OPTIMIZING THE IT BUSINESS SUPPLY CHAIN UTILIZING CLOUD COMPUTING

Donald F. Ferguson  
CA Technologies, NY, USA  
donald.ferguson@ca.com

Ethan Hadar  
CA Technologies, Herzlia, Israel  
ethan.hadar@ca.com

Keywords: Business-IT alignment, Enterprise Engineering, Cloud Computing.

Abstract: Information technology applications and systems are essential to businesses and enterprises. IT applications govern and optimize operations and retail, manufacturing and other industries. IT applications implement the business model in others industries, for example, financial services. Optimizing Return-on-Investment (ROI) in the IT area is essential to the business performance. Reducing cost is one component of ROI, however the predominant value is increasing revenue. IT is essential to the enterprise agilely to exploit new business opportunities. Cloud computing is emerging as a technology for optimizing IT costs and supporting agility. Enterprises are incrementally moving to cloud computing in an exploratory, ad hoc manner. Software solutions implementing best practices for transforming to cloud computing are essential. Enterprises think in terms of IT Services that IT provides to the business. An IT service is an interconnected set of hardware and software resources, and is conceptually similar to a manufacturer or retail supply chain. Exploiting cloud computing is a supply chain management problem for IT services using cloud computing. This paper describes the architecture requirements and implementation of a set of components for optimizing the IT supply chain using cloud computing.

1 INTRODUCTION

Modern, non-trivial business applications are composite applications [8], integrating various technologies. A composite application exploits a myriad of technologies (Figure 2), for example: PHP web applications, relational database servers, business process execution language (BPEL), automation engines, collaboration portals, mainframe transaction problems, web service calls to external systems for credit card processing, and Microsoft-office plug-ins for personal productivity. Business transactions and end-to-end requests flow through the application components and support hardware and software resources. For example: submit shopping cart composite application and its underlying a composite IT system.

IT service as a term defines the composite application and supporting hardware and software resources. Virtually all IT services and supporting hardware and software resources are “in the logical datacenter.” There can be multiple datacenters and reliance on hosting providers, but the conceptual model is a single, company owned datacenter. This approach to IT is similar to vertically integrated manufacturing with all resources (factories, warehouses, manufacturing devices, etc.) inside the enterprise.

Intrinsic characteristics of cloud computing are dynamic changes, growing offerings, competition, costs reduction and new pricing models, flexible quality-of-service and more. The emerging economy and marketplace of cloud computing services holds the promise of optimizing IT services in a way similar to how supply chain management optimized manufacturing, retail and other industries. The optimizing is cost effective and agile. (Figure 3).
Cloud computing for business applications applies the supply chain management model in a way that transforms IT services from vertically integrated to an IT supply cloud chain model. The enterprise maps the IT service’s resource requirements to various cloud service providers, that can include [3]:

- **Software-as-a-Service (SaaS):** Complete, cloud delivered applications.
- **Business-Service-as-a-Service (B-SaaS):** Web callable (programmable web) application interfaces (APIs) for performing specific business operations. For example FedEx APIs for shipping or Dunn and Bradstreet APIs for credit ratings. This approach expands the internal SOA application model to the Web/cloud.
- **Platform-as-a-Service (PaaS):** Online application enablement tools and runtimes for developing and delivering applications. For example, SalesForce.com, RedHat’s JBoss delivered via PaaS.
- **Infrastructure-as-a-Service (IaaS):** Low-level, usually hardware resources, for running the applications and software stack that the consumer supplies. Amazon EC2 is an example.
- The current focus of cloud computing is on IaaS and SaaS [4]. The focus will evolve over time to increase exploitation of PaaS and B-SaaS. There are many positive motivations for the increased exploitation; for example, almost all SaaS use requires platform functions to integrate with other business systems, hiding the PaaS elements. Further, enterprises that offer online applications are incentivise to surface their business capabilities with web callable APIs for inclusion in their customers’ applications. For example, an enterprise can directly integrate FedEx capabilities into their applications, replacing manual interaction with the embedded web user interface. Other requirements originate from Platform vendors (IBM WebSphere, Oracle Fusion) where customers are requesting the benefits of SaaS for their infrastructure. Cost requirements pose another difficulty. IaaS providers can drive higher margins by offering preconfigured packages of hardware and software for specific use cases, for example, collaboration, complex data analysis, and private out-of-the-box cloud.

The move to PaaS and B-SaaS benefits enterprises that evolve to cloud computing. Platforms optimized for application requirements reduce complexity and increase the efficiency of deploying an application relative to IaaS. Enterprises can reuse the platform published APIs instead of writing and developing their modules.

Enterprises will incrementally implement and evolve IT services by selecting cloud service providers and creating the IT supply chain, followed by deploying the application and managing the composite application and systems.

We use the term **IT Cloud Service (ITCS)** for an IT service following this hybrid, automated, managed deployment model. Exploiting the new model requires a completely new type of software system – **cloud service optimizer (CSO)**, which is analogous to enterprise resource planning (ERP) systems for manufacturing. This paper explains:
The requirements for the CSO software systems.

The service lifecycle and architecture capabilities model for CSO, including short description on actual implementation.

Figure 2 – transitioning of composite applications elements with cloud based services

2 CLOUD SERVICE OPTIMIZER REQUIREMENTS

Enterprises use a conceptual lifecycle model to identify, prioritize, develop modify, deploy, manage and retire IT services. For example, ITIL V3 [5] is a set of processes that define codifying best practices for IT service lifecycle management. Here, we focus on the inner lifecycle process to develop, deploy and manage the applications. This paper uses the term Composite Application Business Service Lifecycle (CABS-LC, Figure 3).

Several enterprises systematically apply the CABS-LC model, while others opportunistically manage IT service lifecycles. In either case, understanding the model helps to understand the CSO requirements, architecture and implementation.

Model

There are three facets to CABS-LC: the IT service modelling, the business process modelling (BPM), and the composite IT system modelling.

IT service modelling implements several elements. These include identifying services to develop or modify and prioritizing these projects and IT relevant investment. Additional steps are defining financial goals and service level contracts and agreements, allocating and managing investment and assets, and fading out services, or ones that reach their end-of-life. Examples for IT service modelling decisions are deciding which projects receive budget allocation for purchasing of hardware infrastructure, or choosing between composite applications alternatives of in-house development or subcontracting to outsourced development. An example for a business process is managing approval process of key stakeholders (CFO, line-of-business executive, Chief Security Officer). An example for a business process cost and billing is defining chargeback to the Line-Of-Business (LOB) for resource usage for the service. The ahead proposed CSO architecture replaces ad-hoc decision making with a more systematic approach to optimizing the IT service lifecycle.

Unlike CSO, there are many commercial BPM. BPM is broader than “process modelling”, and typically includes modelling document or business objects, rules and policies, organization and roles or capabilities, superimposed with key performance indicators (KPI). The primary interaction between BPM and CSO is KPIs. A CSO complements optimization during IT service modelling with dynamic optimization of resource allocations for deployed IT services executing business transactions. KPIs define the optimization goals and constraints. Aligned, the consumer can evaluate the service offering, and how it is performing, thus can be compared and overtime enhanced, replaced, or removed.

Finally, defining and configuring a composite IT system that supports an application is usually a manual, tedious and error-prone process. A core element of the CSO is to support productive, intuitive and reliable modelling for defining and
automatically modifying composite IT systems. The CSO’s support for modelling composite IT systems and subsequent automation, improves productivity, reusability, and reliability by applying repeatable best-practices and sharing between cloud service consumers and providers.

Many enterprises do not fully understand their existing IT systems and interactions (figure 2). The systems emerged over time through independent decisions, technology shifts, mergers and acquisitions and other factors. For example, identifying applications and transactions effected by a server or router failure is often impossible. IT service modelling and optimization is not feasible without insight into existing IT systems, applications and resource allocations. Comparing cloud services to on-premise resources is impossible without this insight. Providing this insight is a core function and value proposition of a CSO because to design the evolution of the IT services architecture of existing systems, ones must understand the current status and make a knowledgeable decision concerning the value of the change, rational, cost, quality, ease of change, and other insights.

Develop/Assemble/Configure

As stated before, modelling defines the logical, high-level structure and behaviour of the IT service and composite application. Producing a working application and system requires more complete specification, usually involving programming and IT administration. An example is developing a business process executable language (BPEL) that begins from the business process executable notation (BPMN) process definition.

The assemble phase composes of three sub-elements; Develop, Assemble, and Configure. Using Service Oriented Architecture (SOA) terms, Develop implements new services or modifies existing services. Assemble uses higher layer abstractions and configuration of integrations, wiring and consumption modes like BPEL, portlets, rules, orchestration etc. to assemble the composite application. The application assembly also sets policy and quality-of-service requirements for the deployed application. J2EE deployment descriptors are an example. The Service Component Architecture [2] and Windows Communication Foundation [1] have similar concepts.

A motivation for SOA is agility through reuse. There is a similar requirement to define and configure composite IT systems. Currently, there is little reuse of working configurations, virtual machine templates and other IT resources, systems configuration or definition information. There are almost no intuitive tools for such purposes. The state of the art is scripts, editing configuration files and primitive automation tools.

Agility is impossible without a significant improvement in defining the composite IT system that supports a composite application. The problem is especially important when mapping composite applications to a mix of on-premise resources and off-premise remote cloud services.

These observations generate several requirements for a CSO. Some of these requirements are defining canonical building block templates (virtual machines, application servers, etc.) and providing tools to assemble composites from the templates that turn into basic building blocks. A cardinal additional requirement is constraining connections and bindings between templates to help ensure that the result complies with policies, rules and best practices for producing a robust configuration.

The Configuration step instantiates templates to produce a deployable system. The core function of this step is setting customization parameters and attaching policies, including security policies. An important function of configuration is constraining the properties and policies of templates to a predefined, fixed set. The constraints preclude possibly hundreds of configuration options and reduce the optimization problem space. Constraining the variability is essential to ensure scalability, availability, security, efficiency and successful deployment to exploit cloud services. The vast majority of deterioration of such quality of service attributes derives from faulty configuration.

One perspective on the composite IT system assembly is for the CSO to implement a domain specific language [9] to define and assemble IT systems. Such capabilities require an intuitive visual tool with precise notation for creating and editing IT systems. Furthermore, support for defining patterns, templates and best practices for building cloud enabled systems is also required. Consequently, these requirements will provide precise, unambiguously defined configurations for deployment and change management systems.

Provision (deploy)/Change

Provision/Change receives directives from the IT system assembly tool and the Analyse and Optimization subsystem. The directives trigger automation processes (Figure 1), with possible human interaction steps, to implement the directive.
The basis of the automation is the concept of an abstract container. An abstract container supports a WS-Management standard interface and possible interfaces for other abstract container types for managing purposes. For example, a container that manages a pool of machines allocated to running virtual machines and a physical server farm may support a common, abstract interface. An operation on this container might be `deploy()`, with the virtualization manager accepting virtual machines images and the server farm manager accepting boot images.

An important requirement is to support extensible notation (activity types, properties, etc.) The notion is to support advancements in technology by extending the content, rather that to replace the technology.

The automation processes relies on the abstract containers to implement low-level functions for the resources they manage. This is especially important for cloud services. An IaaS service might support provisioning a virtual machine through a well-defined interface. The IaaS would typically not surface interfaces allowing direct management of physical servers, network devices, storage, etc. Another example is provisioning security and identity information. The workflow would provision the identity information in Active Directory or to a Lightweight Directory Access Protocol (LDAP) system, but not propagate the information to the individual repositories throughout the relevant sub elements of the composite applications.

Monitor

The Monitor phase’s role is self-explanatory. Core requirements and steps are: 1) customizable event filtering, 2) normalizing disparate events formats into a common event type and taxonomy, 3) event correlation, 4) a high-performance event-condition-action rule, namely a complex event processing capabilities [7], that invokes a management process and operation execution based on a defined procedure.

Cloud services introduce requirements that expand on commonly available monitoring systems [6]. Specifically, optimizations for monitoring services remotely over the cloud and monitoring cloud services that do not support embedded instrumentation. Such constraints may include synthetic transactions or robots that sample the actual use of the service and infer information. Other unique requirements are traversing firewalls and network traffic on the boundaries between on-premise and enterprise internal domains, and to maintain security requirements due to the highly federated cloud environment compared to traditional “all in the datacenter” models.

A final challenge is correlating and integrating infrastructure events with the higher layer business activity management and monitoring system. Consider a simple example of “submit shopping cart” failures. The failure might occur because of IT reasons or because the discount rules and validation rules cannot process the shopping cart. The IT infrastructure alone cannot determine that the “submit shopping cart” actually failed. A well-designed application catches exceptions and returns meaningful information to the user, which is a correct behaviour for lower level IT management system. Yet, at the IT systems levels, the monitoring applications must correlate the information into higher-layer business systems to put events in the business process and transaction context.

Analyse/Optimize

Careful analysis of appropriate sensors provided metrics aids in reasoning on the root-cause of a problem, optionally leading to a resolution. An example can be that the system correlates “unresolved requests”, with “server transactions load”, and analyzes it to “My web application server is aborting requests under load”. The triggered resolution of this analysis will be to provision additional capacity for use when the server transaction load increases. The goal of traditional analysis tools in IT is to understand the underlying reason and react to it. Modern and future analysis tools provide forecasting of IT needs, prior to the requests, to proactively prevent the problem. Such analysis can be based on past experience with the same service or solution, but more importantly, correlating trends that are based on multiply sensing IT domains, within the organization and outside. Global effect of power failure in one geographical territory can trigger the immediate backup of
systems and move of services to other locations, because it is likely that systems in the affected area will request response in the active ones, creating additional load on the systems that run in the active area. Intelligent systems cross-analyze multiple domains, using collaborative social networks and systems. Proactive sensing is based on observing unplanned anomalous behaviour on the network and system, and correlating activities that occur on similar systems and yet not manifested on the monitored systems. Proactive analysis is the understanding of the phoneme. Intelligent optimization decides on the probable action and adjusting accordingly. Operation research study paves the way from feasible solution within a limited pool of resources, to optimized solution in presumably, endless resource pool of cloud services. Artificial business intelligence or optimization of resources algorithms are currently the main stream in providing such a solution. Proactive collaborative systems that require federated and deep search capabilities are the foundations to future ones that can enable a larger set of decision domains. Profiling systems behaviour, use patterns, transactions contextual patterns are all part of the required analysis, where making a human or machine based decision are the optimization part.

The Analysis and Optimization lifecycle element is a proactive, forecasting phase aimed at generating a comprehension of the systems current and needed state. The optimization part defines not just the end goal, but rather the set of changes that must occur, to gradually progress to an improved systems state. In complex systems, a change in one area might generate a propagated affect in another, it is fundamental to understand the existing structure and its quality levels as captured by the automated discovery service of the modelling phase. Optimization is based not on changing local self-adaptive capabilities, rather on balancing between feasible modelled alternatives, in a holistic manner. Furthermore, the optimization roadmap may be iterative in nature due to the problem size. Consequently, automation of the change, rapid discovery of the new state, monitoring the trend of the new state, and overall balancing the conditions against alternatives is required.

Shared Information

The different stages of the service lifecycle involve a cross-application, complex information. However, contrary to shared data models that all applications produce and consume information from, an enterprise environment and services are constantly evolving. At a certain time, adding more services requires backward compatibility and regression testing that add management overhead. Furthermore, the different applications and services provide capabilities and offerings of their own, and support self-behaviours as a standalone system. This implies that the composite applications cannot rely on a single point of failure. When dealing with such requirements, one must add the needs for unified logical information that is unambiguous, precise, searchable, reusable, contextual, extensible, expandable, and exportable. This information must be simple enough to learn, self-descriptive, with unified semantic. Yet, the communication mechanism of synchronizing the information elements must be scalable, on-demand, and delivered only to the requestors of that information.

3 CONCEPTUAL ARCHITECTURE

The proposed implementation blocks are technologies that interact and share information, catering for most of the above capabilities. These modules encapsulate value proposition that is reusable in other scenarios than the composite application for cloud IT services, thus, aggregated as such.

3.1 Unified Service Model

The technology caters for the shared information of the Unified Service Model, aimed at assisting in the data transformation between the technologies. It is based on a hub-and-spoke architecture pattern, employing a logical canonical data model pattern, and using publish-subscribe pattern messaging. For the IT services to collaborate, exchange information and maintain system integrity, the passing data must support semantically correct transactions content, and entities identity such as computing resources and configuration items (CI). Yet, creating a single repository of all domains is extremely inefficient and even unpractical in large enterprises and disconnects vendors on the supply chain. Virtually centralized, but federated to distributed applications across the chain, the transported data is transforming in terms of contextual meaning, logically integrating the different applications within the composite application. Furthermore, the exposed IT services employ the Unified Service Model taxonomy, enabling future ease of connectivity.
3.2 Discovery

The set of discovery technologies provide the capabilities of the IT topological discovery modelling captured as Configuration Management DB. As part of the monitoring tools that assure service quality, technologies such as infrastructure management monitoring tools enable to identify a configuration item that use the Data collecting agents or even access and probe the information from a remote cloud location into the on-premise assets. In many cases, the information is missing, and the status of the systems needs verifying, catering for the Proactive Response capability, fixing systems and by informing on missing resources or system starvation. The discovered systems can be active, such as captured on CMDB, or be potential, in a Service Catalogue [4] of alternative services, enabling selective optimized assembly capability to be fulfilled. IT topological discovery modelling of components (commonly referred to as Configuration Items or CIs) such as applications, servers, and managed devices, is aimed at capturing the topological structure of dependencies between these entities to support and enable modelling of existing services. By probing existing systems or ad-hoc joining systems and networks, the model self-evolves and mutates as long as new systems or existing ones are provisioned or de-provisioned from within the environments. These configuration items, existing or planned, are stored in the Configuration Management System (CMS according to ITIL 3 [4]), that is captured in a Configuration Management DB (CMDB), leveraging discovered IT Asset data stored in the repositories associated with IT Asset Management, Network and Infrastructure Management, Secured servers, Application Transactions Modelling, Service Registries, etc.

3.3 Cloud Insight

To understand the status of existing systems, and the potential implications of a replacement, the Cloud Insight Technology enables to select and define Metrics and Measurements Modelling capability, and construct required Service Level Modelling as well as the Service Contract Management capabilities. Integrating with Project and Portfolio Management tools, the relative Cost Assembly capability can be added, enabling overall Benchmarking alternatives using the provided Metrics repositories. These capabilities, as well as producing on-going high level service measurement indications of the active composite applications relative to the alternatives, the Selective optimized assembly capability is fulfilled, and informing the end user by a Flexible Dashboard and Reporting capability.

3.4 Cloud Compose

Defining the structure of the composite application (figure 5) modelled to model, is the fundamental element of optimization logic. The technology provides the ability for Contextual Composite Applications modelling and modelling structure patterns of

![Composite Application](image)

Figure 5 – 3-Tera cloud compose

Infrastructure. Further, reusable patterns are provided to the end-user that fulfils the patterns assembly capability, and consequently, according to deployment requirements and constraints, the selective optimized assembly capability is activated.

3.5 Cloud Automation

Automating the change process and synchronizing all technologies for a single change as part of the self-triggered approach of optimization, is
imperative. Using preconfigured modelling process patterns capabilities, the automated abstract infrastructure deployment of 3-Tera (figure 5) is aligned with configured knowledge of self-adaptive and self-service situation detection. These automated processes, assist in delegating changes in hardware and software resources, and also with identity management and federated identify management capabilities, to ensure a proactive response capability of the entire system maintained.

3.6 Service Assurance

Automating the change and aligning the IT composite application with the changing business requirements requires constant closed-loop control, to validate that the changes improve the system overall operational performance according to the service levels. Service Spectrum Assurance technology helps visualize the contextual composite Applications modelling capability aggregating all the collected data from the Data collecting agents and Metrics repositories, providing a monitored structure of the composite application using a simple operational Flexible Dashboard and Reporting.

4 CONCLUSIONS

Business-IT alignment in enterprises, in a cloud era, requires optimized automation. The vast offerings and options for operational and capital efficiency changes daily and reactive management slows down the business is not acceptable. IT is moving from reactive to proactive, from human intense to intelligent automated system, using cloud computing offerings from SaaS, PaaS and IaaS vendors. Analyzing the requirements from such optimized IT service, and its underlying sub-services is required to optimize the supply chain of IT cloud services. Smart technologies, with rich capabilities are the fundamental answer for such a need.

This paper presented both the analyzed capabilities of the major steps of a composite IT cloud service construction, and the underlying IT services technologies that assist in achieving such a task. Optimization of IT engulfs the entire domain of IT. As such, smart integrated technologies, with unified semantics, self-service driven, automated and on-demand are required. As understood, these are just the first steps to fulfill the need to align with the business, or better yet, as stated in this paper, for IT to be a true PART of the business.

REFERENCES


9. Web Site: ITIL 3.0