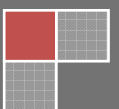


2010

Cloud Computing Infrastructure for Campus Management System

Arguments for cloud computing
infrastructure in UCTI

This paper outlines the current trends in infrastructure development across the industry and makes the case for implementing a cloud computing infrastructure for deployment of Easy Moo campus management system



Contents

Introduction.....3

Easy Moo3

Conclusion5

Reference6

Bibliography.....6

Introduction

The rise of web applications — websites that replace the functions of a software program that was traditionally installed on a personal computer – was one of the hottest topics in the tech industry. Huge numbers of “Web 2.0” startups are competing for user attention, and many observers predict rapid growth for web applications (Rubicon Consulting, Inc. 2007). Usage of a web application can outpace initial expectations. Growth is good for business but creates some real challenges when it comes to trying to keep everything up and running speedily along specially increasing traffic to web applications poses great challenges to database servers. End users are becoming more and more sensitive to the quality of the offered services. This requires addressing issues such as pushing quality of service (QoS) requirements into database processing and providing database system scalability (Ye 2002). But applications suffer from unpredictable load, especially due to events such as breaking news (e.g., Hurricane Katrina) and sudden popularity spikes (e.g., the “Slashdot Effect”) (Amit Manjhi 2009). Investing in a server farm that can accommodate such high loads is not only expensive (particularly after factoring in the management costs) but also risky because the expected customers might not show up. Content Delivery Networks (CDNs) provide such service by maintaining a large, shared infrastructure to absorb the load spikes that may occur for any individual application. However, CDNs currently do not provide a way to scale the database component of a Web application they only provide a way to mitigate network load hence the CDN solution is not sufficient when the database system is the bottle-neck, as such is in many web applications (Rafeeq, 2010).

Easy Moo

EasyMoo, UCTI’s Campus management system is going to be susceptible to such load spikes outlined above especially when new intakes/ Academic year starts. But since these load spikes are temporary investing in infrastructure to absorb this kind of load is wasteful as this capacity is going to be underutilized. Also Since Easy moo is a going to be a localized application the CDN cannot be leveraged for scalability. Furthermore to provide for Database scalability that could absorb load spikes cannot be handled by traditional architecture without having to invest on large number of servers. In order to mitigate these issues it is recommended that a cloud computing infrastructure be implemented.

With a private cloud computing infrastructure server resources can be utilized more efficiently. resources can be shared across different applications smoothly. Resources can be provisioned for critical systems when it is required by allocating resources that has been reserved for other systems. For example at the end of the month when pay roll need to be calculated more server instances can be created after 9:00 clock and resources from other applications such as carpet management system can be assigned to the payroll till the university opens next day. Once university open next day the resources can be given back to carpet system and if more processing capacity is required for the carpet these can be provisioned. These provisioning of resources can be highly automated and would require little human intervention. On a fixed infrastructure deploying of new instances would take more than 20 Minutes for minimal Linux installations and up to one hour for windows server installation and would require lot of human intervention and supervision and cannot be in any way automated. Further more resources such as CPU and RAM cannot be shared across instances while on a cloud infrastructure these can be shared and increased and reduced for instances on the fly.

Another major advantage going for a cloud computing infrastructure is that a public cloud's such as Amazons EC2 can be leveraged when bursts of computing power is required that exceeds the capacity of the private cloud such as when payroll need to be calculated.

Furthermore Management monitoring and maintenance of systems can be automated. Since all the server instances are virtual a base image of the instance is kept at all times in the cloud environment. In case of fault this base instance can be spun up in matter of minutes and take over from the faulty instance smoothly. Furthermore the diagnostics and health servers that built into the cloud environments can automatically monitor the system for faults and automatically restart and redeploy instances with minimal interruption to service. In the current fixed environment it take up to two hours to trouble shoot and fix a faulty system in case of major faults.

Conclusion

With the Advantages outlined it is highly recommended that UCTI deploy a cloud computing infrastructure for deployment of EasyMoo and campus management system. EasyMoo is currently being built from ground up to be cloud aware from the get go as such with a cloud computing infrastructure QOS of the system for all the user is guaranteed with minimal downtime (up to 99.99% availability or 1.3 hour of downtime for every 10,000 hours of operation). Also by depending on private cloud that could connect to a public cloud for more resources computationally intensive task can be completed easy and quickly without having to invest in large number of servers and resources.

Reference

Bibliography

Rafeeq, N. (2010). *Strategies to Mitigate Web Application Database Scalability Issues via the Use of NOSql Solutions*. Kuala Lumpur: Stafford Shire / UCTI.

Interactive Advertising Bureau. (2010). *Internet Advertising Revenue Report 2010*. New york: Interactive Advertising Bureau.

Alexa . (2010, july). *reddit.com*. Retrieved october 1, 2010, from Alexa :
<http://www.alex.com/siteinfo/reddit.com>

Clayton, D. (2010, October 25). *The herald Blog*. Retrieved Nov 1, 2010, from Digg Downsizes Team:
<http://www.blogherald.com/2010/10/25/digg-downsizes-team/>

Cohen, A. (2009, november). *Is there a dump of reddit data available for download like wikipedia dump ?*. Retrieved october 2, 2010, from Reddit.com:
http://www.reddit.com/r/help/comments/9ihua/is_there_a_dump_of_reddit_data_available_for/c0cwmyg

david, K. (2010, March 10). *She who entangles men*. Retrieved April 20, 2010, from Reddit Blog:
<http://blog.reddit.com/2010/03/she-who-entangles-men.html>

Do, C. (2007, June 23). *Seattle Conference on Scalability*. Retrieved april 20, 2010, from google.com:
<http://video.google.com/videoplay?docid=-6304964351441328559#>

Eure, I. (2009, September 9). *Looking to the future with Cassandra*. Retrieved april 26, 2010, from Digg Blog: <http://about.digg.com/blog/looking-future-cassandra>

Evans, E. (2009, May 12). *NOSQL 2009*. Retrieved April 23, 2010, from sym-link: http://blog.sym-link.com/2009/05/12/nosql_2009.html

Fay Chang, J. D. (2006). Bigtable: A Distributed Storage System for Structured Data. *USENIX Symposium on Operating Systems Design and Implementation*. Seattle, WA, USA: Google Inc.

Giuseppe DeCandia, D. H. (2007). Dynamo: Amazon's Highly Available Key-value Store. *Symposium on Operating Systems Principles* (p. 1). Stevenson, WA: Amazon.

Hamilton, J. (2008, July 12). *"Facebook Releases Cassandra as Open Source"*. Retrieved april 2006, 2010, from perspectives:
<http://perspectives.mvdirona.com/2008/07/12/FacebookReleasesCassandraAsOpenSource.aspx>

Lakshman, A. (2008, August 26). *assandra – A structured storage system on a P2P Network*. Retrieved April 25, 2010, from Facebook Engineering's Notes:
http://www.facebook.com/note.php?note_id=24413138919&id=9445547199&index=9

Lawson, L. (2010, Mar 2006). *The Business Pros and Cons of NoSQL*. Retrieved april 23, 2010, from IT Business Edge: <http://www.itbusinessedge.com/cm/blogs/lawson/the-business-pros-and-cons-of-nosql/?cs=40089>

Lynch, N. S. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News v33 Issue 2* , 51-59.

Maxia:, G. (2009). Evolution of “non-relational databases . *ODBMS* , 7.

Mullins, C. S. (2010, april 1). *The Database Report - April 2010*. Retrieved october 20, 2010, from THE DATA ADMINISTRATION NEWSLETTER : <http://www.tdan.com/view-featured-columns/13041>

Oracle. (2010, October 20). *Mysql cluster 7.1*. Retrieved October 20, 2010, from MySQL.com: <http://www.mysql.com/products/cluster/>

Peirce. (1877). *Collected Papers of Charles Sanders Peirce*.

Quinn, J. (2010, March 7). *Saying Yes to NoSQL; Going Steady with Cassandra*. Retrieved April 23, 2010, from Digg : <http://about.digg.com/node/564>

Schonfeld, E. (2010, Nov 1). *The Times UK Lost 4 Million Readers To Its Paywall Experiment*. Retrieved Nov 5, 2010, from Techcurnch : <http://techcrunch.com/2010/11/02/times-paywall-4-million-readers/>

Schonfeld, E. (2010, Nov 2). *The Times UK Lost 4 Million Readers To Its Paywall Experiment*. Retrieved Nov 5, 2010, from TechCrunch: <http://techcrunch.com/2010/11/02/times-paywall-4-million-readers/>

Sherry, G. (2010, jun 10). *Replication, Clustering, and Connection Pooling*. Retrieved oct 12, 2010, from PostgreSQL.org: http://wiki.postgresql.org/wiki/Replication,_Clustering,_and_Connection_Pooling

Shoup, a. (2008, May 7). *Scalability Best Practices: Lessons from eBay*. Retrieved 4 24, 2010, from Infoq: <http://www.infoq.com/articles/ebay-scalability-best-practices>

Vogels, W. (2008). Eventually Consistent: BUILDING RELIABLE DISTRIBUTED SYSTEMS AT A WORLDWIDE SCALE DEMANDS TRADE-OFFS—BETWEEN CONSISTENCY AND AVAILABILITY. *Association for Computing Machinery* , 30-37.

Zicari, R. V. (2009). On NoSQL technologies. *ODBMS* , 9.

Amit Manjhi, Phillip B. Gibbons, Anastassia Ailamaki , Charles Garrod. Invalidation Clues for Database Scalability Services. National Science Foundation, 2009, 1-8.