Cloud Computing Architecture for Social Computing - A Comparison Study of Facebook and Google

[1]Bo-Wen Yang, [2]Wen-Chih Tsai, [3]An-Pin Chen, [4]Singh Ramandeep
[1,2,3]*Institute of Information Management*
National Chiao Tung University
Hsinchu, Taiwan
1001 Ta Hsueh Road, Hsinchu, Taiwan, R.O.C.
nike4859@gmail.com
[4] *Asia Pacific Management-ICT*
Ritsumekan Asia Pacific University
Japan
ramasi06@apu.ac.jp

Abstract

As far as we know, cloud computing is a new business model and the cloud computing architecture is the famous topic recently. Today, everyone enjoy the innovative search engine or social network application for new Internet services no longer require the large capital outlays in hardware to access those service or the human expense to operate it. Currently, Google is the largest search engine and Facebook is the largest social network in the Software as a Service (SaaS). But how them can support the huge requests from world thought each personal computer, mobile device, and smart phone. In this paper, we will try to analysis their backend cloud computing architecture to support future SaaS especially in large social network.

1. Introduction

Recently, millions of users share details of their personal life with vast networks of friends, and onften strangers. The growth and popularity of Social Networks (SNs) [9] is unprecedented and pose unique challenges in terms of scaling, management and maintenance. Hence the cloud technology is very important in SNs domain. Some examples include managing and processing on a network consisting of hundreds of millions of edges on a single machine (e.g.LinkedIn) [5], distributing status updates to millions of users (e.g. Twitter, Facebook) [4, 6] and managing and distributing user generated content (UGC) to millions of users spread geographically [4, 2].

Traditional framwork is scaling up architecture. But in SNs, the scaling up is in general a non-trivial endeavor and, the problem is particularly acute due to the rapid growth that they can potentially experience. Best practice would advice to start with a fully distributed architecture to avoid traditional scalability problems. This is, however, not always an option due to resource scarcity: there is a tradeoff between the application functionality versus future scalability. Consequently, common practice is to implement a prototype of an SN application to run in a scaling out paradigm and then scale it up whenever the application – if ever – takes off.

Postponing scalability is dangerous, specially for SNs system that can experience an extreme growth. For instance, Facebook experienced a growth of 1382% in one month (Feb-May 06) [9], on top of the sustained rapid growth during previous months. Facebook realized by means of considerable down-times that their initial architecture was not adequate to sustain traffic generated by millions of users. The transition involved a continuous redesign and re-implementation of their initial system [6] until it finally resembled the prototypical distributed architecture depicted in Fig. 1 (left side). The typical distributed architecture consists of a number of layers: Application logic (Ruby on Rails, Scala), caching (Memcache, SQL query caching) and database backend (RDBMS clusters, CouchDB, Google’s BigTable or Amazon’s Dynamo) that interact through asynchronous message passing. In this highly distributed architecture, each layer consists of a number of machines devoted to perform their respective tasks and scaling up can be addressed by increasing the number of servers in each layer.

978-0-7695-4375-8/11 $26.00 © 2011 IEEE
DOI 10.1109/ASONAM.2011.79

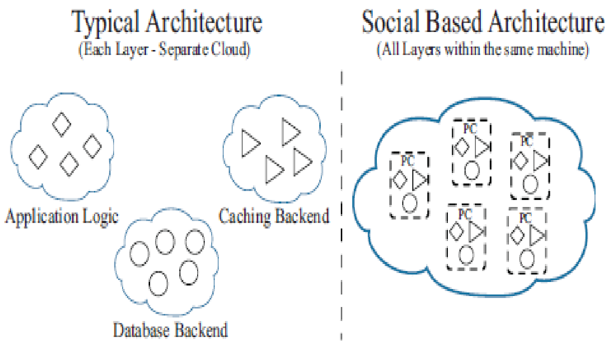

Figure 1: Typical DB Architecture vs Social Based Architecture

2. Background

2.1. Google and Facebook

In Table 1, Google and Facebook are come form education institute. In Fig.2 ,in 2004 , Facebook may have a fraction of the employees that Google does, the company is gaining increasing attention a potential rival to Google as it's valuation has skyrocketed beyond $30 billion and it appears to be organizing (and making accessible via search) the semantic web. In 2006, there are numerous articles published comparing the two companies, we thought it would be fun to create a graphic depicting the growing tension between the two internet giants.

Table 1: Background introduction

	Google	facebook
Found	1997 (From Stanford University)	2004 (From Harvard University)
IPO	2004 (EPS USD26.31 in 2010)	2011 (Plan)
Market Cap. (2010)	USD150.80B (IBM: 162.14, Intel 105.12)	USD30-40B (Projected)
Revenue (2010)	USD29.3B	USD1.5B (Projected)
Major Service	Search Service (162M searches/daily) Gmail Service (193M active users)	Social Network (500M active users)

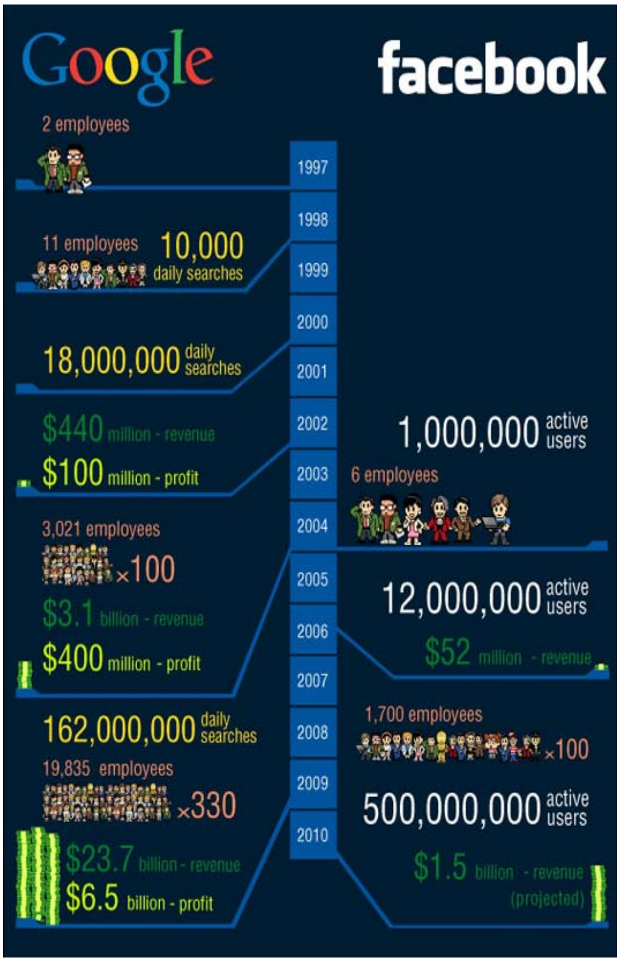

Figure2: Google vs. Faceboo's history

3. Cloud Architecture Comparison

In order to compare Google and Facebook's cloud technology, we will follow below items to compare this two cloud service provider that include the business model, server hardware, data center and distributed architecture. We try to thought the comparison to know the future cloud technology trend in the social network domain.

3.1 Software as a Service (SaaS) Business Model

In Table 2, this two service provider have the same revenue source. However, in Feb, 2010 Facebook toppled Google as the top online destination from internet traffic report. That's mean the social network is more attractive than search engine.

Table 2: SaaS business model comparison

Items	Google	facebook
Service	Search engine	Social network
Source of Revenue	Online advertisement	Online advertisement
Targeted Customer	General customers	Selected customers (Filter by location, interest, and age etc)
Charging Model	Cost-per-click(CPC): USD$0.6	Cost-per-click(CPC): USD$0.6

Table 3: Server H/W comparison

Item	Google	facebook
PC	Commodity PC Server (***No redundant components)	Commodity PC Server (****No redundant components)
CPU	Quart cores	Quart cores
Memory	16GB	16GB
HDD	2TB	USB
Storage Server	No	Tearing (MemCache->DB->HDFS)
Network	1GB	1GB

Figure 3: SaaS business model comparison

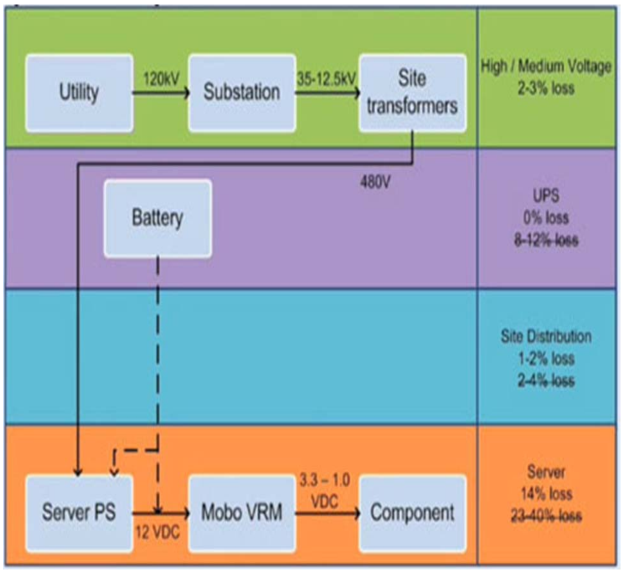

Figure 4: Included battery H/W to save the UPS cost

3.2 Commodity PC Solution

In Table 3, these two service providers have the similar commodity PC server hardware. They used the no redundant component PC. They used the application redundant and scale-out architecture to fix the hardware crash issue. Hence, they use the commodity PC to save the hardware cost and investment. The commodity use the Direct Current(DC) to avoid the DC to Alternating Current transform waste. Additionally, the commodity PC embedded the battery to save the UPS investment and maintain cost.

3.3 Scaling Out Data Center

In below Table 4, these two service providers have the similar data center design. They have the low measurements of Power Usage effectiveness (PUE). So they can fully utilized their power in data center. They used the water cooling solution in their rack of data center. No UPS to avoid 10%-16% the DC to AC power loss as Fig. 6.

Table 4: Data Center comparison

Item	Google	facebook
Measure PUE	1.3	Target 1.15
Manage air flow	Water cooling	Water cooling
Adjust the thermostat	26 C	26 C
UPS	No UPS (Battery)	No UPS (Battery)
Use free cooling	River cooling/ Sea water/Wind cooling	River cooling/ Wind cooling

3.4 Distributed file system

They all used the Distributed File System (DFS) on the commodity PC. So the have the lowest storage cost. Facebook can save the daily 2TB transaction data that include photo, messages and log. Google used the Google File System (GFS) to save the global web in 3 copies to avoid the data crash. Due to the social network can leverage the social group concept to partition the data. Hence, Facebook introduce data stored in memory architecture for data in 2-day to accelerate the performance and store the old data (>2-day) in external storages

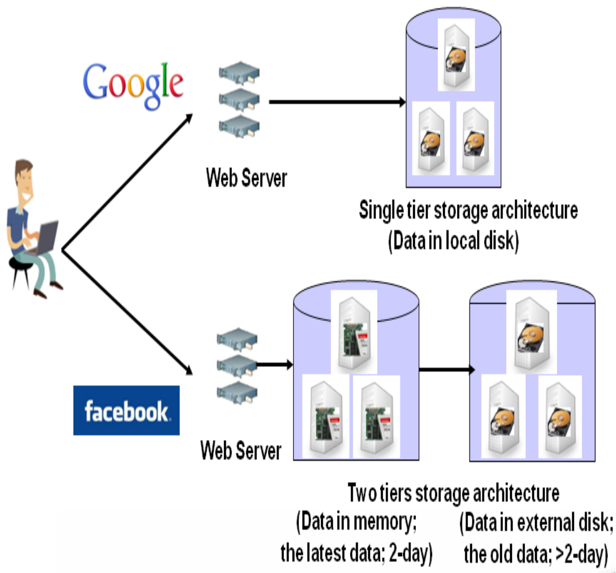

Figure 5:Distributed File System

4. Conclusion

This paper has presented how Google and Facebook work in cloud technology. Since, in this paper comparison reulst, we can conclude that break the above conundrum by maintaining the early advantages of common practice without incurring in the costs of transition to a fully distributed architecture in the case the SN application becomes successful.

5. References

[1] Kenneth Church, Albert Greenberg, and James Hamilton. On delivering embarrassingly distributed cloud services. In ACM HotNets VII, 2008.

[2] Facebook. Cassandra. http://www.facebook.com/note.php?noteid=24413138919.

[3] M.S. Granovetter. The Strength of Weak Ties. The American Journal of Sociology, 78(6):1360–1380, 1973.

[4] James Hamilton. Geo-replication at facebook. http://perspectives.mvdirona.com/2008/08/21/GeoReplicationAtFacebook.aspx.

[5] James Hamilton. Scaling linkedin. http://perspectives.mvdirona.com/2008/06/08/ScalingLinkedIn.aspx.

[6] highscalability.com. Twitter architecture. http://highscalability.com/scaling-twitter-makingtwitter-10000-percent-faster.

[7] Nikolaos Laoutaris, Pablo Rodriguez, and Laurent Massoulie. Echos: edge capacity hosting overlays of nano data centers. SIGCOMM Comput. Commun.Rev.,38(1):51–54, 2008.

[8] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. CoRR, abs/0810.1355,2008.

[9] Nielsen Media. Growth of twitter. http://blog.nielsen.com/nielsenwire/onlinemobile/twitters-tweet-smell-of-success/.

[10] Alan Mislove, Massimiliano Marcon, Krishna P.Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pages 29–42.ACM,2007.

[11] M.E.J. Newman and J. Park. Why social networks are different from other types of networks. Phys. Rev. E,68:036122, 2003.

[12] Josep M. Pujol, Vijay Erramilli, and Pablo Rodriguez. Divide and conquer: Partitioning online social networks. CoRR, abs/0905.4918, 2009.
[13] Rich Miller, http://www.datacenterknowledge.com/archives/2011/01/03/will-facebook-funding-finance-facilities/ , January,2011
[14] Rich Miller,http://www.datacenterknowledge.com/archives/2011/01/03/will-facebook-funding-finance-facilities/ , October, 2010
[15] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia,Above the Clouds: A Berkeley View of Cloud Computing, February 10, 2009