CARLETON UNIVERSITY

WEB CONFERENCE SYSTEM SCALABILITY:
DIMENSION AND MEASUREMENT

By

Adefemi Michael Debo-Omidokun

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfilment of the requirements for the degree of

Master of Applied Science in Technology Innovation Management

Department of Systems and Computer Engineering

Carleton University

Ottawa Ontario, Canada,

K1S 5B6

June, 2012

Your file Votre référence
ISBN: 978-0-494-93481-4

Our file Notre référence
ISBN: 978-0-494-93481-4

Abstract

Software developers, design architects and technical managers of web conference system applications require a more accurate and dependable scalability dimensions and measurement metrics. Few empirical studies have examined web conference system scalability and dependable and accurate scalability dimensions with measurement metrics of web conference systems are lacking.

A controlled experiment was used to test data from different versions of a web conference system application. Results suggest that data size, speed and cost of scaling, should be considered when scaling a web conference system application and particular attention should be paid to extensible architecture in early stages of software development process. Findings have implications for managers of software groups developing real time web conference systems.

Acknowledgements

The Most High God for His goodness, mercy and faithfulness. My male mentor, Late Prince 'Debo Omidokun, my mum, Deaconess Bolaji Omidokun, my siblings and in-laws, the Onishiles.

Finally, I like to thank my supervisor Professor Tony Bailetti for his support, guidance and leadership to me. Also, my classmates who have impacted my life and acted as destiny helpers to me on the TIM program.

Dedication

To my soul mate Moyosola and the godly seeds in my life

'Kiishi, 'Kiiye and 'Kiitan

Also to my late younger brother, Akintunde 'Debo-Omidokun

who moved to glory in the course of my program at Carleton University.

Table of Contents

Approval page....................ii

Abstract....................iii

Acknowledgements....................iv

Dedication....................v

Table of Contents....................vi

List of Tables....................vii

List of Figures....................viii

1. INTRODUCTION....................1

1.1 Deliverables....................2

1.2 Contributions....................4

1.3 Relevance....................4

1.4 Organization....................6

2. LITERATURE REVIEW....................7

2.1 Web conference system....................7

2.2 Scalability....................9

2.3 Dimensions....................11

2.4 Scalability measurement....................15

2.5 Factors that affect web conference system scalability....................16

2.6 Scalability Models....................19

2.6.1 Amdahl's Law Scalability Model....................20

2.6.2 Gustafson's Law Scalability Model....................22

2.6.3 Linear Scalability....................24

2.7 BigBlueButton 25

2.8 Lessons learned 27

3. METHOD 30

3.1 Objective 30

3.2 Approach 30

3.3 Case studied 31

3.4 Unit of analysis 31

3.5 Research method 32

3.5.1 Identify key factors that affect web conferencing system scalability 34

3.5.2 Identify different versions of BigBlueButton developed overtime 34

3.5.3 Collect logs (data) of different versions of BigBlueButton 34

3.5.4 Test data obtained for relationship between different variables 35

3.5.5 Use regression analysis model to analyze results obtained from tests 37

3.5.6 Insights development 38

4. RESULTS 39

4.1 Results of 5 tests carried out 39

4.1.1 Results of Test 1. CPU usage against fixed number of users 39

4.1.2 Results of Test 2. CPU usage against number of users 41

4.1.3 Results of Test 3. System throughput against varied number of users 45

4.1.4 Results of Test 4. System throughput against CPU utilization 47

4.1.5 Results of Test 5. Audio quality 48

4.2 Answer to research question 1 51

4.2.1 Data size 52

4.2.2 Speed 52

4.2.3 Cost 53

4.3 Answer to research question 2 54

4.3.1 Lines of Code (LOC) 54

4.3.2 Number of Modules (NOM) 55

4.3.3 Defect metrics 55

4.3.4 Number of Functions (NOF) 55

4.4 Insights gained for software managers, architects and developers 56

5 DISCUSSION OF RESULTS 58

5.1 Discussion of results of five tests 58

5.2 Relating results to extant literature 63

6. CONCLUSION, LIMITATIONS AND FUTURE RESEARCH 68

6.1 Conclusions 68

6.2 Limitations 69

6.3 Future Research 69

REFERENCES 69

List of Tables

Table 1: Scalability dimensions....12

Table 2: Factors that affect web conference system scalability....16

Table 3: Research method steps....32

Table 4: CPU usage with 100 users by BigBlueButton versions....39

Table 5: CPU usage with varied users by BigBlueButton versions....42

Table 6: Regression analysis summary output for CPU usage with varied users....44

Table 7: Anova analysis for CPU usage with varied users....44

Table 8: Coefficients of CPU usage vs. varied users....44

Table 9: Residual output for CPU usage vs. varied users....45

Table 10: Concurrent users vs. throughput at different times....45

Table 11: Regression analysis residual output....46

List of Figures

Figure 1: Web conference system vendors ecosystem..19

Figure 2: Amdahl`s law explanation..20

Figure 3: Gustafson`s law counter argument to Amdahl`s law..22

Figure 4: Linear scalability graph..25

Figure 5: BigBlueButton main interface..26

Figure 6: CPU utilization vs. fixed users by BigBlueButton version..40

Figure 7: Area chart of CPU utilization vs. fixed users by BigBlueButton version..........40

Figure 8: Regression of CPU usage vs. fixed users by BigBlueButton version.............41

Figure 9: BigBlueButton residual plot for CPU utilization vs. fixed users.......................41

Figure 10: CPU usage vs. varied users by BigBlueButton version................................42

Figure 11: Chart of CPU usage vs. varied users by BigBlueButton version..................43

Figure 12: CPU usage vs. varied users at different times..43

Figure 13: Throughput with varied user at different times..46

Figure 14: Residual plot of varied users vs. throughput at different times.....................46

Figure 15: Throughput vs. CPU utilization with varied user at different times...............47

Figure 16: Area chart of CPU utilization with varied user at different times..................47

Figure 17: System performance in BigBlueButton version 0.7..48

Figure 18: Unstable memory usage in BigBlueButton version 0.7................................49

Figure 19: Unstable system thread in BigBlueButton version 0.7................................49

Figure 20: System memory performance in BigBlueButton version 0.71.......................50

Figure 21: Stable memory usage in BigBlueButton version 0.71..................................50

Figure 22: Stable system thread in BigBlueButton version 0.71..................................51

1. INTRODUCTION

The objective of this research is to answer two research questions:

1. What are the dimensions of web conference system scalability?
2. What metrics should be measured in web conference system scalability?

For the purpose of this research, web conferencing system scalability is defined as the ability of a real-time communications (RTC) web system to accommodate changes in transaction volume, handle growing amounts of work successfully or ability to be enlarged to accommodate growth without major changes to the system. Web conferencing system scalability is a characteristic of a real-time communication web system that describes its capability to cope and perform under an increased or expanding workload and functionalities with additional features, (Mackenzie, 2002).

Dimensions are specific ranges of factors that determine a system's scalability. These include: ability to increase total number of users, number of concurrent users, number of users' locations, extent of the data store, transaction volume, output volume, response time, and adding new applications or customers to the system.

Measurement of web conference system scalability requires putting certain metrics or criteria in place. This requires analysis of certain system parameters like system operations, functionalities and performance after the system has been scaled. Metric is a parameter or standard of quantitative assessment used for gauging, comparison or tracking of performance or set goals like timelines and other factors that impacts web conference system scalability as a multi-sided property of the system which can be captured, (Woodside, 2009).

Typically, system scalability metrics are data or parameters which relates to system software development. They focus on measurement of system performance and functionalities against defined and specified system requirements and value before and after the system has been scaled. Metrics could also be used to assess or determine efficiency of the system by measuring criteria like: quality of service, levels of resource saturation, percentage of system up time, system throughput, number of real time users, CPU utilization — number of users supported versus CPU utilization, response time, and overall usability.

1.1 Deliverables

This research produced three deliverables:

- Dimensions of web conference system scalability
- Web conference system scalability measurement metrics
- Insights relevant to technical managers and software developers

Dimensions of web conferencing system scalability are frameworks which guide the approach to be used when scaling a web conference system, and the understanding of web conference system scalability subject matter. Since poor system performance often translates into substantial costs and other functional challenges, system scalability is one of the most important quality attributes of today's software systems. However, despite the importance of system scalability in web conference systems, it is poorly understood. Hence, there is no generally accepted definition or approach for web conference system scalability.

To deliver these scalability dimensions, Gustafson's law (Smith and Williams, 2004) is employed as it is appropriate for evaluating web conference system application scalability. John Gustafson, (1988) refined Amdahl's law (Amdahl, 1967) by considering large-scale resources and tasks. Gustafson's law states that, with increasing data size, speedup obtained through parallelization increases, because parallel work increases with data size.

Web conference system scalability measurement metrics focuses on parameters or standards of quantitative assessment used for gauging and comparing system performance or tracking output result of web conference system. These include density of real-time users per server instance, concurrent user population and response time. Metrics are also used by analysts to compare performance of system application against set of system goals. Insights gained by working on this research are made available to technical managers and software developers. Developers will find these insights helpful when scaling web conference systems.

1.2 Contributions

There are at least two contributions from this research. Firstly, the research introduces an approach which simplifies scalability method of web conference system. It emphasizes the need to pay attention to extensible architecture of any software application from inception of product development process to allow for flexibility. Paying attention to this saves developers time during product development, add value to existing users and attract potential users of the product.

The second contribution of this research focuses on importance of parallelism in the distribution of work load in a scalable system instead of sequential distribution of work load. Parallelism helps achieve certifiable quality goals that can be analyzed in order to accomplish cost-effective and efficient web conference system scalability.

1.3 Relevance

This research is relevant to at least three groups:

i) Academia and researchers, ii) Software developers, and iii) Technical Managers of software developing organizations

E-Learning programs are a main driver of web conference system scalability research. However, few studies have applied analytic research to support web conference system scalability model. This research will be of interest to academic community as the models it applies to study web conference system scalability can be used to examine other real-time communication applications.

Software developers will also find this research relevant. Results derived from this research can be used by software developers to guide their design and development and collaborative efforts in the case of open source software. Developers play a critical role in the evolution of emerging and existing real-time communication systems, especially web conference systems (Batty,2010). In many cases, ability to build complex software systems outstrips ability to plan and carry out rigorous analysis which predicts the behaviour of these systems under conditions of increasing load (Moyle, 2010).

Technical managers and managers of software development companies will also find this research relevant. This research provides insight into dimensioning, measurement and web conference system scalability models which empowers managerial decisions and shape strategies within an organization. The better management teams understand system scalability models, the better it is for all in an organization. Hagel (2008), noted that "well-executed shaping strategies mobilize masses of players to learn from and share risk with one another – creating a profitable future for all."

1.4 Organization

This thesis is organized into six chapters. Chapter 1 is the introduction. Chapter 2 reviews relevant literatures. Chapter 3 presents the method used to undertake this research. Chapter 4 presents data analysis and results. Chapter 5 discusses the results. Chapter 6 presents the conclusion, research limitations, and future research opportunities.

2. LITERATURE REVIEW

The first seven sections of Chapter 2 provide a review of relevant literature streams. Sections 2.1 to 2.7 review the following literature streams: i) Web conference system; ii) Scalability; iii) System dimensioning; iv) Scalability measurement; v) Factors that affect web conference system scalability; vi) Scalability models; and vii) BigBlueButton. Section 2.8 provides lessons learned from literature review.

2.1 Web conference system

Today, there are many leading proprietary and open source web conferencing systems. They are used for meetings, webinars, and eLearning. They offer advanced accessibility features to promote a positive and effective experience for all users with real-time collaboration via a combination of data streaming, voice and video. However, care must be taken to differentiate between web conferencing system and video conferencing. Web conferencing system allows participants to communicate in real - time using internet, optional microphone and a web cam while video conferencing requires that participants physically travel to a specific location to use cameras and equipment that communicates with similar equipment at another location (Schroeder, 2007). Web conferencing and video conferencing afford participants to be active and be involved in discussions as they are generally planned as a 2-way communication.

The cost of having to physically attend a meeting, conference or lectures actually drove the development of web conference systems in the 1990s. Web conference system offers to match (and perhaps exceed) customary strengths of face-to-face meetings. A web conferencing system allows for both synchronous and asynchronous communications; it provides opportunity for a rich learning environment and imitates face to face experience with features and processes that records sessions and can be viewed at a later date (Foreman & Jenkins, 2005).

The evolution of web conference systems overcame many conventional barriers to real-time communications, and demand for better technology is outpacing the capacity of existing infrastructure. Unacceptably long response times, bad audio and video quality, and other shortcomings provide evidence that existing infrastructure require upgrade or expansion (Datta, Dutta, Thomas, & VanderMeer, 2003).

Web conference applications that support real-time interactions with more people and meet exceptional users expectations, is still faced with diverse challenges, among which is scalability of the system which has hindered much desired expansion and evolution of this technology. For instance, the immediate demand for real-time collaboration and interactions between virtual meeting participants, lecturer and remote students in educational institutions that offer online degree or eLearning programs require system scalability. In some situations, live streaming of contents makes it difficult to translate multimedia information into text or to provide alternatives in many cases.

Smith and Williams (2000), warn that any web application system that is not able to meet users requests or needs will lose such users to other providers of the same application. Responsiveness is an essential quality of service of web conference system applications. It is expedient for software developers to be able to ascertain whether users' needs and requirements are being met or there is a need to scale in order to meet projected demand.

2.2 Scalability

Scalability is a characteristic of a web conference system. According to Duboc (2009), scalability lacks a precise definition and a systematic, uniform and consistent treatment which can be applied across application domains and system designs. Lack of a consistent, uniform and systematic treatment of scalability makes it difficult to identify and avoid scalability problems.

Scalability is an attribute that is often used to describe multiprocessor systems (Hill, 1990). However, some are of the opinion that scalability adds more to marketing brochures than to technical systems (Hill, 1990; Duboc, 2009).

A generally accepted definition for scalability does not exist. The word scalability refers to the ability to smoothly and economically grow a system. An increase in capacity of a system should have an acceptable incremental cost per capacity unit without encountering limits that forces either disruptive system upgrade implementation or replacement, or compromise system requirements (Winter, 2009).

The only instance in which the term scalability is used consistently is in parallel computing literature stream (Luke, 1993; Kumar & Gupta, 1994), and it refers to ability of a system to function proficiently and with sufficient quality of service (QoS) over a given range of goals and configurations.

Software developers perceive web conference system scalability as a design challenge because foundational platforms and technologies are not all inherently accessible (Antona, Basdekis, Klironomos, & Stephanidis, 2006).

According to Duboc,(2007), a system architecture is scalable if it has a linear (or sub-linear) increase in physical resource usage as capacity increases. This is represented by the equation below:

$$speedup(n,x) = \frac{time(1,x)}{time(n,x)}$$

Where time *(n, x)* is the time required to solve a problem of size[2] *x*. The speedup on a problem of size *x* with *n* processors is the execution time on one processor divided by the time on *n* processors:

$$speedup(n,x) = \frac{time(1,x)}{time(n,x)}$$

One important factor that is related to speedup is efficiency, which is speed divided by the number of processors:

$$efficiency\ (n,x) = \frac{speedup(n,x)}{n}$$

$$= \frac{time(1,x)}{time(n,x)} \Big/ n$$

In general, the best possible efficiency is one[3], implying best speedup is linear, $speedup(n,x) = n$

Therefore, a system is scalable if,

$$efficiency^{*}(n,x) = 0([n^{\frac{1}{3}} log\ n])^{-1}$$

for all algorithms, number of processors *n* and problem sizes *x* (Hill, 1990).

System scalability is also faced with diverse limitations that include scope of scaling, system design or architecture constraints, and minimum short-term cost. However, the cost involved in system scalability is redeemed over the life cycle of the system. Thus, the choice of what scalable solutions used by a developer determines the overall cost and long-term effect on the system or product, whether the system is able to achieve its goal.

2.3 Dimensions

Scalability dimensions can be organized into the following: performance, physical size, economics, and load. Gustavson (1994), argues that a system architecture must consider these dimensions for it to be successful over a significant time period after the system is scaled. Table 1 shows the different scalability dimensions that are considered in a scalable system:

Dimension	Definition	Notes on important points
Performance	System performance is the observation of the trend in system efficiency in response to increasing demands (Butler,2009). Performance is how rapidly an operation completes a request (Shoup, 2008; Little, 1992).	The outcome of increase in the number of processors is directly proportional to the system throughput, which results in steady cost value for the system (Gustavson, 1994; Lin, 2001). Performance measurement requires event processing time, batch throughput, transaction rates, database size, users perception and resources used to service a request (Gunther, 2011; Shoup 2008). Highly scalable systems can cause degradation if resources available for the scalability process are constrained and extra demands beyond the available resources and its capacity is placed on the system (Butler, 2009). System performance depends on many design and implementation choices in the SW layers that are used (Gerrit 2011). Little`s law also confirms consistent system throughput is a sign of good performance (Little, 1992)
Physical size	Physical size is the quantitative attribute of the physical components that constitutes the scalable system and its interface with other systems (Gustavson, 1994).	Interconnectivity barriers of scalable system components could limit the appropriate interface required for scalable systems. For system components to work together in a scalable architecture, interconnectivity barrier has to be overcome. Therefore, technology dependent boundaries should be ignored and more fundamental restrictions should be focused on for a scalable system to have a significant scalable life cycle (Gustavson, 1994). In web conference system, the size of a bus, known as its width, is important

		because it determines how much data can be transmitted over a distance at any given time. Hence, as operating frequencies rise, the practical size of a bus must decrease because time available for signal propagation varies inversely with frequency (Gustavson, 1994).
Economics	Economics is the observation of the system trend in relation to the cost required to maintain performance in response to the increasing demands on the system (Butler, 2009).	A system is considered economically scalable if the cost of maintaining its performance, reliability, or other characteristics increases slowly as compared with increasing loads (Butler, 2009). The design of a scalable system architecture for cost-effective high-end interconnects must be based on components used within high-volume low-cost applications. The interconnectivity of the cost-effective high end system makes scalable system attractive to the low-end high-volume market, and the architectural features make it attractive to the high-end low-volume market. In this way the benefits of the high volume production accrue to a wide spectrum of users and markets (Gustavson, 2004) . Cost may be quantified in many ways, including but not limited to response time, processing overhead, space, memory, or even money. A system that does not scale well adds to labour costs or harms the quality of service. It can delay or deprive the user of revenue opportunities (Bondi, 2000).
Load	Load is the ability of a system to function	Load scalability can be undermined by factors like i) the scheduling of a shared resource, ii) the

	without undue delay and unproductive resource consumption or resource contention at light, moderate, or heavy loads while making good use of available resources. (Bondi,2000)	scheduling of a class of resources in a manner that increases its own usage, and iii) inadequate exploitation of parallelism (Bondi, 2000). Poor load scalability occurs in a system when one of the resources performance measure is an increasing function of itself. This is called selfexpanding. Selfexpansion occurs in scalable system when the holding time of a resource is increased by contention for a like resource, whose holding time is increased by the delay incurred by the customer wishing to free it. Care must be taken to avoid self-expansion when scaling a system because it diminishes scalability in systems by reducing the traffic volume at which saturation occurs (Bondi, 2000; Duboc, 2009).

Table 1: Scalability dimensions

2.4 Scalability measurement

Measurement is often regarded as being only a means by which observations are expressed numerically in order to investigate causal relations or associations (Moghaddam & Moballeghi, 2008; Osayomi, 2007). Measurement provides a means to assess the status of software program to determine if there are errors or in need of corrective action and process improvement. Measurement assessment must be based on up-to-date measures that reflect current program or system status, both in relation to the program plan and to models of expected performance drawn from historical data of similar programs and system (Smith, 2000).

Kuhn (1961), argues that measurement often plays a more important role in system scalability. Measurement recognizes the importance of gauging set parameters in a scalable system and how careful and sensitive developers and designers must be when drawing conclusions on web conference system scalability measurements and provides benefits at the strategic, program, and technical levels in a scalable system (Elshoff, 1978). Measurement determines the result of a system scalability whether it was successful or not, as a good measurement program is an asset to the success of scalability by facilitating early detection of problems, and by providing quantitative insights on significant development or design issues, as developers are able to identify, resolve, and limit risk issues before they surface. Thus, for the scalability of the web conference system to be effective, the number of network device metrics collected and the number of events that the web conference system could process must be used to measure the performance of the system (Campbell, Luke, & Koster, 1995).

Scalability cannot be measured just based on the number of physical systems monitored, but must also take into account the number of application instances being managed. The total number of metrics collected from the system like CPU availability and usage, is a better indicator of web conference system scalability than the number of servers and applications monitored, because this helps to predict system behaviour in the face of varying resources and an increasing workload (Gunther, 2007). It is very important also to pay attention to the time the scalability of a system is measured, because if the measurement is obtained at a wrong time, it will be difficult to distinguish between bad scalability or a slowdown in workload (Carvalho & Pereira, 2010).

2.5 Factors that affect web conference system scalability

The following factors were identified as affecting web conference system scalability:

i) system performance (Bondi,2000); ii) system extensibility (Bershad & Grimm, 1995), and iii) system expandability (Plummer, 2004).

Factors that affect scalability	Definition of factors	Notes on important points
System Performance	System performance is the accomplishment of a given task measured against pre-set known standards of accuracy,	Many web conference systems fall short of meeting the performance objectives after they are scaled (Iyengar, Squillante & Zhang, 2000). Some other systems perform satisfactorily with a small number of users but do not scale to support increased usage. Most performance failures are due to a lack of consideration of performance issues early in the development process,

	completeness, cost, and speed in a scalable system (Bondi 2000; Miller, Albert, Lam, Konstan, & Riedl, 2003).	not giving thought to the system performance in early phases of the design of the system, either in the development process, in the architectural phase or when planning for system expansion. System failures lead to poor performance as a result of problems in the architecture or design rather than the implementation, which then result in damaged customer relations, productivity loss for users, revenue loss, cost overruns due to tuning, redesign or re-scaling, and missed market windows (Smith & Williams, 2002). Many systems designers and performance analysts have an intuitive feel for scalability, but the determining factors are not always clear. (Cardoso, Sheth, & Miller, 2002). Performance is functionality plus usability (Whitworth & Fjermestad, 2005) According to Clements (1996): *"Performance is largely a function of the frequency and nature of inter-component communication, in addition to the performance characteristics of the components themselves, and hence can be predicted by studying the architecture of a system", (*Smith et al., 2002).
System extensibility	System extensibility refers to the ability of the scaled system to allow and accept significant extensions or integration of web conferencing system	Conscious efforts should be made to create a highly extensible and flexible solution that integrates well with other systems. Figure 1 shows how several web conference system vendors, especially Adobe, built a unique extension ecosystem that provides integrations to other systems and mobile support. Overall, providing extensibility in a scalable system does not introduce significant overheads and does not limit the system

	capabilities by users or developers, without major rewrite of the code or changes in its basic architecture (Bershard,1995; Schermerhorn, Minerick, Rijks, & Freeh, 2001).	scalability (Schermerhorn et al., 2001). Scalable system applications must be built for expanding/enhancing the system with new capabilities without having to make major changes. The extensible scalable system provides application interfaces to integrate web communication capabilities with the application network, including integration with portals, reporting applications, CRM systems, content management systems, and other corporate systems (Flouris, 2009; Flouris, Lachaize, & Bilas, 2001)
System expandability	System expandability is the ability of a system to accommodate additions to its capacity or capabilities (Abachi & Amiripour, 2007; Plummer, 2004).	System expandability is one of the scalable system architecture's greatest strength. It supports flexibility for developers to easily add/remove/modify system architectural components. In a scalable system, system architecture supports expandability and allow developers to more easily expand the system to accommodate additions to its capacity (Plummer, 2004). System expandability is very important in keeping development time and costs down, and it ultimately results in a better quality system upgrade (Plummer, 2004). System expandability is also an essential OS feature, as it allows the reconfiguration of the kernel, as well as the provisioning of the wireless sensor network concept in the kernel (Cha, Choi, Jung, Kim, Shin, Yoo, Yoon, 2007). It also includes the ability of the system to support more or larger numbers of concurrent users, greater number of request or requirements from users

		and the throughput time (Datta et al., 2003).

Table 2: Factors that affect web conference system scalability

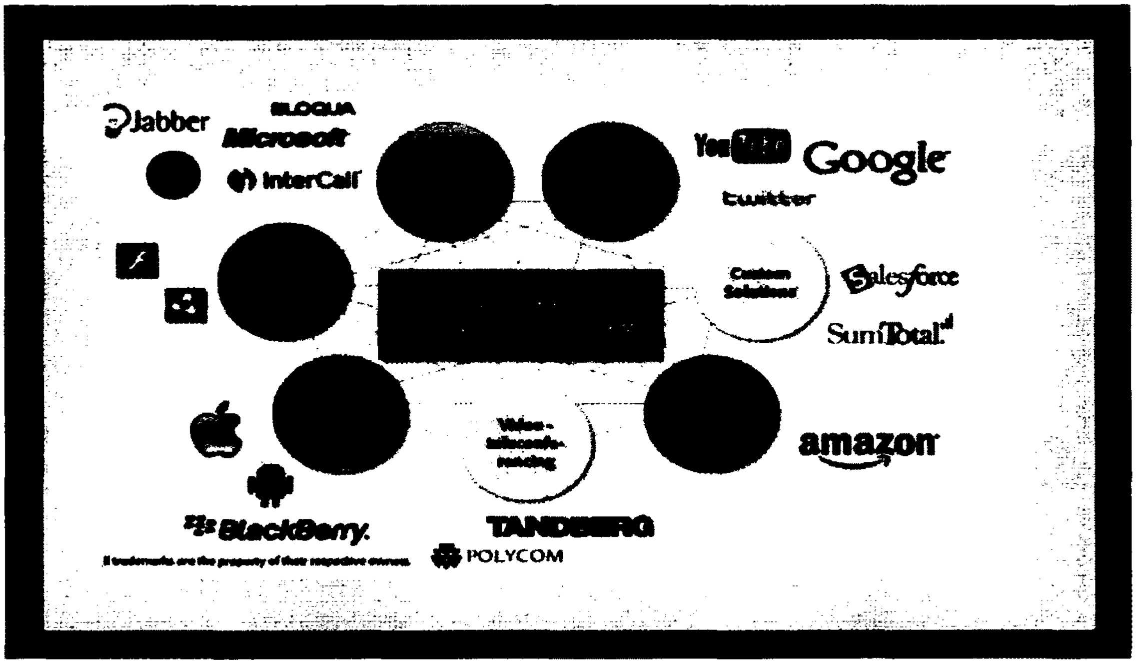

Figure 1: Web conference system vendors ecosystem ***(Adobe.com, 2011)***

2.6 Scalability Models

Scalability models refer to the framework which fashions out the approach to be used when scaling web conference system (Gunther, 2008).

This section reviews three scalability models identified by William and Smith (2004): Amdahl's law, Gustafson's Law, and Linear scalability. However, many more scalability models exist (Smith & William, 2004).

2.6.1 Amdahl's Law Scalability Model

The Amdahl's Law Scalability Model states that the overall speed up that can be achieved by any process is limited by the weakest link in the process (Cavasos, 2007). The maximum speedup obtainable from an infinite number of processors is $1/\sigma$ where σ is the fraction of the work that must be performed sequentially (Amdahl, 1967).

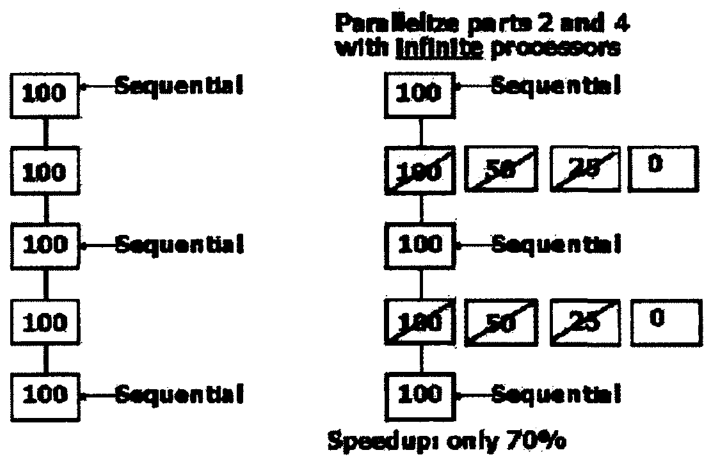

Figure 2: Amdahl's law explanation (Cavasos, 2007)

Thus, if *p* is the number of processors, *ts* is the time spent by a sequential processor on the sequential parts of the program, and *tp* is the time spent by a sequential processor on the parts of the program that can be executed in parallel, Speedup for Amdahl's Law, can then be written as:

$$Speedup = S_A = \frac{T(1)}{T(p)} = \frac{t_s + t_p}{t_s + t_p/p}$$

$$S_A = \frac{1}{\frac{t_s}{t_s + t_p} + \frac{t_p}{t_s + t_p}\left(\frac{1}{p}\right)} = \frac{p}{\sigma + \pi/p}$$

Here, σ is the fraction of the time spent on the sequential parts of the program and π is the fraction of time spent on the parts of the program that can be executed in parallel. Since $\pi = 1 - \sigma$:

$$S_A = \frac{p}{1 + \sigma(p - 1)}$$

Amdahl (1967), argues that given the different limitations of system scalability, a fast single-processor machine is more cost-effective than a multiprocessor machine. For instance, instead of adding a second processor to a system, he argues that replacing the single processor with one twice as fast, enhances the throughput to be exactly twice that with the slower processor. This is because the faster processor reduces the time required for both the serial and parallel portions of the workload. For this reason, it is practically more cost effective to use a faster single processor than to add processors to achieve increased throughput in cases where Amdahl's Law applies. In summary, Amdahl's law assumes work load to be fixed, even when there is hardware or resource available for more parallelism.

2.6.2 Gustafson's Law Scalability Model

Gustafson's law scalability model states that increasing processors gives linear speed up. More processors allow larger dataset size (Cavasos, 2007). Gustafson (1988), proposed that if we have a fixed time and the work load is increased, the serial parts of the system will have diminishing effect in reducing the overall speedup in a parallel environment. He shows that as processors grow the problem size is scaled and this scaling results in a substantial increase in the parallel parts of program as compared to the serial parts.

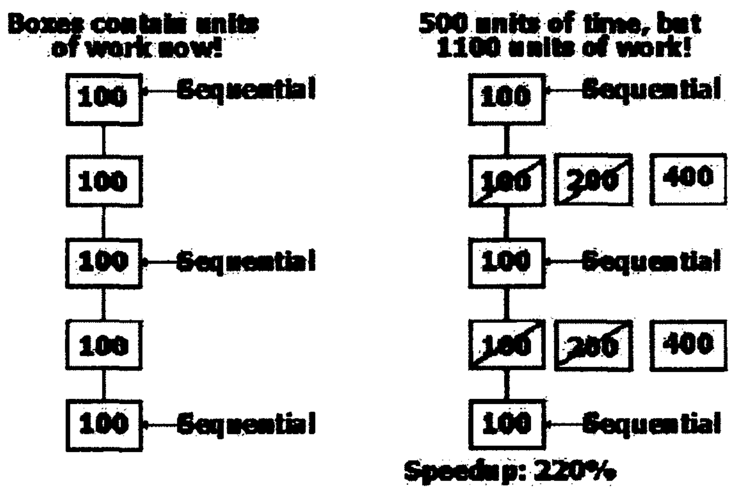

Figure 3: Gustafson's law counter argument to Amdahl's law (Cavasos, 2007)

Applying Gustafson's law to certain applications show that speedups greater than that predicted by Amdahl's Law are possible. A good example is the systems that undertake a speedup of more than 2,000 on a 2,024 processor hypercube (Smith & Williams, 2004). Gustafson in his work,(Gustafson, 1988) stated clearly that Amdahl's Law assumes that the parallel fraction of the application ($\pi = 1 - \sigma$) is constant, i.e., independent of the number of processors. Yet, in many cases, it was discovered that the amount of parallel work increases in response to the presence of additional computational resources, but the amount of serial work remains constant. In view of this, it goes that with more computing power, better web system quality can be achieved with higher number of processors and lower memory usage in the same amount of time (Gustafson, 1988). In these cases, π (and, therefore,) σ is actually a function of the number of processors. Thus, if *ts* and *tp* are the times required to execute the serial and parallel portions of the workload on a parallel system with *p* processors, a sequential processor would require a time of *ts* + (*tp* x *p*) to perform the same work. This is termed *scaled speedup* (Gustafson, 2000), which is described in the Gustafson equations stated below:

$$\textit{Scaled Speedup} = S_G = \frac{t_s + (t_p \times p)}{t_s + t_p}$$

$$S_G = p + \sigma'(1 - p)$$

where σ' is the serial fraction of the work performed on *p* processors. This law describes *fixed-time* speedup while Amdahl's Law describes *fixed-size* speedup (Smith & Williams, 2004). The above equation further describes the speedup as the ratio of the time required to execute the workload on a system with *p* processors to that required to execute the same amount of work on a single processor. However, this is not a ratio that is likely to be measured, as this thesis is more interested in measurements of the maximum throughput at various numbers of processors in a scalable web conference system. Thus, expressing Gustafson's law for web conference system application scalability, it has to be done in terms of $C(p)$, the ratio of the maximum throughput with p processors to the maximum throughput with one processor. The demand with one processor is $ts(1) + tp(1)$ and the maximum throughput is given as:

$$X_{max}(1) = \frac{1}{t_s(1) + t_p(1)}$$

Gustafson's law assumes that the parallel portion of the workload increases as the number of processors increase (Smith & Williams, 2004). Thus, the total demand with p processors is $ts(1) +(tp(1) \times p)$. However, this demand is spread over p processors, so the average demand per processor is:

$$D_b(p) = \frac{t_s(1) + (t_p(1) \times p)}{p}$$

Note that the average demand per processor is a decreasing function of p.

2.6.3 Linear Scalability

Linear scalability refers to a state where the degree of parallelism in the system application is such that additional resources provided by scaling the system can be fully utilized (Smith & Williams, 2004). For example, in a scalable web conference system application, when the system receives requests or instructions from multiple users, it processes it and prepares it for additional processing, depending on the request. This requests is then processed in parallel to increased capacity. However, in order for this application to scale linearly, the processed requests must not interfere with each other to contend for shared resources (Smith & Williams, 2004). If there is any contention for shared resources by the processed requests, it will lead to reduction in the system scalability below linear.

Then, with linear scalability the relative capacity, $C(p)$, is equal to the number of processors, p. i.e., $C(p) = p$.

Therefore, for a system that scales linearly, a graph of $C(p)$ versus p produces a straight line with a slope of one and a y-intercept of zero [$C(0)=0$].

Figure 4 shows a graph of the equation $C(p) = p$.

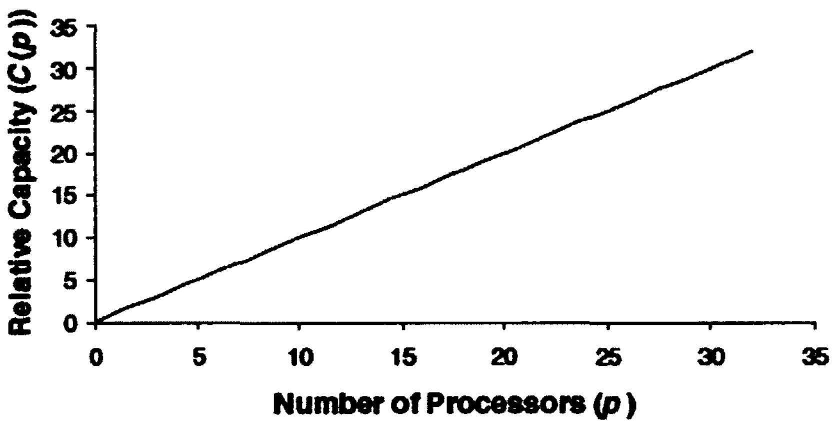

Figure 4: Linear Scalability graph

2.7 BigBlueButton

BigBlueButton is an open source web conferencing system developed primarily for remote eLearning. It supports the sharing of slides (PDF and PPT), video, whiteboard, chat, VoIP (using FreeSWITCH), and desktops in a real-time environment (Dixon 2011). This product architecture was built based on open source components like : Ubuntu, Flex SDK, Grails, Ghostscript, Asterisk and many more which makes it run successfully on Mac, Unix, and PC computers installed either from source code or from Ubuntu packages. BigBlueButton is similar to OpenMeetings and uses red5, an open source implementation of Adobe's Flash Media Server, to support its real-time collaboration. It is downloadable as a Virtual Machine (VM) that runs within VMware Player on PC and Unix computers and within VMware Fusion on Macs and its components are open source. This web conferencing system has witnessed improvement in development and acceptability by the end users especially with the system's scalability over the years and this is one reason it is used as a case study is this research (http://www.bigbluebutton.org/). Figure 5 shows the main interface caption of BigBlueButton.

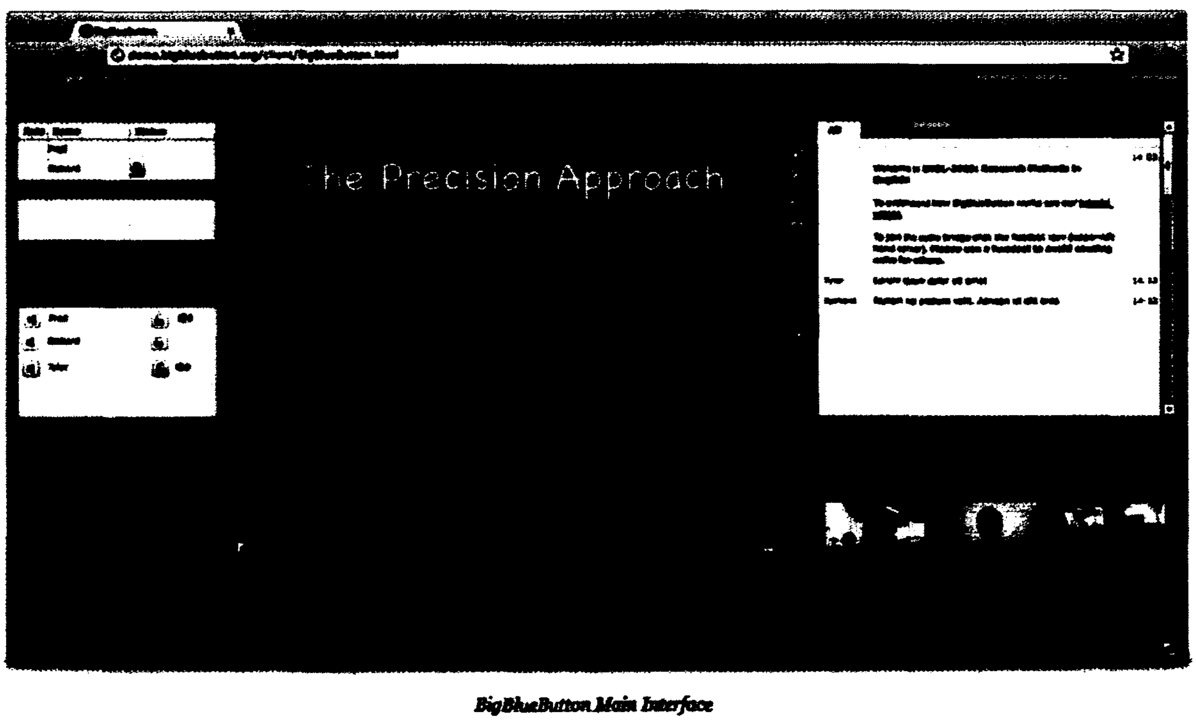

Figure 5: BigBlueButton main interface

2.8 Lessons learned

The lessons learned from literature review are below:

What is Scalability?

There is no one or generally accepted definition for scalability, it refers to frequently-claimed attribute of microprocessor system (Hill,1990) amongst which are: Scalability describes the capability for a system to handle larger loads, providing the same level of service (response times, numbers of requests per second, and so on) through an increase in hardware (Mackenzie, 2002).

Scalability is a property and a measure of an application system's ability to without modification, cost effectively provide increased throughput, reduced response time and/or support more users when hardware resources are added (Smith & Williams, 2004).Scalability is the ability of a real-time communications (RTC) web system to operate efficiently with adequate quality of service, over a given range of configurations. This means being prepared when the load on the system increases, having ways to react without resorting to drastic changes or redevelopment (Schroeder, 2007).

What are the ways to scale a system?

There are two ways to scale a web system application: Strong scaling and Weak scaling (Bennett & McGuigan, 2005). Strong scaling refers to how efficiently a parallel implementation solves a problem of a constant total size compared with a serial implementation running the same problem [Ananthanarayanan, Esser, Simon, and Modha, (2009);Weatherley, Boros, and Abe, (2012)].

Weak scaling refers to a measure of how efficiently a parallel implementation handles increasing total problem size: as the total problem size increases, more CPUs are assigned to maintain a constant problem size per CPU and, ideally, a constant computation time. (Weatherley, Boros, and Abe, (2012)).

Scalability dimensions

The scalability of web conference system scalability can be measured using various dimensions: i)Performance (Lin, 2001; Butler, 2009; & Gunther, 2001),
ii)Physical size (Gustavson,1994), iii)Economics (Winter, 2009; Butler, 2009; & Gustavson, 2004), and iv) Load (Bondi, 2000 & Duboc, 2009). Table 1 provides the definitions of these dimensions.

Scalability and prototyping

Scalability refers to the ability of a system to expand in a chosen dimension without major modification to its architecture and be able to perform well as the offered traffic increases (Bondi, 2000), while prototyping is a sample or model built to test or validate the development requirements of a concept or software system process (Sommerville, 2000). Scalability is an important requirement to be considered when designing a web conference system and having a prototype which describes how the system is developed and conforms to the rules enhances a better performance of web conference system.The principal use of prototype is to help customers and developers understand the requirements for the system and it reduces system requirements risks as it can reveal errors and omissions in the design (Sommerville, 2000). Scalability focuses on the future growth of the system as it accommodates increasing requests from users. (Carvalho & Pereira, 2010, Gunther,2006).

3. METHOD

Chapter 3 describes the method used in this research. It is organized into 5 sections. Section 3.1 provides the objective of this research. Section 3.2 describes the approach used in this research. Section 3.3 presents the case studied. Section 3.4 provides the Unit of analysis and section 3.5 describes the research method used in this thesis.

3.1 Objective

This research carries out an experiment using the logs of various versions of one web conference system to identify the dimensions of the scalability of a system and the metrics that can be used to measure scalability.

The objectives of this research are to answers two research questions and identify insights that may be relevant to real-time system developers.

The two research questions are:

- What are the dimensions of web conference system scalability?
- What metrics should be measured in web conference system scalability?

3.2 Approach

The research method used by Easterbrook, Singer, Storey, Damian, (2008) was used to answer the two research questions. A controlled experiment was used to examine data, where one or more independent variables were manipulated to measure the effect on one or more dependent variables.

The experiment allows the identification of cause-effect relating to one web conference system and describes the processes of analyzing and interpreting data obtained from experiments (Prechelt & Tichy 1998).

3.3 Case studied

The experiment used the logs of the various versions of a web conference system known as BigBlueButton (http://www.bigbluebutton.org/). These versions were released since the inception of the system.

The reason BigBlueButton was chosen as the system to examine was convenience. Carleton University's Technology Innovation Management program spun out BigBlueButton as an open source system in 2009. Strong relationships exist between the developers of BigBlueButton and the graduate students in the graduate program offered by Carleton. Strong local knowledge about BigBlueButton exists with the faculty and students in the graduate program.

3.4 Unit of analysis

The unit of analysis is a version of the BigBlueButton system. Nine different versions of BigBlueButton were examined in a controlled experiment. The primary data source are the logs of BigBlueButton versions 0.4 released mid 2009 to 0.72 (0.71a) released in 2011.

Each new version release improves on the previous, reflecting the desire by the core development team to build a solid product and better meet the needs of the growing end users (Dixon, 2011).

3.5 Research method

Table 3 identify the steps used in the method carried out to complete this research.

S/No	Steps	Dominant activity	Output
1	Identify key factors that affect web conferencing system scalability.	Extant literatures were reviewed to identify the factors that affect scalability of a web conference system (Table 2, Section 2.5)	Factors were identified, based on findings from reviewed literature.
2	Identify various BigBlueButton versions that have been developed	BigBlueButton release notes were obtained through the BigBlueButton website, and BigBlueButton managers were contacted to validate the information released: (http://code.google.com/ p/bigbluebutton/wiki/ReleaseNotes)	Various versions of BigBlueButton developed from 2009 to 2011
3	Collect logs(data) of BigBlueButton versions developed from 2009 to 2011.	BigBlueButton configuration files and logs of performance, system uptime and downtime (errors) for versions developed from 2009 to 2011 were obtained from the configuration and log files backup directories of the system.	Collected logs were tested and analyzed.
4	Test data for relationship	Test environment server was set up with the following configuration: Core 2	Test results obtained and the relationships

	between the different variables and how it affect a web conference system scalability.	Dual Xeon 3.2GHz processor and 8GB RAM running on Ubuntu 32 bit Operating System with BBB version 0.72 . The server runs on Sun Java 6 Application Server Platform for the web conference system framework. Jconsole test tool was installed to measure variables like memory usage, percentage system uptime, number of real time users, and amount of resources available to the server. Different tests were carried out on the logs as shown in section 3.5.4	between different variables measured were established and shown graphically.
5	Analyze test results obtained	Test results were analyzed using Excel 2007 regression analysis model tool. The results were validated by comparing the outcome for different variables measured and the relationship between the tested variables.	Scalability dimensions and metrics to be measured were identified.
6	Provide insights relevant to technical managers and software developers	Showcase every steps taken, observations made, list challenges or limitations faced and document relevant findings.	Publish and make available the research findings

Table 3: Research method steps

Detail description of the steps in Table 3 are stated in sections 3.5.1 to 3.5.6

3.5.1 Identify key factors that affect web conferencing system scalability

Extant literatures were reviewed to identify the factors that affect scalability of a web conference system (Table 2, Section 2.5). The different literatures reviewed identifies diverse factors that influence the outcome of a system after it has been scaled. The identified factors play a vital role in determining what and how decisions are made by design architects and developers when a system or software is been developed or needs to be scaled.

3.5.2 Identify the different versions of BigBlueButton developed overtime

BigBlueButton release notes were obtained through the BigBlueButton website, and a follow up with BigBlueButton managers was done to get more useful information relating to the various versions released within the years under consideration. A comprehensive release note stating the release name and year, issues fixed with the new release and other relevant information was obtained through the website: (http://code.google.com/p/bigbluebutton/wiki/ReleaseNotes). A validation of the release note was done with two of the developers of the web conference system to validate and obtain more insight regarding the development of the software and how the new release emerges.

3.5.3 Collect logs (data) of different versions of BigBlueButton

Access permission to the archived backup log files directory which contains the configuration files and performance (error) logs were obtained from the managers of BigBlueButton. With this access, configuration files, performance logs, system uptime and downtime (errors) logs for versions developed from 2009 to 2011 were obtained by issuing the command: */var/log/nginx/bigbluebutton.access.log* for all access information to the Web and */var/log/nginx/error.log* for Web log of errors generated by nginx. More logs were obtained with other command lines like:

/usr/share/red5/log/sip.log to obtain audio/voice logs from the sip red5 application, which is the red5 server component for supporting voice over IP.

/usr/share/red5/log/video.log provided access to video logs from video red5 application, which is the red5 server component for supporting video

3.5.4 Test data obtained for relationship between different variables

Test environment server was set up. The server has the configuration: Core 2 Dual Xeon 3.2GHz processor and 8GB RAM running on Ubuntu 32 bit Operating System with BigBlueButton version 0.72 . The server runs on Sun Java 6 Application Server Platform for the web conference system framework. Jconsole test tool was installed to measure variables like memory usage, percentage system uptime, number of real time users, and amount of resources available to the server.

In carrying out five(5) empirical tests on the data, different controlled conditions were implemented and tested in a test environment set up using a server machine on the BigBlueButton test environment and a client accessing it with the configuration as stated in the first paragraph of this section. The test environment has both client and server running on the same host, although the client was implemented to be remotely accessible to the server through Putty. The empirical tests carried out are as follows and the outcome shown in sections 4.1.1 to 4.1.5.

i) CPU usage against fixed number of users- The number of fixed concurrent users was first set to 100 users for each version of BigBlueButton, using the configuration files to change settings and set policies on the system through the command line: */etc/nginx/sites-enabled/bigbluebutton* and the actual CPU use was measured through Jconsole tool and the result is shown in Table 4,Section 4.1.1.

ii) CPU usage against varied number of users- Test 2 measures CPU usage with varied concurrent users on the web conference system. The command line : */etc/nginx/sites-enabled/bigbluebutton was used and the CPU usage measured*. The test process was observed and the outcome result is shown in Table 5, Section 4.1.2.

iii) System throughput against varied number of users- Test 3 measures the system throughput as against the users that logged on to the system with different requests. Test 3 measures the average rate of successful message delivery based on different requests from the users. The output result is shown in Table 10 & 11 in Section 4.1.3 as obtained using Jconsole graphical tool.

iv) System throughput against CPU utilization- Test 4 examines the average rate of successful request delivery in the system against the utilization of the CPU when different number of users log on to the Web conference system. The result obtained through the use of Jconsole tool is shown in Section 4.1.4.

v) Audio test- Test 5 measures the audio quality of BigBlueButton under different conditions, with particular attention on versions 0.7 and 0.71. Two servers were set up to measure the audio quality. Server 1 (Figure 17, 18, & 19, Section 4.1.5) with I.P address 173.195.48.101 runs 0.70 version of BigBlueButton, while Server 2 (Figure 20, 21, & 22, Section 4.1.5) with I.P address 173.195.48.113, runs 0.71 version of BigBlueButton. The audio quality test was done over two days with more than 20 users accessing the system. Both audio and video features were used by the users and the output result as obtained through Jconsole tool are shown in Section 4.1.5.

3.5.5 Use regression analysis model to analyze results obtained from tests

Regression analysis model was used to analyze the test results and estimate the quantitative effect of the independent variables upon the dependent variables.

Excel 2007's built-in regression tool was used to analyze whether the outcome of the tests obtained in section 4 can predict the scalability of the BigBlueButton and what dimensions are noticed.

The result from each test is arranged into a tabular form as seen in Tables 4 - 11, Section 4.1.1 - 4.1.5.

The steps followed while using Excel 2007's built-in regression tool to analyze the result obtained from the tests are as follows:

i) Arrange the data - The independent variables were located together in the worksheet. No blank columns or columns with non-relevant data interrupting the range independent variables was allowed. The dependent variables were arranged in a single range as required by Excel's built-in regression tool. ii) Open the regression analysis tool in Excel

2007- the regression analysis tool was opened and " data analysis" was selected. Under the data analysis dialog box, "regression" option was selected and opens up the sole interface to Excel's regression tool, where all interaction with Excel's regression tool takes place. iii) Complete the "Regression" Dialog Box- The required information on the regression dialog box were filled in, specifying what each column stand for, either independent variable of dependent variable. It was also specified that Excel should to put results on a new worksheet by choosing the option "New Worksheet Ply". Residuals, and Standardized Residuals options were selected for the outputs. iv) Run the regression analysis and view the results. After completing steps i) to iii), the OK button was clicked and the regression was executed. The regression outputs were then located by Excel 2007 on a new seperate worksheet that it created. The steps were repeated for each tests and the outcomes are shown in section 4.

3.5.6 Insights development

Insights development focuses on how to showcase every steps taken in this research, what observations were made, what are the challenges faced and what limitations hindered what would have been a better research work. The insights are documented and made available to software managers and developer as stated in section 4.4.

4. RESULTS

Chapter 4 provides the results of the research. Chapter 4 is organized into 3 sections. Section 4.1 provides the result of the 5 tests. Section 4.2 provides the answer to research questions 1. Section 4.3 provides the answer to research question 2. Section 4.4 shows the insights gained in this research for software managers and developers.

4.1 Results of 5 tests carried out

4.1.1 Results of Test 1. CPU usage against fixed number of users

In Test 1, the relationship between concurrent users and CPU usage was measured. The purpose is to determine how the system performs with fixed number of users and examine what relationship exists between the traffic on the system and the CPU usage. The number of concurrent users was first set to 100 users and the actual CPU use was measured. Table 4 shows that the CPU usage varied for different versions of BigBlueButton when the number of users was fixed at 100.

	BigBlueButton Version								
	0.4	0.5	0.61	0.62	0.63	0.64	0.7	0.71	0.72
No of Users (X)	100	100	100	100	100	100	100	100	100
CPU Usage (Y)	64	70	72	75	75	66	60	57	55

Table 4: CPU usage with 100 users by BigBlueButton version

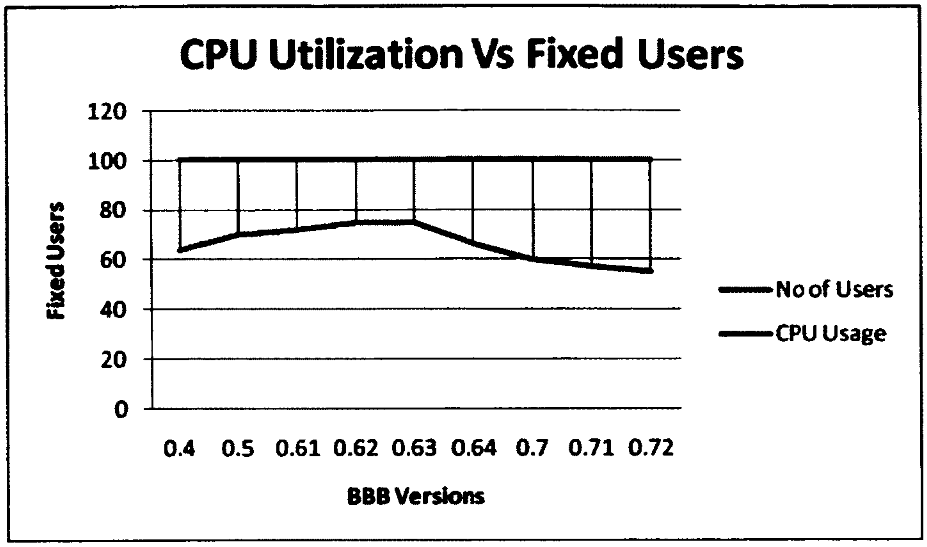

Figure 6. CPU Utilization Vs. Fixed Users by BigBlueButton versions

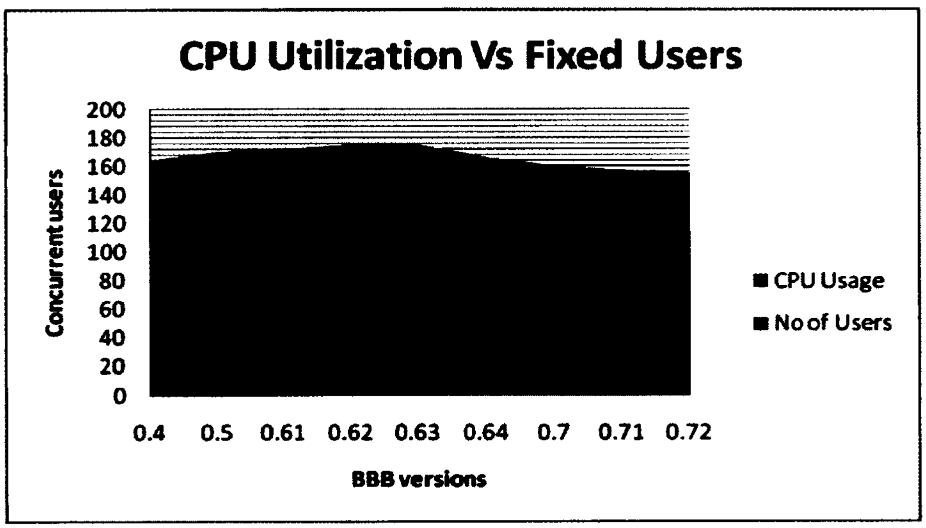

Figure 7. Area chart of CPU Utilization Vs. Fixed Users by BigBlueButton versions

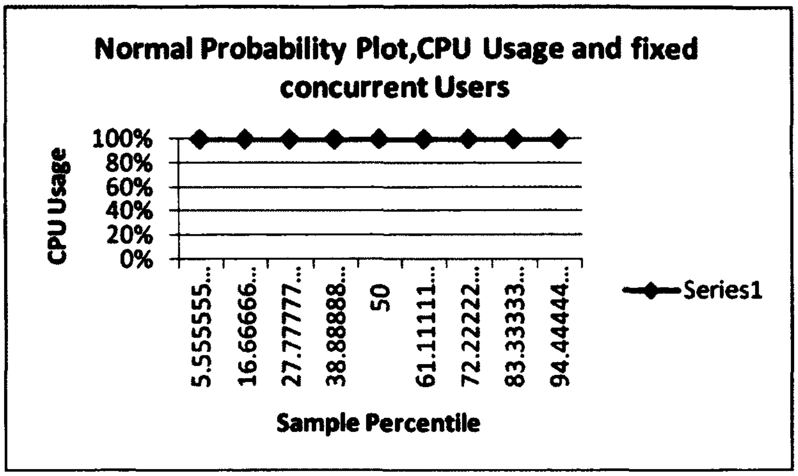

Figure 8. Regression analysis of CPU Usage Vs. Fixed Users by BigBlueButton versions

Figure 9 displays the residual (y) and independent variable (X) as a residual plot, showing inputs and outputs from linear regression.

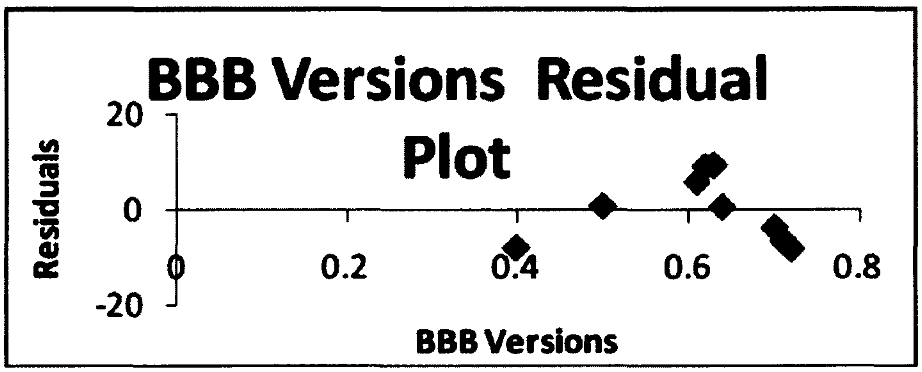

Figure 9. BigBlueButton residual plot showing residual (y) and independent variable (x)

4.1.2 Results of Test 2. CPU usage against number of users

For each version of BigBlueButton, Test 2 measures CPU usage as a function of concurrent users. Figure 10 shows the residual plot obtained, while Figure 11 shows the regression probability plot for the test. Table 6 and Table 7 are the predicted residual output of system CPU usage and probability output of varied user vs.CPU usage respectively.

	BigBlueButton Version								
	0.4	0.5	0.61	0.62	0.63	0.64	0.7	0.71	0.72
No of Users (X)	5	8	32	18	30	28	45	68	64
CPU Usage (Y)	35	46	50	61	68	68	75	86	89

Table 5. CPU usage with varied users by BigBlueButton versions

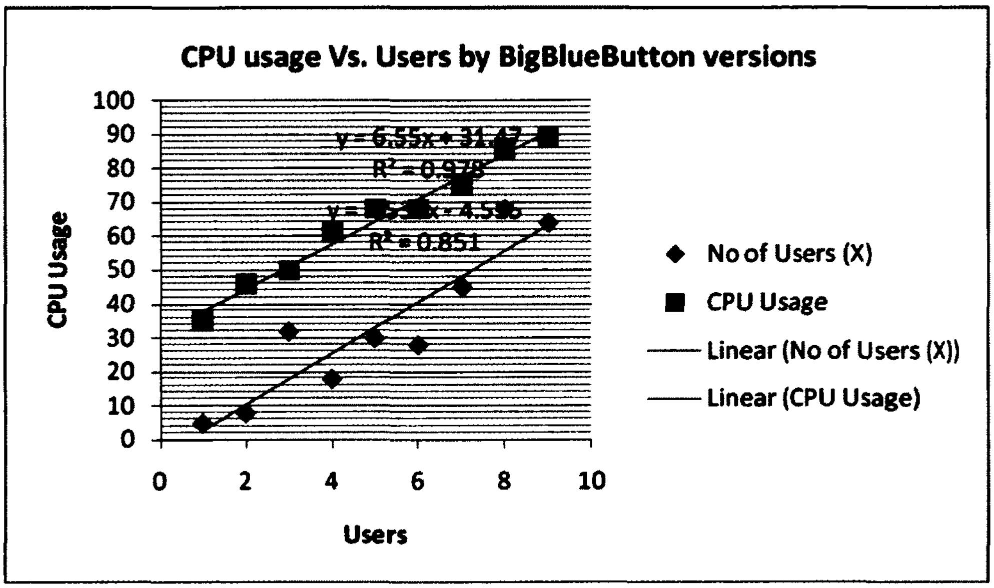

Figure 10. CPU usage Vs. Varied Users by BigBlueButton versions

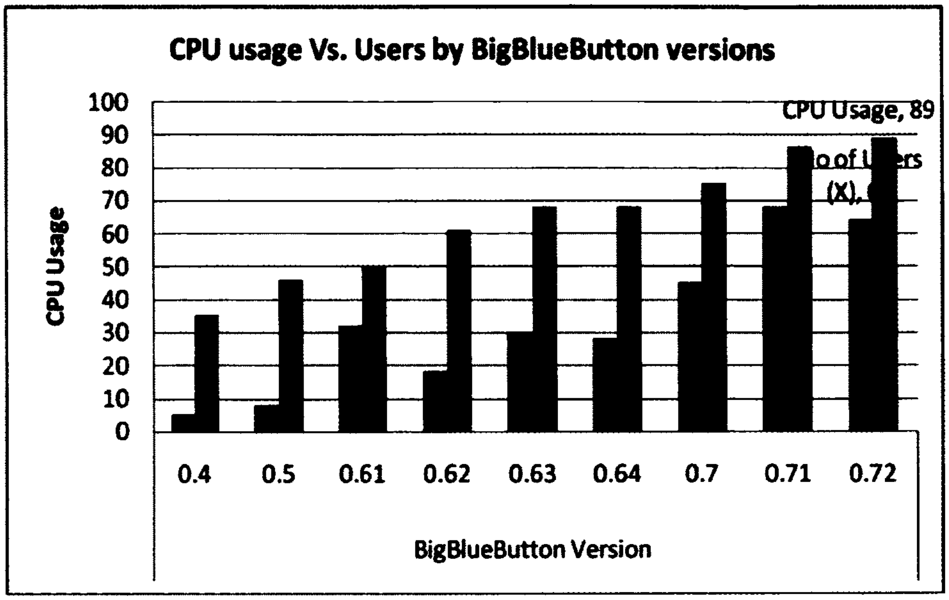

Figure 11. Chart showing CPU usage Vs. Varied Users by BigBlueButton versions

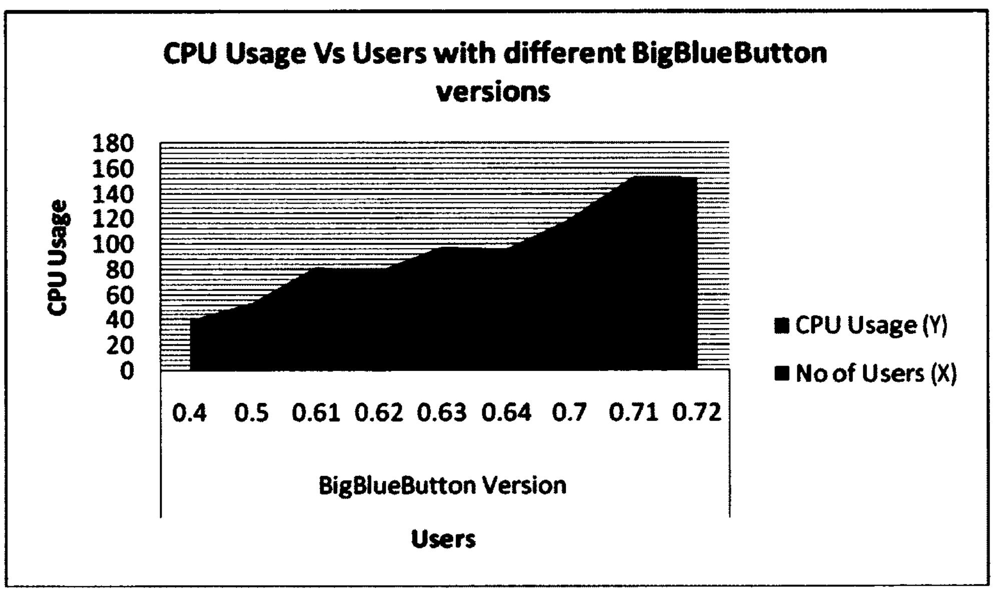

Figure 12. CPU Usage Vs. Varied User at different times

Regression Statistics	
Multiple R	0.895638034
R Square	0.802167488
Adjusted R Square	-1.333333333
Standard Error	7.422414235
Observations	1

Table 6. Regression analysis summary output for CPU Usage Vs. Varied Users

	df	*SS*	*MS*	*F*	*Significance F*
Regression	8	1340.3216	167.5402	24.3287	0.04
Residual	6	330.5534	55.09223		
Total	14	1670.875			

Table 7. Anova analysis for CPU Usage Vs. Varied Users

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept							3.30E-284	3.27E-284
5							-5.00E-280	4.93E-280
8							1.70E+171	1.73E+171
32							-5.00E-280	4.76E-280
18							1.70E+171	1.73E+171
30							-2.00E-274	1.88E-274
28							-2.00E-274	1.88E-274
45	43.830781	5.536208	7.9171	0.0002	30.28	57.3774	30.28417	57.377396
68	0.6564974	0.133098	4.9324	0.0026	0.331	0.98218	0.330817	0.9821782

Table 8: Table of Coefficients for CPU Usage Vs. Varied Users

Observation	Predicted 35	Residuals	Standard Residuals
1	3022.50894	-2976.50894	-3.74166

Table 9. Residual Output for CPU Usage Vs. Varied Users

4.1.3 Results of Test 3. System throughput against varied number of users

Test 3 was carried out to measure the system throughput with varied users, as shown in Table 10. Figure 13 shows the graphical representation of the result, while Figure 14 shows the residual plot for this test. Table 11 is the residual output from regression analysis of the test data and throughput is measured in terms of transactions per second (tps).

1	10	100
50	500	100
100	1200	83.33
150	2200	68.18
200	4000	50

Table 10: Concurrent Users Vs. Throughput at different times

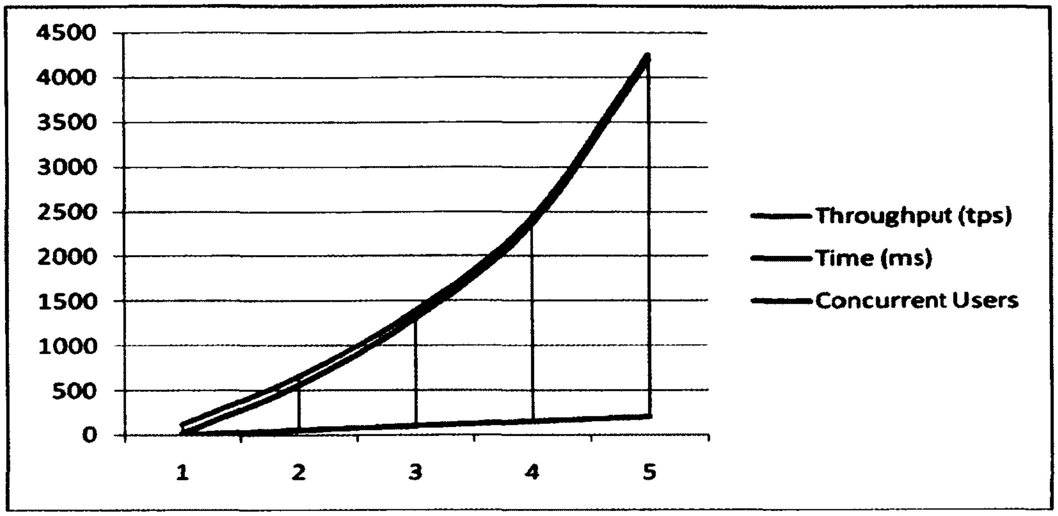

Figure 13. Throughput with Varied Users at different times

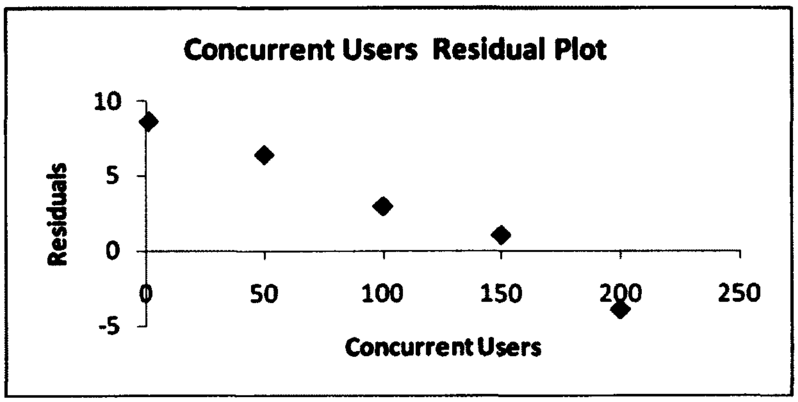

Figure 14: Residual plot of Varied Users Vs. Throughput at different times

Observation	*Predicted Throughput (tps)*	*Residuals*	*Standard Residuals*
1	106.58696	8.586964	-1.2606
2	93.603958	6.396042	1.224061
3	80.355992	2.977008	0.569734
4	67.108026	1.073974	0.205535
5	53.86006	-3.86006	-0.73873

Table 11: Regression analysis residual output

4.1.4 Results of Test 4. System throughput against CPU utilization

Test 4 measures the total amount of items processed by the system over the defined period of time against the utilization of the system CPU. Figure 15 and Figure 16 respectively show the result obtained.

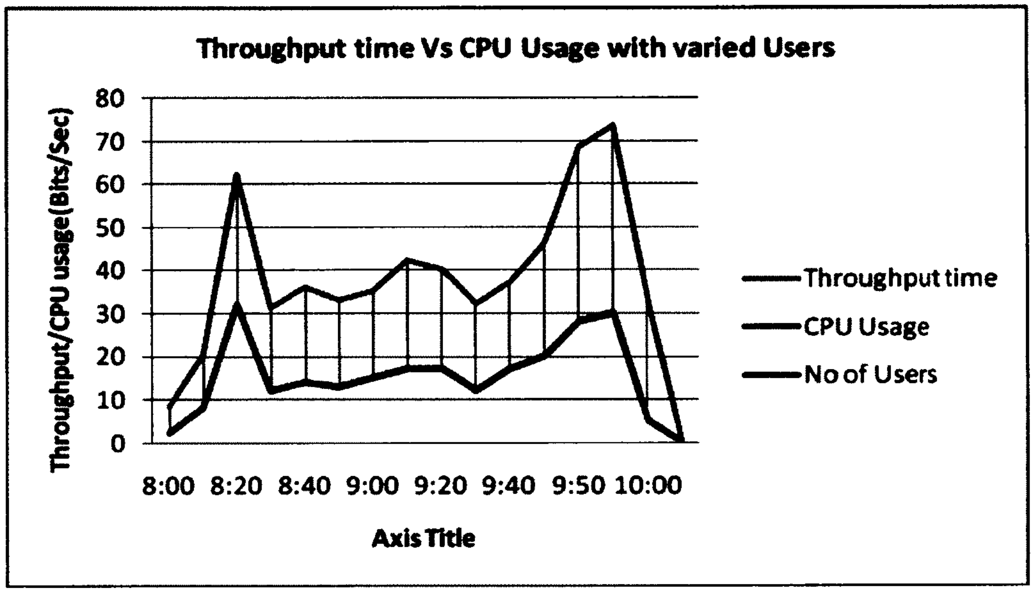

Figure 15: Throughput Vs.CPU utilization with varied users at different times

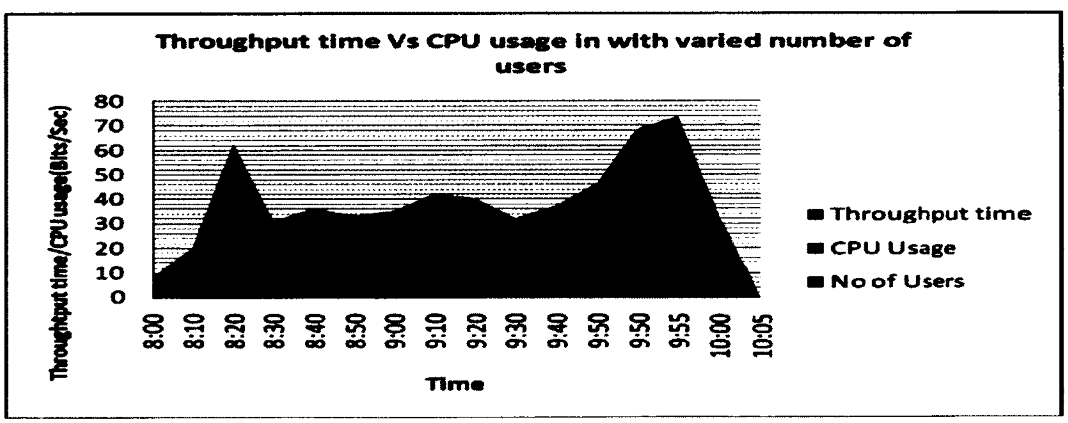

Figure 16: Area chart of Throughput Vs.CPU utilization with varied users at different times

4.1.5 Results of Test 5. Audio quality

Test 5 examines the audio quality of BigBlueButton versions 0.7 and 0.71. The output result shown (Figures 17, 18, 19, 20, 21 & 22), reveal tremendous difference in audio performances in relation to the BigBlueButton versions tested and the number of users that accessed the system on both servers 1 and 2 (Section 3.5.4).

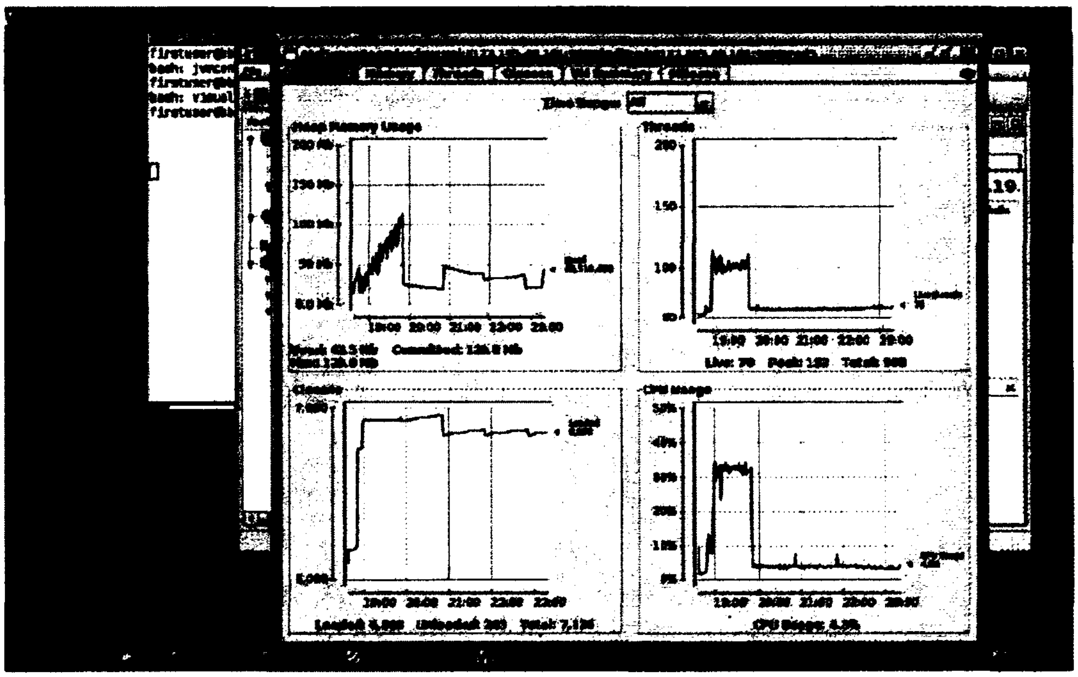

Figure 17: System performance in BigBlueButton version 0.7

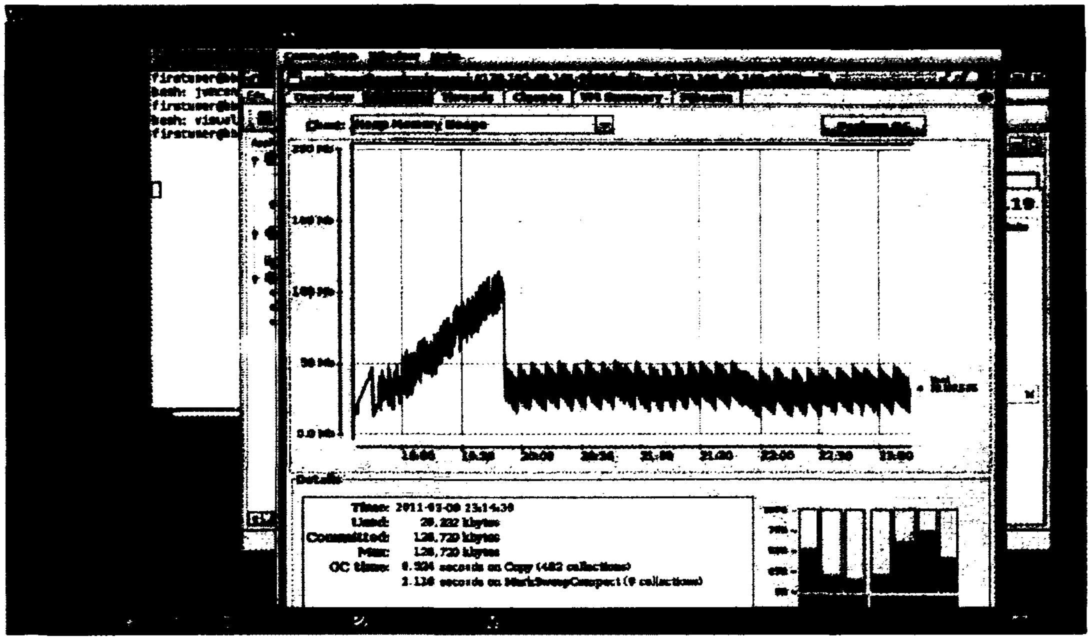

Figure 18: Unstable memory usage in BigBlueButton version 0.7

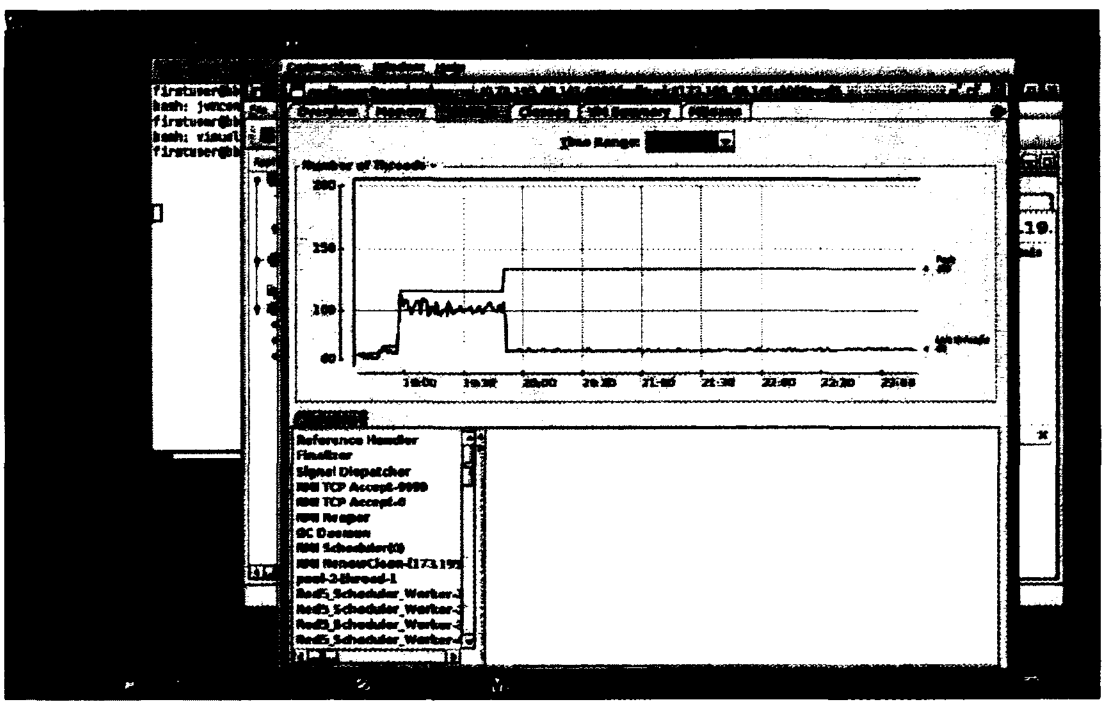

Figure 19: Unstable system thread in BigBlueButton version 0.7

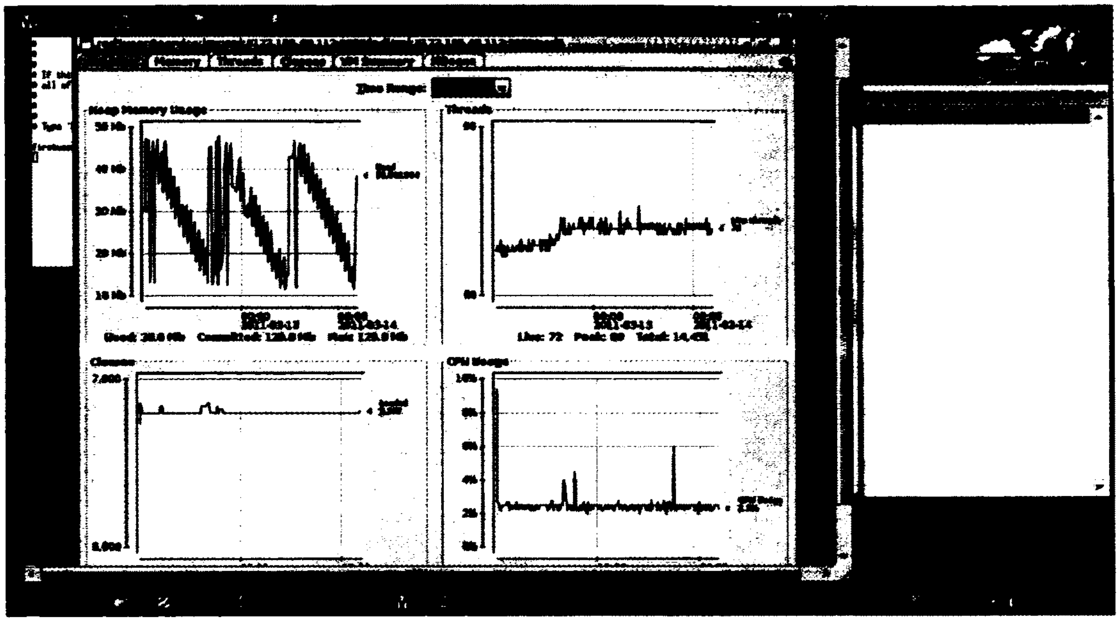

Figure 20: System performance in BigBlueButton version 0.71

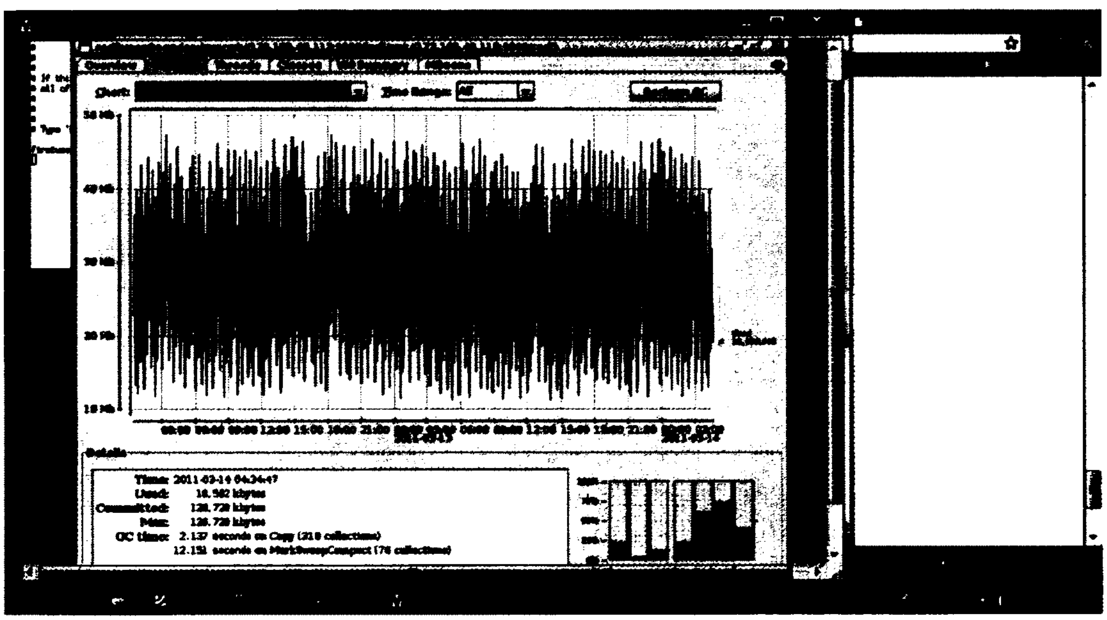

Figure 21: Stable memory usage in BigBlueButton version 0.71

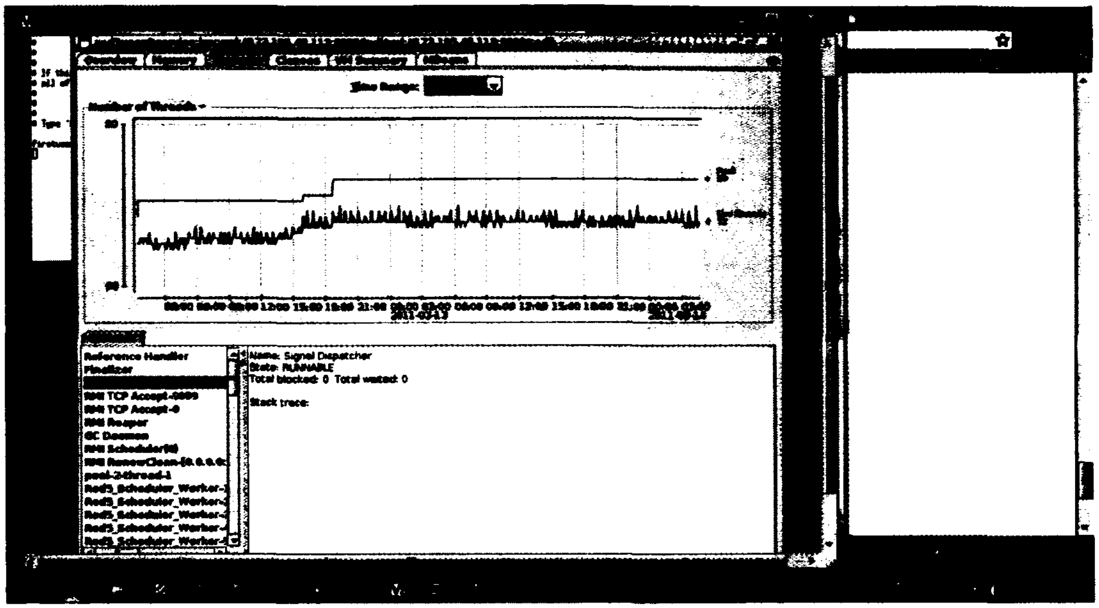

Figure 22: Stable system thread in BigBlueButton version 0.71

4.2 Answer to research question 1

The first research question was: What are the dimensions of web conference system scalability?

The outcome of the tests carried out on this research shows that for a web conference system to be scaled successfully, system scalability dimension is a very importance attribute that must not be ignored. Many scalable systems experience system failure due to the inability of the developers to put in place the necessary dimensions which helps to guide the process of system scalability. A major challenge of web conference system applications is designing for scalability.

Hence, with more computing power, better web system quality can be achieved with higher number of processors and lower memory usage in the same amount of time. In view of this, the following scalability dimensions should be taken in order to have a successful web conference system scalability.

4.2.1 Data size

Theoretically, the relationship between the number of users and CPU usage is given by a linear line (Figures 6 & 7,10 &11, Section 4.1.1- 4.1.2). The output result of the tests carried out shows that for a constant web conference system configuration like BigBlueButton, increase in data size causes the speedup obtained through parallelization to increase. That is, if the data size increases by a factor of *x*, then in a scalable system, the query response time based on the number of users will increase by at most a factor of *x*. The noticeable increase in processed data size (Figure 13 & 14, Section 4.1.3), is due to the parallel work increases with data size and by implication, increase in the number of processors, which gives linear speed up as more processors will always allow larger dataset size. This also implies that if the data volume is increased, increasing processors will give linear speed up as noticed in Test 1 & 2, as more processors allow larger dataset size. Thus, data size is a scalability dimension that should be considered when scaling a web conference system.

4.2.2 Speed

Speed is another scalability dimension that needs to be given consideration when scaling a system. In Test 4 (Figure 15 &16, Section 4.1.4), the result obtained indicates that when the capacity of the hardware configuration of a web conference system increased by a factor of *x*, the system query response time will decrease by no less than a factor of *x*. From the test carried out (Test 4, Figure 15 &16, Section 4.1.4), observation reveals that as BigBlueButton scales from one version to the other, the throughput obtained from the system indicates better speed, time and performance. In view of this, it goes that with more computing power, better web system quality can be achieved with higher number of processors and lower memory usage in the same amount of time. This confirms that speed is a vital dimension to be considered when scaling a web conference system because, the overall scalable system performance measure rating cannot be determined without considering the speed or processing time of request from users to the system.

4.2.3 Cost

Cost is a very crucial and strategic dimension of web conference system scalability. It is therefore very expedient for technical managers of software development organisations, architects, software developers and users to know that scalability is not only a technical issue, but also an economic as well. Cost played a vital role in the process of BigBlueButton scalability and growth, in that it is a strategic measure that impacts meeting the system performance requirements. From Test 1 to Test 5 carried out in this research (Sections 4.1- 4.5), obtained results indicates that the importance of cost in meeting performance requirements is implicit in the inclusion of cost figures when designing a scalable system. Adding more resources could mean adding costs for additional system resources. Therefore, all costs associated with the system scaling strategy is essential while designing a scalable web conference system.

4.3 Answer to research question 2

The second research question was: What metrics should be measured in web conference system scalability?

Process management requires effective measurement and quantification. Hence, web conference system scalability metrics provide a measurable and quantitative basis for the development and validation of scalability dimensions of the web conference system development process. This is then used to improve software productivity and quality. From test carried out in this thesis (Test1 - 5, Sections 4.1.1 - 4.1.5), it is observed that the new generation of high performance software systems puts more emphasis on efficiency and high productivity (Figure 6 & 7, rather than high performance as against what obtained in the past. These new generation systems do not only meet the traditional requirements of web conferencing system scalability performance, but also address the ongoing technical challenges in the software system development domain (Figure 17 & 20, Section 4.1.5).

Therefore, a successful web conference system scalability process should measure the listed metrics, most especially as they relate with the software design.

4.3.1 Lines of Code (LOC)

Line of code indicates the number of nonblank and non-comment source code lines in the web conference system software. The result of the tests carried out (Test 1-5, Section 4.1.1- 4.1.5) indicates that using line of code metric as a basis for comparison between BigBlueButton versions requires the software to be written in the same programming language.

4.3.2 Number of Modules (NOM)

Number of modules is another useful metric, which is somewhat less influenced by programming languages and line-level coding styles. It is used to compare scalability of softwares written in different languages. Number of modules is very vital as BigBlueButton is an open source software, and the software development is done by the member of the software community.

4.3.3 Defect metrics

Defect metrics is a parameter which measures the number of defects in the software design, number of code changes made and number of code changes that will be required. Additionally, when considering web conference system scalability characteristics such as maintainability, flexibility, source code quality and so on, architectural design metrics must also be given adequate attention.

4.3.4 Number of Functions (NOF)

Number of functions is used to measure module size in terms of the operations it implements. Figure 20, 21 & 22 of section 4.1.5 reveals the system performance graphs, the stable memory and thread. Number of functions is also used to identify the reuse potential of a module because generally, modules with a large number of functions are more difficult to reuse as they tend to be less cohesive. Hence, it is recommended that a module should not have an excessive number of functions.

4.4 Insights gained for software managers, architects and developers

The insights gained in this research work are key to the success of software managers, developers and architecture. The desire of every manager or developer is to have a solution that is able to grow with the organization, more secure, cost-effective , reliable, and scalable. Insights from this thesis show that most performance failures are due to a lack of consideration of performance, usability and scalability issues early in the development process, and in the architectural phase. However, performance failures results in, productivity lost for users, lost revenue, cost overruns due to tuning or redesign, and in some cases, the entire product may have to be redesigned.

In view of this, extensible architecture that supports cost-effective integration with existing and evolving infrastructure investments should be considered by software managers and developers. Furthermore, best architecture of software modules that can run concurrently on a parallel processor should be determined from the very beginning of the software design.

This research work also reveals that best software architecture enhances better software productivity and quality as system features based on software functionality, rather than other characteristics, are becoming the most important factor in competing on the market. System attributes like scalability, usability, robustness, implementation and integration become the most important features of software. Software today, is increasingly becoming large and complex, thus, to adapt quickly to these changes, reusability and system scalability becomes expedient, especially in a web conference system.

5 DISCUSSION OF RESULTS

This chapter discusses the results of the experiments and compared emergent findings with extant literature to extend or replicate existing theory and identify anomalies. The chapter is organized into the following sections; section 5.1 outlines the relationship between the different variables tested as seen in the experiments, section 5.2 further discusses the test results and other findings in comparison to the extant literatures reviewed in chapter 2.

5.1 Discussion of the results of five tests

The results of the five tests carried out are discussed extensively in this section. The theoretical relationship between the number of users and CPU usage is given by the line in Figure 6 and Figure 7. The result shows that given a perfect situation, where number of concurrent users and CPU usage could be measured accurately, the observed relationship between these two variables would lie exactly on the straight line. However, in reality it is not possible to have a fixed number of users constantly log on to the web conference system and measure the exact CPU usage as users, in most cases sign in and out of the web conference at will for reasons best known to them, and also considering other factors that influence the usage. In Figure 8, the regression analysis probability plot of CPU usage against the users is shown. The probability plot shows that even with fixed users, the CPU usage varies considerably, and because we have a fixed number of users (normal distribution),the outcome falls on an straight line.

The result explains the features that are accessible to users during a web conference determines the usage of the CPU because a linear (or sub-linear) increase in physical resource usage always lead to capacity increase of the system (Duboc, 2007). The residual plot shown in Figure 9 is a fairly random patterned plot, the first residual is negative, the next is zero, the third, fourth and fifth are positive, the sixth is negative and the last three residuals are negative. This random patterned plot suggests that there is a straight-line relationship between the two variables, and a linear model will provide a decent fit to the data, as an increase or decrease in the value of one variable causes the value of the other variable to either increase or decrease also.

Figure 10 (Test 2, section 4.1.2), shows the scattered plot obtained when CPU usage was measured against varied users. Figure 11 show the activities of the web conference system when the CPU usage against varied users were measured under different versions of BigBlueButton. The scatterplot in Figure 10 suggests a functional relationship between the CPU usage and the users. Each point on the plot represents both the number of users measured and the CPU usage respectively. Not surprisingly, the scatter plot shows that CPU usage and the number of concurrent users are closely related. Also, there seems to be a linear growth pattern, within the given BigBlueButton version range (Figures 11 & 12, Section 4.1.2).

From Table 6, R square is the squared multiple correlation coefficient between the CPU usage and the concurrent users when measured under different BigBlueButton versions. R square is the Coefficient of Determination of the relationship between the tested variables. For a perfect scalability, $R^2=1$ while $R^2=0$ for a non scalable system. However in an ideal situation as seen in this test, R^2 is never observed to be exactly 0 the same way the sample correlation coefficient is never exactly 0. R^2 obtained from Test 2 regression analysis (Table 6) is 0.8021675, which is a good result, and this shows that BigBlueButton scales well and effectively.

Multiple R is the multiple correlation coefficient and square root of R^2, it is the correlation between the predicted and observed values. In linear regression, R equals the magnitude correlation coefficient between X and Y (Table 5, Section 4.1.2). Therefore, the correlation between concurrent users and CPU usage is 0.8021675. The Standard Error of the Estimate is the Root Mean Square Error and the square root of the Residual Mean Square. It is the standard deviation of the data about the regression line, rather than about the sample mean.

The Analysis of Variance (ANOVA) table (Table 7, Section 4.1.2) describes how the regression equation accounts for variability in the response variable of Test 2.

The column labeled Sum of Squares (SS) describes the variability in the response variable, Y. Each sum of squares (SS) has a corresponding degrees of freedom (DF) associated with it. Total df is *n-1*, one less than the number of observations. The Regression df is the number of independent variables in the considered in this test, which is 1 (Number of users). The last column of Table 7, Section 4.1.2 has the significance F which shows how importance of the relationship between the independent and dependent variables tested. From this result, Significance F obtained is 0.04. This result shows there is a good relationship between both independent and dependent variables, as any value greater that 0.05 would have suggested that the system did not scale well, and the independent variable has no relationship with the dependent variable as it did not give a prediction of the dependent variable.

The standardized coefficients (Table 8, Section 4.1.2) are the regression coefficients of variables A & B.

A- Users, while B- CPU usage. The regression equation is then given thus:

CPU usage = 43.830781+0.6564974(Users)

The predicted CPU usage for 5 users is 43.830781+0.6564974(5)

=> 44.4872784(5)

=> 222.436

However, for longitudinal data like that in Test 2, the regression coefficient is the change in B (CPU usage) per unit change in A (Number of users). Here, CPU usage differs by 0.6564974 units for every version of BigBlueButton.

The Standard Errors are the standard errors of the regression coefficients. For example, the standard error of the CPU usage coefficient is 0.1330987. A 95% confidence interval for the regression coefficient for CPU usage is given as (0.6564974 $\pm$ *k* 0.219), where *k* is the appropriate percentile of the t distribution.

The residuals from Table 9 in Test 3 (Section 4.1.3), are the differences between the regression's predicted throughput value and the actual throughput value. The final value is the standardized residual (the residuals adjusted to ensure that they have a standard deviation of 1; they have a mean of zero already). The plot of the residuals in Figure 16 has a straight-line appearance, which is a linear pattern. This residual plot implies that a linear model is the appropriate model for predicting the dependent y (throughput) values. There is also a strong linear association between throughput and users under different BigBlueButton versions. It shows with known number of users, you can get a fairly good estimate of the throughput and vice versa. Furthermore, the result indicates that the throughput drops for every newer BigBlueButton version. One main reason for this could be the addition of resources like processors and memory increase as the system is scaled.

The result obtained in Test 4 (Figure 15 &16, Section 4.1.4), indicates that the throughput time increases with higher CPU utilization, though the increase pattern varies from one user to the other. The outcome result also provides indication of overall growth pattern, which is not quite surprising, as it shows the throughput is closely related to the CPU utilization, because, the more features users are connected to, the more of CPU will be utilized.

One good observation in Test 5 (Figure 17,18,19,20,21 & 22, Section 4.1.5), is that the number of concurrent users and the installed java applet software have such a great influence on the audio performance of the web conference system. The test confirms that as the number of concurrent users increased from one BigBlueButton version to the other, there is laudable increase in the audio quality and the performance in general. The result of this test was analyzed to determine if the difference in the relationship result between the variants (Control and Treatment) is statistically significant and what the principal cause of variation in measured response is. Observed result shows the audio quality on server 1 is not good compared to server 2. Audio lags were noticed on server 1, but not on server 2. Audio lags are not found on server 2 because BigBlueButton version 0.71 has voice java applet installed on it to take care of the audio lag effect in version 0.7.

Figure 21 shows the memory usage was more stable and no leak identified on 0.71 compared to 0.7. The result confirms better voice quality in BigBlueButton version 0.71.

5.2 Relating results to extant literature

The result of this study has several assertions regarding the behaviour of a Web conference system scalability based on reviewed literatures in sections 2.1 to 2.7. The result obtained in test 1(Table 4, Figure 6 & 7,Section 4.1.1), which examines the CPU usage with fixed number of concurrent users shows that performance is largely a function of the frequency and nature of inter-component communication, in addition to the performance characteristics of the components themselves (Smith et al., 2002).

Observations from Test 1 (Section 4.1.1), indicates that CPU usage varied for different versions of BigBlueButton when the system is accessed by a fixed number of users (Table 4, Section 4.1.1). The result shows a linear relationship between the number of users and CPU usage. The linear relationship observed here indicates that the system architecture is scalable, and an increase in the number of processors as the web conference system scales gives linear speed up and increase in system capacity.

Addition of more processors to a system, allows larger dataset size (Cavasos, 2007). Test 1 (Figure 6, Section 4.1.1) shows that as number of processors grow from one BigBlueButton version to the other, the problem size is scaled and this scaling results in a substantial increase in the parallel parts of the system as compared to the serial parts (Gustafson, 1988). Thus, in a perfect situation where number of concurrent users and CPU usage could be measured without error, all observations would lie exactly on the same line. This assertion is further proved with the equation below. Assuming time *(n, x)* is the time required to solve a problem of size[2] *x*. The speedup on a problem of size *x* with *n* processors is the execution time on one processor divided by the time on *n* processors (Duboc, 2007).

$$speedup(n,x) = \frac{time(1,x)}{time(n,x)}$$

Similarly, if we consider the outcome of Test 2 which examines the varied number of concurrent users and CPU usage, the scatterplot in Figure 10 suggests a functional relationship between the CPU usage and the users. Each point on the plot represents both the number of users measured and the CPU usage respectively. This result confirms the importance of increasing the capacity of the system hardware configuration for speed by a factor of *x*. It shows that as the number of user grows, and the system is scaled, the query response time decreases by no less than a factor of *x*. The decrease in query response time speeds up the processing time and gives opportunity for more resources to be utilized (Figure 10, Section 4.1.2) (Gunther, 2011). It then follows that addition of CPU or memory to the system enhances better performance.

Performance enhancement requires event processing time, batch throughput, transaction rates, database size, users perception and resources used to service a request, and increasing the system workload by adding resources to the web system enables effective virtualization technology , as it provides more resources for the hosted set of operating system and application modules to share (Shoup, 2008). One important factor that is related to speedup performance is efficiency, and this is speed divided by the number of processors (Hill, 1990).

$$efficiency\,(n,x) = \frac{speedup(n,x)}{n}$$

$$= \frac{time(1,x)}{time(n,x)} \Big/ n$$

In Figure 14 (Test 3, Section 4.1.3), where the residuals are on the vertical axis (throughput) and the independent variable (varied concurrent users) on the horizontal axis, both speedup and Scaleup are two sides of the same coin, though they appear to be different metrics (Gustafson 2000). Thus, if transactions can be executed more quickly after increasing the hardware capacity, then, Scaleup can be expressed as a ratio of the capacity with *p* processors to the capacity with one processor, (Gunther 2000]). In test 3, it was discovered that the amount of parallel work increases in response to the presence of additional computational resources, but the amount of serial work remains constant. This result then supports the assertion that with more computing power, better web system quality can be achieved with higher number of processors and lower memory usage in the same amount of time (Gustafson, 1988).

Furthermore, comparing the results of test 3 (Figure 13 &14, Section 4.1.3), it is valuable to check the consistency of the independent and dependent variables tested and analysed. The result actually confirms that the throughput time is consistent with expectation, as adding more resources (both software and hardware), is directly proportional to the system throughput (Lin, 2001). Also, Little (1992) states that the average number of requests in a system must equal the product of the throughput of that system and the average time spent in that system by a request, that is if a system contain an average of N users, and the average user spends R seconds on the system, then the throughput X of the system is given by: X = N/R. Little's law can be applied at many different levels of a web conference system. The key to success is consistency: the definitions of concurrent users, system time taken and throughput has to be compatible with one another as seen in Test 3, (Figure 13 &14, Section 4.1.3). This indicates the timesharing of system resources at the different levels of web conference system.

In Test 4, (Figure 15 &16, Section 4.1.4), the output result shown graphically indicates that Little`s law holds for this test. The result proves that concurrent users corresponds to the utilization of the resource. In this test, the resource is utilized whenever there is a request present; thus resource utilization is equal to the proportion of time there is one request present, which is also equal to the average number of requests present.

Consequently, throughput corresponds to the rate at which the resource (CPU) is satisfying requests, and time taken the system corresponds to the average provision requirement per request at the resource, given by N=XR (Little, 1992).

Test 5 examines the audio quality against the number of concurrent users that accessed the web conference system at about the same time. The observed results indicates audio performance measurement became necessary in order to investigate causal relations or associations between the BigBlueButton version and the resources available to user to meet their needs (Moghaddam & Moballeghi, 2008; Osayomi, 2007). This causal relationship provides the required means to assess the status of software program running on the web conference application. Therefore, being able to add the right software that corrects the lags in BigBlueButton version 0.7 confirms the assertion that poor load scalability occurs in a system when one of the resources performance measure is an increasing function of itself, which is called selfexpanding. Selfexpansion occurs in scalable system when the holding time of a resource is increased by contention for a like resource. Self-expansion diminishes scalability in systems by reducing the traffic volume at which saturation occurs (Bondi, 2000; Duboc, 2009). Therefore, keeping a sustainable check on the performance and its quality to avoid software error is vital to a successful web conference system scalability (Smith, 2000), as system performance depends on design and implementation choices in the software layers that are used (Gerrit 2011).

6. CONCLUSION, LIMITATIONS AND FUTURE RESEARCH

This chapter is organized into three sections. Section 6.1 provides the conclusions of this research. Section 6.2 describes the limitations of this research. Section 6.3 provides suggestions for future research.

6.1 Conclusions

This research examines the different dimensions of a major attribute of web conference system, which is its scalability. I demonstrate the applicability of scalability dimensions to the BigBlueButton Web conference system and provide insights that may be relevant to technical managers.

I conclude that the scalability of a web conference system has three (3) more dimensions than are usually recognized in literature. These dimensions are: data size, speed and cost. Also, I conclude that parallelism in work load distribution offers a better and more cost effective web conference application compared to sequential distribution of work load. Similarly, particular attention should be paid to the extensible architecture of a web conference system in early stages of product development, so as to allow for flexibility in system scalability as the system application evolves.

6.2 Limitations

This research has at least two limitations. First, the research uses only one application to make conclusions. This research only focused on BigBlueButton, not other conference systems.

The second limitation is that the results presented in Chapter 4 relied on my interpretation of test results from the logs obtained for the various versions of BigBlueButton.

The third limitation is that the variables tested were selected based on my knowledge of the BigBlueButton Web conference system.

6.3 Future Research

Two suggestions for future research are offered. First, identify the attributes that an open source components should have to support system level scalability.

Second, research work should be undertaken to develop a technique for quantifying the quality of system scalability. It would be desirable to find a metric to measure the quality of scaling a system.

REFERENCES

- Abachi, H. & Walker, J. 1996. "Network Expandability and Cost Analysis of Tours, Hypercube and Tree Microprocessor Systems" 28th IEEE South-eastern Conference on System Theory, Louisiana, U.S.A, pp. 426-430
- Abachi, H. & Amiripour, M. 2007, Hardware Design, Expandability, System Cost And Mean Inter-Node Message Distance Of Augmented Hypercube Torus And Master-Slave Star-Ring Augmented Hypercube Architectures Dept. of Electrical and Computer Systems Engineering Monash University Australia
- Ananthanarayanan, R., Esser, S.K., Simon, H.D,& Modha, D.S. 2009. The Cat is Out of the Bag: Cortical Simulations with 109 Neurons, 1013 Synapses, IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, 2Lawrence Berkeley National Laboratory, One Cyclotron Road, Berkeley, CA 94720
- Antona, M. Basdekis, I., Klironomos. I., & Stephanidis, C. 2006. White Paper: promoting Design for All and e-Accessibility in Europe Universal Access in the Information Society Volume 5 Issue 1
- Avesani, P., Bazzanella, C., Perini, A., and Susi, A. 2005. Facing scalability issues in requirements prioritization with machine learning techniques, in RE'05:Proceedings of the 13th IEEE International Conference on Requirements Engineering, IEEE Computer Society, Washington, DC, USA, pp.297–306
- Batty, M. 2011. Integrated models and grand challenges. *ArcNews,* Winter 2010/2011 Available at http://www.esri.com/news/arcnews/winter1011articles/integrated-models.html
- Batty, M., Heppenstall, A.J., Crooks, A.T., See, L.M., 2011, Agent-Based Models of Geographical Systems, Dordrecht: Springer
- Becker, D., Bershad, B.N., Chambers, C., Eggers, S., Fiuczynski M, G¨un Sirer E., Pardyak, P. & Savage S. 1995. Extensibility, Safety and Performance in the SPIN Operating System. In *Proceedings of the 15th Symposium on Operating Systems Principles*, pages 267–284, Copper Mountain, Colorado

- Beckett, D. 2001. Semantic Web Scalability and Storage: Survey of Free Software / Open Source RDF storage systems, SWAD-Europe Deliverable 10.1 http://www.w3.org/2001/sw/Europe/reports/rdf scalable storage report
- Bennett, R. & McGuigan, M. 2005, Parallel Heisenberg spin model performance on supercomputer architectures. ACM 2006 Symposium on Principles and Practice of Parallel Programming
- Bershad, B. N. & Grimm, R., 1995. Security for Extensible Systems Dept. of Computer Science and Engineering University of Washington Seattle, WA 98195, U.S.A
- Bondi André B. 2000. Characteristics of Scalability and Their Impact on Performance, in WOSP'00: Proceedings of the 2nd International Workshop on Software and Performance, ACM Press, pp.195–20
- Butler, J. 2009. Dimensions of Scalability. Http://drjbutler.wordpress.com/2009 /04/19/dimensions-of-scalability/
- Campbell, Luke & Koster B. 1995. "Software Metrics: Adding Engineering Rigor to a Currently Ephemeral Process," briefing presented to the McGrummwell F/A-24 CDR course
- Cardoso, J. Sheth, A. & Miller, J. 2002. Workflow Quality of Service Technical report, LSDIS Lab, Computer Science, University of Georgia, Athens GA USA, March 2002
- Carvalho, N.A. & Pereira J. 2010. Measuring Software Systems Scalability for Proactive Data Center Management. Computer Science and Technology Center Universidade do Minho Braga, Portugal
- Cha, H., Choi, S., Jung, I., Kim, H., Shin, H., Yoo, J., Yoon, C. 2007. RETOS: Resilient, Expandable, and Threaded Operating System for Wireless Sensor Networks, Department of Computer Science Yonsei University Seoul 120-749, Korea
- Chen, Y., & Sun, X. 2006. STAS: A Scalability Testing and Analysis System, Proceedings of 2006 IEEE Conference on Cluster Computing pp.1–10
- Clements, P.C., & Northrup, L.M. 1996. "Software Architecture: An executive overview," No. CMU/SIE-96-TR-003,Software Engineering Institute, Carnegie Mellon University
- Clements, P., Kazman, R., & Klein, M. 2002. Evaluating Software Architectures: Methods and Case Studies, Addison-Wesley Professional

- Datta, A., Dutta, K., Thomas, H., and VanderMeer, D., 2003. World wide wait: A study of Internet scalability and cache-based approaches to alleviate it. Management Science © 2003 INFORMS Vol. 49, No. 10, pp. 1425–1444
- Dixon, F. 2011. Development Road map for BigBlueButton 1.0, http://code.google.com/p/bigbluebutton/wiki/RoadMap
- Duboc, L., Rosenblum, D., & Wicks, T. 2006. A Framework for modeling and analysis of software systems scalability, in ICSE '06: Proceedings of the 28th international conference on Software engineering. Doctoral Symposium, ACM, New York, NY, USA, pp.949–952
- Duboc, L., Rosenblum, D., & Wicks, T. 2007. A framework for characterization and analysis of web conferencing system scalability. In: Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering.(pp. pp. 375-384), New York, USA
- Duboc, L., Letier, E., Rosenblum, D., & Wicks, T. 2008. Case study in eliciting scalability requirements, in RE'08: Proceedings of the 2008 16th IEEE International Requirements Engineering Conference, Barcelona, Spain
- Easterbrook, S., Singer, J., Storey, M., & Damian, D., 2008. Selecting Empirical Methods for Software Engineering Research, in Guide to Advanced Empirical Software Engineering. Page 285-311
- Elshoff, J. L. 1978. "An Investigation into the Effects of the Counting Method Used on Software Science Measurements."*ACM SIGPLAN Notices 13*, 30-45
- Flouris, M. D. 2009, Extensible Networked-Storage Virtualization with Metadata Management at the Block level, Department of Computer Science University of Toronto.
- Flouris, M., Lachaize, R.,& Bilas, A.2001. Shared & Flexible Block I/O for Cluster-Based Storage. Institute of Computer Science, Foundation for Research and Technology, P.O. Box 1385, Heraklion, GR-71110, Greece
- Gunther, N.J. 2006: 5, Evaluating Scalability Parameters. In: Guerrilla Capacity Planning: A Tactical Approach to Planning for Highly Scalable Applications and Services. Springer-Verlag New York, Inc., Secaucus, NJ, USA

- Gunther, N.J. 2007. Guerrilla Capacity Planning: A Tactical Approach to Planning for Highly Scalable Applications and Services, XX, 253 page 108
- Gunther, N.J. 2011. Evaluating Scalability Parameters: A Fitting End, Copyright ©2011 TeamQuest Corporation, www.teamquest.com
- Gustafson, J. L. 1988. "The scaled-sized model: A Revision of Amdahl's Law," ICs Supercomputing '88. L. P. Kartashev and S . I. Kartashev eds., International Supercomputing Institute Inc., 11, 130-133
- Hill, M. D. 1990. What is Scalability?, ACM SIGARCH Computer Architecture News 18(4), 18–21
- Iyengar, A. K., Squillante, M.S., Zhang, L. 2000. Analysis and Characterization of Large-Scale Web Server Access Patterns and Performance IBM Research Division Thomas J. Watson Research Center Yorktown Heights, NY 10598
- Kong, J., Bridgewater, J., & Roychowdhury, V. 2006. A general framework for scalability and performance analysis of DHT routing systems, DSN'06: International Conference on Dependable Systems and Networks pp.343–354
- Kuhn Thomas S. 1961. The Function of Measurement in Modern Physical Science Author(s): Reviewed work(s):Source: Isis, Vol. 52, No. 2, pp. 161-193Published by: The University of Chicago Press on behalf of The History of Science Society
- Larsen, P. 2008. Regression and analysis of variance http://statmaster.sdu.dk/courses/st111
- Lin, B. 2001. Scalability Management for e-business solution: A resource based view, Louisiana State University, Shreveport
- Little, J.D.C. 1992. ``Are there `Laws` of Manufacturing Systems: Foundations of World-Class Practice, edited by J.A Heim and W.D. Compton, National Academy Press, Washington, D.C, 180-188
- Mathew, J., & Vijayakumar, R. 2011. The Performance of Parallel Algorithms by Amdahl's Law, Gustafson's Trend, © 2011 ACEEE Computer Science and Information Technology

- Miller, J. A. & Seila, A. F. 2000. "The JSIM Web-Based Simulation Environment." Future Generation Computer Systems: Special Issue on Web-Based Modeling and Simulation 17(2):119-133
- Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A.,& Riedl, J. 2003. Movie Lens unplugged: Experiences with a recommender systems on four mobile devices. In *Proceedings of the 2003Conference on Intelligent User Interfaces*
- Moghaddam, G. & Moballeghi, M. 2008. How Do We Measure Use of Scientific Journals? A Note on Research Methodologies Razi Metallurgical Research Center (RMRC), Tehran, IRAN
- Moyle, K. 2010, Building Innovation: Learning with technologies. Australian Education Review, Australian Council *for* Educational Research
- Osayomi, T. 2007. International Marketing Research: Research Paper ATLANTIC INTERNATIONAL UNIVERSITY North Miami, Florida October
- Plummer, J. 2004. A Flexible and Expandable Architecture for Computer Games, Arizona State University
- Prechelt, L & Tichy, W.F. 1998. A Controlled Experiment to Assess the Benefits of Procedure Argument Type Checking. IEEE Transactions on Software Engineering, vol. 24 (4), April 1998, 302-318
- Sarkar, V., Harrod, W., & Snavely, A., 2009. Software challenges in extreme scale systems V. http://www.cs.rice.edu/~vs3/PDF/ Sarkar-Harrod-Snavely-SciDAC
- Schermerhorn, P. W., Minerick, R. J., Rijks, P.W., & Freeh, V.W. 2001. User-level Extensibility in the Mona File System. In *Proc. of Freenix*, pages 173–184
- Schroeder, Barbara: Microsoft Live Meeting 2007: Web conferencing system for virtual classrooms, retrieved February 20, 2009 from Http://www.microsoft.com/education/highered/whitepapers/conferencing/WebConferencing.aspx
- Smith, C. U. & Williams, L. 2002. Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software, Reading, MA, Addison-Wesley
- Smith, C.U., & William, L. 2004. Web application scalability: A Model-Based Approach, in Computer Measurement Group Conference (CMG), 2004, 215-226

- Smith, J.C. & Krajewski, L.L. 1971. Expandability and Collection wise Normality Reviewed work(s): Transactions of the American Mathematical Society, Vol. 160 pp. 437-451Published
- Smith, L.W. 2000. *CrossTalk: The DoD Journal of Software Engineering* Software Estimation, Measurement & Metrics GSAM Version 3.0, Chapter 13
- Sommerville, Ian 2000, Software Prototyping, Software Engineering 6th Edition, Chapter 8
- Thomson, L. 2009. Scalability and Performance Best Practices. http://www.omniti.com
- Weatherley, D., Boros, V.,& Abe, S. 2012, Performance benchmarking results for a parallel implementation of the discrete element method, Geophysical Research Abstracts Vol. 14, EGU2012-7996-1, 2012 EGU General Assembly
- Weatherley, D., Boros, V., Hancock, W. and Abe, S. (2010), Scaling benchmark of ESyS-Particle for elastic wave propagation simulations, Sixth IEEE Int. Conf. eScience, 277–283
- Whitworth, B., Fjermestad, J. 2005. The Web of System Performance (WOSP) for balanced Information System Design and Evaluation
- Whitworth, B., Fjermestad, J., & Mahinda, E. 2005. The Web of System Performance: A Multi-Goal Model of Information System Performance in Press, Communications of the ACM