Evolution of the IBM Cloud: Enabling an enterprise cloud services ecosystem

A. Kochut
Y. Deng
M. R. Head
J. Munson
A. Sailer
H. Shaikh
C. Tang
A. Amies
M. Beaton
D. Geiss
D. Herman
H. Macho
S. Pappe
S. Peddle
R. Rendahl
A. E. Tomala Reyes
H. Sluiman
B. Snitzer
T. Volin
H. Wagner

Cloud computing is a new paradigm that is transforming the information technology (IT) industry and reshaping the way enterprise services are developed, deployed, sold, delivered, and consumed. Instead of managing complex IT systems, customers can focus on the core competence of their enterprise while obtaining all required IT functions as a service. From the perspective of a cloud provider, remaining competitive and realizing full potential of economies of scale that the cloud paradigm promises require extreme levels of standardization, automation, and optimization. This paper describes the evolution of the Common Cloud Management Platform (CCMP), a management system providing business and operations support for cloud services. We cover its initial implementation and applications, discuss the latest challenges faced when adapting enterprise solutions to the cloud, and introduce the exploratory research topics to which this work led. We address the business services aspects, including framework-based integration of the catalog, and customer and revenue management, as well as the operational aspects, including novel approaches for scalable virtual machine provisioning and adaptive workload placement optimization. We discuss architecture, design, and implementation details of key CCMP components and highlight the challenging aspects of providing such architecture while promoting scalability, modularity, and reuse.

Introduction

Widespread adoption of virtualization, ubiquitous broadband network access, and a rapidly growing Internet user base led to the emergence of the cloud computing model [1] in which information technology (IT) functions are sold and delivered as a service from large-scale hosted infrastructures. Delivered in this manner, functionality can be either at the infrastructure level, at the platform level, or at the application level. Several vendors in the market offer, for instance, Infrastructure as a Service (IaaS) [2, 3], where users can obtain virtual machines (VMs) on a pay-as-you-go basis. IBM offers IaaS services for enterprise customers [4]. This service allows users to obtain almost instantaneous access to significant compute power with no capital investment. IaaS allows small service providers to rapidly scale out and handle surges in service demand, thereby radically lowering barriers for market entry and spurring new innovation. The term *scale out* refers to increasing system capacity by adding new nodes rather than, as is the case with *scale up*, replacing the system with another one having higher capacity. In recent years, scale-up growth has been limited and has led to increased interest in alternative scale-out approaches such as cloud computing. Platform as a Service (PaaS) provides access to middleware such as a database or application server. An example is DB2* as a service, in which customers can connect to a database instance that provides required capacity and performance characteristics while a service provider assumes responsibility for running and managing the system. Software as a Service (SaaS) is another model of cloud computing. An example of SaaS is IBM LotusLive* [5], which provides access to web conferencing as a subscription-based service. Instead of installing and managing a dedicated Lotus server, customers can instantly obtain the required functionality and scale as their end-user population changes.

Digital Object Identifier: 10.1147/JRD.2011.2170920

0018-8646/11/$5.00 © 2011 IBM

The above examples illustrate the significant potential of the cloud computing paradigm to radically alter the IT industry. When applied to enterprise workloads, principles of economies of scale and automation can result in radical simplification, quality improvements, and cost reduction. However, delivery of large-scale enterprise IT service has been traditionally a labor-intensive task, requiring significant capital investments and usually involving complex processes. There are a few examples of the application of the cloud paradigm to the enterprise services domain, and there are several reasons for the difficulty of migrating enterprise workloads to the cloud, i.e., much stricter security requirements, service-level agreement demands, and expectation to customize the service and related business processes to enterprise customers' needs. These requirements generally reduce the potential for resource sharing and weaken economy-of-scale effects. In addition, adapting enterprise services to deliver SaaS is a very challenging task. It requires either reengineering the applications or extending the cloud platform to expose more flexible interfaces that better support legacy applications. Moreover, interoperability of enterprise cloud services must be enabled, both in terms of operational service composition and providing business support functions in a consistent, reusable, and automated manner. These, in turn, enable an enterprise cloud services ecosystem.

We view a cloud services ecosystem as a logical progression of the cloud model itself. While the cloud is intended to lower the costs of service delivery on a service-by-service basis, the ecosystem is intended to lower both service management and transactional costs among services, enabling composite services to be delivered at a lower cost. For example, a SaaS solution can offer rapid customer on-boarding (a process in which new customers begin to use the system) by making use of PaaS, which is, in turn, enabled by elastic compute and storage infrastructure services provided by IaaS. In the ecosystem, services are able to consume services from one another, enabled not simply by programmer-coded interoperability but through on-the-fly business interconnections. Moreover, through semantically rich offering definitions, offering managers are able to develop new assemblages of services, where the assemblages are logically described instead of through low-level programming. These logical descriptions are interpreted by infrastructure services, which provide runtime composition and scaling.

IBM offers several enterprise-class cloud services [4], including IBM SmartCloud* Enterprise, IBM SmartCloud Desktop and IBM LotusLive Collaboration Suite. These services are being integrated into the underlying management system called Common Cloud Management Platform (CCMP). The key elements of CCMP are Business Support Services (BSS), Operations Support Services (OSS), front-end portals, and access application programming interfaces (APIs) (see **Figure 1**). BSS implements all of the functions related to the business lifecycle of the cloud service, such as catalog, customer, and entitlement management and usage accounting, rating, and billing. OSS implements all operational functions needed to effectively manage data center infrastructure, such as virtualized servers, storage, and networking equipment. It realizes the services by provisioning and maintaining all of the required infrastructure and software components. For example, in the case of SmartCloud Enterprise, BSS manages the image catalog that includes all VM software stacks available for purchase, accounts for VM and storage usage, and provides both web-based and APIs to order and manage the cloud resources. The OSS provisions and deprovisions VMs and storage shares, manages resource allocation, enforces security and isolation, and performs all other tasks related to data center management. The following section provides a more detailed overview of CCMP functions and implementation.

CCMP, as described above, provides a platform supporting IaaS cloud services. However, it is only a first step on the path to cloud adoption in the enterprise context. The longer term objective of delivering a variety of complex enterprise services from the cloud requires extensions and modifications to CCMP. This paper discusses a set of key extensions that spans both the BSS and OSS aspects of CCMP. In the case of BSS, it is crucial to expose its functionality as a service such that cloud services hosted on CCMP can reuse all of the functionalities related to offering, customer, entitlement, and revenue management. This objective is realized by a compositional BSS framework. Another aspect addressed is the ease of the service lifecycle management. To this end, we propose an ontology-based service composition and a graph-based service representation for creation, deployment, and delivery management. In the case of OSS, we propose an extension to virtual-disk streaming technology that allows rapid deployment of VM instances and also reduces the disruption experienced by other network flows in the data center. We also discuss resource allocation aspects, focusing on the effects of workload predictability and the time and cost of resource reallocation actions on the effectiveness of dynamic resource allocation strategies for the cloud. This provides important insights into the design of cloud resource managers.

The remainder of this paper first discusses architecture and implementation highlights of the current CCMP release. Subsequently, we outline key research contributions to challenges identified while working on providing scalable and efficient enterprise services from the cloud.

Common Cloud Management Platform

CCMP was established to provide a basis upon which various IT cloud solutions could be used to build the required functions. By providing a common set of composable

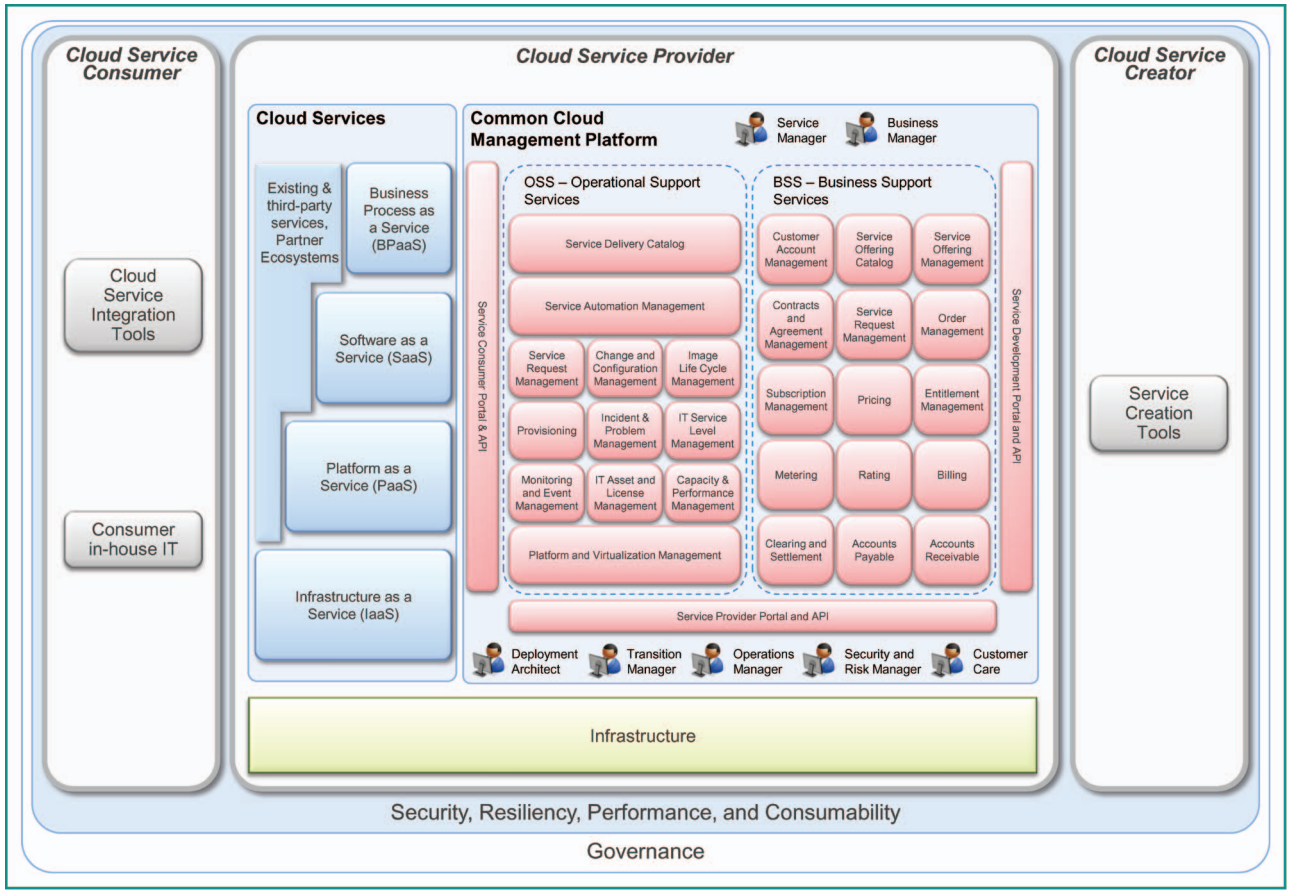

Figure 1

Architecture of CCMP. (The figure is adapted from the IBM CCMP Reference Architecture [6].)

support components, CCMP is able to provide common functionality and integration points across clouds, consistent consumer experiences with the common services, streamlined audit support (due to common code and interface support), shared awareness of the customer across multiple cloud services (such as enterprise accounts and users), and base cloud operational services on which to build other cloud capabilities on. As shown in Figure 1, the CCMP includes two major groups of functions, namely, BSS and OSS, with a surrounding set of interfaces (both graphical user interface and API) to support various cloud roles across the Cloud Developer, Cloud Provider, and Cloud Consumer domains.

Business Support Services

The current CCMP BSS functionality was developed for two IBM enterprise cloud offerings, namely, the LotusLive collaboration services, and SmartCloud Enterprise where customers can use IBM software products on a per-hour basis. Providing BSS for enterprise services entails significant complexities, as we detail below. Our descriptions cover selected BSS services in Figure 1.

Contract-based purchasing

Using contract-based purchasing (*Customer Account Management* and *Contracts and Agreement Management* in Figure 1), rather than purchasing from a standard catalog as individual consumers do, enterprises may purchase services through contracts. Generally, contracts specify the offerings subscribed to and their prices, including any one-time charges (e.g., on-boarding services). Contract-based purchasing has implications on catalog management, i.e., each customer has its own catalog, prices, and price expiration dates. Contract-based purchasing also has implications for rating (calculation of billing charges), i.e., each customer may require a personalized set of algorithms to be performed for its bill. The catalog management and customer account management tools in the Service Provider Portal for SmartCloud Enterprise implement these enterprise-oriented requirements.

Additionally, SmartCloud Enterprise supports self-service image-specific instance customization by making use of an asset management service. The asset manager stores content files and metadata about each image, including a

description of the parameters that the Service Consumer Portal presents to the user when an instance is requested. As an example, the image-specific parameters for a database server image include credentials for a user account that should control the database service. The image metadata also includes a "getting started" guide that is presented in the portal to the user alongside instances provisioned from that image.

The catalog further supports three levels of image privacy relevant for enterprise customers, i.e., private, enterprise-wide, and public. Images in the private catalog are owned by and visible to a single user in the system, and that user has full self-service control over the images therein. Enterprise-wide catalog supports open collaboration among all users associated with a particular enterprise customer account. Images in the public catalog are visible to all users, but changes to this catalog are strictly limited to the catalog manager.

Subscriber management

Using subscriber management (*Subscription Management* and *Entitlement Management* in Figure 1), the IBM enterprise cloud BSS accommodates enterprise requirements for subscriber management, such as on-boarding subscribers in large batches and enabling customer staff to manage subscription resources rather than individuals. In SmartCloud Enterprise, individuals can self-create and self-manage their instances, but only within limits set by their company's subscription manager. BSS is responsible for maintaining the contracted limits and tracking each user's usage within these limits.

Back-office integration

Using back-office integration (*Billing*, *Clearing and Settlement*, *Accounts Payable*, and *Accounts Receivable* in Figure 1), revenue from cloud services is, in the end, no different than revenue from other lines of business such as software sales and service engagements. As such, BSS for cloud services must adapt to preexisting taxonomies and data rules of the enterprise offering them. As cloud services delivered to an enterprise may be only one set of dealings with that enterprise, BSS must be able to obtain and use preexisting customer data of an enterprise. In addition, BSS must be able to provide adaptation to existing back-office functions of the service provider. It should enable existing rules for customer contracts to be structured in a way that ensures consistent financial measurement, promotes transactions to the back office using preexisting financial rules and coding schemes, and enables the cloud business office to handle exceptions that occur during the promotion of transactions to the back office.

Order management

The *Order Management* component in Figure 1 integrates with the OSS via an adapter layer that communicates with the OSS Service Request Management component to submit requests and receive the results of those requests.

Operations Support Services

The OSS component is responsible for the management of resources within the CCMP. This includes the assignment of resources, both physical and virtual, as well as reporting and monitoring of the resources. Typical resources managed by an OSS are central processing unit (CPU), memory, and disk and network resources such as *Internet Protocol* (IP) addresses and virtual local area networks. We discuss in detail a subset of key OSS services depicted in Figure 1.

Service Automation Management is responsible for orchestrating the other various components of the OSS. Service templates are used to model a variety of services running on top of CCMP. They can represent low-level IaaS services, offering compute power, high-level SaaS offerings, and other services with varying levels of complexity. Each service template contains context data needed for orchestration of the other OSS components. Provisioning, deprovisioning, or management actions for the services are all expressed as workflows. They specify the control and data flows of a management action.

Platform and Virtualization Management for compute resources is a pool of hypervisors managing guest instances. Most of the network resources are managed by the virtualized infrastructure for compute resources; however, for complex network architectures, a separate component controlling the network infrastructure is necessary. *Platform and Virtualization* Management for compute resources is responsible for managing the runtime access of guest instances to the physical CPU, memory, and local disk resources of the hosts. In addition, this component offers APIs to create, change, or delete guest instances. The interface is used by the BSS layer to realize users' requests.

Image Life Cycle Management is responsible for storing the binary content of the disk representing the image, as well as the metadata needed for a correct provisioning process. Examples of metadata include the operating system type and special OS driver information; hence, the hypervisor and the running guest OS can optimize the runtime performance of the guest.

Monitoring and Event Management is responsible for monitoring of the physical and virtual resources of the service. When a physical resource is down or service is degraded, monitoring provides CCMP with information required to mitigate the impact. This can result in a ticket inserted into a ticketing system so that operators can analyze and fix the root cause. An event can also trigger management plans of the Service Automation, leading to an automated change in the Platform and Virtualization.

Capacity and Performance provides data needed to control the cloud from a provider's perspective. Daily reports of used resources, provisioning, and runtime data allow the provider to, among other actions, plan and adjust capacity and isolate and resolve problems. It also implements resource allocation and optimization strategies.

Providing a scalable enterprise services cloud ecosystem

Making the enterprise services cloud more efficient, scalable, and cost effective requires innovation and extensions to the basic cloud paradigm. Several challenges have been identified in this space during the development of CCMP detailed above. One of the key distinguishing features of enterprise cloud services is the requirement for complex BSS functionality. In order to provide BSS in an easily reusable manner, we propose a modular BSS framework that promotes reuse of BSS functions and provides a foundation layer for service integration and compositionality. It exposes key BSS functions as a service and makes it possible for cloud services to compose their BSS functions and integrate them with external services, as well as within the cloud platform itself. As a result, functions such as rating, billing, and reporting can be provided in a consistent and cost-effective manner.

A critical BSS function in an enterprise cloud is offering management. As cloud services evolve, there needs to be a robust mechanism to manage their lifecycle, including creation, modification, deployment, and delivery, and also to enable fast composition of complex enterprise cloud services. We propose complementary graph-based and ontology-based service representations that allow expressing the dependencies between the services, the mapping to IT resources in data centers in view of cost assessment, the processes to provision and delete them in the data center upon customer request, and other related features. This component is key in achieving a cloud services ecosystem.

In the area of service delivery and data center management, it is critical for cloud providers to ensure that the physical resources are appropriately assigned as to meet all of the stringent enterprise service-level agreements while minimizing running costs. We present two aspects of this issue, i.e., one related to rapid VM provisioning and the other related to dynamic resource reallocation for previously provisioned services. Rapid provisioning of VMs requires transferring of significant amounts of data and, if not properly executed, can result in poor provisioning performance and can also adversely affect other network transfers. We propose a fast virtual disk (FVD), which makes it possible to rapidly start new VMs and transfer all of the required data in a controlled fashion, thereby reducing disruptions.

Another challenge, once resources are provisioned, is to perform resource reallocation in the face of dynamic workload changes. The strategies for reallocating resources must be robust with respect to workload variability and capable of determining which resource reallocation actions are most likely to give lasting benefit to systems performance. For this purpose, we formulate a dynamic VM reallocation problem and propose a stochastic model to estimate the gain from VM reallocation given the properties of the workload and virtualization infrastructure. The model considers variability and predictability of VM workload, as well as time and cost of resource reallocation actions.

Compositional BSS framework

To enable service providers to more easily commercialize their offerings and to lower the transaction costs of service providers consuming the services of other service providers, the ecosystem we envision provides a framework of BSS functions as a core service of the ecosystem services platform. Built on top of the CCMP BSS components, this BSS framework provides a service with four main attributes.

The first attribute is *multitenancy*. Service providers are themselves customers and are represented as independent business entities. The second attribute is the *online attribute*. The system accepts new core parties and entities, i.e., service providers, service products, offerings, offers, and customers, and configuration of these core entities, in an "online" fashion, with no need for restart.

The third attribute involves *service-provider APIs*. To enable service providers to make BSS requests programmatically, the framework provides APIs that provide secure authenticated access. These may be used for one service to automatically subscribe to another service for more resources or for a service to check entitlements for individual subscribers.

The final attribute is *self-service configuration*. Service providers are provided with tools (such as a web user interface) that enable them to configure the BSS functions for their service products themselves. Because BSS encompasses a wide range of functions, the BSS framework will provide a variety of self-service configuration tools. These tools configure the BSS components shown in Figure 1. These tools reflect the fact that while BSS for different cloud services all involve the same basic entities of customers, contracts, subscriptions, offerings, invoices, etc., the nature of these will differ from one service provider to another. Thus, the service-provider tools in the BSS framework will enable providers to define templates or other forms of definition for these core entities.

The BSS framework consists of the key components described in the following subsections.

Profile manager

Service providers configure account information, define authorization roles, and assign users to them. The BSS framework provides standard roles such as *Offer Manager*,

which are used to enable/disable functions of the configuration tools. Roles are also used in the process service and in the tasks service, as described below. Service providers can define roles that are specific to their service and define processes and tasks that relate specifically to those roles.

Service registration

Service providers use the service registration tool to register their products (including the data on service components to which the BSS will provide provisioning events) and to define the various offerings of the products.

Rating plan developer

Service providers will be able to select from a range of pricing plans and configure the parameters of each plan.

Catalog definition

Service providers may define a range of catalogs, including those used for development and testing, and those used for staging and production (i.e., the catalog that customers see). It provides presentation-level functions for defining catalog structure and functions serving lifecycle management, such as a function for exporting offerings from one catalog to another. Once a catalog is defined, this tool enables service providers to populate it with offerings.

Billing process manager

With the Billing Process Manager tool, the service provider defines how sets of charges are presented for payment. Of particular importance is the definition of how the billing process connects to the service provider's back-office functions, as described earlier.

Contract developer

The Contract Developer tool enables service providers to define electronic contracts for their services, which can be filled and submitted online by service-provider staff. Contract templates may contain a provider's standard terms and would include a process defined by the provider (see below) for approving the contract.

Process manager

Many BSS functions have procedural elements such as customer on-boarding, approval of new offerings, migrating offerings from staging to production, or obtaining approval of charges before invoicing. These processes may vary from one service provider to another. The BSS framework, therefore, provides, as an integral service, a process service and associated scripting and monitoring tools that enable service providers to script processes and attach them to various events in the system. For example, a service provider may specify that a new-customer registration be first checked against a denied-party list and then sent to a manager for approval.

Report developer

The Report Developer enables service providers to select from a range of predefined reports, or to define new reports, to be used in processes.

Task manager

Many BSS processes involve human tasks. The framework provides a service that enables processes to create role-specific tasks and to wait for users to complete tasks, and then to act according to the outcome of the task.

The purpose of providing these self-service define-your-own-BSS tools, and our vision for the BSS framework, is to provide BSS functions that are as accessible as the simple billing services offered by Amazon, Google, or Microsoft but provide the flexibility and features required for enterprise-class services.

Graph-based representation of services

Our graph-based representation of the services, as well as their related business processes and constraints, addresses critical limitations of manual business support processes. It enables 1) a hierarchical visualization of the features of catalog services from IT and business subcomponents and dependencies on operation constraints and their mapping to prospective operational infrastructure; 2) optimization of availability through resource assignment into one or more data centers from which, eventually, the service will be provisioned at delivery time; 3) evaluation of, at service deployment time, the service costs for various customer requirement models and locations before actual subscriptions occur; 4) matching of actual customer service requirements against the resource types and availability of data centers; and 5) selection of valid data centers for customer provisioning.

An example catalog service graph based on our representation is illustrated in **Figure 2**. Such a service catalog graph can be presented for manipulation in the user interface in the same way Rational* software tools present the software plug-ins for composition. However, in addition to what software composition tools offer, the building blocks relate to an operational infrastructure in data centers, whereas the service deployment and potential resources assignment into the infrastructure is done by optimizing the placement of service components in data centers (more details in [7]). Blue ovals in Figure 2 represent composite offerings. Green ovals represent simple offerings, and pink ovals represent business offerings. A service developer may specify scalability rules. For instance, in the web hosting case where the deployment requirements vary with the load increase, the scalability rules express the mapping of increasing service loads to various IT resource types.

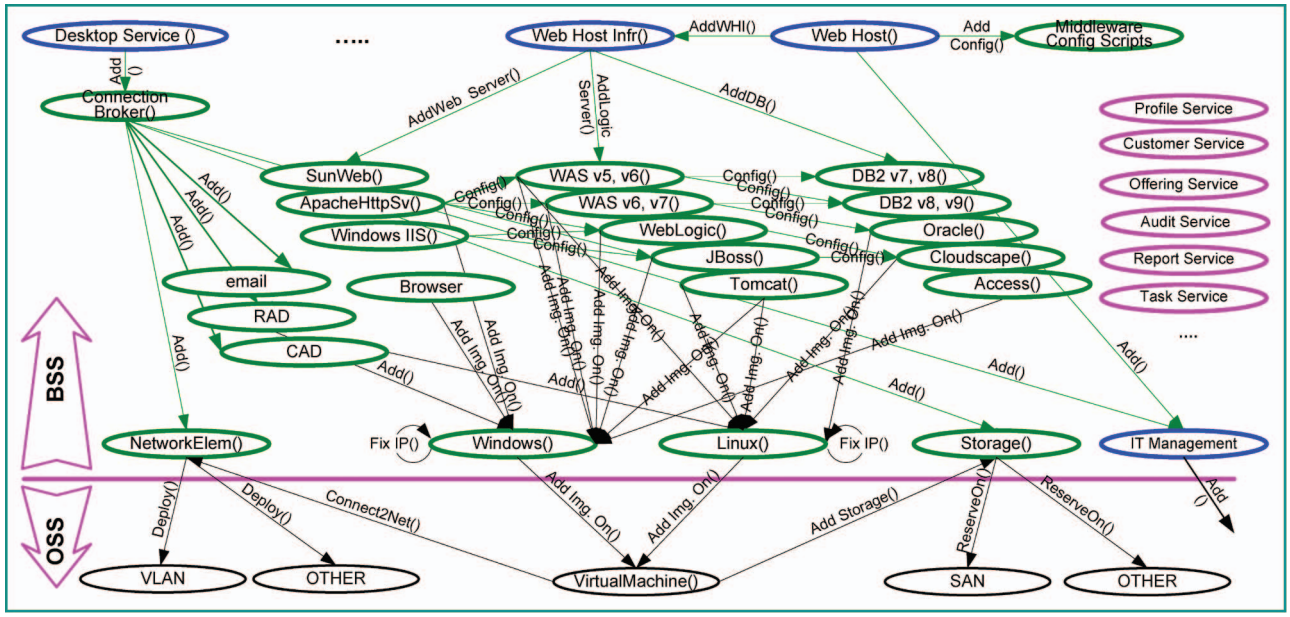

Figure 2

Example of a service catalog as a graph for service creation support. (RAD: Rational Application Developer; CAD: computer-aided design; VLAN: virtual local area network; web host: web hosting service; web host infr: web hosting infrastructure; SAN: storage area network.) (The figure is adapted from [7].)

Additional IT expert knowledge may be reflected in the graph as limitations, for instance, known service dependencies such as "web application server *M* version *a.b.c* works with database server *N* version *x.y.z*" or "web application server *M* version *a.b.c* works on Linux** SUSE** version *n.m*." The green Config() arrows in Figure 2 indicate such configuration dependencies. Lack of an arrow indicates either unfeasible or unsupported dependence. Configuration conflicts are another potential type of limitations. Examples of software conflict constraints are "Windows Defender* that has issues on Windows Vista*" or "WebSphere* AS v5 on an AIX* operating system that conflicts with Oracle Web Services Manager." For brevity, note that not all valid dependencies and constraints are shown in Figure 2.

Service creation

To illustrate the creation of a new service, let us consider the creation of a desktop service. We can make use of existing OSS and BSS services as building blocks for its subcomponents. Desktop service is a new way of delivering virtual desktops where VMs executing desktop operating systems run in a remote data center operated by a service provider. The key component of the desktop service is its connection broker. It is a management component mediating user connections, establishing identity and entitlements, and managing back-end infrastructure in response to changes in user activity and demand for desktop services. The connection broker requests new VMs from the OSS running actual desktops (and later assigned to users). The decision regarding where to place those desktop VMs is based on multiple factors including network latency and throughput between the data center and user's location, required CPU and memory, security and governance constraints, and also the cost considerations.

Although there is significant uniqueness to the desktop service, it can still reuse BSS framework services as exemplified in the previous section. In addition, most of the OSS requests of the connection broker (such as VM provisioning, operating system deployment, and virtual local area network setup) can be fulfilled using the basic OSS services. Thus, a new service creation implies a new service description materialized as graph links into existing subservices and new requirements, as described in the examples above. For example, customer on-boarding, account activation, rating, billing, and cost accounting can be composed based on existing BSS services.

Service deployment

The deployment of a new service involves its potential assignment to resources into one or more data centers from which, eventually, the service will be provisioned at delivery time upon customer subscription to that service. It is also the step at which the service can become available in the BSS Services Catalog in case of successful resource assignment. Typically, a service provider manages services

across multiple data centers. The resource assignment identifies data center(s) equipped with the appropriate type of resource to provide the service, regardless of whether utilized or not. (Resource availability will be verified at provisioning time.) The IT requirements populated during a new service creation are subjected here to the IT constraints of the data centers.

Service subscription

Publishing a new service in the BSS Services Catalog makes it available for customers to subscribe to it. The actual placement and provisioning take place at that time. An optimal resource allocation is critical to service performance and cost savings. Upon service subscription, the values of the customer requirements and the costs associated with the different aspects of the service are collected. Once all the service requirements are acquired, we can proceed with the matching process against the resource types and availability of the data centers. The valid data center (if any) with minimum cost is selected for service placement and provisioning.

The goal of our optimization formulation presented in [7] is to allocate resources to a given new service (at service deployment time) and to its customers' subscriptions (at delivery time) in one of multiple data centers. The constraint that we place upon the allocation is that the allocation cost should be minimized while the data center and services requirements are matched. Our future research plans include extending the model with the scalability rules capturing nonlinear (with respect to load) service requirements.

The catalog service graph presented above lacks catalog user-action representations and their related deployment processes that would trigger actual provisioning of resources in response to a customer subscription. An ontology-based solution that addresses this limitation is described below.

Ontology-based catalog representation

In today's marketplace for cloud services, service providers may offer packages, or compositions, of cloud services. An example is IBM LotusLive, in which customers may buy combinations of email, meeting, and collaboration services (as mentioned, we call such combinations composite offerings). When such offerings are purchased, operations required to provision them may need to be sequenced in a certain order due to interservice dependencies. Because these combinations are defined by the service provider, the operations involved in provisioning them can be anticipated and controlled. However, as the cloud service marketplace grows and the number and variety of services increase, it becomes desirable to enable other parties such as resellers or customers themselves to define the available combinations. In this case, it becomes problematic to ensure that the combinations are correctly provisioned. Here, we describe a mechanism based on knowledge-representation technology that enables provisioning and other operations (e.g., upgrade) on top of complex service offerings to be correctly sequenced.

We propose an ontology-based approach to formally model the service offerings and associated operations while paying attention to system robustness. Ontology has been adopted in many services-related areas. For example, the Web Service Modeling Language (WSML) [8] provides a syntax and semantics for modeling a web service. Recently, Youseff et al. [9] have proposed a unified ontology for cloud computing to classify cloud services.

In our ontology, we model cloud service offerings with common requirements on catalog actions such that service providers or even customers can easily manage their catalogs of service offerings. One of the features we model is the composability of cloud service offerings. As mentioned, *composability* refers to the ability to compose a service offering from one or more other offerings, taking operational support into account. By examining common actions (e.g., on-boarding, subscription, and upgrade) taken by a user in a catalog across cloud service offerings, we cover the following concepts in our ontology model: 1) a simple offering that is defined through properties [e.g., capacity, availability, and input/output (I/O) speed of a storage offering] and basic operations; 2) a basic operation that is a catalog action or internal atomic process; 3) a composite offering that makes use of operations provided by simple or composite offerings; and 4) a composite operation that is defined on top of simple or composite operations and corresponds to a user action when related to catalog offerings. More details can be found in [10].

Figure 3 shows an example of a catalog ontology, adapted from the one in [10]. The links in the figure without any labels represent the *rdfs:subClassOf* property. Specifically, *SmallStorage* and *LargeStorage* are simple offerings, and *StorageUpgrade* is a basic operation between these two offerings. In addition, *BronzeDesktop* is a composite offering based on *SmallStorage*, whereas *SilverDesktop* is a composite offering based on *LargeStorage*. The upgrade operation *DesktopUpgrade* between *BronzeDesktop* and *SilverDesktop* can be defined as a composite operation making use of *StorageUpgrade*.

Prior research has addressed a rich range of aspects when dealing with ontology and processes for services. However, the safeness and reversibility in sequencing operations into complex processes have remained open questions. We introduce the notion of safe sequences to our ontological model and show how to select and sequence a set of delivery operation processes to automatically fulfill catalog requests for complex offerings in two steps. The first step harnesses the ontological representation, particularly the composite and dependence relationships between offerings and processes. The second step enforces system robustness through

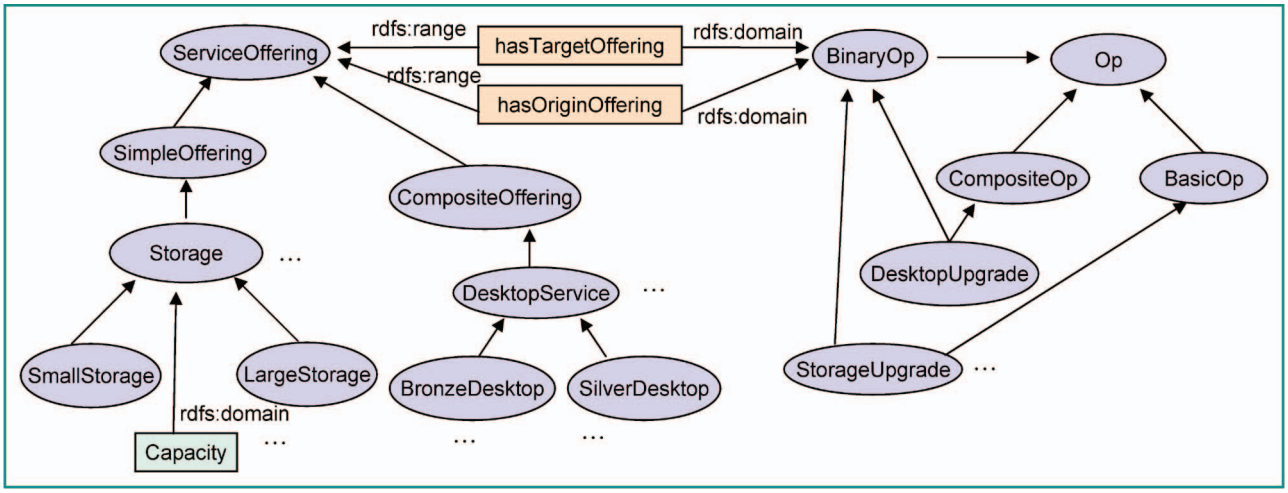

Figure 3

Example of an ontology-based service catalog. (Op: operation; rdfs: Resource Description Framework schema.)

postcomposition reordering, thus limiting the situations that require costly labor-intensive repairs.

Fast virtual disk

In a cloud, the storage space for the virtual disks of VMs can be allocated from multiple sources, i.e., the host's direct-attached storage (DAS, i.e., local disk), network-attached storage (NAS), or a storage area network (SAN). These options offer different performance, reliability, and availability at different prices. DAS is at least several times cheaper than NAS and SAN, but DAS limits the availability and mobility of VMs. To achieve the best use of the different technologies, a cloud usually offers a combination of block-device storage services to VMs, including both ephemeral storage (on DAS) and persistent storage (on NAS).

DAS is simple and inexpensive. The aggregate storage space and I/O bandwidth of DAS linearly scale as hosts are added. However, using DAS slows down the process of VM creation and migration and diminishes the benefits of an elastic cloud. The discussion below uses Kernel VM (KVM) [11] and QEMU [12] as examples, as the IBM Cloud is based on KVM.

In a cloud, VMs are created based on read-only image templates, which are stored on NAS and accessible to all hosts. The virtual disk of a VM can use different image formats. The RAW format is simply a byte-by-byte copy of the full content of a physical block device stored in a regular file. If a VM uses the RAW format, the VM creation process may take a long time and cause resource contentions because the host needs to copy a complete image template (i.e., gigabytes of data) across a heavily shared network in order to create a new RAW image on DAS.

QCOW2 [13] is another image format supported by QEMU. It does copy-on-write (CoW), i.e., the QCOW2 image only stores data modified by a VM, whereas unmodified data are always read from the backing image. QCOW2 supports fast VM creation. The host can instantly create and boot an empty QCOW2 image on DAS with the backing image pointing to an image template stored on NAS. Using QCOW2, however, limits the scalability of a cloud because a large number of VMs may repeatedly read unmodified data from the backing image, generating excessive network traffic and I/O load on the shared NAS server.

Our solution to this problem is the FVD image format and the corresponding driver. In addition to CoW, FVD also performs copy-on-read (CoR) and adaptive prefetching. CoR avoids repeatedly reading a data block from NAS by saving a copy of the returned data on DAS for later reuse. Adaptive prefetching uses idle time to copy, from NAS to DAS, the rest of the image that has not been accessed by the VM.

Figure 4 compares the VM creation processes with and without FVD. This example creates three VMs concurrently. In Figure 4(a), using the RAW image format, the process has to wait for a long time until an entire image template is copied from NAS to DAS, and then boots the VM. Note that most of the copied image data are not needed during the VM booting process and may even never be accessed throughout the lifetime of the VM. In Figure 4(b), FVD boots the VM instantly without any image data on DAS and copies data from NAS to DAS on demand as they are accessed by the VM. In addition, the prefetching mechanism of FVD finds resource idle time to copy from NAS to DAS the rest of the image that have not been accessed by the VM. Prefetching is conservative in that if FVD detects a

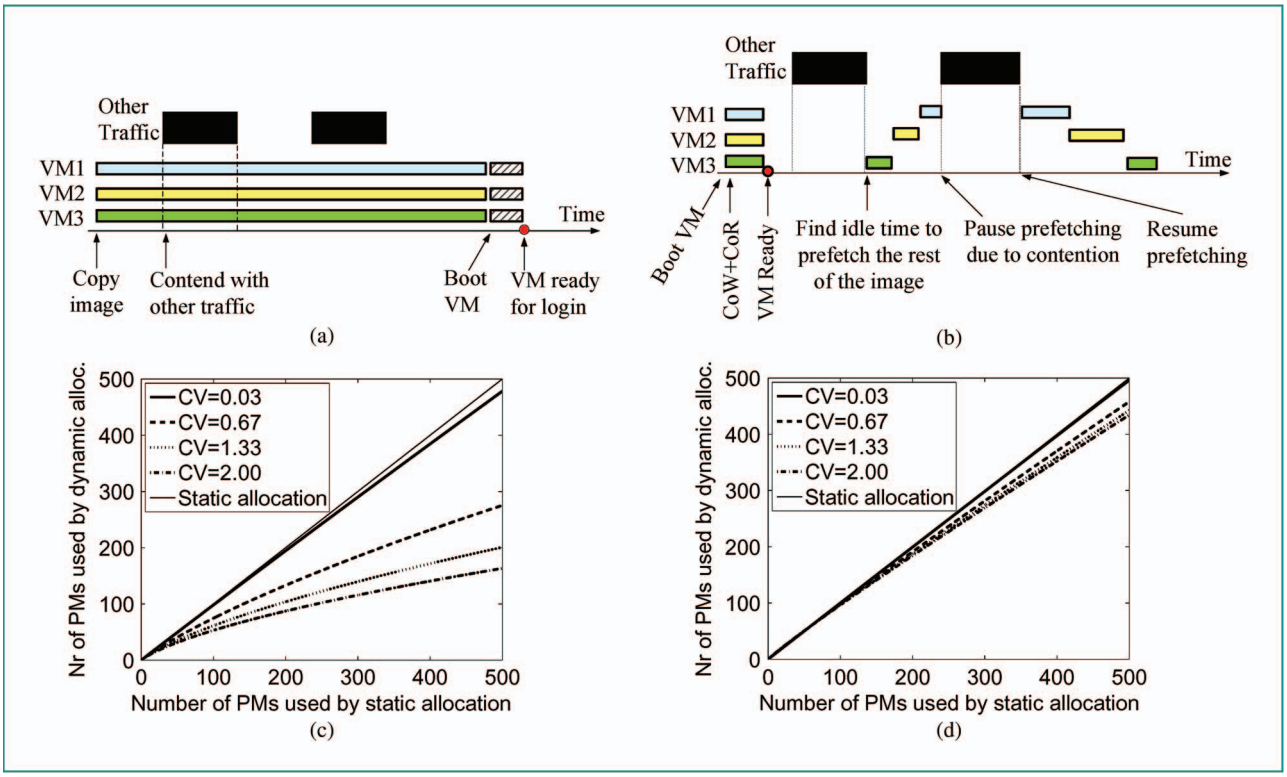

Figure 4

FVD [(a) and (b)] and dynamic resource allocation [(c) and (d)]. (a) VM creation using RAW images. (b) VM creation using FVD images. (c) Impact of low-frequency variation. (d) Impact of high-frequency variation. (Parts (c) and (d) are adapted from [14].) (CoW: copy-on-write; CoR: copy-on-read; PM: physical machine, i.e., server; CV: coefficient of variation.)

contention on any resource (including DAS, NAS, or a network), FVD pauses prefetching temporarily and resumes prefetching later when congestion disappears.

Dynamic workload reallocation

Cloud computing promises a significant efficiency improvement due to making use of elasticity and economies of scale. However, one of the key requirements to realize this promise is creating an agile resource management system that adjusts the resources allocated to cloud workloads according to their changing needs and global optimization objectives. An important aspect of the resource adjustment process in a virtualized system is its cost and latency. For example, if the cloud resource manager is to reallocate a VM between two hosts, it should consider both the resources consumed by the migration process (such as network bandwidth and CPU on source and target hosts) and also the time spent in performing the migration.

Therefore, the cloud resource manager algorithm needs to take into account the tradeoff between the benefits of the reallocation action and the costs and time required to perform such an action. In particular, we may ask up to what extent the utilization of a consolidated data center can be increased by employing dynamic reallocation of VMs. As discussed above, since the reallocation has inherent costs and degrades the overall performance of the system, it cannot be performed too often. Thus, the amount of gain that can be realized from reallocating the VMs is not easily quantifiable. As a simple example, consider the case of resource demand for which most of the variability is contained in frequencies higher than the VM migration frequency. In this case, moving does not improve overall efficiency because the system will not be able to keep up with the changing load. However, if the majority of signal power is located at lower frequencies, then dynamic migration can be a viable option, potentially reducing the number of required physical servers.

We formulated an analytical model [14] of the virtualized data center and used it to derive relationships between important parameters that influence the amount of gain expected from the dynamic reallocation of VMs. The parameters include resource demands of VMs, time between reallocations, and desired level of capacity overflows. Resource demands of VMs are represented using stochastic

processes (with autoregressive properties of a process used to quantify the variability on the migration timescales). Although the actual gain from the dynamic reallocation approach depends on the details of resource demand distributions, we manage to provide a closed-form solution that solely depends on the first two moments of those distributions and the values of their autocorrelation functions. This is possible by the use of a central limit theorem, which provides very good approximation for larger data center sizes. We verified the model using extensive data center simulations and used it to discover relationships between the aforementioned parameters and the amount of gain in average data center utilization. The gain is defined as the reduction in the number of physical machines (PMs) required to host the workload, as compared to the number required by static server consolidation while assuming the same requirement on the maximal rate of capacity overflows. As an example, consider Figures 4(c) and 4(d), which show gain in terms of time-averaged number of used PMs to support the same workload with an identical capacity overflow probability of 0.05 for various levels of workload variability with high-frequency variation (c) and low-frequency variation (d). Quantification of high versus low frequency is with respect to the maximum frequency of resource reallocation.

Conclusion

We have presented IBM CCMP, a key component of IBM Cloud Architecture, which provides both business and operational support services for cloud IT service offerings. We have also discussed a set of novel extensions to the platform designed to increase its flexibility, reusability, and efficiency and transform it into a foundation for enterprise cloud services ecosystem. The proposed extensions span business support compositionality, advanced offering management and contract optimization, service provisioning, and resource allocation optimizations.

*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 Linus Torvalds, Novell, Inc., Microsoft Corporation, or Sun Microsystems in the United States, other countries, or both.

References

1. A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, and M. Armbrust, *Above the Clouds: A Berkeley View of Cloud Computing*. Berkeley, CA: Univ. California, 2009.
2. Amazon Inc., *Amazon Elastic Compute Cloud*. [Online]. Available: http://aws.amazon.com/ec2/
3. Rackspace Hosting, *Cloud Computing, Cloud Hosting and Online Storage*, May 2011. [Online]. Available: http://www.rackspace.com/cloud/
4. IBM Corporation, *IBM Cloud Computing*. [Online]. Available: http://www.ibm.com/ibm/cloud/
5. IBM Corporation, *IBM Lotus Live*. [Online]. Available: https://www.lotuslive.com/en/
6. IBM Corporation, *IBM Cloud Computing Reference Architecture 2.0*, 2011. [Online]. Available: https://www.opengroup.org/cloudcomputing/uploads/40/23840/CCRA.IBMSubmission.02282011.doc
7. A. Sailer, M. Head, A. Kochut, and H. Shaikh, "Graph-based cloud service placement," in *Proc. Int. Conf. Services Comput.*, Miami, FL, 2010, pp. 89–96.
8. WSML Working Group, *WSML Language Reference*, 2008. [Online]. Available: http://www.wsmo.org/TR/d16/d16.1/v1.0/
9. L. Youseff, M. Butrico, and D. Silva, "Towards a unified ontology of Cloud computing," in *Proc. Grid Comput. Environ. Workshop*, Budapest, Hungary, 2008, pp. 1–10.
10. Y. Deng, M. R. Head, A. Kochut, J. Munson, A. Sailer, and H. Shaikh, "An ontology based approach for cloud services catalog management," in *Proc. 8th Int. Conf. Service Oriented Comput.*, San Francisco, CA, 2010, pp. 680–681.
11. A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguor, "KVM: The Linux virtual machine monitor," in *Proc. Linux Symp.*, Ottawa, ON, Canada, 2007, pp. 225–230.
12. F. Bellard, "QEMU, a fast and portable dynamic translator," in *Proc. USENIX FREENIX Track*, Anaheim, CA, 2005, pp. 41–46.
13. M. McLoughlin, *The QCOW2 Image Format*. [Online]. Available: http://people.gnome.org/~markmc/qcow-image-format.html
14. A. Kochut, "On impact of dynamic virtual machine reallocation on data center efficiency," in *Proc. IEEE Conf. Meas. Simul. Comput. Telecommun. Syst.*, Baltimore, MD, 2008, pp. 1–8.

Received February 8, 2011; accepted for publication March 10, 2011

Andrzej Kochut *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (akochut@us.ibm.com)*. Dr. Kochut received the Ph.D. degree in computer science from the University of Maryland, College Park, in 2005. He is a Research Staff Member and Manager at the IBM T. J. Watson Research Center. His research interests include computer systems, networking, stochastic modeling, and performance evaluation with a recent focus on virtualization and cloud computing. Dr. Kochut is an active author and member of conference program committees, was recognized with a Best Paper Award, and received eight IBM awards.

Yu Deng *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (dengy@us.ibm.com).* Dr. Deng is a Research Staff Member at IBM T. J. Watson Research Center. Her research interests include data and knowledge management, ontologies, semantic analysis, and machine learning. She received her Ph.D. degree in computer science from the University of Maryland, College Park, in 2006, after which she joined IBM Research. Her research work at IBM has contributed to cloud systems and vertical enterprise search engines.

Michael R. Head *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (mrhead@us.ibm.com).* Dr. Head is a Senior Software Engineer at IBM T. J. Watson Research Center in Services Research. He received the Ph.D. degree from Binghamton University, Vestal, NY, in 2009 and his M.A. degree from Brandeis University, Waltham, MA, in 2004, both of which are in computer science. His software runs production services including the Smart Business Development and Test on the IBM Cloud. His research interests include service automation, algorithms for concurrent programming, performance analysis, and software engineering.

Jonathan Munson *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (jpmunson@us.ibm.com)*. Dr. Munson received the Ph.D. degree in computer science from the University of North Carolina, North Carolina, in 1997. He joined the IBM T. J. Watson Research Center in 1998 and has

worked on virtual-reality-based collaboration systems, systems for distributing and protecting digital content, and location-based services for wireless networks. He has recently worked in the areas of automotive telematics, spatially oriented collaboration systems, and business-support systems for cloud computing.

Anca Sailer *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (ancas@us.ibm.com).* In 2000, Dr. Sailer received her Ph.D. degree in computer science from Pierre et Marie Curie University of Paris, Paris, France. She was a Research Member at the Networking Research Laboratory at Bell Labs from 2001 to 2003, where she specialized in Internet services traffic engineering and monitoring. In 2003, Dr. Sailer joined IBM where she is currently a Research Staff Member in the Service Products department. Her interests include cloud computing management, self-healing technologies, and information analysis.

Hidayatullah Shaikh *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (hshaikh@us.ibm.com).* Mr. Shaikh is a Senior Technical Staff Member and Senior Manager at the IBM T. J. Watson Research Center. He is currently the IBM lead for Business Support Services for Cloud offerings. His areas of interest and expertise include virtualization, cloud computing, remote services delivery, business process modeling and integration, service-oriented architecture, grid computing, e-commerce, enterprise Java**, database management systems, and high-availability clusters.

Chunqiang Tang *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (ctang@us.ibm.com).* Dr. Tang received the Ph.D. degree in computer science from the University of Rochester in 2004. He is a Research Staff Member and Manager at the IBM T. J. Watson Research Center. His research interests lie primarily in the systems area (including services computing, distributed systems, operating systems, computer networks, and storage systems), and secondarily in novel applications of information retrieval (including consumer-centric health informatics and peer-to-peer information retrieval). He holds 22 patents, published 42 papers, won two Best Paper Awards, and received nine IBM awards.

Alex Amies *IBM Global Technology Services Development Lab, Zhongguancun Software Park #19, Haidian District, Beijing 100193, China (aamies@cn.ibm.com).* Mr. Amies is a Senior Software Engineer acting as an architect in the Development and Test Cloud Business Support Services design team. His area of focus at present is cloud application programming interfaces and customer and catalog management. His other areas of interest include security, identity management, user interface technologies, web services, and monitoring.

Murray Beaton *IBM Global Technology Services, IBM Toronto Lab, Markham, ON L6G 1C7, Canada (beatonm@ca.ibm.com).* Mr. Beaton is a Senior Software Developer on the IBM Smart Business Development and Test Cloud. He is driving the design and delivery of tooling responsible for customization of virtual machine instances.

David Geiss *IBM Global Technology Services Architecture, Research Triangle Park, Durham, NC 27709 USA (drgeiss@us.ibm.com).* Mr. Geiss is a certified Master IT Architect focused on the integration of the Common Cloud Management Platform with IBM fulfillment and financial systems. His area of interest and expertise involves application architecture, process and data modeling, and systems integration.

David Herman *IBM Corporate Headquaters, BT/IT CIO, Southbury, CT 06488 USA (dherman@us.ibm.com).* Mr. Herman is an Executive Architect in the IBM CIO Office supporting the end-to-end (E2E) application architecture for the IBM Services business. His area of focus includes the support for new Global Technology Services Cloud offerings and the integration of the Cloud BSS/OSS with legacy and future IBM Business Systems.

Holger Macho *IBM Global Technology Services (GTS), IBM Germany Lab, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (macho@de.ibm.com).* Mr. Macho is Director of IBM GTS Cloud Development, located in the IBM Germany Lab, leading a worldwide development team. He is currently leading multiple large cloud development projects. His areas of technical interest and expertise include virtualization life cycle management, cloud computing, Information Technology Infrastructure Library (ITIL), service management, system high-availability and disaster recovery, and SAP software on cloud systems. He is also interested in managing diverse multicultural teams and creating a high-performance culture.

Stefan Pappe *IBM Global Technology Services, IBM Deutschland GmbH, St.-Vitus-Gasse 15, Heidelberg 6912, Germany (pappe@de.ibm.com).* Dr. Pappe is an IBM Fellow and an expert in cloud computing and asset-based service development. He currently leads the Specialty Service Area for Cloud Services in the IBM Global Technology Services (GTS) organization, building the architecture for the cloud offerings of GTS and solutioning client projects. He coauthored the "Cloud Computing Reference Architecture," which is a comprehensive technical blueprint guiding IBM employees and clients in solution design and delivery. Dr. Pappe spent most of his more-than 20 years of his IBM career growing the business of Global Services through technical advancements. His assignments ranged from business to infrastructure services, including both project- and managed services. The realization of the asset-based service model within IBM is a consistent thread throughout his career. This concerns standardized production and consumption of services to increase quality and efficiency of services for IBM clients, while allowing for innovation through IBM community developments. Dr. Pappe holds a master's degree in economics from University of Karlsruhe, Karlsruhe, Germany, and a Ph.D. degree in computer science from University of Kaiserslautern, Kaiserslautern, Germany.

Scott Peddle *IBM Global Technology Services, IBM Toronto Lab, Markham, ON L6G 1C7, Canada (peddle@ca.ibm.com).* Mr. Peddle is an Advisory Software Developer on the IBM Smart Business Development and Test cloud. He leads projects focusing on integration of Rational Asset Manager within a cloud environment. His areas of focus include image collaboration and governance.

Randy Rendahl *IBM Global Technology Services, Research Triangle Park, Durham, NC 27709 USA (rr@us.ibm.com).* Mr. Rendahl is a Senior Software Engineer with an architectural role in the Common Cloud Management Platform design team. His current focus is in the areas of billing and rating, business process enablement and BSS data model design. Beyond his current focus, he has a broad background in the cloud BSS technology in addition to a long and varied past in the system management arena.

Angel E. Tomala Reyes *IBM Global Technology Services, New York, NY 10010 USA (aetomala@us.ibm.com).* Mr. Tomala Reyes is a Senior Software Engineer with a user interface architectural role in the Common Cloud Management Platform (CCMP) design team. His current focus is the technical design and implementation of the CCMP web user interface and internal APIs. His interests include software component architecture, geospatial applications development, and several Web 2.0 technologies and frameworks.

Harm Sluiman *IBM Global Technology Services, Toronto Lab, Markham, ON L6G 1C7, Canada (sluiman@ca.ibm.com).* Mr. Sluiman joined IBM in 1974, and has held positions in virtually all aspects of the company during his career. For the last 29 years he has been working in research and development and is currently leading the development of the public cloud computing efforts. He holds more than

25 patents related to software. He was appointed IBM Distinguished Engineer in 2007 in recognition for his leadership and technical contribution to IBM. Mr. Sluiman's work spans the IBM software group brands as well as several open source and standards initiatives in which he contributes code and helps manage. Prior to his current role, he led the implementation of the IBM Development and Test cloud public Beta. Earlier he was a member of the Rational CTO team and responsible for the integration of full IT life cycle processes with Rational offerings, and operational modeling. This included setting the Rational strategy to address the long term challenges in the industry due to the advent and impact of higher density systems, virtualization, and cloud computing.

Brian Snitzer *IBM Global Technology Services Delivery, West Chester, PA 19380 USA (snitzer@us.ibm.com).* Mr. Snitzer is a Distinguished Engineer in Global Technology Services Delivery. He is a member of the IBM Academy of Technology and is a Master Inventor, having filed 16 patents and published many more. In 2004, he received his M.B.A. degree from the Duke University Fuqua School of Business. He is currently the Service Line Leader for delivery of cloud services, and has served as an architect, development lead, and delivery lead for the creation of clouds. He concentrates on product development and on establishing new service capabilities. His work has included a focus on high-volume and large-scale infrastructure design and management, networking, security, operational models and processes, organizational design, and service automation.

Troy Volin *IBM Global Technology Services, Durham, NC 27709 USA (tvolin@us.ibm.com).* Mr. Volin is a Senior Software Engineer in GTS Offerings Technology Development. He currently focuses on the architecture and implementation of the Business Support Services of the Common Cloud Management Platform. He concentrates on identifying and developing cloud computing functionality unique to the enterprise customer. His areas of expertise are in systems management, integration, and automation. He came to the CCMP Development area from the Tivoli CTO Office, where he worked on advanced technology and strategy, developing proof-of-technology projects and integrating emerging technologies into products.

Hendrik Wagner *IBM Development GmbH Germany, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (hzwagner@de.ibm.com).* Mr. Wagner is currently at the IBM Development Laboratory in Boeblingen, Germany. After his graduation in computer science from the Berufsakademie in Stuttgart, Germany, in 1991, Mr. Wagner worked on various IT projects for the automobile industry and telecommunication customers as well as on cloud management software development.