Software defined environments: An introduction

C.-S. Li
B. L. Brech
S. Crowder
D. M. Dias
H. Franke
M. Hogstrom
D. Lindquist
G. Pacifici
S. Pappe
B. Rajaraman
J. Rao
R. P. Ratnaparkhi
R. A. Smith
M. D. Williams

During the past few years, enterprises have been increasingly aggressive in moving mission-critical and performance-sensitive applications to the cloud, while at the same time many new mobile, social, and analytics applications are directly developed and operated on cloud computing platforms. These two movements are encouraging the shift of the value proposition of cloud computing from cost reduction to simultaneous agility and optimization. These requirements (agility and optimization) are driving the recent disruptive trend of software defined computing, for which the entire computing infrastructure—compute, storage and network—is becoming software defined and dynamically programmable. The key elements within software defined environments include capability-based resource abstraction, goal-based and policy-based workload definition, and outcome-based continuous mapping of the workload to the available resources. Furthermore, software defined environments provide the tooling and capabilities to compose workloads from existing components that are then continuously and autonomously mapped onto the underlying programmable infrastructure. These elements enable software defined environments to achieve agility, efficiency, and continuous outcome-optimized provisioning and management, plus continuous assurance for resiliency and security. This paper provides an overview and introduction to the key elements and challenges of software defined environments.

Introduction

Based on the trends toward migrating both mission-critical and performance-sensitive workloads to the cloud deployment model and the convergence of mobile, social, and analytics workloads on the cloud, we see a shift of the value proposition of cloud computing from cost reduction to simultaneous agility and optimization. While cost reduction largely focused on the virtualization of infrastructure (IaaS, or infrastructure as a service), agility focuses on the ability to rapidly react to changes in the cloud environment and workload requirements. Optimization focuses on identifying better outcomes within a cloud environment with respect to the value of the service, the cost of the required resources, and the risk of failure in an unpredictable environment and under constant usage changes. This requires a high degree of automation and programmability of the infrastructure itself. Hence, this shift led to the recent disruptive trend of software defined computing for which the entire system infrastructure—compute, storage and network—is becoming software defined and dynamically programmable. Software defined computing receives considerable focus across academia and enterprises [1–7].

Software defined computing originated from the compute environment in which the computing resources are virtualized and managed as virtual machines [8–11]. This enabled mobility and higher resource utilization as several virtual machines are collocated on the same server,

Digital Object Identifier: 10.1147/JRD.2014.2298134

and variable resource requirements can be mitigated by being shared amongst the virtual machines. Software defined networks (SDNs) move the network control and management planes (functions) away from the hardware packet switches and routers to the server for improved programmability, efficiency, extensibility, and security [12–16]. Software defined storage (SDS), similarly, separates the control and management planes from the data plane of a storage system and dynamically leverages heterogeneous storage to respond to changing workload demands [17, 18]. Software defined environments (SDEs) bring together software defined compute, network, and storage and unify the control and management planes from each individual software defined component. These unified control planes are assembled from programmable resource abstractions of the compute, network, and storage resources of a system (also known as purpose-fit systems or workload-optimized systems) that meet the specific requirements of individual workloads and enable dynamic optimization in response to changing business requirements. For example, a workload can specify the abstracted compute and storage resources of its various workload components—and their operational requirements (e.g., I/O [input/output] operations per second)—and how these components are interconnected via an abstract "wiring" that will have to be realized using the programmable network.

In this paper, we provide an overview of the vision, architecture, and benefit of SDEs, as shown in **Figure 1**. At the top, workload abstractions (shown as graphs of connected services and applications) and related tools provide the means to construct workloads and services based on preexisting patterns and to capture the functional and non-functional requirements of the workloads. At the bottom, heterogeneous compute, storage, and networking resources are pooled based on their capabilities. The workloads and their contexts are then mapped to the best-suited resources. The unified control plane dynamically constructs, configures, continuously optimizes, and proactively orchestrates the mapping between the workload and the resources based on the desired outcome specified by the workload and the operational conditions of the cloud environment. We also demonstrate at a high level how this architecture achieves agility, efficiency, and continuous outcome optimized infrastructure with proactive resiliency and security, but we also refer to the additional papers in this journal for more details and examples. A modeling language for SDEs is described in [19]. A language and runtime for continuous delivery of software and related infrastructure and the validation of workload blueprints is described in [20, 21]. An overview of how service-level agreements are being met in virtualized infrastructures is given in [22]. State of the art software defined efficient and agile storage solutions are described in [23] and advances in software defined networking are described in [24].

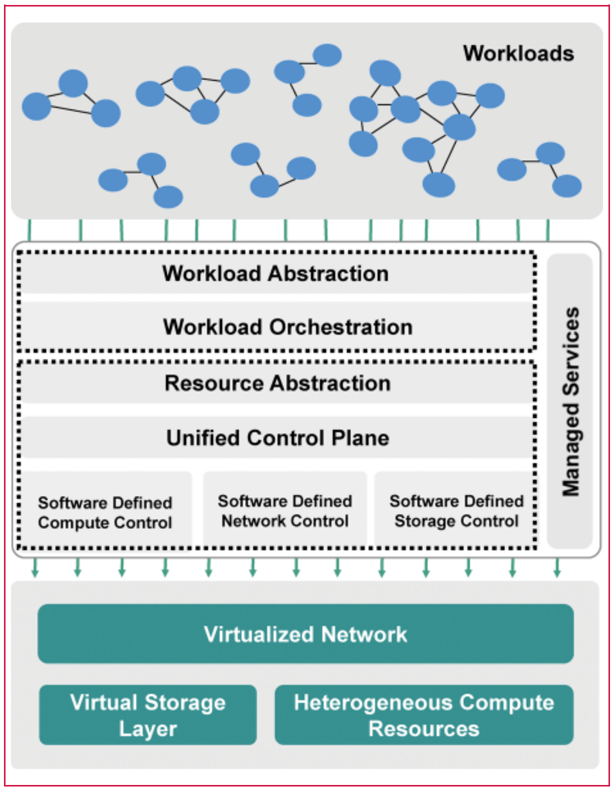

Figure 1

Architecture of software defined environments. Workloads are complex wirings of components and are represented through abstractions. Given a set of abstract resources the workloads are continuously mapped (orchestrated) into the environment through the unified control plane. The individual resource controllers program the underlying virtual resources (compute, network, and storage).

Software defined environments architecture

Current virtualization and cloud solutions only allow basic abstraction of the computing, storage, and network resources in terms of their capacity [25]. These approaches often call for standardization of the underlying system architecture to simplify the abstraction of these resources. The convenience offered by the elasticity for scaling the provisioned resources based on the workload requirements, however, is often achieved at the expense of overlooking capability differences inherent in these resources. Capability differences in the computing domain could range from differences in the instruction set architecture (ISA), e.g., Intel x86 versus IBM POWER* architectures—to different implementations of the same ISA, e.g., Xeon** by Intel versus Opteron** by AMD—to different generations of the same ISA by the same vendor, e.g., POWER7* versus POWER7+* versus POWER8* from IBM, and Nehalem versus Westmere versus Sandy Bridge versus Ivy Bridge from Intel. An additional dimension is added with the availability of general processing units (GPUs) and other accelerators for encryption, compression, extensible markup language (XML) acceleration, or other scalar/vector functions.

The workload optimized system approaches often call for tight integration of the workload with the tuning of the underlying system architecture for the specific workload. These approaches tightly couple the special capabilities offered by each micro-architecture and by the system level capabilities at the expense of potentially labor-intensive tuning required. These workload optimized approaches are not sustainable in an environment where the workload might be unpredictable or evolve rapidly as a result of growth of the user population or the continuous changing usage patterns.

The conundrum created by these conflicting requirements in terms of standardized infrastructure vs. workload-optimized infrastructure is further exacerbated by the increasing demand for agility and efficiency as more applications from *systems of record* and *systems of engagement* require fast deployment while continuously being optimized based on the available resources and unpredictable usage patterns. Systems of record and systems of engagement are defined later in this paper, but refer to client-server ERP (enterprise resource planning) systems "on top of" the Internet—and systems that are more focused on engagement with the large set of end users in the consumer space, respectively.

SDEs are intended to address the challenge created from the desire for simultaneous agility and optimization. SDEs decouple the abstraction of resources from the real resources and only focus on the salient capabilities of the resources that really matter for the desired performance of the workload. SDEs also establish the workload definition and decouple this definition from the actual workloads so that the matching between the workload characteristics and the capabilities of the resources can be done efficiently and continuously. Simultaneous abstraction of both resources and workloads to enable late binding and flexible coupling among workload definitions, workload runtime, and available resources is fundamental to addressing the challenge created by the desire for both agility and optimization in deploying workloads.

Policy-based and goal-based workload abstraction

Workload involves the relationship between the amount of processing capabilities (resources) and the tasks that need to be performed. Workload modeling is the analytical technique used to measure and predict workloads, with the primary objective to achieve evenly distributed manageable workloads, to avoid overload and to satisfy service level objectives. The workload definition has been previously and extensively studied in the context of the OMG (Object Management Group) Model Driven Architecture (MDA) initiative during the late 1990s as an approach to system specification and interoperability based on the use of formal models. In MDA, platform-independent models are described in a platform-independent modeling language such as UML (Unified Modeling Language). The platform-independent model is then subsequently translated into a platform-specific model by mapping the platform independent models to implementation languages such as Java**, XML, SOAP (Simple Object Access Protocol), or various dynamic scripting languages using formal rules.

Workload concepts were heavily used in grid computing, for example, IBM Platform Symphony, for defining and specifying tasks and resources, and predicting and optimizing the resources for the tasks in order to achieve optimal performance. IBM Enterprise Workload Manager (eWLM) allows the user to monitor application-level transactions and operating-system processes, allows the user to define specific performance goals with respect to specific work, and allows adjusting the processing power among partitions in a partition workload group to ensure that performance goals are met.

More recently, workload automation and development for deployment have received considerable interests as the Development and Operations (DevOps) concept becomes widely deployed. These workload automation environments often include programmable infrastructures that describe the available resources and characterization of the workloads (topology, service-level agreements, and various functional and nonfunctional requirements). Examples of such environments include Amazon Cloud Formation, Oracle Virtual Assembly Builder, VMware vFabric, and IBM Workload Deployer (IWD).

A workload, in the context of SDEs, is often composed of a complex "wiring" of services, applications, middleware components, management agents, and distributed data stores. Correct execution of a workload requires that these elements be wired and mapped to appropriate logical infrastructure according to workload-specific policies and goals.

Workload experts create workload definitions for specific workloads, which codify the best practices for deploying and managing the workloads. The workload abstraction specifies all of the workload components including services, applications, middleware components, management agents, and data. It also specifies the relationships among components and policies/goals defining how the workload should be managed and orchestrated [20]. These policies represent examples of workload context embedded in a workload definition. They are derived based on expert knowledge of a specific workload or are learned in the course of running the workload in SDE. These policies may include requirements on continuous availability, minimum throughput, maximum latency, automatic load balancing, automatic migration, and auto-scaling in order to satisfy the service-level objectives. These "contexts" for the execution of the workload need to be incorporated during the translation from workload definition to an optimal infrastructure pattern that satisfies as many of the policies, constraints, and goals that are pertinent to this workload as possible [21].

Capability-based resource abstraction and software defined infrastructure

The abstraction of resources is based on the capabilities of these resources. Capability-based pooling of heterogeneous resources requires classification of these resources based on workload characteristics. Using compute as an example, server design is often based on the thread speed, thread count, and effective cache/thread. The fitness of the compute resources (servers in this case) for the workload can then be measured by the serial fitness (in terms of thread speed), the parallel fitness (in terms of thread count), and the data fitness (in terms of cache/thread).

Capability-based resource abstraction is an important step toward decoupling provisioning heterogeneous resources from the workload. Currently, resource provisioning is only based on capacity, and hence the differences in characteristics of the resource are often ignored.

The *Pfister framework* [26] has been used to describe workload characteristics [7] in a two-dimensional space where one axis describes the amount of thread contention, and the other axis describes the amount of data contention. We can categorize the workload into four categories based on the Pfister framework: *Type 1* (mixed workload updating shared data or queues*),* *Type 2* (highly threaded applications, including WebSphere* applications), *Type 3* (parallel data structures with analytics, including Big Data, Hadoop**, etc.), and *Type 4* (small discrete applications, such as Web 2.0 apps).

Servers are usually optimized to one of the corners of this two-dimensional space, but not all three corners. For instance, the IBM System z* [27] is best known for its single-thread performance, while IBM Blue Gene* [28] is best known for its ability to carry many parallel threads. Some of the systems (IBM System x3950 [Intel based] and IBM POWER 575) were designed to have better I/O capabilities. Eventually there is not one server that can fit all workloads and is good for all types of workloads described above.

This leads to a very important observation: the majority of workloads (whether they are systems of record or systems of engagement or combination of the two) consist of multiple workload types, and are best addressed by a combination of heterogeneous servers rather than homogeneous servers.

We envision resource abstractions based on different computing capabilities that are pertinent to the subsequent workload deployments. These capabilities could include high memory bandwidth resources, high single thread performance resources, high I/O throughout resources, high

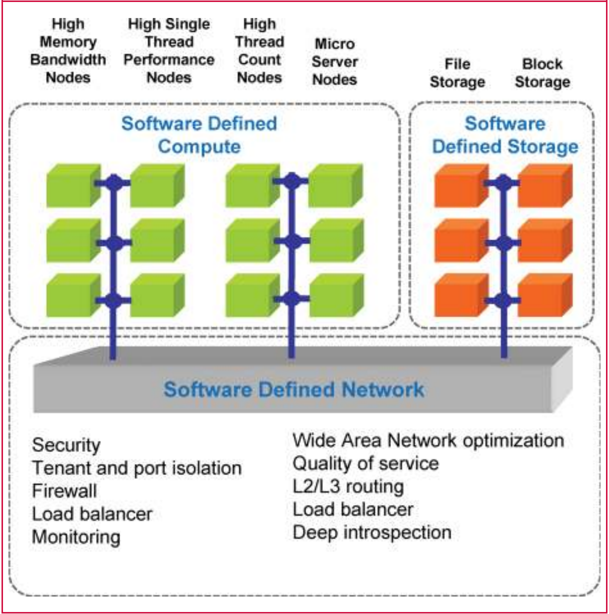

Figure 2

Capability-based resource abstraction.

cache/thread resources, and resources with strong graphics capabilities. Capability-based resource abstraction eliminates the dependency on specific instruction-set architectures (e.g., Intel x86 versus IBM POWER versus ARM) while focusing on the true capability differences (AMD Opteron versus Intel Xeon, and IBM POWER7 versus POWER7+ versus POWER8 may be represented as different capabilities).

Previously, it was reported [29] that up to 70% throughput improvement can be achieved through careful selection of the resources (AMD Opteron versus Intel Xeon) to run Google's workloads (content analytics, Big Table, and web search) in its heterogeneous warehouse scale computer center. Likewise, storage resources can be abstracted beyond the capacity and block versus file versus objects. Additional characteristics of storage—such as high I/O throughput, high resiliency, and low latency—can all be "brought to the surface" as part of storage abstraction. Networking resources can be abstracted beyond the basic connectivity and bandwidth. Additional characteristics of networking—such as latency, resiliency, and support for remote direct memory access (RDMA)—can be brought to the surface as part of the networking abstraction.

The combination of capability-based resource abstraction for software defined compute, storage, and networking forms the software defined infrastructure, as shown in **Figure 2**. This is essentially an abstract view of the available compute and storage resources interconnected by the networking resources. This abstract view of the resources includes the pooling of resources with similar capabilities (for compute and storage), connectivity among these resources (within one hop or multiple hops), and additional functional or nonfunctional capabilities attached to the connectivity (load balancing, firewall, security, etc.). Additional physical characteristics of the datacenter are often captured in the resource abstraction model as well. These characteristics include clustering (for nodes and storage sharing the same top of the rack switches and that can be reached within one hop), point of delivery (POD) (for nodes and storage area network [SAN]-attached storage sharing the same aggregation switch and can be reached within 4 hops), availability zones (for nodes sharing the same uninterrupted power supply (UPS) and A/C), and physical data center (for nodes that might be subject to the same natural or man-made disasters). These characteristics are often needed during the process of matching workload requirements to available resources in order to address various performance, throughput, and resiliency requirements.

Continuous optimization

As a business increasingly relies on the availability and efficiency of its IT infrastructure, linking the business operations to the agility and performance of the deployment and continuous operation of IT becomes crucial for the overall business optimization. SDEs provide an overall framework for directly linking the business operation to the underlying IT as described below.

Each *business operation* can be decomposed into multiple tasks, each of which has a priority. Each task has a set of key performance indicators (KPIs), which could include confidentiality, integrity, availability, correctness/precision, quality of service (latency, throughput, etc.), and potentially other KPIs.

As an example, an e-commerce business operation might include the following tasks: marketing (ad display), search against the catalog, verifying the inventory, invoicing and billing, and fulfillment. Each of these tasks may be measured by different KPIs: display ads may focus on availability; search against catalog may focus on latency and then precision; verification of inventory/invoicing/billing may focus on integrity. The specification of the task decomposition of a business operation, the priority of each task, and KPIs for each task allow trade-offs being made among these tasks when necessary. Using search against catalog as an example, precision might have to be reduced when there is insufficient capacity until the capacity is increased or the load is reduced.

The KPIs for the task often are translated to the architecture and KPIs for the infrastructure. Confidentiality usually translates to required isolation for the infrastructure. Availability potentially translates into redundant instantiation of the runtime for each task using active-active or active-passive configurations. Integrity of transactions, data, processes, and policies is managed at the application level, while the integrity of the executables and virtual machine images is managed at the infrastructure level. Correctness and precision need to be managed at the application level, and quality of service (QoS) (latency, throughput, etc.) usually translates directly to the implications for infrastructures.

Continuous optimization of the business operation is performed to ensure optimal business operation during both "normal time" (best utilization of the available resources) as well as "abnormal time" (ensures the business operation continues in spite of potential system outages). This potentially requires tradeoffs among KPIs in order to ensure the overall business performance does not drop to zero due to outages.

The overall close-loop framework, as shown in **Figure 3**, for continuous optimizing is as follows:

- The KPIs of the service are continuously monitored and evaluated at each layer (the application layer and the infrastructure layer), so that the overall utility function (value of the business operation, cost of resource, and risk to potential failures) can be continuously evaluated based on the probabilities of success and failure, $P(\text{success})$ and $P(\text{failure})$. Deep introspection, i.e. a detailed understanding of resource usage and resource interactions, within each layer is used to facilitate the monitoring. The data is fed into the behavior models for the SDE (which includes the workload, the data (usage patterns), the infrastructure, and the people and processes).
- When triggering events occur, what-if scenarios for deploying different amount of resources against each task will be evaluated to determine whether KPIs can be potentially improved.
- The scenario that maximizes the overall utility function is selected, and the orchestration engine will orchestrate the SDE through the following: (a) adjustment to resource provisioning (scale up or down), (b) quarantine of the resources (in various resiliency and security scenarios), (c) task/workload migration, and (d) server rejuvenation.

Simultaneous agility and efficiency

It was observed in [30] that a fundamental change in the axis of IT innovation is happening in the industry. Prior to 2000, new systems were introduced at the very high end of the economic spectrum (large public agencies and Fortune 500 companies). These innovations trickled down to smaller businesses, then to home office applications, and finally to consumers, students and even children. In this past decade, this innovation flow has been reversed and often started with the consumers, students, and children leading the way, especially due to the proliferation of mobile devices. These innovations are then adopted by nimble

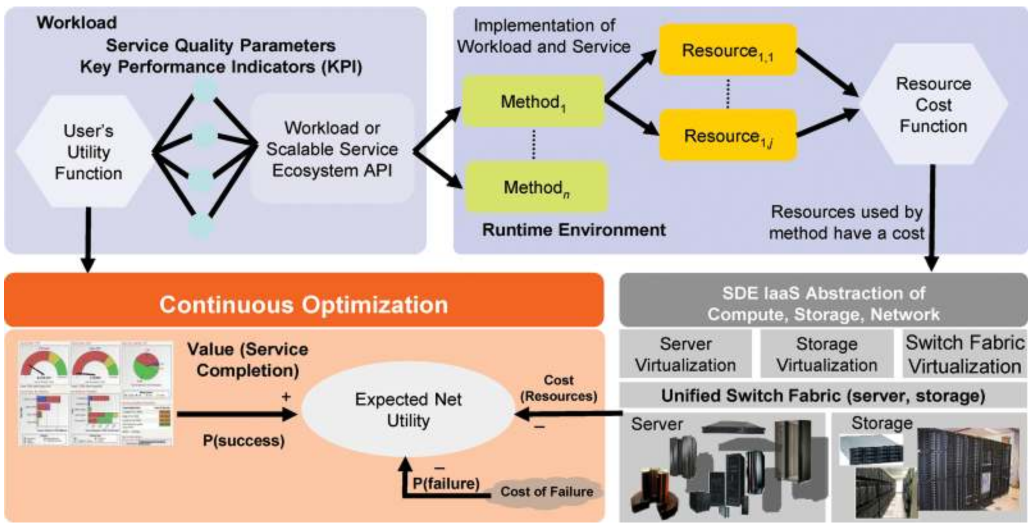
Workload
Service Quality Parameters
Key Performance Indicators (KPI)
User's Utility Function
Workload or Scalable Service Ecosystem API
Implementation of Workload and Service
Method1
Methodn
Resource1,1
Resource1,j
Resource Cost Function
Runtime Environment
Resources used by method have a cost
Continuous Optimization
Value (Service Completion)
P(success)
Expected Net Utility
Cost (Resources)
P(failure)
Cost of Failure
SDE IaaS Abstraction of Compute, Storage, Network
Server Virtualization
Storage Virtualization
Switch Fabric Virtualization
Unified Switch Fabric (server, storage)
Server
Storage

Figure 3

Outcome-optimized framework for software defined environments.

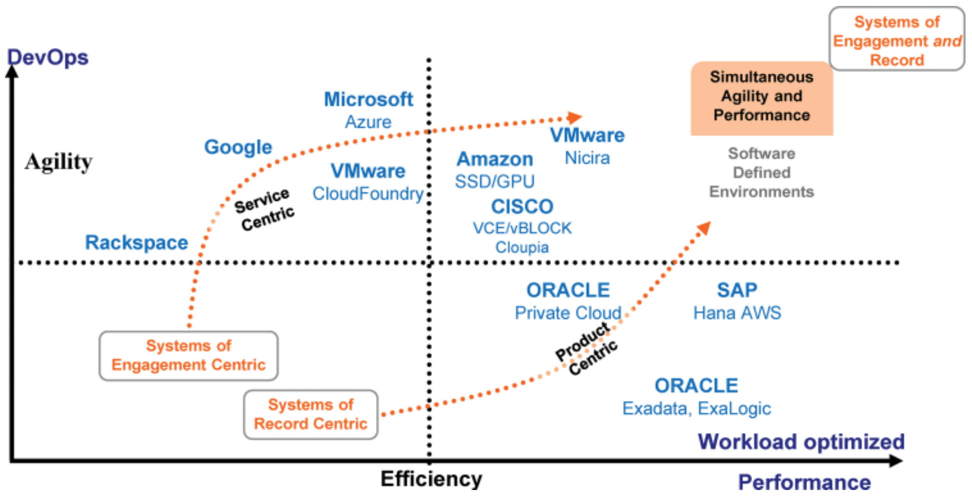

Figure 4

Industry is responding to the trend toward simultaneous agility and efficiency through a product driven and service driven approach. Two potential path are shown toward simultaneous agility and performance (classifications are estimates and for illustration only): (a) service-centric, which is typically followed by offerings that are systems of engagement centric, and (b) product-centric, which is typically followed by offerings that are systems-of-record centric.

small-to-medium-size businesses. Larger institutions are often the last to embrace these innovations.

The author of [30] coined the term "systems of engagement" for the new kinds of systems that are more focused on engagement with the large set of end users in the consumer space. As shown in **Figure 4**, many systems of engagement such as Facebook**, Twitter**, Netflix**, Instagram**, and many others are "born" on the cloud using public cloud services from Amazon Web Services (AWS), Google App Engine, Microsoft Azure**, Rackspace**, etc.

These systems of engagement often follow the agility trajectory. On the other hand, as mentioned, the workload-optimized system concept is introduced to the “systems of record” environment, which occurred with the rise of client-server ERP (enterprise resource planning) systems “on top of” the Internet. Here the entire system, from top to bottom, is tuned for the database or data warehouse environment. Evidence is mounting that both kinds of systems are likely to converge in the upper right quadrant of Figure 4 as systems of engagement are demanding optimal efficiency, while systems of records are demanding improved agility.

Netflix is an example that started as a system of engagement. Its traffic grew 37 times between January 2010 and January 2011. As a result, Netflix’s infrastructure stack went through rapid transitions to address the fast evolution of the environment. The key approach adopted by Netflix is to instrument and automate all aspects as much as possible to achieve continuous delivery and agile deployment. Since 2012, Netflix also began its relentless pursuing of efficiency. When AWS came out with instances equipped with solid-state drives (SSDs) during July 2012, Netflix quickly determined that the use of these types of AWS instances eliminate the need for MemcacheD, a memory-based caching service, and achieved a two-fold cost reduction in spite of each new type of instance being more expensive, as fewer instances are needed. Animoto, a company specialized in generating videos from photo albums, leverages GPU-equipped instances from AWS to accelerate the video synthesis process and achieved a ten-fold throughput improvement as well as much improved video resolution. Meanwhile, most of the workload-optimized systems in the industry, such as Oracle Exadata/Exalogic [31] and SAP Hana [32], are being provided in the context of private cloud environments to facilitate virtualization and sharing for improved agility.

Continuous assurance on resiliency and security

The key securities challenges that need to be addressed in an SDE include the following. First, *challenges exist due to SDE’s increased virtualization*. Traditional security architectures are physically based, as IT security often relies on the identities of the machine and the network. This model is less effective when there are multiple layers of virtualization and/or abstractions, which could result in many virtual “systems” being created within the same physical system or multiple physical systems virtualized into a single virtual system. This is further compounded by the use of dedicated or virtual appliances in the computing environment.

Second, *challenges exist due to SDE’s increased agility*. SDEs enable standing up and tearing down computing, storage and networking resources quickly as everything becomes programmable; hence, they break the *long term association* between security policies and the underlying hardware and software environment. The emerging SDE environment requires the ability to quickly set up and continuously evolve security policies directly related to users, workloads, and the software defined infrastructure. There are no permanent associations (or bindings) between the logical resources and physical resources as software defined “systems” can be continuously created from scratch, can be continuously evolved, and destroyed at the end. As a result, the challenge will be to provide a low-overhead approach for capturing the *provenance* (“who” has done what, at what time, to whom, in what context), to identify the suspicious events in a rapidly changing virtual topology.

Third, *challenges exist due to SDE’s increased resource heterogeneity*. In order to accommodate heterogeneous compute, storage, and network resources in an SDE, resources are abstracted in terms of capability and capacity. This normalization of the capability across multiple types of resources masks the potential differences in various nonfunctional aspects such as the vulnerabilities to outages and security risk.

The overall framework on continuous assurance of resiliency and security is directly related to the continual optimization of the services performed within the SDEs, taking into account the value created by the delivery of service, subtracting the cost for delivering the service and the cost associated with a potential failure (weighted by the probability of such failure). This framework allows us to properly calibrate the value@risk for any given service, so that the overall metric will be risk adjusted cost performance. The key elements for providing continuous assurance, to ensure resiliency and security, as shown in **Figure 5**, include the following. The first element is *deep introspection*, which works with *probes* (often in the form of agents) inserted into the governed system to collect additional information that cannot be easily obtained simply by observing network traffic. These probes could be inserted into the hardware, hypervisors, guest virtual machines, middleware, or applications. Additional approaches include micro-checkpoints and periodic snapshots of the virtual machine images when they are active. The key challenge is to avoid introducing unnecessary overhead while providing comprehensive capabilities for monitoring and rollback when abnormal behaviors are found.

The second key element is *behavior modeling*. The data collected from deep introspection are assimilated with user, system, workload, threat, and business behavior models. Known causalities among these behavior models as well as the models themselves allow the early detection of unusual behaviors. Being able to provide early warning of these abnormal behaviors from users, systems, and workloads, as well as various cyber security threats, is crucial for

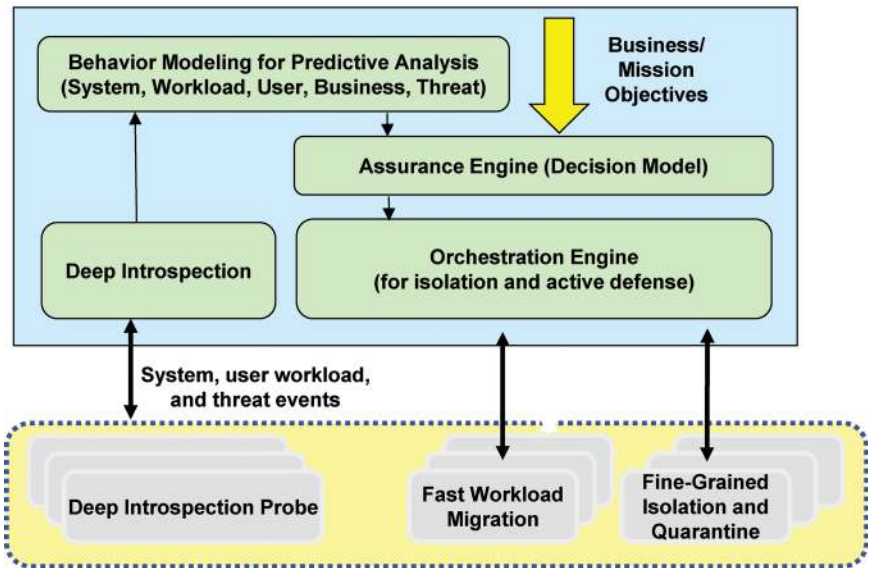
Behavior Modeling for Predictive Analysis (System, Workload, User, Business, Threat)
Business/ Mission Objectives
Assurance Engine (Decision Model)
Deep Introspection
Orchestration Engine (for isolation and active defense)
System, user workload, and threat events
Deep Introspection Probe
Fast Workload Migration
Fine-Grained Isolation and Quarantine

Figure 5

Continuous assurance for resiliency and security helps enable continuous deep introspection, advanced early warning, and proactive quarantine and orchestration for software defined environments.

taking proactive actions against these threats and ensuring continuous business operations.

Third, the *assurance engine* will continuously evaluate the predicted trajectory of the user, system, workload and threats and compare against the business objectives and policies to determine whether proactive actions need to be taken by the orchestration engine.

Finally, the *orchestration engine* receives instructions from the assurance engine and orchestrates defensive or offensive actions including taking evasive maneuvers as necessary. Examples of these defensive or offensive actions includes: fast workload migration from infected areas, fine-grained isolation and quarantine of infected areas of the system, server rejuvenation of those server images when the risk of server image contamination due to malware is found to be unacceptable, and IP (Internet Protocol) address randomization of the workload, making it much more difficult to accurately pinpoint an exact target for attacks.

Conclusion

As the industry is quickly moving toward converged systems of record and systems of engagement, enterprises are increasingly aggressive in moving mission-critical and performance-sensitive applications to the cloud. Meanwhile, many new mobile, social, and analytics applications are directly developed and operated on the cloud. These converged systems of records and systems of engagement will demand simultaneous agility and optimization, and will inevitably require SDEs for which the entire system infrastructure—compute, storage and network–is becoming software defined and dynamically programmable and composable.

In this paper, we described an SDE framework that includes capability-based resource abstraction, goal/policy-based workload definition, and continuous optimization of the mapping of the workload to the available resources. These elements enable SDEs to achieve agility, efficiency, continuously optimized provisioning and management, and continuous assurance for resiliency and security.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of Intel Corporation, Sun Microsystems, Apache Software Foundation, Facebook, Inc., Twitter, Inc., Netflix, Inc., Instagram, Inc., Microsoft Corporation, Rackspace, Inc., or Linus Torvalds in the United States, other countries, or both.

References

1. Data Center and Virtualization. [Online]. Available: http://www.cisco.com/en/US/netsol/ns340/ns394/ns224/index.html
2. R. Prodan and S. Ostermann, “A survey and taxonomy of infrastructure as a service and web hosting cloud providers,” in *Proc. 10th IEEE/ACM Int. Conf. Grid Comput.*, 2009, pp. 17–25.
3. RackSpace. [Online]. Available: http://www.rackspace.com/
4. Amazon Web Services. [Online]. Available: http://aws.amazon.com/

5. *IBM Cloud Computing Overview*, IBM Corporation, Armonk, NY, USA. [Online]. Available: http://www.ibm.com/cloud-computing/us/en/
6. *Cloud Computing with VMWare Virtualization and Cloud Technology*. [Online]. Available: http://www.vmware.com/cloud-computing.html
7. J. Temple and R. Lebsack, "Fit for purpose: Workload based platform selection," *J. Comput. Resourc. Manage.*, no. 129, pp. 20–43, Summer 2011.
8. S. E. Madnick, "Time-sharing systems: Virtual machine concept vs. conventional approach," *Modern Data*, vol. 2, no. 3, pp. 34–36, Mar. 1969.
9. G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures," *Commun. ACM*, vol. 17, no. 7, pp. 412–421, Jul. 1974.
10. P. Barman, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proc. ACM SOSP*, Oct. 2013, pp. 164–177.
11. E. Bugnion, S. Devine, M. Rosenblum, J. Sugerman, and E. Y. Wang, "Bringing virtualization to the x86 architecture with the original VMware workstation," *ACM Trans. Comput. Syst.*, vol. 30, no. 4, pp. 12:1–12:51, Nov. 2012.
12. C.-S. Li and W. Liao, "Software defined networks [guest editorial]," *IEEE Commun. Mag.*, vol. 51, no. 2, p. 113, Feb. 2013.
13. M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. Gude, N. McKeown, and S. Shenker, "Rethinking enterprise network control," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1270–1283, Aug. 2009.
14. D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proc. 2nd ACM SIGCOMM Workshop Hot SDN*, 2013, pp. 55–60.
15. W. Stallings. (2013, Mar.). Software-defined networks and openflow. *Internet Protocol J.* [Online]. *16(1)*. Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_16-1/161_sdn.html
16. *Security Requirements in the Software Defined Networking Model*. [Online]. Available: http://tools.ietf.org/html/draft-hartman-sdnsec-requirements-00
17. *ViPR: Software Defined Storage*. [Online]. Available: http://www.emc.com/data-center-management/vipr/index.htm
18. A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: Integration and load balancing in data centers," in *Proc. ACM/IEEE Conf. SC*, Piscataway, NJ, USA, 2008, pp. 53:1–53:12, IEEE Press.
19. G. Breiter, M. Behrendt, M. Gupta, S. D. Moser, R. Schulze, I. Sippli, and T. Spatzier, "Software defined environments based on TOSCA in IBM cloud implementations," *IBM J. Res. & Dev.*, vol. 58, no. 2/3, Paper 9, Mar./May 2014.
20. M. H. Kalantar, F. Rosenberg, J. Doran, T. Eilam, M. D. Elder, F. Oliveira, E. C. Snible, and T. Roth, "Weaver: Language and runtime for software defined environments," *IBM J. Res. & Dev.*, vol. 58, no. 2/3, Paper 10, Mar./May 2014.
21. W. C. Arnold, D. J. Arroyo, W. Segmuller, M. Spreitzer, M. Steinder, and A. N. Tantawi, "Workload orchestration and optimization for software defined environments," *IBM J. Res. & Dev.*, vol. 58, no. 2/3, Paper 11, Mar./May 2014.
22. G. Kandiraju, H. Franke, M. D. Williams, M. Steinder, and S. M. Black, "Software defined infrastructures," *IBM J. Res. & Dev.*, vol. 58, no. 2/3, Paper 2, Mar./May 2014.
23. A. Alba, G. Alatorre, C. Bolik, A. Corrao, T. Clark, S. Gopisetty, R. Haas, R. I. Kat, B. S. Langston, N. S. Mandagere, D. Noll, S. Padbidri, R. Routray, Y. Song, C.-H. Tan, and A. Traeger, "Efficient and agile storage management in software-defined environments," *IBM J. Res. & Dev.*, vol. 58, no. 2/3, Paper 5, Mar./May 2014.
24. C. Dixon, D. Olshefski, V. Jain, C. DeCusatis, W. Felter, J. Carter, M. Banikazemi, V. Mann, J. M. Tracey, and R. Recio, "Software defined networking to support the software defined environment," *IBM J. Res. & Dev.*, vol. 58, no. 2/3, Paper 3, Mar./May 2014.
25. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, Feb. 10, 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html
26. F. Darema-Rogers, G. Pfister, and K. So, "Memory access patterns of parallel scientific programs," in *Proc. ACM SIGMETRICS Conf. Meas. Model. Comput. Syst.*, 1987, pp. 46–58.
27. J. Haas and R. Wallner, "IBM zEnterprise systems and technology," *IBM J. Res. & Dev.*, vol. 56, no. 1/2, pp. 1–6, Jan./Feb. 2012.
28. "IBM Blue Gene," *IBM J. Res. & Dev.*, vol. 49, no. 2/3, Mar. 2005.
29. J. Mars, L. Tang, and R. Hundt, "Heterogeneity in "Homogeneous" warehouse-scalecomputers: A performance opportunity," *Comput. Architect. Lett.*, vol. 10, no. 2, pp. 29–32, Jul. 2011.
30. J. Moore, "System of Engagement and the Future of Enterprise IT: A Sea Change in Enterprise IT," in *AIIM White Paper*, 2012. [Online]. Available: http://www.aiim.org/~/media/Files/AIIM%20White%20Papers/Systems-of-Engagement-Future-of-Enterprise-IT.ashx
31. Oracle Exadata. [Online]. Available: http://www.oracle.com/us/products/database/exadata/overview/index.html
32. SAP Hana. [Online]. Available: http://www.saphana.com/welcome

Received October 3, 2013; accepted for publication October 29, 2013

Chung-Sheng Li *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (csli@us.ibm.com).* Dr. Li's research interests include cloud computing, security and compliance, digital library and multimedia databases, knowledge discovery and data mining, and data center networking. He has authored or coauthored more than 130 journal and conference papers and received the best paper award *from IEEE Transactions on Multimedia* in 2003. He is both a member of IBM Academy of Technology and a Fellow of the IEEE. He received a B.S.E.E. degree from National Taiwan University, Taiwan, R.O.C., in 1984, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1989 and 1991, respectively.

Brad L. Brech *IBM Systems and Technology Group, Rochester, MN 55901 USA (blbrech@us.ibm.com).* Mr. Brech's development career includes work in logic design, virtualization, communications, system architecture, and energy-efficient computing. He is known through several product releases, papers and presentations, and his industry work. He is a member of IBM Academy of Technology and Board member of The Green Grid. He received a B.E.E.E. degree from Steven's Institute of Technology in 1982.

Scott Crowder *IBM Corporation, Armonk, NY 10504 USA (scrowder@us.ibm.com).* As Vice President, Technical Strategy at IBM, Dr. Crowder is responsible for supporting the development of technical and business strategy for IBM's growth initiatives, including Smarter Planet*, Business Analytics and Optimization, Cloud, and Systems Leadership. He was previously a development executive at IBM's Semiconductor Research and Development Center. He is an author or co-author of 14 patents and 16 technical papers. He received his Ph.D. degree in electrical engineering from Stanford University in 1995 and has master's degrees in economics and in electrical engineering and bachelor's degrees in international relations and electrical engineering.

Daniel M. Dias *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA.* Dr. Dias is director of cloud innovation technologies at the IBM T. J. Watson Research Center. Prior to this, he was the inaugural lab director of IBM Research—Brazil, with responsibility for leading and creating IBM's first research lab in the southern hemisphere. Earlier, he led the internet infrastructure and computing utilities department at the IBM T. J. Watson Research Center, where he and his team built one of the first scalable and highly available web servers, with key patents and papers, forming the basis for IBM offerings and products in this area. Dr. Dias is co-inventor on more than 45 issued U.S. patents, and coauthor of more than 90 refereed publications. He received his M.S. and Ph.D. degrees from Rice University, Houston, Texas, both in electrical engineering. He is an IBM Distinguished Engineer, a member of the IBM Academy of Technology, and a Fellow of the IEEE.

Hubertus Franke *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (frankeh@us.ibm.com).* Dr. Franke is a Research Staff Member and Senior Manager for Software Defined Infrastructures at the IBM T. J. Watson Research Center. He received a Diplom degree in computer science from the Technical University of Karlsruhe, Germany, in 1987, and M.S. and Ph.D. degrees in electrical engineering from Vanderbilt University in 1989 and 1992, respectively. He subsequently joined IBM at the T. J. Watson Research Center, where he worked on the IBM SP1/2 messaging, scalable operating systems, multi-core architectures, scalable applications, and most recently cloud platforms. He received several IBM Outstanding Innovation Awards for his work. He is an author or coauthor of more than 30 patents and over 100 technical papers. He is a Master Inventor and a Member of the IBM Academy of Technology.

Matt Hogstrom *IBM Software Group, Research Triangle Park, NC 27709 USA (hogstrom@us.ibm.com).* Mr. Hogstrom is an IBM Distinguished Engineer In his current role, Mr. Hogstrom is the CTO (chief technology officer) for the architecture and strategy for software defined environments (SDE). His prior roles included architect of the IBM Workload Deployer (preciously known as WebSphere CloudBurst Appliance), chief architect of WebSphere AppServer Community Edition, and architect of the WebSphere performance team across all runtime environments.

Dave Lindquist *IBM Software Group, Tivoli, Durham, NC 27703 USA (lindqui@us.ibm.com).* Mr. Lindquist, an IBM Fellow and Vice President, is an IBM Software Group CTO (chief technical officer) responsible for the strategy and architecture of IBM's Cloud technology (SmartCloud* software), Smarter Infrastructure, and Integrated Service Management solutions. He has more than 25 years of experience as a global technology leader in cloud computing, management systems, web infrastructure, networking, and large systems architecture. Appointed an IBM Fellow in 2007, IBM's highest technical honor, Mr. Lindquist began his career with IBM in the Server Group, specializing in large-systems architecture, performance and database systems. In 1990, he joined the IBM Software Group, where he has focused on cloud, web infrastructure, content delivery networks, mobile and wireless technology, and Internet products. His research has led to 48 patents, recognition as an IBM Master Inventor, and election into the IBM Academy of Technology.

Giovanni Pacifici *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (giovanni@us.ibm.com).* Dr. Pacifici joined IBM Research in 1995. He currently serves as the Director of Distributed Computing, leading a research program on foundation technologies for cloud computing, information, and service integration. His personal research interests include language and tools for representing and manipulating workload components and the use of analytics to optimize and manage workloads. Dr. Pacifici's research led to the creation of a number of IBM products including PureApplication* System, IBM Workload Deployer, WebSphere Virtual Enterprise, and Deployment Planning and Automation delivered as an extension of Rational Software Architect. Dr. Pacifici served as Technical Program Co-Chair of IEEE INFOCOM (International Conference on Computer Communications) in 2001 and is a past editor of the *IEEE/ACM Transactions on Networking*, the Elsevier Science's *Computer Networks Journal*, and two special issues of *the IEEE Journal on Selected Areas in Communications*. He received a Laurea in electrical engineering and a research doctorate in information science and telecommunications from the University of Rome "La Sapienza." Dr. Pacifici is a senior member of IEEE, a member of ACM, and a member of the IBM Academy of Technology.

Stefan Pappe *IBM Global Technology Services, Cloud Architecture and Solution Design (pappe@de.ibm.com).* Dr. Pappe is an IBM Fellow and an expert in cloud computing and asset-based services. He currently oversees the architecture and design of IBM Global Technology Services' cloud offerings and client solutions in IBM's Global Technology Services. Dr. Pappe received a master's degree in economics from University of Karlsruhe, Germany, in 1986, and a Ph.D. degree in computer science from University of Kaiserslautern, Germany, in 1990. Dr. Pappe spent most of the 25 years of his IBM career fueling the services business through technical advancements. His assignments ranged from infrastructure to business services, including both project and managed services. He is author or coauthor of several patents and technical papers, including the "IBM Cloud Computing Reference Architecture," a comprehensive technical blueprint guiding cloud design and delivery.

Bala Rajaraman *IBM Software Group, Durham, NC 27703 USA (balar@us.ibm.com).* Dr. Rajaraman has been with IBM since 1992 and is currently a Distinguished Engineer with responsibilities including the strategy, architecture, and design of Cloud and Service Management solutions. He is the Chief Architect for the PaaS (platform as a service) aspects of the Next Generation Cloud Architecture. He is involved in the strategy, architecture, and design of several products for cloud and service management and supports the strategy and implementation of solutions at several large clients. In the recent past, he has been responsible for various aspects of Cloud solutions and DevOps; service management including ITIL (Information Technology Infrastructure Library), process automation and integration, and provisioning and automation tools; and performance of IBM System z communication stacks. His areas of professional interest include communications technologies, systems performance, autonomic computing, and service and systems management. He holds over 25 patents in these areas. He received a Ph.D. degree in computer engineering from Clemson University, South Carolina, and a master's degree in computer engineering from Drexel University, Philadelphia.

Josyula Rao *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (jjrao@us.ibm.com).* Dr. Rao leads Security Research at IBM. Based in IBM's Thomas J. Watson Research Center, the global team comprises over a hundred researchers who work in the areas of cryptography, cybersecurity, cloud, mobile security and secure platform technologies. Dr. Rao works closely with customers, academic partners and IBM business units to drive new and innovative technologies into IBM's products and services and definitive industry standards. The goal of his research is to significantly "raise the bar" on the quality of security while simultaneously easing the overhead of developing and deploying secure solutions. Dr. Rao has published widely in premier security conferences and workshops and holds numerous U.S. and European patents. He is a member of both the Industry Advisory Board of the Georgia Tech Information Security Center and the prestigious IBM Academy of Technology, as well as emeritus member of the IFIP (International Federation for Information Processing) Working Group 2.3 (Programming Methodology). Dr. Rao obtained his doctorate degree from the University of Texas at Austin in 1992, an M.S. degree in computer science from the State University of New York at Stony Brook, 1986 and a B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, 1984.

Radha P. Ratnaparkhi *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (radha@us.ibm.com).* Ms. Ratnaparkhi is the Vice President for Software Defined Environments at IBM Research where she leads a global team of over 100 researchers worldwide. Just prior to this, she was the Vice President for IT and Wireless Convergence at IBM Research where she was responsible for defining the IBM strategy in this space. Previously, she was the Director of Commercial Systems where she led the research initiative in the area of Cloud Computing. She has also led and managed the commercialization efforts of a high-end text analytics solution WebFountain. Her experience at IBM further includes her development leadership effort for IBM's flagship database product DB2* on the mainframe. Prior to IBM, she led the Java products development team at Informix Software. Her experience at Informix spans all products, namely, Informix Dynamic Server, Connectivity, and Tools products. She started her career in Mumbai, India with Tata Consultancy Services (TCS)—India's premier services consulting firm, after completing her Masters of Technology degree from the Indian Institute of Technology (IIT) in Delhi.

Rodney A. Smith *IBM Software Group, Atlanta, GA 30328 USA (rod.smith@us.ibm.com).* Mr. Smith is an IBM fellow and Vice President of the IBM Emerging Internet Technologies organization. He leads a group of highly technical innovators who are developing solutions to help organizations realize the value of big data. His early advocacy in the industry has played an important role in the adoption of technologies such as J2EE (Java 2 Platform Enterprise Edition), Linux**, web services, extensible markup language), rich Internet applications, and various wireless standards. He is helping lead IBM's efforts around big data analytics and the application of IBM Watson*-like technologies to business solutions, helping companies make better decisions more quickly for improved business outcomes. He is a computer science graduate of Western Michigan University, and holds M.A. and B.A. degrees in economics with a concentration in math from Western Michigan University.

Michael D. Williams *IBM Systems and Technology Group, Poughkeepsie, NY 12601 USA (mdw@us.ibm.com).* Mr. Williams is a Distinguished Engineer and member of the IBM Academy of Technology. He joined IBM in 1989 after graduating *magna cum laude* from the State University of New York at Oswego with a bachelor's degree in computer science. Throughout his career, he has had a broad set of client-facing and development responsibilities in the area of systems and software. His current focus is software architecture and design for IBM's next generation systems with an emphasis on software defined environments, cloud, and virtualization.