Migrating Relational Databases to the Cloud
Rethinking the Necessity of Rapid Elasticity

Kevin Williams
Center for Information Systems & Technology,
Claremont Graduate University, 150 E. 10th Street, Claremont, California, U.S.A.

Keywords: Transactional Processing Relational Databases, Cloud Computing, Scalability, Rapid Elasticity, Elasticity, Horizontal Scalability, Vertical Scalability, Inelastic Systems, Incarnational Scalability, ACID Compliance, System Efficiency.

Abstract: Rapid Elasticity is often described as an essential characteristic of cloud computing, but there are some good reasons to rethink how it is described and implemented – especially as it relates to transaction processing relational databases, which are broadly used in many organizations. These types of relational databases, which support transaction processing, strictly adhere to what has been called the ACID compliance model, where the Atomicity, Consistency, Isolation, and Durability of transactions are guaranteed to ensure a reliable transaction system. Databases in the cloud often sacrifice one or more of these essential ACID properties to achieve the desired Rapid Elasticity. This conflict between Rapid Elasticity and ACID compliance explains why relatively few existing transactional processing relational databases have been deployed to the cloud without undergoing significant revision. This paper argues for an expanded definition of the essential characteristic of cloud computing on which the underlying goal of Rapid Elasticity is based, but where the ACID compliance remains intact and many of the advantages of cloud computing can be utilized.

1 INTRODUCTION

Rapid Elasticity (nearly automatic unlimited scaling of computer resources upon demand) is often described as an essential characteristic of cloud computing (Mell and Grance 2011), but there are some good reasons to rethink how this quality is described and implemented – especially as it relates to transaction processing relational databases, as broadly used in many organizations. Relational databases which are ACID compliant (supporting the qualities of Atomicity, Consistency, Isolation, Durability) might not be able to support Rapid Elasticity, but does this mean that they are ineligible to be considered cloud computing? This paper will seek to articulate the goal behind Rapid Elasticity and evaluate whether Rapid Elasticity is the only way to meet this underlying scalability goal. The goals for cloud computing should be scalability and efficiency, not Rapid Elasticity as an essential characteristic. This paper will then suggest an alternative method for cloud computing systems to achieve the goal behind Rapid Elasticity that might have value for transaction processing relational databases and other systems, which are not natively rapidly elastically compliant.

This paper will discuss scalability, elasticity, and efficiency as they relate to rapid elasticity. A comparison will be made between rapidly elastic and inelastic systems. Two types of scalability methods will be discussed: Horizontal Scalability (adding additional systems) and Vertical Scalability (adding more power to existing systems), highlighting some of the advantages and limitations of each. A scalability method for transactional processing relational databases in the cloud will be proposed called Incarnational Scalability. A brief discussion will highlight how Incarnational Scalability could be implemented and what its benefits might be. Finally, the paper concludes with the need to broaden the definition of rapid elasticity as an essential characteristic of cloud computing.

Williams K..
Migrating Relational Databases to the Cloud - Rethinking the Necessity of Rapid Elasticity.
DOI: 10.5220/0004965202050210
In *Proceedings of the 4th International Conference on Cloud Computing and Services Science* (CLOSER-2014), pages 205-210
ISBN: 978-989-758-019-2

2 SCALABILITY, ELASTICITY, AND EFFICIENCY

Scalability is the quality of a system to handle increased workload (Weinstock and Goodenough 2006). Thus as more workload is deployed to a system; the system is able to perform without failure or within acceptable levels. Scalability is a very desirable quality in a system, but overall service quality is based on a number of factors including reliability and responsiveness (Pitt, Watson et al. 1995, Buyya, Yeo et al. 2009).

One of the ways that cloud computing is expected to help solve the challenge of scalability is through the concept of elasticity which is defined by (Herbst, Kounev et al. 2013) as:

> Elasticity is the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manger, such that at each point in time the available resources *match* the current demand as closely as possible.

The appearance of "infinite computing resources" in cloud computing (Armbrust, Fox et al. 2010) seems to hold the promise that all scalability concerns can be resolved through the elasticity that is possible with cloud computing. The distinction between scalability (a goal) and elasticity (a method to reach a goal) can obscure the definition of what a cloud computing system is, since this then makes the claim that all scalability problems can be solved by elastically provisioning and de-provisioning resources.

Alternatively, another goal for systems is that of efficiency, defined as "the amount of resources consumed for processing a given amount of work" (Herbst, Kounev et al. 2013). More work being done by fewer resources has a higher efficiency and also has implications for the ability of the system to scale better. Thus efficiency and scalability are closely related. The greater the efficiency then the greater the ability of the system to handle increased workload.

2.1 Rapid Elasticity

One of the essential characteristics of cloud computing according to (Mell and Grance 2011) is Rapid Elasticity, defined by NIST as:

> Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

This definition suggests that the essential characteristic or goal behind Rapid Elasticity is the unlimited, rapid, and efficient scalability of system capabilities. The problem that is being addressed by the "Rapid Elasticity" seems to be that the demand on the system is dynamic and random and therefore the cloud computing system must be able to respond quickly to the changes in demand, but without wasting resources. In the NIST definition of Rapid Elasticity there is the statement that "capabilities can be provisioned and released," but the word "capability" may not be the best word for what is happening as capability indicates a potential ability, not the usage of that ability.

2.2 Inelastic Systems

The type of system, which performs well when adding additional resources, can be called rapid elastically compliant. There are however, many systems, which are not elastically scalable including transaction processing relational databases. The relational data model when supporting transactional processing creates some dependencies that often get in the way of rapid scalability, including locking, latching, and deadlocks. Locking is meant to protect the integrity of the system; the integrity is considered to be more important than the system's scalability.

Jim Gray described the set of properties in a transactional processing system necessary for a reliable transactional processing database and these became known as ACID compliance: Atomicity, Consistency, Isolation, Durability (Gray 1981). There are few instances of large transactional processing relational databases as broadly used in organizations being moved to cloud systems and supporting the Rapid Elasticity model.

There is a mistaken idea that adding resources to a slow system will improve its performance. Cary Millsap indicates that more or faster resources will only improve database performance if the initial problem was slow or insufficient resources (Millsap and Holt 2003). Millsap continues that on transactional processing relational databases such as Oracle, adding additional resources can in fact exacerbate the performance and scalability problems as the system gets to the root culprit faster.

While there are many successful and highly elastic NoSQL database systems deployed on cloud systems, they often have to relax one or more of the

ACID compliance guarantees. Cassandra, a popular NoSQL database, describes ACID compliance as:

> Unlike relational databases, Cassandra does not offer fully ACID-compliant transactions. There is no locking or transactional dependencies when concurrently updating multiple rows or column families. But if by "transactions" you mean real-time data entry and retrieval, with durability and tunable consistency, then yes.

When the consistency guarantee is relaxed, this means that at some point in time the database can be in an inconsistent state. For example, in an ATM banking transaction, the money must be removed from the account at the same time that the money is released from the ATM. If either of the parts of the transaction fails, then both parts should fail. However, in a system that is eventually consistent, one part might succeed and the other part could fail. This would be disastrous for banking transactions – either for the client or for the bank.

Some efforts to address this conflict between the ACID compliance model (transactional support) and rapid scalability have been attempted (Das 2011) by developing a relational cloud:

> Statements and transactions spanning multiple nodes incur significant overhead, and are the main limiting factor to linear scalability in practice. (Curino, Jones et al. 2011)

Therefore, the awareness of the dependencies of the data relationships was not present when the applications were originally written. This may give some explanation for the difficulty in migrating existing transactional processing relational databases to the cloud and achieving Rapid Elasticity.

3 APPROACHES TO SCALABILITY

There are three main approaches to scalability in cloud computing: horizontal scalability, vertical scalability, and efficiency improvement. Horizontal and vertical scalability have been discussed, but improving the efficiency of the system is an important method to improve scalability – especially for systems that are inelastically scalable, such as transactional processing relational databases.

3.1 Horizontal Scalability

In the NIST definition of computing the idea of "provisioned and released" resources indicates that things are added or removed from the system, but this presupposes that the inability for the system to handle the additional load can be solved by additional resources. Currently, additional resources are added for horizontal scalability or vertical scalability, but only horizontal scaling is able to scale as to "appear to be unlimited."

Horizontal scaling is described as:

> Horizontal scaling is applicable for applications that have a clustered architecture with a gateway or a master node that distributes requests between the worker nodes (or VMs). If the workload increases, additional nodes are added to the cluster. During decrease in workload intensity, some nodes are removed from the cluster freeing up resources. In typical clustered architectures, the gateway maintains a list of nodes that are part of the cluster. The reconfiguration cost of horizontal scaling varies between applications and depends on the ease with which nodes can join or leave the cluster. (Dutta, Gera et al. 2012)

Therefore, the concept of Rapid Elasticity seems to be synonymous with the concept of horizontal scalability.

3.2 Vertical Scalability

Another method for increasing the ability of a system to handle load is vertical scalability. Vertical scaling is described (Dutta, Gera et al. 2012) as:

> Virtualization enables another way to add or remove resources to a virtual machine. Modern hypervisors support online VM resizing allowing one to add CPU or memory resources to a VM without bringing it down. Vertical scaling is used to denote the addition/deletion of resources to a virtual machine.

One of the features of cloud computing is that instead of buying a certain amount of hardware as capital expense, the resources are leased as operational expenses. This permits the rapid and dynamic movement of the system from less powerful hardware to more powerful hardware. While the scalability is not unlimited, the system can be moved to hardware that is able to handle increased load. Additionally, with vertical scalability, existing software doesn't have to be re-written to scale across multiple nodes (as in the case of horizontal scalability). This is of great benefit to systems that are not elastically scalable – such as transactional processing relational databases.

The vertical scalability available with cloud

computing resolves three important problems in organizations: buying too large a system initially, experiencing a delay when adding capacity, and encountering the problem of sunk costs. With vertical scalability, only the resources that are actually needed are allocated and then changed as needed. Depending on the system, the system might adjust resources throughout the day to handle increases and decreases in load. This is not Rapid Elasticity since the number of resources that can be added to an individual virtual machine bound the scale up.

The second benefit with vertical scalability as available with cloud computing is the relatively short delay in new system deployment and adding capacity. When servers are purchased in a capital expenditure scenario, the delays can be substantial, as the capital expenditure must go through the process of a cost benefit analysis, ordering, hardware delivery, and systems configuration.

The third benefit may be the elimination of sunk costs. Sunk costs will have no place in situations where capital expenditures are replaced with operational expenses, as in cloud computing.

Therefore, even for systems, which are not rapidly elastic, there are some very good reasons to move those systems to the cloud including the ability to take advantage of vertical scalability.

3.3 Improving Efficiency

The operation of a system is a collaboration and/or interaction between the system designers, the programmers, the hardware, the system elasticity, and the users, among others. Much of the efforts in cloud computing have seemed to focus on agnostically enabling Rapid Elasticity and ignoring the changes that can be made in systems to improve their performance, efficiency, and scalability.

Rather than concluding that transactional processing relational databases cannot be migrated to the cloud using current technology, this paper argues for a slightly different scalability model. This model uses the resources and benefits of cloud computing to improve scalability and efficiency of systems through improved testing initiatives.

In many organizations, besides the production systems, there are generally one or more development and test systems. In order to have valid tests, in many cases the development and test systems are equivalently sized to that of production. If the systems are less powerful than the production systems, there is the concern that the testing might not behave identically to the way production behaves in the same situation.

In order to have a reliable test environment, the test systems and the production systems must be made as similar as possible. However, the shrinking or sub-setting of full sized environments is difficult in practice. The cost of production-sized environments for testing and/or development can also be prohibitive.

The demand for the use of test systems is not consistent. Often test systems are some of the most highly scheduled systems in an organization. During some periods, such as pre-release testing, the demand for the test system might peak and multiple identical test systems would be required. Then, after testing has concluded, the demand may drop to zero.

The nature of test system demand seems to fit well with the scalability model of cloud computing, where multiple full-sized copies of the production system can be deployed for testing, but are only available during the actual testing. Once the testing has been completed, the test systems can be de-allocated.

3.4 Incarnational Scalability

The use of cloud copies of production systems for testing in order to improve system efficiency is both economically efficient (as it only incurs necessary expenditures) and scalable (as many or as few copies can be created as needed). Therefore, this paper proposes another method for reaching the goals of scalability and efficiency, different from vertical or horizontal scaling: Incarnational Scalability, where separate incarnations of the full production system are deployed for testing. To do effective testing one must test one change at a time so as to isolate whether that change improves the system or not. Unfortunately, most test environments do not have this luxury and multiple changes are tested at the same time because of system availability limitations. Testing in such an environment always leaves the results in doubt as to what change (or changes) actually have the most benefit. However, testing in the cloud, using as many multiple identical systems (incarnational scalability) as needed, allows one to test each change separately and determine its effect. In this model, the use of the cloud does not benefit existing systems directly through the addition resources, instead helps better use what resources you already have through better testing.

Incarnational Scalability would be implemented by instantiating multiple copies of the system in the cloud and then using a technique called A/B Testing.

A/B Testing is used to evaluate iterative optimization changes in websites:

> Using A/B, new ideas can be essentially focus-group tested in real time: Without being told, a fraction of users are diverted to a slightly different version of a given web page and their behavior compared against the mass of users on the standard site. If the new version proves superior—gaining more clicks, longer visits, more purchases—it will displace the original; if the new version is inferior, it's quietly phased out without most users ever seeing it. A/B allows seemingly subjective questions of design—color, layout, image selection, text—to become incontrovertible matters of data-driven social science (Christian 2012).

Incarnational Scalability would function like A/B Testing, but for whole systems. Multiple incarnations of identical test systems with playback of synthetic or recorded system load can be iteratively evaluated to determine the better version between:

> two versions of an element (A and B) and a metric that defines success (Chopra 2010).

An element might be a new index, statistics change, new release of code, or some other change. By evaluating changes in individual elements instead of as a massive set of changes, it should be possible to find the optimal elements for the success metrics.

Therefore, Incarnational Scalability can be used to solve several important testing challenges including:

1. Availability – test systems can be allocated and de-allocated dynamically.
2. Cost – test systems are only paid for when in use.
3. Granularity – testing can now iteratively test multiple potential configurations to find the optimal ones instead of testing a cluster of interrelated changes.
4. Increased thoroughness – since multiple tests for different elements can be evaluated simultaneously, the testing doesn't need to be prioritized for only the most critical changes.
5. Hardware rightsizing – ability to evaluate the benefits of adding more CPUs, memory, etc.

The experimentation made possible by Incarnational Scalability can support a more scientific approach to system optimization and thereby improves scalability.

4 CONCLUSIONS

The goal for cloud providers is a multi-tenant model that can provide services to a multitude of consumers (Rimal, Eunmi et al. 2009). Analysis of different cloud providers suggests that cloud computing will not be held back by definitions, but will continue to expand, as it has, in the past, as new services are invented and deployed. Nevertheless, where we are able to refine definitions, we can improve the precision of our science.

Some benefits of deploying transactional processing relational databases in a cloud model include the ability to perform vertical scaling (adding resources to a single node), which seems to be different from the widely accepted characteristic of Rapid Elasticity (Mell and Grance 2011). The use of incarnational scalability to support improved testing efforts (and thereby improved system efficiency) also seems to be a powerful alterative method to achieving what has been, until now, the goal of Rapid Elasticity – that is, scalability. Therefore, the essential quality of cloud computing should be broadened from mere Rapid Elasticity to any process that uses cloud-based resources and methods to improve scalability and the efficiency of the system.

REFERENCES

"Planet Cassandra FAQ." Retrieved January 13, 2014, from http://planetcassandra.org/Learn/FAQ.

Armbrust, M., et al. (2010). "A view of cloud computing." Commun. ACM **53**(4): 50-58.

Buyya, R., et al. (2009). "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Future Generation Computer Systems **25**(6): 599-616.

Chopra, P. (2010). The Ulitmate Guide to A/B Testing. Smashing Magazine. Online.

Christian, B. (2012). The A/B Test: Inside the Technology That's Changing the Rules of Business. Wired, Condé Nast.

Curino, C., et al. (2011). Relational Cloud: A Database-as-a-Service for the Cloud.". 5th Biennial Conference on Innovative Data.

Systems Research, Asilomar, California.

Das, S. (2011). Scalable and Elastic Transactional Data Stores for Cloud Computing Platforms. Computer Science, UCSB**:** 278.

Dutta, S., et al. (2012). SmartScale: Automatic Application Scaling in Enterprise Clouds. Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on.

Gray, J. (1981). The transaction concept: Virtues and

limitations. VLDB.
Herbst, N. R., et al. (2013). Elasticity in Cloud Computing: What It Is, and What It Is Not. ICAC 13. San Jose, CA.
Mell, P. and T. Grance (2011) The NIST Definition of Cloud Computing. The NIST Definition of Cloud Computing 3.
Millsap, C. and J. Holt (2003). Optimizing Oracle Performance, O'Reilly Media.
Pitt, L. F., et al. (1995). "Service quality: a measure of information systems effectiveness." MIS quarterly: 173-187.
Rimal, B. P., et al. (2009). A Taxonomy and Survey of Cloud Computing Systems. INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on.
Weinstock, C. B. and J. B. Goodenough (2006). On system scalability, DTIC Document.