to those that are compute-intensive, while sustainability and support will change in the future.

The landscape of the cloud—at different levels of the stack (hardware, services), as well as roles (developers, operators, users)—will differ substantially from the one today. At the hardware layer new technologies will enable greater scale, requiring increased automation and new reliability techniques. Operating these types of evolving clouds and their services will require frequent updating, an understanding of business trends, and the ability to perform what-if analysis. Development of new services will increasingly be the result of the composition with continuous roll-outs. Most cloud users will be mobile, and many new users will be from developing countries; these powerful user segments

Table 3 Summary of trends impacting future of cloud management

Layers of the stack	State of the art	Research direction
Cloud users	Traditional Internet users, increased mobile access limited from developed and emerging areas. Some mash-up ability of limited number of users. Some ability to customize and personalize accounts.	Dominantly mobile access, development countries growth towards 8 B users, especially through mobile. Extensive mash-ups through user composed services. Extensive personalization and customization.
Cloud services developers	Small number for traditional and mobile services Few releases annually, careful testing some service location awareness New services through development.	Through composition, integration, large % of developers; continuous roll-out of new releases, agile development. Full location awareness; integrate with local services available ubiquitously.
Cloud management operators	Cloud and cloud service operators (small %) increasing updates to services mobile devices some high level dashboard, analytics reporting, some prediction.	Merging role with cloud developers (large %) frequent updates to mobile services, access devices, detailed business dashboards, visual analytics what-if analysis, prediction business outcomes.
Hardware and *its impact on support*	Disks, early adoption of SDDs, 10 Gb/s Ethernet, early adoption of optical interconnect, 16–24 Core CPUs, 100,000 server data centers, air cooling, very limited use of water cooling, *high resource redundancy, reactive and delayed support, field engineers, complex software repair.*	NVRAM adoption, broad optical interconnect, deployment, 1000+ Core CPUs, with sophisticated, photonics off-on chips, 10^{12}+ server data centers, ambient cooling (commodity), liquid cooling (high-end), *self-healing, proactive support, customer self-repair, repair moving up the stack, restartable services.*

will drive innovation and cloud services pricing models—and therefore cloud management. (See also Table 3.) Cloud management is fertile ground for fundamental research in systems, applications, and services.

Acknowledgements We would like to thank organizers of the FOME event, Gordon Blaire and Valerie Issarny, for their leadership in forming this event, as well for reviewing our paper. Their feedback significantly improved the content and the form. We are also thankful to Fabio Kon for continued guidance. Figure 3 was derived from the similar figure originally created by Sue Charles. Figure 5 was created by Abhishek Gupta as a part of his internship at HP Labs. Sue Charles and Puneet Sharma reviewed submitted document and provided valuable feedback.

References

1. Armbrust M et al. Above the clouds: a Berkeley view of cloud computing. Tech report UCB/EECS-2009-28 (2009)
2. Gartner, http://www.gartner.com/it/page.jsp?id=1389313; http://www.gartner.com/it/page.jsp?id=1454221
3. IDC, http://www.idc.com/research/cloudcomputing/index.jsp
4. Amazon AWS, http://aws.amazon.com/
5. GoGrid, http://www.gogrid.com/
6. RackSpace, http://www.rackspacecloud.com/
7. HP Cloud, http://www8.hp.com/us/en/solutions/solutions-detail.html?compURI=tcm:245-300983&pageTitle=cloud
8. Google Apps, http://www.google.com/apps/intl/en/business/index.html
9. IBM Cloud, http://www.ibm.com/ibm/cloud/
10. Microsoft Cloud, http://www.microsoft.com/en-us/cloud/
11. Eucalyptus, http://www.eucalyptus.com/
12. RightScale, http://www.rightscale.com/
13. Open Stack, http://www.openstack.org/
14. Moreno-Vozmediano R, Montero RS, Llorente IM (2009) Elastic management of cluster-based services in the cloud. In: ACDC'09.
15. Kozuch M et al (2009) Tashi: location-aware cluster management. In: ACDC'09, Barcelona, Spain, June 2009
16. RESERVOIR, www.reservoir-fp7.eu
17. Avetisyan A et al (2010) Open cirrus a global cloud computing testbed. IEEE Comput 43(4):42–50
18. Open Cloud Consortium, http://www.opencloudconsortium.org/
19. Hindman B et al. (2011) Mesos: a platform for fine-grained resource sharing in the data center, NSDI 2011, March 2011
20. Wang C et al (2011) A flexible architecture integrating monitoring and analytics for managing large-scale data centers. In: ICAC
21. Kutare M et al (2010) Monalytics: online monitoring and analytics for managing large scale data centers. In: ICAC
22. Kumar S et al (2009) vManage: loosely coupled platform and virtualization management in data centers. In: Proceedings of 6th ICAC, Barcelona, Spain, June 2009
23. Kephart J et al (2007) Coordinating multiple autonomic managers to achieve specified power-performance trade-offs. In: Proc of the 4th ICAC, IEEE CS
24. Cook N, Milojicic D, Talwar V (2011) Managing the cloud infrastructure. In: Migrating to the cloud: for developers, and technologists. Elsevier, Amsterdam
25. Wang C et al (2011) Statistical techniques for online anomaly detection in data centers. In: IM
26. Viswanathan K et al (2012) Ranking anomalies in data centers. In: NOMS (to appear)
27. Adams R, Brett P, Iyer S, Milojicic D, Rafaeli S, Talwar V (2006) Scalable management. In: Autonomic computing: concepts, infrastructure, and applications. CRC Press, Boca Raton
28. Becker K, Pruyne J, Singhal S, Lopes A, Milojicic D (2010) Automatic determination of compatibility in evolving services. Int J Web Serv Res, 8(1):21–40
29. Chen Y, Iyer S, Liu X, Milojicic D, Sahai A, Decomposition SLA (2008) Translating service level objectives into system level thresholds. Clust Comput 11(3):299–311
30. Vambenepe W, Thompson C, Talwar V, Rafaeli S, Murray B, Milojicic D, Iyer S, Farkas K, Arlitt M (2007) Dealing with scale and adaptation of global web services management. J Web Serv Res 4(4):65–84
31. Talwar V, Milojicic D, Wu Q, Pu C, Yan W, Jung G (2005) Approaches for service deployment. IEEE Internet Comput 9(2):70–80
32. Bardhan S, Hidangmayum R, McGeer R, Milojicic D, RN V, Feldhaus F, Roeblitz T, Yahayapour R (2011) Practical federations. In:

Proceedings of the fifth open cirrus summit, Moscow, IEEE co-sponsored, June 2011

33. Cai Z, Chen Y, Kumar V, Milojicic D, Schwan K (2007) Automated availability management driven by business policies. In: Proc. of the 10th IFIP/IEEE symposium on integrated network Mgmt, IM'07, Munch, pp 264–273
34. Rivaldo R, Chen Y, Milojicic D, Adams R, Model-based SML (2007) Management. In: Proceedings of the 10th IFIP/IEEE symposium on integrated network management (IM 2007), Munich, pp 761–764
35. Shankar C et al (2006) Specification-enhanced policies for automated management of changes in IT systems. In: Proc of 20th USENIX LISA'06
36. Bash C et al Sustainability Dashboard Cloud (2011) Dynamically assessing sustainability of data centers and clouds. In: Proceedings of the fifth open cirrus summit, Moscow, IEEE co-sponsored, June 2011
37. Connelly C et al (2009) Reiki: serviceability architecture and approach for reduction and management of product service incidents. In: Proc. IEEE ICWS, Jul 2009, pp 775–782
38. Issarny V et al (2011) Service-oriented middleware for the future Internet: state of the art and research directions. JISA—J Int Appl Serv 2(1):23–45