NoSQL Application Redesign may be Unnecessary for Most Corporation Cloud Migration Deployments

Curtis J. Muller, MS&T, Rolla, MO, USA
Aaron Christopher Davis, MS&T, Rolla, MO, USA

ABSTRACT

A great deal of cloud research implies that application redesign is a necessity in order to take full advantage of migration to the cloud, and developers are urged to consider migrating legacy application onto various NoSQL architectures. It may be argued however, that many enterprises have non-web-centric business models and relatively few of their applications may need the database scaling capability that cloud deployment with NoSQL affords. The authors contend that these corporations, in the process of cloud migration, are less motivated by the promise of NoSQL than by gaining greater operational efficiencies and cost-effectiveness. Several Cloud Infrastructure-as-a-Service (IaaS) migration scenarios for five top cloud vendors were studied using cost versus effectiveness ratios (CE). The results show that the greatest cost-benefit was gained by simply moving off physical resources and onto the cloud. This was coined the ‘simple yet effective’ model, as opposed to continuing along the technology curve to fully implement NoSQL solutions, coined the ‘complex and robust’ model.

KEYWORDS

Application, CEA, Cost, Disaster, DR, DRaaS, Effectiveness, IaaS, Model, NoSQL, Recovery Saas, TCO

INTRODUCTION

In this paper the authors first present a narrative review of key issues facing many corporate enterprises as they migrate key legacy applications to the cloud. Selected literature sources are used as support to present a practical viewpoint that the authors believe should be considered as the consider application redesign during migration analysis. A key issue for the purposes of this paper is to identify how the NoSQL database paradigm may or may not be beneficial for taking advantage of cloud computing, and finding a cost metric to help decide whether or not it is worth the effort for a majority of ‘typical’ corporations to redesign their legacy applications. Towards the end of the paper an approach is offered based on cost effectiveness to support the author’s conclusion that redesign to incorporate NoSQL may in fact be unneeded for most corporations who may be able to realize the majority of cost benefits simple from cloud migration and server virtualization.

Thesis: Re-design to incorporate NoSQL is not a paramount consideration for a large number of corporations and their legacy application migration plans based on cost effectiveness. Most enterprises have non-web-centric business models, are non-multinational in nature, and have a large number of legacy applications that do not need the scaling capability that cloud deployment with NoSQL affords, nor can such legacy applications tolerate the trade-offs between availability and consistency under the
NoSQL model. Such corporations generally, are trying to capture operational cost-effectiveness with the added benefits of robust business continuity planning and disaster recovery, since their primary concern lies in the cloud’s ability to provide operational cost-effectiveness with the added benefits of robust business continuity planning and disaster recovery.

**Cloud Computing May Be Viewed as a Natural Evolution from Client-Server Models**

Corporations have spent a great amount of resources over the years in developing applications and infrastructures before the advent of cloud-computing. Programmers first created applications for stand-alone computer (PC and mainframe) systems. Throughout much of the 1980’s and 1990’s programmers created single applications to be run on a single system or as a file-server, so called one-tier systems. A list by PC Magazine in June 6, 1992 showed the top 5 applications of the time included spreadsheets, word processing applications and the Microsoft Windows 3.1 upgrade (“Ranking of PC “, 1992, p. 33). As technology progressed, developers focused on distributed programming methodologies and the two-tier or the ‘client-server’ model in an increasingly networked world (Miller, 1994).

In two-tier systems the data is not distributed and load balancing and failover systems are generally added for increased reliability and fault tolerance. Truly distributed system development was introduced later in the mid 2000’s with the ‘n-tier’ application model. This model utilizes shared server resources organized into tiers (layers) to separate out the client, business logic, and data management functions (Schuldt, 2009). Separation in this way has allowed a certain degree of scalability as server resources may be added to the business and data management tiers as needed up to a point. For example, (RAC) could be employed to run multiple database instances on additional servers as needed in a RAC cluster against a shared database (Michalewic, 2013). N-tiered systems however are not infinitely scalable. After the ‘easy-wins’ of vertically scaling (extending/expanding) out the physical resources per tier server, the more difficult task of horizontal scaling (e.g. adding new servers to a tier to allow it provide uninterrupted service) may or may not be possible. This is because architectural design choices may prevent the use horizontal scaling best practices such as distributed caching, the use of CDN (Content delivery network), re-coding for asynchronous function/data calls, parallelization, etc. (Gulati, 2013).

In the mid to late 2000’s, several key enabling technologies had matured to allow cloud computing to advance, notably server virtualization (Poelker, 2009; Scroggins, 2013), and the coming of age of widely available broadband access (Kolko, 2010). With the announcement (IBM News Release, 2007) for a “University Initiative to Address Internet-Scale Computing Challenges”, the race was on to define and develop a foundation for what would become cloud services as we know it today, led by industry leaders such as Amazon, Google, IBM, and Microsoft.

Multiple definitions have been offered to describe cloud computing, and summaries are provided by Vaquero et al. (2009) and Vuyyuru et al. (2012), however in essence cloud computing is a shared pool of configurable computing resources broadly divided into three categories or ‘stacks’ of services built on top of one another: Infrastructure-as-a-Service (IaaS), Platform as-a-Service (PaaS), and Software-as-a-Service (SaaS). All these could service can be characterized as services that ‘are available regardless of where you are’, ‘when you need them’, and are ‘pay as you go (PAYG)’. Cloud computing continues to evolve at a fast pace (Wladawsky-Berger, 2015). Overviews can be found by (Vuyyuru et al., 2012; Singh, 2015). A more detailed history was done by (Neto, 2014; “30 years of accumulation”, 2013). The differences between cloud computing and grid computing also have been examined by Hashemi and Bardsiri (2012).
Related Content

A Risk Integration Framework for the Service-Oriented Enterprise
www.igi-global.com/article/a-risk-integration-framework-for-the-service-oriented-enterprise/206894?camid=4v1a

A Survey on MapReduce Implementations
www.igi-global.com/article/a-survey-on-mapreduce-implementations/144188?camid=4v1a
The King is Naked: Discovering that Frequent Customers May Not be Your Best Friend
www.igi-global.com/chapter/king-naked-discovering-frequent-customers/31264?camid=4v1a

Sponsored Search as a Strategic E-Service
www.igi-global.com/article/sponsored-search-strategic-service/2154?camid=4v1a