Proceedings of The National Conference
On Undergraduate Research (NCUR) 2018
University of Central Oklahoma
Edmond, Oklahoma
April 5-7, 2018

Computational and Network Utilization for Virtual Training Using Thin Clients

Chandler Lattin
Institute for Simulation and Training
University of Central Florida
3100 Technology Parkway
Orlando, Florida 32826 USA

Faculty Advisor: Dr. Glenn A. Martin

Abstract

Typically, training using virtual environments uses a client-server or a fully distributed approach. In either arrangement, the clients used are full computers (PCs) with an adequate processor, memory, and graphics capability. These are reasonably costly, require maintenance, and have security concerns. In the office desktop environment, the use of thin clients is well known; however, the application of thin clients with cloud-based servers to virtual training is relatively new. Thin clients require less initial cost, require less setup and maintenance, and centralize the virtual environment configuration, maintenance, and security to virtual cloud servers. Rather than housing an expensive computer (a so-called thick client) at each station, functionality is replaced using a streaming protocol, a remote server, and a thin client to allow the user to interact. This paper reviews two game-focused streaming protocols running across a set of four thin clients (of various capability and cost) from both local and remote cloud-based data centers. Data were gathered to measure latency and network and computational utilization across each client using two scenarios in both local and remote conditions. Results of these experiments indicate that thin clients for use in virtual training is viable regardless of local or remote server location.

Keywords: Thin Clients, Virtual Training, Cloud Computing

1. Introduction

Virtual training refers to the use of an interactive, computer-generated world where a user learns and practices a task (or tasks). Current virtual training systems use one of two methods: a server-client arrangement (most game-based systems) or a fully-distributed approach (where no single computer maintains the main copy of data, but rather a copy of data exists on each node). In either approach, the client stations run applications that are fairly intensive in terms of computational and graphical load. Of course, this requires that computers be used that can adequately support those applications. Virtual Battlespace 3 (VBS3), a game-based training application used by the U.S. military, recommends a computer with an Intel Core i5-2300 or AMD Phenom II 940 CPU, 8 GB RAM, and an Nvidia GeForce GTX 560 or AMD Radeon HD 7750 (with 1024 MB VRAM) 0. "Optimal" recommendations are even higher. Such a system may be reasonable for a home user or a single training station; however, the cost of these systems multiply quickly when installing multiple stations within a training center. These high-fidelity computers (also termed "thick clients") may also be supported in multiple (geographically-separated) training centers, making security and maintenance an additional cost and challenge.

Many training centers (such as those used within the U.S. military) are already investigating reducing overall hardware footprints by developing a new cloud-based infrastructure 0. This infrastructure would use a "software as a service" (SaaS) methodology using a service-oriented architecture. This approach can provide the simulations to less-

capable clients (so-called "thin clients"), thereby addressing weaknesses in cost, security, maintenance, and deployment.

As alluded to above, the use of a cloud-based approach begins to provide the infrastructure needed to reduce client hardware footprint (in terms of capability required, allowing the use of thin clients). In addition to providing the server side (in client-server game-based simulation), cloud servers could also provide the client side (host the user's game itself). In this case, the trainee (or operator) receives the view through some mechanism, and input from the trainee transmitted back. Doing so would centralize the significant computational resources and allow for easier deployment and maintenance, reducing cost and enhancing security. In addition, the clients in the cloud efficiently leverages the application of virtual machines, providing additional cost savings.

With client applications provided from the cloud, one open question is the presentation of the application to the user. Various "remote desktop" technologies already exist. Virtual Network Computing (VNC) 0 and Microsoft's Remote Desktop 0 are two well-known examples. In addition, companies such as Citrix and VMware have developed their own proprietary approaches.

1.1. Game-based Streaming

"Remote desktop" approaches provide a generic capability for display of any kind of content (whether game-based simulations or office applications such as word processing or spreadsheets). Other companies and groups have developed approaches focused on the remote display of *interactive, 3-D graphical* content, and others continue to do so. As a comparison to the typical desktop approaches, how might these graphical streaming protocols perform to achieve the needs for streaming content for virtual training? We initially investigated multiple protocols in a functional sense ("do they install and work?"). While developers continue to develop and leverage technologies such as HTML5 (through a system such as Guacamole) 0 and GamingAnywhere 0, these are not yet capable enough for game-based virtual training. However, Nvidia's GameStream 0 and Steam's In-Home Streaming 0 do appear promising and they quickly became the focus of this work.

GameStream uses H.264 encoding to send video from server to client. Nvidia has optimized the encoding in terms of both speed and size. As a part of this process, the server leverages the GPU; in fact, both Nvidia's GeForce GTX and GRID cards include special hardware for H.264 encoding/decoding. While not required, if the clients have an Nvidia GPU (even the mobile Tegra GPU), the decoding process uses it in order to increase its speed. Figure *1* shows the architecture used. Nvidia releases GameStream software that works directly on GeForce GTX hardware; unfortunately, the same software does not operate directly on Nvidia GRID cards although Nvidia does release Software Development Kits (SDKs) to allow developers to create their own software.

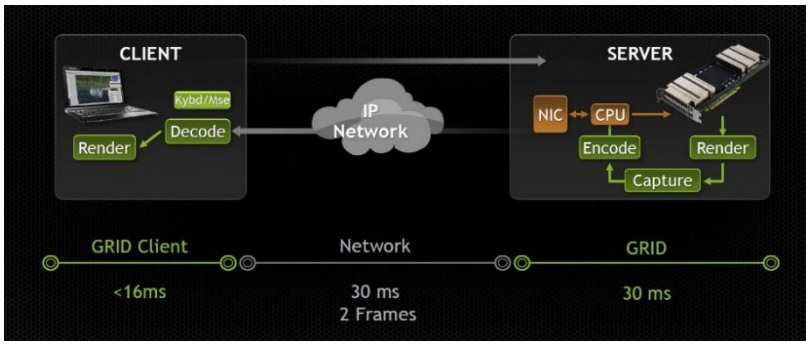

Figure 1. Nvidia GameStream architecture (from 0).

Steam In-Home Streaming uses a similar approach. This is apparent in its conceptual design diagram, shown in Figure *2*. One exception is that Steam In-Home Streaming, primarily designed for the home user, focuses on the local network. Therefore, servers and clients must be within the same network (e.g. broadcast domain), potentially requiring Virtual Private Network (VPN) capabilities if not. This becomes important when considering streaming from a cloud-based source as VPN may be required to make the two sides of the connection appear as a single network.

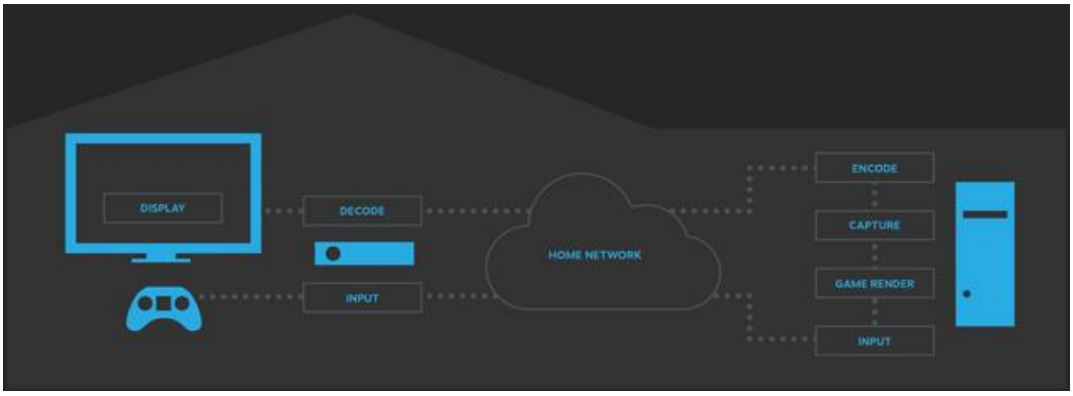

Figure 2. Steam In-Home Streaming architecture (from 0).

During initial functionality testing, the performance of both GameStream and Steam In-Home Streaming was evaluated through just qualitative user experience ("Does it feel good enough?"). These early anecdotal results show comparable performance between GameStream and Steam In-Home Streaming although GameStream seems slightly superior. In addition, open source clients that use GameStream have been developed, which allows the support of additional thin client hardware. For example, Moonlight 0 has a number of GameStream-compatible clients including ones for iOS, Android, and even the Raspberry Pi. A Java-based Chrome browser plug-in is also available.

1.2. Thin Clients

The streaming protocol (and capability) from the cloud server to a thin client is important, but no discussion of reducing the hardware footprint would be complete without some discussion of the client hardware itself. Some of the more commercially-supported approaches have specialized clients available. Nvidia's Shield product implements the client-side of GameStream, and Valve's SteamLink product similarly implements the client-side of Steam In-Home Streaming. As alluded to earlier, the open-source GameStream client, Moonlight, is available on multiple platforms including iOS, Android, and Raspberry Pi, and even runs within the Chrome browser using an extension 0. The latter allows any old desktop or laptop computer (regardless of operating system) to extend its lifespan by use as a thin client. This aspect may be particularly useful to many domains where a wide range of computers are available throughout.

2. Computational and Network Utilization

While the aforementioned game streaming protocols and thin clients appear promising, their computational and network utilization must be measured to determine if they can address the needs for expanding virtual training. To evaluate the client footprint, four very inexpensive thin clients were studied as test cases: the Nvidia Shield (~$200), the Valve Software SteamLink (~$50), the Dragon Touch X10 Android tablet (~$100), and the Raspberry Pi 3 (~$50).

The experiment used two scenarios within Unreal Tournament 4. First, the Weapons Training course was used as a 10-minute, fixed path scenario; second, a "death match" exercise was used as a 20-minute, random combat scenario. The four thin clients were each tested individually with both scenarios. The Nvidia Shield used its own proprietary GameStream software, the SteamLink ran its own proprietary Steam In-Home Streaming software, and the Dragon Touch X10 and Raspberry Pi each used the Moonlight client, which is an open-source application compatible with the GameStream protocol. With the experimental testbed defined, a review of computational load, network bandwidth and network latency was completed.

2.1. Computational Load

Given the limited capabilities of these thin clients, we studied the computational load within each thin client during each scenario in the experiment. The Nvidia Shield and the Dragon Touch X10 run the Android operating system, and the Valve SteamLink and the Raspberry Pi run Linux. Each operating system reports load on the central processing unit (CPU) differently, making a direct comparison difficult. However, we can review each in turn.

The Android-based clients reported data as a percentage of the CPU capacity actually used. Table *1* shows the results for those clients. The Nvidia Shield (using the GameStream software) performed very well and still had ample capacity for other tasks; conversely, the Dragon Touch X10 (Android Tablet using the Moonlight open source client) devoted more CPU to the streaming task although still did not reach maximum utilization. Both clients used less of the CPU for the "Death Match" condition than for the "Weapons Training" condition. In particular, the Dragon Touch X10 used significantly less. We have not yet explored the cause of these reductions, but note that both Android-based thin clients perform within their capacity under both experimental conditions.

Table 1. CPU load for Android-based clients

CPU Load	Weapons Training	Death Match
Nvidia Shield	24.000	20.173
DragonTouch X10	89.867	43.205

The Linux-base clients (Valve SteamLink and Raspberry Pi) report data as a value of CPU over/under utilization. Specifically, the value is an average number of processes executing or waiting in the queue over time (one second). Table 2 shows the results for those clients. The Raspberry Pi (using the Moonlight open source client) performed exceedingly well and reported very little load. Note that, despite being relatively inexpensive, the Pi has a quad-core processor, which is fairly capable. The SteamLink load is higher than desired (given that the SteamLink is believed to have a single-core processor); however, observations have shown that the SteamLink also maintains a load around 0.9 even when idle. Both clients had increased load under the "Death Match" condition (compared to the "Weapons Training" condition), which is contrary to the results seen in the Android-based clients. However, again both Linux-based thin clients perform within their capacity under both experimental conditions.

Table 2. CPU load for Linux-based clients

CPU Load	Weapons Training	Death Match
Valve SteamLink	2.161	2.459
Raspberry Pi	0.273	0.313

These results show the thin clients considered perform adequately for streaming game content for virtual training. While this is not surprising for the Nvidia Shield and Valve SteamLink (since they are designed and built for such streaming), this is an important result for the Dragon Touch X10 and Raspberry Pi, which are general-purpose devices being utilized as thin clients.

2.2. Network Bandwidth

All streaming protocols, of course, use some quantity of network bandwidth. As far as bandwidth, the requirements depend on screen resolution. For example, Nvidia's GameStream approach requires 10 Mbps, and it recommends 20 Mbps for 720p/60fps quality and 50 Mbps for 1080p/60fps quality.

Network capability will clearly be an important issue as virtual training moves to cloud-based approaches. Training centers may vary from very heavily used sites with ample network capability to only medium-duty sites with still only a 100 Mbps network. Careful consideration of network capability will be important, as it will drive potential location of cloud servers.

If we assume Nvidia's recommendations and the use of 1080p displays, each thin client will use up to 50 Mbps. Within a local location, this should be addressable. Even an older 100 Mbps network will support that requirement. However, if the servers are at a remote location, careful consideration is required as multiple clients may funnel through one or few links to the servers. Designing and implementing a network with an appropriate bandwidth (and topology) will allow servers and clients to transmit the data necessary.

2.2.1. local condition

To test bandwidth utilization, as discussed earlier, we used Unreal Tournament 4 and tested two scenarios, and tested in a local setting. Each scenario was run on the thin clients, each in their default settings. Results for the average bandwidth were as shown in Table 3.

Table 3. Network bandwidth in local condition (server on the local network)

Network Bandwidth				
Mbps	Weapons Training		Death Match	
	Transmit	*Receive*	*Transmit*	*Receive*
Nvidia Shield	0.145	35.313	0.152	37.200
Valve SteamLink	0.060	13.340	0.090	15.640
Raspberry Pi	0.056	10.988	0.065	12.096
DragonTouch X10	0.081	42.968	0.053	44.048

Obviously, the bandwidth for received data is significantly larger. Of note, both the SteamLink and the Raspberry Pi had significantly reduced bandwidth utilization due to the lower frame rate of the streamed video. However, this works very well for virtual training and should allow upwards of 8 clients within a 100 Mb/s network. The Nvidia Shield and Dragon Touch Android tablet results show the potential cost of increasing to 60 frames/sec; however, both devices support lowering down to 30 fps, which may be worth reducing for the potential bandwidth savings.

2.2.2. remote condition

Since the Steam In-Home Streaming software can work on rack-mount servers with GPUs (in this case, an Nvidia GRID card), we also performed the same bandwidth test by running the same two scenarios but within a virtual machine running on an Amazon Web Services (AWS) instance. Recall that Steam In-Home Streaming requires a VPN connection due to a limitation that server and client must be on the same Layer 2 network. The results were as shown in Table 4.

Table 4. Network bandwidth in remote condition (server on the remote network)

Network Bandwidth				
Mbps	Weapons Training		Death Match	
	Transmit	*Receive*	*Transmit*	*Receive*
Valve SteamLink (30fps)	0.098	15.181	0.116	16.441

While bandwidth utilization increased running from AWS, it is a very slight increase. Therefore, it certainly seems possible that a cloud-hosted training regimen could function and provide many enhancements for virtual training. Throughout both sets of bandwidth data, it is clear that the utilization is of an adequate level to allow streaming even on a 100 Mbps network and should work well within a limited network, provided that multiple installations "funnel" into larger connections (e.g. multiple 100 Mbps training centers connect to a future data center through a gigabit-capable or higher network).

2.3. Network Latency

For interactive virtual training, network latency is also an important issue. Depending on the source quoted, to provide a satisfactory experience for human use, the latency must be somewhere less than 60 ms 0 or less than 150 ms 0. We take the "tougher" 60 ms figure here. Of course, locally within a given facility it is a given to achieve latencies under this recommendation. In the remote data center scenario, we used Amazon as a model of cloud data centers. Amazon runs four main data centers within the United States. As an informal study on latency, we collected latency ping times from various colleges and universities (both small and large) across the nation (see Table 5).

Each campus had at least one Amazon region within the 60 ms latency threshold recommended. These campuses are fairly dispersed and of various sizes. While not a definitive result, we feel this shows it is likely capable to design a network that can meet the latency threshold needed.

Table 5. Ping Times of Various College and University Campuses to Amazon Regions.

Amazon Region	University of Central Florida (large, metropolitan university)	University of Maryland (large, metropolitan university)	University of Minnesota (large, metropolitan university)	Bowdoin College (small, liberal arts college)	St. Olaf College (small, liberal arts college)
US-East (Virginia)	32 ms	15 ms	30 ms	29 ms	37 ms
US-West (California)	110 ms	89 ms	87 ms	91 ms	61 ms
US-West (Oregon)	91 ms	94 ms	52 ms	85 ms	49 ms
Europe (Ireland)	119 ms	105 ms	113 ms	125 ms	111 ms
Europe (Frankfurt)	136 ms	108 ms	111 ms	91 ms	113 ms
Asia Pacific (Mumbai)	253 ms	297 ms	388 ms	300 ms	438 ms
Asia Pacific (Seoul)	200 ms	205 ms	217 ms	215 ms	175 ms
Asia Pacific (Singapore)	261 ms	328 ms	292 ms	282 ms	309 ms
Asia Pacific (Sydney)	226 ms	229 ms	272 ms	251 ms	280 ms
Asia Pacific (Tokyo)	197 ms	176 ms	195 ms	220 ms	142 ms
South America (Sao Paulo)	147 ms	351 ms	208 ms	186 ms	178 ms

Given the findings on bandwidth and latency, we have found that both requirements can be met. Bandwidth utilization of both GameStream and Steam In-Home Streaming are reasonable and sufficiently low, and a network that provides latency to regional data centers that is within the 50 ms recommended limit is possible. The numbers found here provide guidance on the design requirements of such a network. Special consideration should be given to local facilities and how multiple instances funnel into any such data center.

3. Conclusions and Future Work

The use of cloud-based systems and thin clients for virtual training addresses many concerns in cost, security, maintenance, and point of need. However, their use includes requirements in terms of networking and the thin clients. Streaming interactive, 3-D graphical content to thin clients requires those clients possess sufficient computational capability and requires the network to handle the bandwidth requirements with appropriate latency for consumption by users. We have explored two streaming protocols built for video games and verified the computational load, network latency and network bandwidth utilization across four representative thin clients. We parameterized the CPU and network needs and these metrics show the viability of cloud-based virtual training.

Unfortunately, due to the limitation that Nvidia's GameStream software does not directly work on their GRID cards, we could only test Valve's Steam In-Home Streaming (via the SteamLink thin client) in the remote condition (e.g. the server hosted at an AWS region). We are currently exploring options for hosting GameStream-compatible approaches

on AWS nodes. This includes Nvidia's Video Capture SDK. With such a capability, we will perform the experiment (both scenarios) for the remaining thin clients in the remote condition.

In addition, to this point we have focused on single displays per user. Some virtual training (such as vehicle simulators) require multiple displays for a single user. Would a single thin client or multiple clients drive these? If the latter, how would they be synchronized? Similarly, we have also focused on keyboard and mouse input. What issues arise from other peripherals such as steering wheels?

Finally, while our focus has been on virtual training, constructive training requires attention as well. While we hypothesize that any solution for virtual training should be sufficient for constructive training, such verification requires appropriate research and experimentation.

4. References

1.VBS3 Release Notes (Version 3.9.2). Obtained on January 18, 2017 from https://manuals.bisimulations.com/vbs3/3-9/manuals/#Release_Notes/Release_Notes.htm%3FTocPath%3DVBS3%2520Release%2520Notes%7C_____0.

2. Dumanoir, Paul, Mike Willoughby, Burt Grippin, Richard Crutchfield, Rob Wittman and Sean Barie. "Live Synthetic Training and Test & Evaluation Infrastructure Architecture (LS TTE IA) Prototype." Interservice/Industry Training, Simulation and Education Conference, Orlando, FL, 2015.

3. Virtual Network Computing (VNC). Obtained July 13, 2017 from https://en.wikipedia.org/wiki/Virtual_Network_Computing.

4. Microsoft Remote Desktop Connection (RDC). Obtained July 13, 2017 from https://www.microsoft.com/en-us/cloud-platform/desktop-virtualization?WT.srch=1&WT.mc_id=AID622874__SEM_inbb38GL&utm_source=Google&utm_medium=CPC&utm_term=microsoft%20remote%20desktop&utm_campaign=Enterprise_Mobility_Suite&gclid=EAIaIQobChMI06O02bWG1QIVj4KzCh1NYwxnEAAYAiAAEgI5DfD_BwE.

5. Apache Guacamole Manual. Obtained January 18, 2017 from http://guacamole.incubator.apache.org/doc/gug/.

6. Huang, Chun-Ying, Kuan-Ta Chen, De-Yu Chen, Hwai-Jung Hsu and Cheng-Hsin Hsu. "GamingAnywhere: The First Open Source Cloud Gaming System." ACM Transactions on Multimedia Computing, Communications and Applications, Vol 10, No. 1, January 2014.

7. GameStream. Nvidia, Inc. Obtained January 18, 2017 from http://shield.nvidia.com/game-stream?utm_campaign=Oct_sale&utm_medium=Owned&utm_source=nvidia.com.

8. Nvidia, Inc. "Cloud Gaming with Nvidia GRID Technologies." Game Developer's Conference, 2014.

9. Steam In-Home Streaming. Valve, Inc. Obtained January 18, 2017 from http://store.steampowered.com/streaming/.

10. Steam In-Home Streaming Architecture. Valve, Inc. Obtained July 6, 2017 from https://support.steampowered.com/kb_article.php?ref=3629-RIAV-1617.

11. Moonlight. Obtained January 18, 2017 from http://moonlight-stream.com/.

12. DCV Administration Guide. Nice Software. Obtained January 18, 2017 from http://www.nice-software.com/storage/nice-dcv/2016.0/docs/nice-dcv-guide-2016.0-16811.pdf.

13. "Blast Extreme Display Protocol in Horizon 7." VMWare, Inc. Obtained January 18, 2017 from https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-horizon-7-view-blast-extreme-display-protocol.pdf.

14. Moonlight Streaming. Obtained July 13, 2017 from http://moonlight-stream.com/.

15. Tolia, Niraj, David G. Andersen and M. Satyanarayanan. "Quantifying Interactive User Experience on Thin Clients." IEEE Computer, Vol. 39, No 3 (March 2006).

16. Nvidia Video Codec SDK. Retrieved on January 18, 2017 from https://developer.nvidia.com/nvidia-video-codec-sdk.