The Impact of Configurations on the Performance of Server Based Computing

Renatus Michael, Faith Shimba, Daniel Koloseni
Computer Science Department
The Institute of Finance Management
Dar es Salaam. Tanzania

ABSTRACT: *Server based computing performance can be affected by several factors. Some previous researches on performance of Server based computing came up with factors which falls to the category of server itself and on the network infrastructure. This research explores the performance behaviour of Thin Client terminals by measuring the performance while changing the performance optimisation configurations on the client side. These measurements are measured at default and with performance configurations set while analysing the response of performance metrics to observe their impact. The analysis shows that compression, caching and merging tends to have significant impact to the overall performance of server based computing. The measurement approach used is slow-motion benchmarking. In some cases, performance behaviour is compared with the previous performance measurement research done nearly 8 years ago for more discussions.*

Keywords: Server Based Computing, Performance, Thin Client Terminals, Performance Metrics, Slow-motion Benchmarking

Received: 10 January 2012, Revised 23 March 2012, Accepted 2 April 2012

1. Introduction

Thin Client computing has gained a significant popularity in many organisations. The reasons behind relies on the its architecture where each applications are executed in server and accessed by Thin Clients which are diskless dumb terminals [3,6]. The interests on the performance behaviour attracts attention to several group of people especially researchers in Network computing, vendors and IT personnel for better understanding and decisions support. This paper analyses the performance behaviour of Thin Clients at different bandwidth levels in triggered performance configurations. The experiment is done at default settings and when the performance configuration settings were set. The main target was to find out the impact of performance configuration on overall performance of server-based computing. The experiment was conducted by the use of a modifiedversion of slow-benchmarking technique previously designed by [7] and results were collected and analysed.

The rest of this paper is organised as follows: section 2 presents the related research, section 3 shows the problem statement followed by the experiment design in section 3. Thin Client performance metrics are described in section 4 followed by section 5 which discuss different metrics of performance in network computing and section 6 shows the performance optimization techniques employed in Thin Client computing. Section 7 presents the evaluation of the experimental environment while section 8 is where the results and discussions are presented. The section 9 shows the conclusion of this research report.

2. Related Research

Previously,[21] proposed and measured the interactive performance of Stateless, Low-level Interface Machine (SLIM) system.

They provides a quantitative analysis of two aspects which are; the extent to which today's interconnect can support graphical displays of interactive programs and the advantages of sharing resources in Thin Client computing. On the other hand, [7] used slow-motion benchmarking technique to measure the performance of several Thin Client architectures. They did this experiment in Network Computing Lab. In Columbia University and compared the obtained set of results with the conventional benchmark ones. Also, [4] conducted comparison experiment on several Thin Client architectures by analysing different performance metrics at several bandwidth levels. Lastly, [8] did a research to assess the capability of Thin Client computing model by measuring the performance of six different platforms running on a range of network bandwidths. In addition, they analysed the differences in the various approaches of the underlying remote display protocols on overall performance. The results also were capable of quantifying the impact of existing display encoding primitives, display caching, compression techniques and display update policies across a wide range of Thin Client systems.

3. Problem Statement

As seen on the previous researches on Thin Client performance analysis, the last one was done on 2002. There is no updated published document that analyses the performance factors on Thin Client computing despite of a dramatic improvement of computing performance in all aspects of computing starting on the network area (as edges) to the computing processing power (as nodes). For example, Network speeds have increased dramatically such that now days there are computers with promising specifications that may, probably, return different results when performance measurement research is conducted on them. The table below shows the description of hardware and software used in the previous research and the new sets which were used in this research.

The main contribution of this paper is to provide the current trend of the performance behaviour of Thin Clients through different bandwidth levels in different Thin Client performance configuration settings. The demand for this research, as compared to the one done by [4] is that it uses the current sets of hardware and software and therefore having a chance of coming up with more updated and correct results as compared to those of nearly 10 years ago.

4. Expermental Design

The design of a test-bed follows the slightlymodified version of slow-motion benchmarking technique [4,7,8]. The test-bed in Figure 1 is a modified version of the original test-bed which measured the performance of Thin Clients nearly 10 years ago which implements slow-motion benchmarking technique [3].

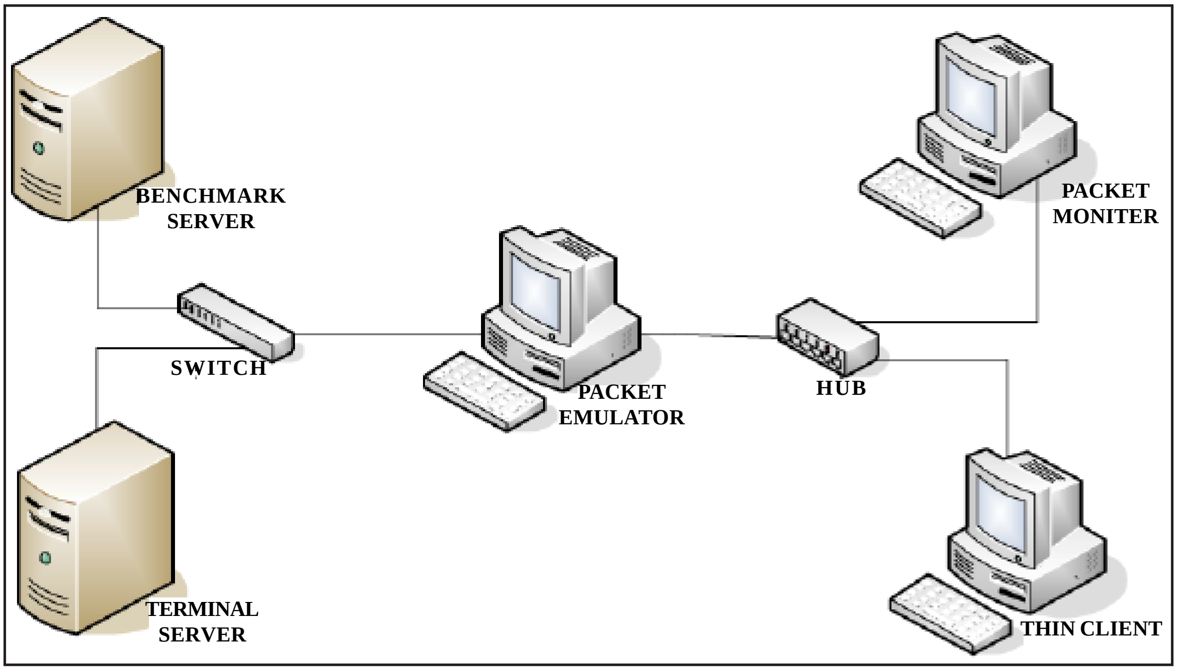

Figure 1. A designed version of Test-bed for this experiment (Adopted from (Nieh 2000))

The set up of this experiment is as seen in Figure 1. It is a modified-version of a slow motion benchmarking technique performed by [7]. The applications used to test the performance are of two types; multimedia application, to analyse how this technology performs upon multimedia applications and web application, to analyse the performance behaviour upon execution of web pages in different network bandwidths. A benchmark server was set to run windows 2003 server operating system and it is the one that hosts web and multimedia applications. In this experiment, ZiffDavis i-Bench benchmark suite version 5.0 [9] was used. The packet monitor machine runs WildPackets' Omnipeek 5.0 [5] and its function is to trap and record all network traffic which moves from Server (Benchmark Server) to Client (Thin Client). The Packet emulator varies the bandwidth of the network that links Client and Server so as to be able to get readings at different bandwidths. It was set to run Bandwidth Controller Standard [1]. A Terminal server is a normal Windows Server 2003 having Remote Desktop Connection (RDP) version 6.0 while a Thin Client is a normal PC running windows XP professional. For the case of this experiment, the benchmark server runs within the terminal server. The connection is established through RDP connection between a Terminal Server and a Thin Client.

In order to get optimum performance, it could be expected to use a switch instead of a hub simply because hub has its shortcomings such as packet broadcasting. We used hub in this test bed because we need to compare the trends of these results in different bandwidths for default settings and when the performance configuration settings are triggered. Moreover, this analysis is done through comparing the obtained trend with the trend and configuration settings being triggered. On the other hand, switch that will fit this purpose which were collected nearly 10 years ago and see any interesting observations. We assumed that currently, most of switch which support mirroring [2]. However, a hub is also capable because we are much interested with the trends of the performance metrics and not real values.

5. Thin Client Performance Metrics

This research was conducted through measuring some key performance metrics used in Thin Client computing. These key metrics of Thin Clients are used for analysis of interactive performance that comprises of speed of execution of applications and the visual display aspects. These metrics are used when web-based or multimedia applications are used as benchmarks for performance measurement.

The first metric is Latency, a synonym for delay. This is an expression of how much time it takes for a packet of data to travel from one designated point to another [14,10]. In some usage, latency is calculated by sending a packet that is returned to the sender; and the round-trip time is considered as the latency. Normally, as the total available network bandwidth is consumed, both TCP connections continue to fight for the bandwidth, continually pushing the network in and out of congestion. Therefore, the latency should be as small as possible for a network to have good performance. For web page applications, per-page latency of less than one second has been shown to be desirable to ensure that the flow of a user's browsing experience is not interrupted [14]. The performance measurement in Thin Client computing should, therefore, include total latency of execution of web pages and per page latency. The total latency of Thin Client is time duration of start to the end of client-server operation based on the operations involved.

Another metric which is commonly used is the amount of data transferred on a network from client to server. Normally, the more the data transferred to a network from client to server, the high the performance a Thin Client have. This metric can also be influenced by some performance optimisation techniques For example; caching web pages in Thin Clients might reduce the amount of transmitted data as it allows local accessing of pages instead of requesting them to travel from server [11]. In Thin Client computing, the correct measurement of amount of data transferred is through a network [7]. The measurement requires special network monitoring tools that capture the packets on a switch or hub, which passes the traffics, from server to client.

The quality of a multimedia display can also be used as performance metric. The quality of display of multimedia contents determines the performance of Thin Client under measurement. But, the amount of data transferred during execution of video playback. For example, is not enough to quantify the video quality of such platform. According to [7], the best approach to measure the video quality is to use the following formula in equation (1).

Equation i: A generic formula to manipulate video quality.

$$\mathrm{VQ} = \frac{\left[\dfrac{\text{DataTransferred (24 fps) / PlaybackTime (24 fps)}}{\text{IdealFPS(24 fps)}}\right]}{\left[\dfrac{\text{DataTransferred (1 fps) / PlaybackTime (1 fps)}}{\text{IdealFPS(1 fps)}}\right]} \qquad (1)$$

Where:
Data transferred (24fps) is the amount of data transferred in normal condition (in ideal used is 24fps) as a video playback is executed. Data transferred (1fps) is the amount of data transmitted as a slowest possible rate of video execution, which is believed to be the most possible amount of data. Playback time (1 fps) and playback time (1 fps) are the time duration for execution of a video at normal speed and the duration for slowest rate respectively. The ideal fps values are the integer values that correspond to 1 and 24 respectively. The video quality metric involved in the above formula takes into account only the amount of discarded data. However, it does not take into consideration tendency of some of video data to be more important to the overall multimedia display quality of a particular video sequence than others. For example, if the video frames are looking the same as the previous or next one in a frame sequence does not affect the user perceived video quality when compared to the case where the updates that are not similar to each of the case in the neighboring frames are discarded.

Moreover, bandwidth utilization can also be used as a performance metric. [20] defines effective bandwidth of a network as the maximum amount of meaningful data that can be transferred per unit time, exclusive of factors such as headers, padding and stuffing. Some of factors that affects the availability and utilisation of the network bandwidth are the sampling rates at which the various devices send information over the network, the number of elements that require synchronous operation, the data or message size of the information and the medium access control sub layer protocol that controls the transmission of information [12].

The last metric used is the CPU Utilisation on Client and Server. Both server and client should have enough power to process the execution of activities. Normally, if the processing power of client and server are not utilised to higher percentage, then it indicates that the performance is not caused by less power of the systems. This conclusion is drawn through taking into account that the large the free portion of utilised CPU the less the processing dependence on that particular computing machine [13]. This research includes the CPU utilisation metric as part of performance analysis. The measurement of the percentage utilisation of CPU is simple to measure as it just involves taking a reading from a respective computer.

6. Performance Optimisation Techniques

The main performance optimization techniques employed on Thin Client computing are discussed here. Some of the configurations have a lot of impacts on the outcome of performance behaviour of Thin Clients. These optimisation techniques are employed for the purpose of improving the remote display performance of Thin Client systems when the bandwidth is limited. Some of the common techniques are discussed below:

6.1 Caching

As the number of active Client users increase, the demand of network bandwidth to connect clients to a server increases as well. On the other hand, trying to scale network and server bandwidth to keep up with client demand is an expensive strategy. In Web-based applications, a proxy server provides caching services where it effectively migrate copies of popular documents from servers closer to client. As a result, Web client users see shorter delays when requesting for URL [15]. Network managers see less traffic and Web servers see lower request rate [11]. In Thin Client computing, a cache can be used to store the elements of display such as fonts and bitmaps that enables user to frequently obtain them locally instead of requesting them multiple times from the serve [7].

Normally, to effectively measure the performance of Thin Clients, the cache settings should be disabled in order to get correct results. On the other hand, according to [8], caching may not always beneficial. For example, Thin Client terminals like Independent Computing Architecture (ICA), enabling caching tends to reduce the amount of data transferred.

6.1.1 Compression

Data Compression shrinks down a file so that it consumes less space. Normally this is done for data storage and data communication. Storage space on disks is expensive so a file which occupies less disk space is "*cheaper*" than an uncompressed file [16]. Moreover, smaller files are also desirable for data communication since the smaller the file the faster it can be transferred. A compressed file appears to increase the speed of data transfer over an uncompressed file [17]. With respect to Server side computing, compression is a technique of reducing the size of the network packets for efficient transportation to the network. Algorithms such as zlib or run-length encoding can be applied to display updates to reduce bandwidth requirements with a smallest acceptable extra processing overhead [7].

Run-length encoding technique can be used to in Platforms such as ICA and RDP [8]. They use this technique to compress cache fonts and bitmaps in memory and on disk at the client system. Some of Thin Clients are capable of changing the compression settings are ICA, AIP and RDP and the previous researches shows that when compression is enabled, there is a extensive reduction in the amount of data transferred for at least a factor of two in all cases [8]. Moreover, the advantages of compression will become enormous due to advances in very large scale integration (VLSI) technology as it makes possible to open more application fields to large number of users while arising the necessity of rising of video encoding standards [18].

6.2 Merging

Merging is the act of putting the signals in a form of queue on the source host before sending them to the destination host. In the case of Thin Client computing, the signals are the screen updates. If two updates change the same screen region, merging will hold back the older update and only send the more recent update to the client. This technique proved to be successful in 802.11 WLANs where the scheme involving application layer merging of consecutive VoIP packets in the same host has been investigated to improve the capacity of Voice over IP (VoIP) over 802.11 WLANs [19]. In Thin Client computing, display updates are sent asynchronously during execution of application and decoupling application rendering of visual output on the server from the actual display of the output on the client [8].

7. Evaluation of the Experiment

This test-bed was shown to few people who have experience in the area of network computing through normal discussions whereby they provided some inputs. In general, this designed version was acceptable under the following assumptions:

Firstly, the hub was used instead of a switch to enable network traffics travelling from Thin Client to server to be monitored as they are broadcasted in real time. Although the hub have drawbacks like packet broadcasting since it does not have intelligence to know the destination address of its carrier. The alternative to hub was to use a switch. But to be able to monitor traffics from server to client, a port connected to packet monitor would require listening to all the traffic which would require a switch that supports port mirroring. Conversely, the principle aim of this research is to investigate the performance trends of Thin Client architecture even hub is can be useful.

Secondly, the delay caused by Packet emulator wasmeasured and found to have a delay of a value of 0.5ms. This delay has a negligible impact on the display on the Thin Client side especially due to the fact that the noticeable delay noticeable by human being is between 50ms to 150ms (Shneiderman 1992). But, since the collected results are compared with a set of results collected in similar experiment which involved a packet emulator with a delay of around 0.6ms, then the analysis of performance trends is still possible. That is, since the aim of experiment as to analyse the performance trends metrics in Thin Client computing (and not the magnitude of values), this delay has a negligible impact on the intentions of the research.

Thirdly, the network emulator software could be configured to operate in either distributed mode or gateway mode. In this experiment, one could expect a distributed mode to be applied as it fits this kind of experiments since it is much simple to operate and needs few clients on a network. But, the previous research conducted by [7] deployed a gateway mode approach to collect the performance results which are used as references for this research. The assumption is that the use of another mode of network emulator (distributed mode) might result into slightly different trend of results that can lead to incorrect interpretation.

Lastly, the delay time for a web benchmark (time elapsed between two successive web page executions) was configured to be approximately 2 seconds in the web benchmark. This delay was inserted so as to ensure that the individual web pages are completely displayed on the client side before the next one start to execute on the server side. This delay also ensures that all the data has been transferred from the server to client since some of Thin Clients have a tendency of discarding data while maintaining display quality in low bandwidths [4].

Moreover, this delay time is made by assuming also that, in real environment, this is the smallest amount of time for users to view different parts of a page and probably reading some interesting sections hence supporting the overall interactive process with the user. Although, the main shortcoming of this assumption is that, it assumes that the user must take a look on the pages for this delay interval before clicking any of the other links in which realistically it may not be true. For example, user might have requests a wrong page and immediately wants to request the other one while has already initiated the first page where he might not wait for these 2 seconds which are rare cases in most cases.

8. Results and Discussions

8.1 Latency with page caching disabled

The conclusion made by [7] shows that caching has very small impact to the latency of data transfer. They conducted experiment with browser's default settings and with disabled page caching and came with such conclusion in their paper. This research conduct the same approach to find the impact of page caching to the latency of data transfer to see if there is any notable observation. To start with, the sets of data collected when browser was in default settings are combined with a new set of data. This new set is obtained by running the same experiment but after disabling the page caching configurations on the server side. The results are illustrated in Figure 2 and the observations are as follows:

The first observation is that, the overall trend of latency in different bandwidth performance is not impacted by caching. That is, the behaviour of Thin Clients towards web applications is the same regardless of caching settings.

[7] draw a conclusion that caching has a very small impact on latency in RDP and VNC Thin Clients. This research deduces that the impact of caching is very small in large bandwidth but the impact starts to be significant as bandwidth of the network approaches low values. This research therefore suggests that web page caching in Thin Client computing helps to reduce the latency of transfer of pages especially in low bandwidths as it has minor effect at higher bandwidths. This research predicts that in the near future where the bandwidth is expected to rise to a number of Giga Bytes per second, caching will have no effect on the latency of execution of web pages.

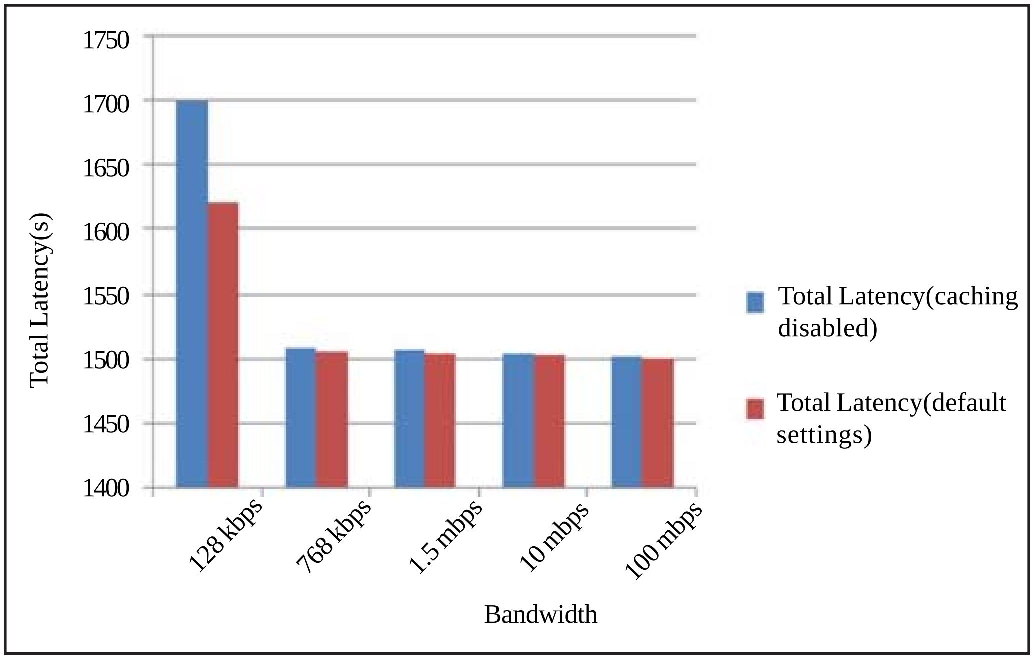

Figure 2. Total Latency of web benchmark with default settings and disabled caching combined (Source: Author)

8.2 Amount of Data Transferred with page caching disabled

In section 6.1, caching is was found to have some impact on the performance of the web page execution, especially on latency and amount of data transferred metrics. A previous research done by [7] suggested that caching had a very small impact to the amount of data transferred by RDP Thin Clients. Although, due to the advancement of computing power over the past 10 years, there might exist some interesting observations on the performance trends on the This section analyses the impact of caching to the amount of data transferred and discuss any observation through the use new sets of hardware and software.

Figure 3 illustrates the trend of amount of transferred data at different bandwidths when page caching is disabled and with default settings.

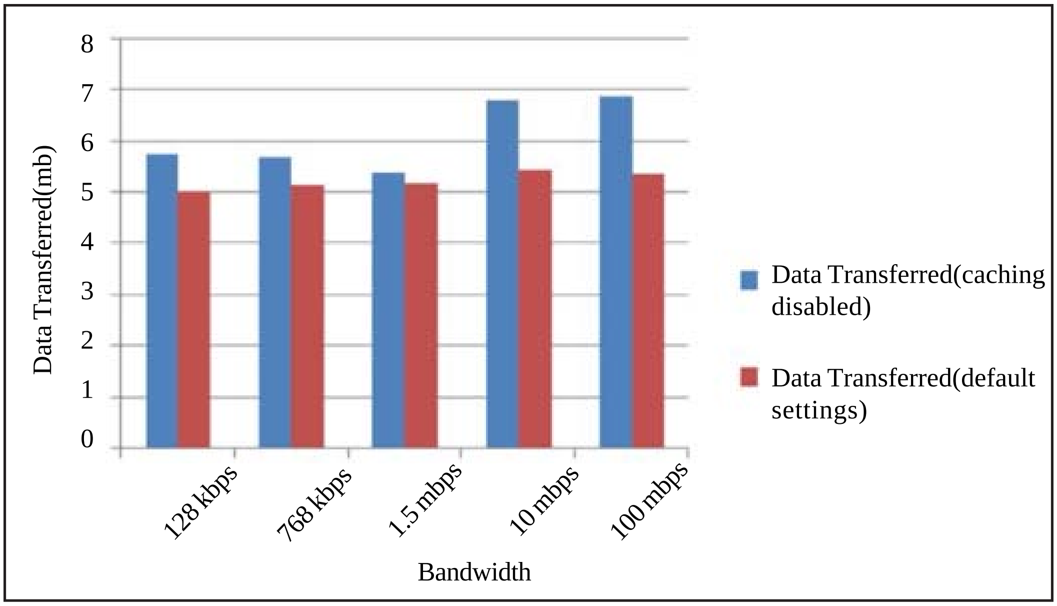

Figure 3. The amount of data transferred by a web benchmark with default settings and with caching disabled combined (Source: Author)

The analysis shows that the impact of caching on the amount of data transferred becomes significant when comparing the two trends in Figure 3. That is, more data seem to be transferred on the network the when the cache is enabled. However, these results suggest that the effects of caching on RDP Thin Clients tends to be more significant on the amount of transferred data at higher bandwidths than when the computer processing power was low. This is so interesting observation since the research conducted by [7] suggested that there is was minor impact of caching on RDP Thin Clients.

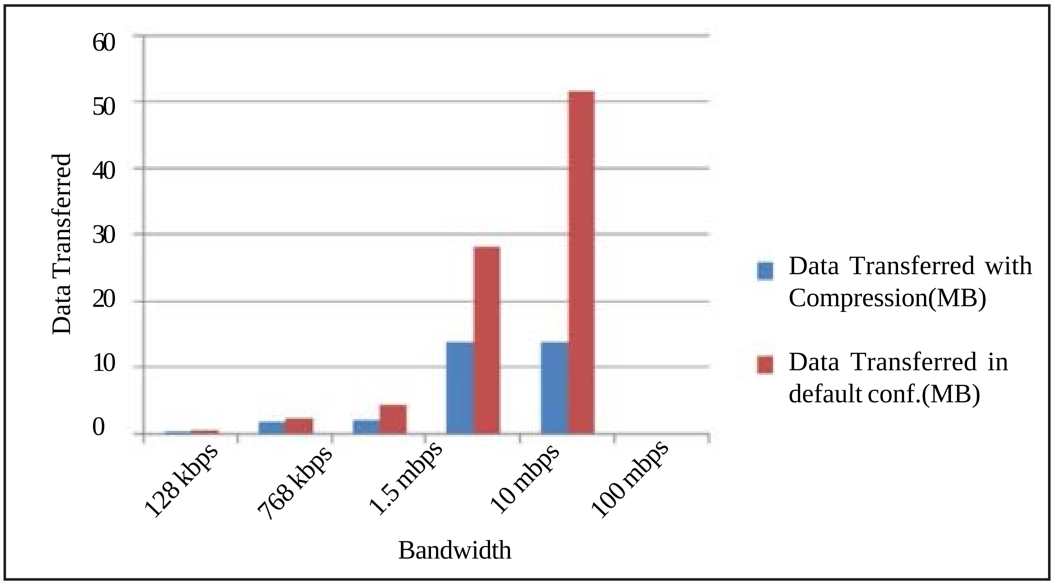

Figure 4. The amount of data transferred with and without compression combined (Source: Author)

8.3 Video benchmark with compression enabled

The compression tends to reduce a size of a file before sending it to a network. Based on the previous experiment done by Yang et al. (2002), the compression was found to reduce the amount of data transferred to a network. The analysis in that research did not show the impact of amount of data sent to a network versus the bandwidth of a network. The aim of this section is to analyse the impact of compression on the amount of transferred data, the trend is the most important thing here.

Figure 4 shows that compressing video files is accompanied by a significant reduction of amount of data transferred on a network. On the other hand, the reduction of amount of transmitted data is accompanied with the reduction of video quality. Therefore, the compression approach is not adviced in Thin Client computing unless for the case where quality is taken for granted.

Further analysis shows that the differences between the amount of data transferred with compression and default settings is so significant in higher bandwidths. It can be seen in Figure 4 that at a frequency of 100mbps, the gap between the amount of transferred data between default configuration and caching is much higher than in the low bandwidth. This suggests that once compression is done, the amount of data transferred does not need high bandwidth to be sent.

9. Conclusion

This paper presents some performance details of Thin Client computing by concentrating on their behaviour in various bandwidth levels. It analyses the main performance metrics in Thin Client computing through measuring them in a designed experiment and executing the desired version of applications and measuring the network behaviour. The discussion relies on the findings of the experiment in default settings and with performance configurations triggered. The discussions are limited to three main performance optimisation techniques which are caching, merging and compression.

References

[1] Bandwidth_Controller, (2007). Bandwidth Controller Standard. Retrieved 20th July 2009, from http://www.brothersoft.com/bandwidth-controllerstandard- 73207.html.

[2] Curtis, J. (1998). Port mirroring: The duplex paradox. Network World Fusion.

[3] Golick, J. (1999). Network computing in the new thinclient age. netWorker 3(1) 30-40.

[4] Nieh, J., Yang, S. J., Novik, N. (2000). A comparison of thin-client computing architectures. Network Computing Laboratory, Columbia University, Technical Report CUCS-022-00.

[5] WildPackets, (2009). WildPackets Omnipeek Software. Retrieved 10th June 2009, from http://www.wildpackets.com/products/distributed_network _analysis/omnipeek_network_analyzer/network_monitoring.

[6] Wyse. (2004). Solving Business Problems at the Point of Data Access. Retrieved 30th June 2009, from http://www.wyse.com/resources/whitepapers/.

[7] Yang, S. J., Nieh, J., Novik, N. (2001). Measuring thinclient performance using slow-motion benchmarking. ACM Transactions on Computer Systems (TOCS).

[8] Yang, S. J., Nieh, J., Novik, N., Selsky, M., Tiwari, N. (2002). The performance of remote display mechanisms for thin-client computing. *In*: Proceedings of the 2002 USENIX Annual Technical Conference.

[9] ZiffDavis. (1996). i-Bench Media Benchmark suite. Retrieved 13rd June 2009, from http://www.pcmag.com/.

[10] Perkins, D. D., Hughes, H. D., Owen, C. B. (2002). Factors affecting the performance of ad hoc networks. IEEE International Conference on Communications, ICC.

[11] Abrams, M. (1995). Caching proxies: Limitations and potentials, Dept. of Computer Science, Virginia Polytechnic Institute and State University.

[12] Koren, Y., Pasek, Z. J., Ulsoy, A. G., Benchetrit, U. (1996). Real-time open control architectures and system performance. CIRP Annals-Manufacturing Technology, 45 (1) 377-380.

[13] Buzen, J. P. (1976). Fundamental Laws of Computer Systems Performance. Centre for Research in Computing Technology, Havard University.

[14] Nielsen, J., Meck, R. L. (1994). Usability inspection Methods, Wiley.

[15] Rabinovich, M., Spatscheck, O, (2003). Web caching and replication. SIGMOD Record, 32 (4) 107.

[16] Richardson, T., Stafford-Fraser, Q., Wood, K. R., Hopper, A. (1998). Virtual network computing, *IEEE Internet Computing*, 2 (1) 33-38.

[17] Ladino, J. N. (1996). Data Compression Algorithms. Retrieved 14th August 2009, from http://www.ccs.neu.edu/groups/honorsprogram/ freshsem/19951996/jnl22/jeff.html

[18] Sikora, T., Berlin, H. H. I. (1999). MPEG digita l video coding standards. Compressed Video over Networks, p. 51- 75.

[19] Sathanur, A. V., Sridhar, G., Sridhar, V. (2006). Application Layer Packet Merging to Improve Call Capacity in a 802.11 Based Intercom, IEEE Computer Society Washington, DC, USA.

[20] Lian, F. L., Moyne, J. R., Tilbury, D. M. (2001). Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet, *IEEE Control Systems Magazine,* 21(1) 66-83.

[21] 21 Schmidt, B. K., Lam, M. S., Duane, J. (1999). The interactive performance of SLIM: a stateless, thin-client architecture.