Block Level Streaming Based Alternative Approach for Serving a Large Number of Workstations Securely and Uniformly

F. Khan[1], M. Quweider[1], M. Torres[2], C. Goldsmith[2], H. Lei[1], L. Zhang[1]*
[1]The University of Texas Rio Grande Valley, Brownsville, Texas
[2]The University of Texas System, Austin, Texas
**Corresponding Author: fitra.khan@utrgv.edu*

***Abstract*—There are different traditional approaches to handling a large number of computers or workstations in a campus setting, ranging from imaging to virtualized environments. The common factor among the traditional approaches is to have a user workstation with a local hard drive (nonvolatile storage), scratchpad volatile memory, a CPU (Central Processing Unit) and connectivity to access resources on the network. This paper presents the use of block streaming, normally used for storage, to serve operating system and applications on-demand over the network to a workstation, also referred to as a client, a client computer, or a client workstation. In order to avoid per seat licensing, an Open Source solution is used, and in order to minimize the field maintenance and meet security privacy constraints, a workstation need not have a permanent storage such as a hard disk drive. A complete blue print, based on performance analyses, is provided to determine the type of network architecture, servers, workstations per server, and minimum workstation configuration, suitable for supporting such a solution. The results of implementing the proposed solution campus wide, supporting more than 450 workstations, are presented as well.**

***Index Terms*—Block Level Streaming, Open Source, virtualized computing, large number of clients, campus computing, Windows clients, Linux.**

I. Background

Our legacy institution, The University of Texas at Brownsville (UTB), was undergoing separation in 2012 through 2015 from its partnering institution, Texas Southmost College (TSC), causing a significant budget cut, thereby having to reduce Information Technology (IT) personnel by about 50% along with significant decrease in funds available for maintenance and licensing [1]. The newly separated institution, referred to as UTB$^{2.0}$, provided funds to its IT unit to modernize IT services such that IT services continue at the same level given half the IT workforce and maintenance/licensing budget cuts. This called for supporting more than 450 computers campus wide with a very few field technicians, and at the same time making software available uniformly on all the computers. IT and Computer Science (CS) pooled their resources to come up with the proposed solution presented in this paper, which at the same time became the groundwork for a thesis for a Master's student in CS [2].

II. Traditional Computing Environments

In a traditional computing environment, a client involves a local nonvolatile storage, such as a Hard Disk Drive (HDD), Random Access Memory (RAM), a Central Processing Unit (CPU), network connectivity and software. A client could be

- an autonomous workstation,
- a thin client in Virtualized Desktop Infrastructure (VDI),
- a fat client in VDI,
- a web based virtual desktop user, or
- a web based virtual app user.

The traditional model of computing could not be supported in UTB$^{2.0}$ setting given the reduced IT workforce and maintenance/licensing budget cuts. The new required model of computing had to address the drastically reduced number of field technicians, and the need to make more than 450 workstations across campus to offer exactly the same software in order to avoid having specialized computer labs.

III. Salient Features of the Proposed Solution

The cause of a computer hardware failure can be the CPU, RAM or a hard drive. Removing any of these components from a lab workstation would reduce repair tickets. However, one cannot do computing without CPU and RAM being in a workstation, therefore, the only hardware that can be removed from a workstation is its hard drive turning the workstation into a diskless workstation. Another reason for having a diskless workstation is to provide security and privacy among users. When a workstation changes hands from one user to another, no session data from the previous one should be available. This also satisfies one of the university's security regulations requiring one to encrypt a permanent storage device. However, one does not need to encrypt workstation's permanent storage in a diskless workstation since there is none.

Therefore, to keep field maintenance to a minimum and to provide security and privacy to users, hard drive needed to become irrelevant in the proposed solution. This calls for some sort of diskless workstation station serving users via some kind of virtualized computing environment. The proposed solution employs Open Source software to provide virtualized environment to avoid per client or per seat licenses. Note that using Open Source software to provide virtualized environment does not save cost on licenses for software packages. This is to emphasize that the cost of just providing a virtualized

978-1-5386-5762-1/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDIS.2018.00021

environment using traditional solutions, which are not Open Source, is substantial.

After reviewing the state of the university network and modern approaches to serving files or images over the network, Block Level Streaming approach was adopted to serve *snippets* of the image containing Operating System (OS) and applications. It is important to note the use of word "snippets" since that is exactly what the proposed solution involves when streaming. The proposed system only streams what is generally required to get a workstation going, and then dynamically streams more snippets as requested by the workstation as a result of the actions by its user.

Another important factor steering the proposed solution was to have all the computers (more than 450) to have the same look and feel. This is to avoid specialized computer labs causing fragmentation of resources. The idea was to provide more than 450 computers across campus with access to any software needed by a student taking any class. Consequently, the proposed solution has a central core to serve all the workstations. The core needed to have a license manager to keep the use of any software within the allowed legal limits.

In order to meet the security and privacy constraints, the solution had to be such that when a user logged out, the new user would get a fresh image. Therefore, the proposed solution incorporates a Read Only (RO) image that is sent to each requesting workstation. Besides providing security and privacy for users, this approach also provides flexibility to the users by allowing them to install any software during a session without needing to have computer administrators come over to install the required software, thereby, reducing field visits as well.

In summary, below are the salient features of the proposed solution:

1) Open Source software to provide virtualized computing environment
2) Diskless workstations
3) Central core to provide a uniform image to each diskless workstation
4) Block level streaming to provide OS and applications to each diskless workstation
5) A centralized license manager to manage licenses of different applications
6) RO image to provide flexibility, security and privacy at each workstation

IV. DETERMINING RESOURCE USE LIMITS

The goal of the proposed solution is to have a virtualized computing environment in which N client diskless workstations are served from a core of servers using open source block level streaming for serving RO Windows OS image with central license manager for keeping the use of applications within legal limits. It is a network intensive solution heavily tasking CPU resources at the core for serving client workstations. The following considerations are in order for such a solution to be within acceptable parameters, such as acceptable boot time and acceptable application processing response time, to provide user experience at par with traditional computing environments:

- C_S, number of clients or workstations streamed per CPU core server
- C_{SI}, number of clients streamed per server network interface
- C_{VLAN}, number of clients streamed per Virtual Local Area Network (VLAN)
- V_{SI}, number of VLANs per server interface

A CPU resource is concurrent, and most of the applications, including block level streaming, uses the CPU as such. In case of block level streaming, the mechanism is divided into virtual shelves and each shelf has virtual blades. This calls for two more parameters to be determined for the proposed solution to be within acceptable parameters:

- C_B, number of clients streamed per virtual blade
- B_{VLAN}, number of virtual blades per VLAN

As part of the proposed solution, the limits on resource usage are determined to serve as a blueprint for successful implementation.

The above parameters are determined using the state of the art networking and server technology available at the time of experimenting with the proposed solution. Therefore, the networking and CPU resources can be tasked further as the technology improves, for example, when networking moves from 10 Gbps to 100 Gbps to 1 Tbps technology.

V. BOOT PROCESS, PROTOCOLS AND TOOLS

When a diskless workstation is powered on, Open Source version of Pre eXecution Environment (PXE) protocol, iPXE, is used by the diskless workstation to start the booting process over the network. A core server responds to the requesting workstation with a pointer to a bootable Windows image on a Virtual Hard Drive (VHD). This is then followed by the workstation requesting initial set of snippets of the image, containing Windows OS and various applications, required for a startup virtualized environment. As the user progresses in the session, more snippets are dynamically sent over the network to the requesting diskless workstation. Figure 1 shows the basic concept of this interaction.

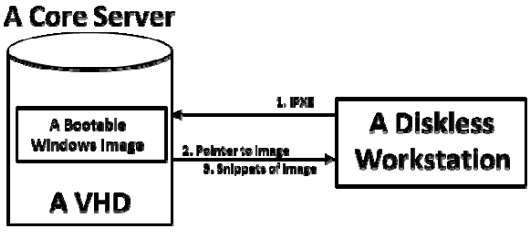

Fig. 1. Boot Process of a Diskless Workstation

Streaming protocol used for transferring snippets of an image is ATA over Ethernet (AoE) [3]. The AoE protocol is generally used for serving files from a network storage as part of a network based storage solution.

In the proposed solution, a Windows image is provided on a VHD as a block storage target. Vblade (Virtual Blade) software, based on AoE protocol, is used to act as a block storage. A virtual blade resides on a shelf in a certain slot. Hence, an image, properly referred to as AoE target, resides on a virtual blade in a specific slot on a specific shelf. For example, $AoE_{5.10}$ refers to an image (or AoE target) on the virtual blade in Slot 10 on Shelf 5. As an example, a sample boot file given to a diskless workstation with an $AoE_{5.10}$ target to get its image from may look like as follows [2]:

```
#!ipxe
dhcp net0
set keep-san 1
sanboot aoe:e5.10
```

DHCP (Dynamic Host Configuration Protocol) is used to provide a usable IP to the workstation which the workstation uses to identify itself on the network to communicate with other devices on the network. As part of the initial DHCP response, a TFTP (Trivial File Transfer Protocol) server is identified that the diskless workstation uses to get initial configuration from in order to get the AoE target information to start the streaming process. As part of the subsequent Windows boot process, another DHCP interchange takes place to get an IP address for accessing network resources as part of normal user computing.

The underlying physical storage containing copies of the uniform image on many virtual blades is on the RAID (Redundant Array of Independent Disks) storage of a core server. Serving directly from a spinning drives is not as fast as serving images from RAM. Vmtouch application is used to move the VHD from the spinning drives to RAM to enhance performance related with streaming snippets of the image to the requesting diskless workstations. Figure 2 summarizes the protocols and tools used in the proposed solution.

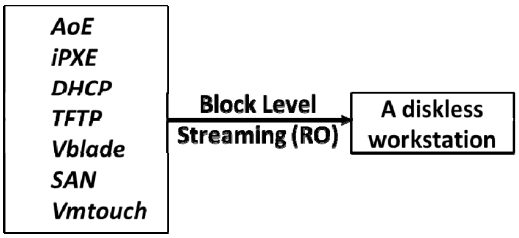

Fig. 2. A summary of Protocols and Tools Used

VI. Block Level Streaming Protocol

At the heart of the proposed solution is the AoE protocol which is used for the block level streaming of the snippets of Windows OS and applications to a diskless workstation. This section is dedicated to dig deeper into the AoE protocol. AoE protocol was created to implement a SAN solution. The protocol is simple but, at the same time, its simplicity is considered to be its weakness from security point of view. To mitigate the risks associated with the protocol's weakness, it is recommended to use Virtual Local Area Networks (VLANs) to segregate traffic [3].

There are two classes of messages pertaining to AoE protocol: ATA and Config/Query. The two classes of messages share a common 24-byte header. This common header is shown in Figure 3.

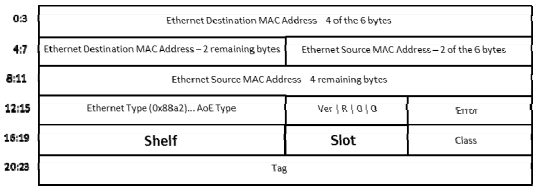

Fig. 3. 24-Byte Common Header for AoE Messages

As illustrated in Figure 3, the common header identifies the source and destination devices involved in the streaming. The header also identifies the target on virtual blade located in a specific slot on a specific shelf. Furthermore, the common header identifies the class of a message and includes any error information.

If the class of a message is ATA, the next 16 bytes of the AoE header contain information regarding whether it is a read or write operation involving data transfer, or no data transfer is involved (solely informational). Figure 4 shows the remaining 16 bytes of the AoE header for an ATA class of message.

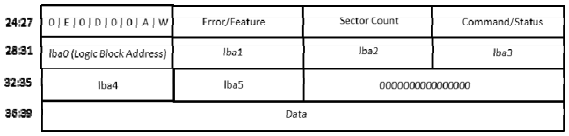

Fig. 4. Last 16 Bytes of AoE Header for ATA Message

If the class of a message is Config/Query, the next 12 bytes of the AoE header contain information regarding Config/Query message via a configuration string consisting no more than 1024 bytes. Figure 5 shows the remaining 12 bytes of the AoE header for Config/Query class of AoE header.

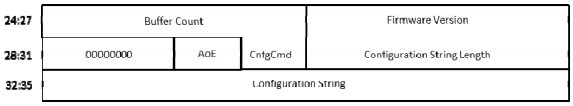

Fig. 5. Last 12 Bytes of AoE Header for Config/Query Message

VII. Performance and Security considerations

Based on comparative tests performed on AoE and other network block device protocols, it is important to enable and allow jumbo Ethernet frames throughout the network from the core to the clients [4]. This conclusion is not limited to AoE protocol. Jumbo frames enhance performance for iSCSI protocol as well [5,6].

It has been shown that AoE protocol's simplicity is its weakness [7]. At the same time, it has been shown that using VLANs to segregate traffic mitigates this risk. Furthermore,

inherently, the protocol is nonroutable, however, according to researchers this is disputed [8]. Again, this is not an issue in the proposed solution since the use of VLANs makes the protocol nonroutable.

VIII. Optimizing AoE Based Environment

The proposed solution uses Windows 7 OS for serving a diskless workstation. Different tests done on the installation of a Windows 7 system reveal that about 0.7% of the OS is needed to boot and 4% of the sectors are required for the log in process [9]. Based on this, the proposed solution streams snippets of the image, containing OS and applications, initially at the boot time. As the session progresses on a diskless workstation, additional snippets required by the workstation are streamed. Since, there is no disk on a diskless workstation, part of the RAM is set aside to act as the local disk.

As mentioned earlier, jumbo frames must be allowed throughout the network involving all VLANs dedicated for AoE implementation. Specifically, for the proposed solution, MTU (Maximum Transmission Unit) of 9000 was used on all NICs (Network Interface Cards) and network switches. It was found out that setting MTU to 9000 necessitated setting of the block size in TFTP configuration file to 8192 to avoid fragmentation.

IX. Network Considerations

Needless to say, network is heavily utilized in streaming of large amounts of data. In the proposed implementation, there are more than 450 workstations requesting snippets of OS and applications from a core of servers through the campus network. A network administrator needs to keep an eye on the network utilization and the reported delays to avoid a breakdown of the network. Ideally, a network should be used up to its available bandwidth without causing congestion. The safe range of network utilization is illustrated in Figure 6 [10].

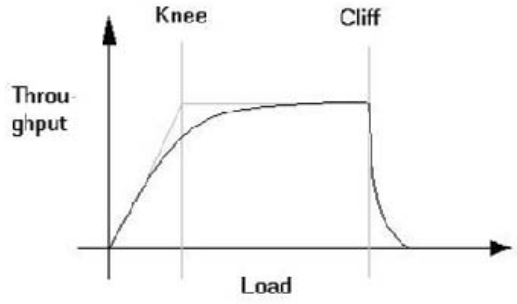

Fig. 6. Throughput as a Function of Load [10].

Network can be marred by four types of delays caused by variety of things on the network. a) Processing delays are introduced by switches, NICs and routers. It gets worse if the hardware and software of these devices are inefficient. b) Propagation delays are introduced by the traffic traversing the physical span between a core server and the clients. c) Queuing delays are caused by the time spent by the network packets in switches, NICs, or routers before these can be sent to the destination. d) Congestion or transmission delay is the time it takes for the network packets to traverse the network path given the existing traffic on the network path. Of course, it gets worse as the traffic increases [11,12,13].

Even though one can do detailed analyses on the four types of delays to know when to upgrade the network in terms of adding more uplinks or adding a switch at a certain location to relieve congestion, the formula below is one of the instructive ways to know whether a network link needs to be upgraded given its sustained utilization:

$$D = D_0/(1 - \mu) \quad (1)$$

where D is the delay at utilization of μ, and D_0 is the delay at 0% utilization. The above simple formula is helpful for network administrators, who are not normally hardcore computer scientists, to gauge the level of congestion on a network path. Since SNMP (Simple Network Management Protocol) based reporting tool easily provides utilization of a network link, a network administrator can use the above formula to put thresholds to generate alarms on utilization of links involving AoE VLANs to be aware of the troubled network links. The thresholds can be set at any point between 61% utilization (a degradation of 1.56) and 81% utilization (a degradation of 4.26) based on the following graph [2]:

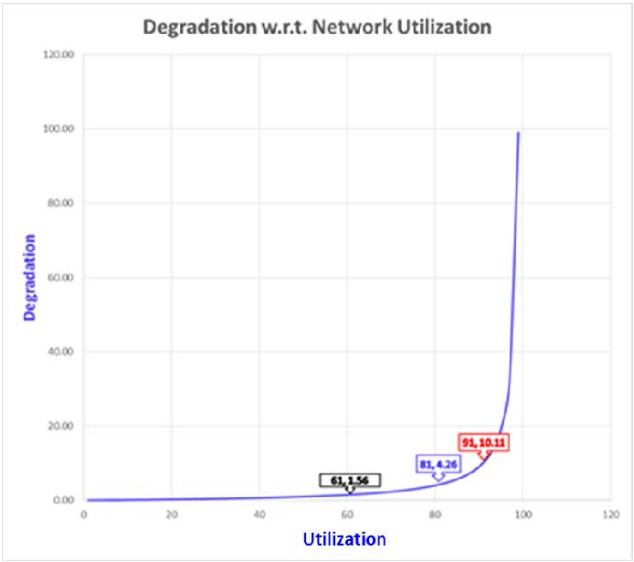

Fig. 7. Degradation as a Function of Utilization [2]

X. Implementation of the Proposed Solution

At the core, for serving images, four servers were deployed, two servers in each of the two geographically dispersed redundant data centers on campus. Each server had four CPUs each having 8 cores sharing 630 GB of RAM and 1.8 TB of RAID 5 spinning drives. Each server had two NICs each having two SFP ports each equipped with a 10 Gbps singlemode fiber transceiver. For redundancy, two transceivers

were connected to the core switch of Data Center #1 (DC #1), and two transceivers to the core switch of Data Center #2 (DC #2). Besides redundancy, this configuration provided increased bandwidth capacity between the core and its clients. In case, one of the data centers goes offline, mostly due to electric power issues, the system still works but with lower bandwidth capacity besides fewer servers being online.

The two redundant data centers were on a campus wide fiber ring. Additional network upgrade was done for the implementation to provide each building with access to the fiber ring (two ways) for redundancy. All uplinks to the core switches from each building and servers were 10 Gbps using singlemode fiber transceivers.

The access layer of the network was based on 48-port switches each having two 10 Gbps signlemode fiber transceivers for connecting to the two core switches, one in DC #1 and another in DC #2, for redundancy. Figure 8 shows the schematic of the campus network for AoE network.

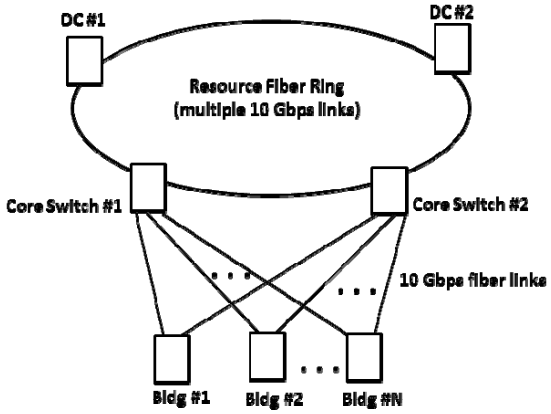

Fig. 8. Campus Network for AoE Network

As indicated by the above architecture, intermediate core switches were avoided to minimize delays. Basically, the design used in the implementation can be expressed by four attributes: two egress points, maximum of two hops to a network core switch, two taps on the campus wide fiber optic ring, and double star network topology.

As will be explained in the result section of the paper, certain aspects of the network design, specifically, 10 Gbps uplink per 48 ports, were determined by a year long testing of campus computer labs before rolling out the implementation. The implementation was tested with 1 Gbps uplinks as well, which produced unacceptable response at the workstations. Also, intuitively, this would make sense since each workstation has 1 Gbps to its network switch which is connecting 47 other computers. Therefore, the uplink needs to be a lot more than 1 Gbps. Given the technology at the time, employing two 10 Gbps uplinks for each 48-port switch was found to be an acceptable solution for providing the required bandwidth and redundancy.

On the client side, several models of computers existed. Newer computers with Intel Core i7 had 24 GB of RAM, and older computers with Intel Core 2 Duo had 8 GB of RAM with a minority of computers having only 4 GB of RAM. All the computers with 10/100 Mbps NICs were upgraded to have 1 Gbps of NICs which was required to get acceptable performance. Of course, the performance of the computers with more RAM was superior for running packages such as AutoCAD. It was found that a computer with 4GB of RAM worked for general applications with no issues. However, one required to have at least 8 GB of RAM to have acceptable performance for packages such as AutoCAD.

XI. Testing of the Design

Images tested for the implementation ranged from 30 GB to more than 150 GB. The evolving image was a result of starting from an image with general applications and incrementally including large specialized packages such as AutoCAD. Depending on making certain packages available at the startup, the boot time and RAM usage varied. The boot time was understood to be the time elapsed from powering a diskless workstation to getting the Windows logon screen.

In the figures presented in this section, it is important to keep the above in mind that with the evolving image different packages were made available at the startup. The goal was to keep the boot time under 150 seconds.

Another goal in the implementation was to keep the RAM usage during the boot time to 2 GB, so that the workstations with 4 GB can have 2 GB as a disk cache during a user session. This worked well for general applications, however, for applications such as AutoCAD this was very limiting. As mentioned earlier, one needs to have at least 8 GB of RAM to run packages such as AutoCAD with acceptable performance. For workstations with 8GB of RAM or more, 4 GB were set aside for initial boot and rest as disk cache.

Figure 9 shows the RAM usage during a boot process. The repeated VHD size mentioned along the x-axis is not a mistake. It simply shows, that different packages were included in the startup for the same image in order to keep the initial RAM usage to a minimum. The peak at 137GB image is not an anomaly in the sense that it simply represents that the number of packages included in the startup caused RAM usage to exceed 2GB. The number of packages included in the startup was then reduced for subsequent images which brought the RAM usage to below 2GB.

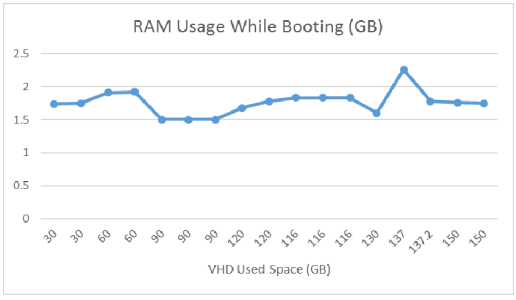

Fig. 9. RAM Usage [2]

Figure 10 shows the boot times for different images.

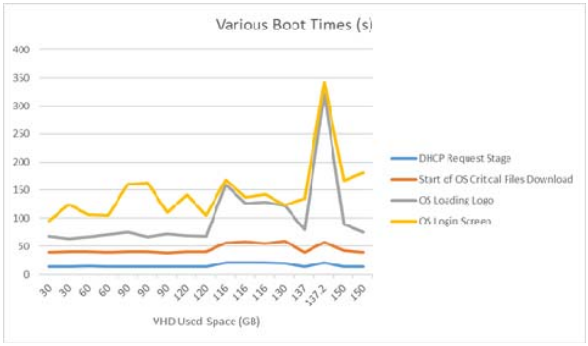

Fig. 10. Boot Times [2]

To see the performance on the server side, Figure 11 shows the CPU core usage for each virtual blade tasked with booting one client after another.

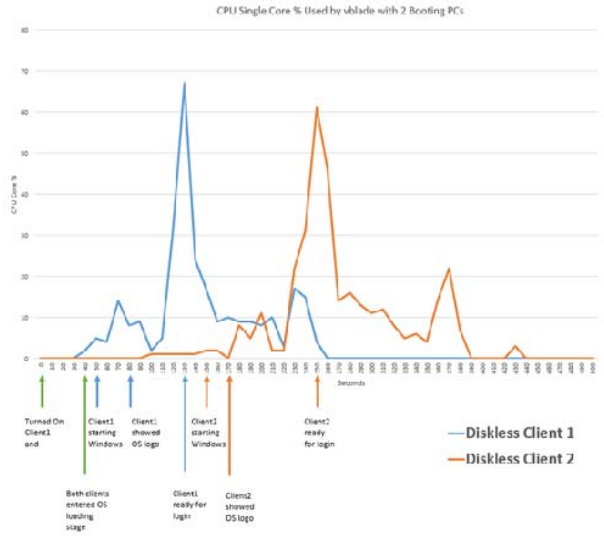

Fig. 11. CPU Core Usage for One Virtual Blade [2]

One can conclude from Figure 11 that the worst time for such an implementation is the simultaneous reboot of computers across campus, for example, when electric power is restored after a campus wide blackout. This is confirmed by Figure 12 which shows boot times exceed 5 minutes just for rebooting 15 workstations (from one virtual blade).

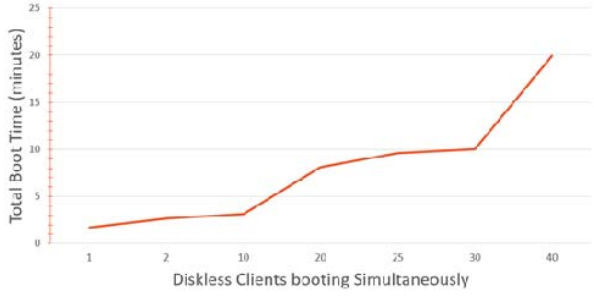

Fig. 12. Simultaneous Client Reboot per Virtual Blade [2]

The level of acceptability of the performance is highly subjective. Ideally, one would want a computer workstation to be available instantly under any circumstance including recovering from a campus wide electric power blackout. However, in this implementation the administration accepted 5 minutes of boot time under the situation of recovering from an electric power blackout. This called for limiting 13 workstations per virtual blade (C_B).

With the above benchmark in mind, other parameters listed in Section IV were tested determined. The following table tabulates the maximum allowed values of other resource parameters.

TABLE I. MAXIMUM RECOMMWENDED VALUES

Description	Maximum Value
C_B, Clients or workstations per virtual blade	13
B_{VLAN}, Virtual blades per VLAN	4
V_{SI}, VLANs per interface of a server	2
C_{VLAN}, Workstations per VLAN	45
C_{SI}, Workstations per server interface	90
C_S, Workstations per server	180

Based on the above table, and given the networking and CPU technology at the time implementing the proposed solution, the proposed design can handle up to 720 workstations, using the four deployed servers, while guaranteeing a maximum boot time of 5 minutes after recovering from a campus wide electric power outage.

XII. Conclusion

The proposed design was implemented campus wide for about two years before the university merged with another institution and new administration took over the reins of governing the institution. The implementation worked very well meeting all its goals while providing original services at 50% of the original IT workforce and drastic budget cuts.

The advantages of such an implementation were observed to be as follows:

a) Uniform software availability across campus
b) Fewer field technicians needed due to less maintenance
c) Relatively secured computing due to RO image
d) Single computer image to manage and maintain
e) Encryption not required (no hard drive)

The disadvantages were known at the time of implementation, and were confirmed to be as follows:

a) Network needs to be upgraded to 10 Gbps
b) High-end servers needed
c) Client computer's RAM ≥ 8 GB
d) Highly skilled IT staff needed
e) Close coordination between network and server groups needed

Technology improves with time, therefore, the first three disadvantages will vanish over time. However, the last two hurdles listed above have proven to be daunting. While this implementation was done, fortunately, the network and server groups were highly skilled and reported to one supervisor with technical knowledge of networks and servers, which made it possible to successfully implement the proposed solution. Any misconfiguration on the server, the network, or the client will have dire consequences.

ACKNOWLEDGMENT

The authors are grateful to the administration of The University of Texas at Brownsville for funding and allowing the implementation of the proposed solution campus wide.

REFERENCES

[1] The University of Texas System, UTB Transition, https://utsystem.edu/utb-transition/faq, 2012.

[2] Max D. Torres, An Alternative Approach to Serve a Large Number of Computer Users Using Block-Level Streaming, Master's Thesis, 2016.

[3] B. Coile, S. Hopkins, The ATA Over Ethernet Protocol, The Brantley Coile Company, Inc., 2009.

[4] J. Aatrokoski, ATA over Ethernet and Network Block Device performance tests, Aalto University MRO PC-EVN Development and Tests, 2007.

[5] D. Murray, T. Koziniec, K. Lee, M. Dixon, Large MTUs and internet performance, 13th IEEE International Conference on High Performance Switching and Routing (HPSR), 2012.

[6] C. He, W. Rao, Modeling and Performance Evaluation of the AoE Protocol, 2009 International Conference on Multimedia Information Networking and Security, 2009.

[7] C. Purvis, M. Marquis-Boire, Access over Ethernet: Insecurities in AoE, security-assessment.com, 2006.

[8] M. Landowski, P. Curran, AoE storage protocol over MPLS network, MSST '11 Proceedings of the 2011 IEEE 27th Symposium on Mass Storage Systems and Technologies, 2011.

[9] D. Clerc, L. Garces-Erice, S. Rooney, OS Streaming Deployment, IBM Research, Zurich Laboratory, International Performance Computing and Communications Conference, 2010.

[10] H. Jasem, Z. Zukarnain, M. Othman, S. Subramaniam, Evaluation Study for Delay and Link Utilization with the New-Additive Increase Multiplicative Decrease Congestion Avoidance and Control Algorithm, Scientific Research and Essays Vol. 5, 2010.

[11] A. Subramanian, J. Curtis, E. Pasiliao, J. Shea, W. Dixon, Continuous Congestion Control for Differentiated-Services Networks, IEEE 51st Annual Conference on Decision and Control (CDC), 2012

[12] M. Maia, M. Rocha, I. Cunha, J. Almeida, S. Campos, Network bandwidth requirements for optimized streaming media transmission to interactive users, WebMedia '06 Proceedings of the 12th Brazilian Symposium on Multimedia and the web, 2006

[13] A. Hassidim, D. Raz, M. Segalov, A. Shaqed, Network Utilization: the Flow View, 2013 Proceedings IEEE INFOCOM, 2013.