Thin-client based remote volume visualization over wide-area networks

Jerry Adams
University of Hawaii – West Oahu
jerrya6@hawaii.edu

Prasad Calyam, PhD.
University of Missouri - Columbia
calyamp@missouri.edu

Eliot Prokop
Washington University in St. Louis
ebprokop@gmail.com

Ronny Bazan Antequera
University of Missouri - Columbia
rcb553@mail.missouri.edu

Abstract

Remote Desktop Access (RDA) applications are becoming increasingly prevalent in distance learning, remote computing and data-intensive computing. RDA applications must have exceptional Quality of Experience (QoE) to provide end users with the best performance possible. A decision tree model uses context awareness, network and application feedback to improve the QoE. In this paper network and application feedback enhancements are encoding selection, server-side GPU acceleration and Software-defined networking by way of an OpenFlow controller. These enhancements are tested on a RDA application as a case study.

1. Introduction

Remote Desktop Access applications allow remote users to view and access computing resources over the internet. This provides users with a "at the server" experience as if the users are present at the server utilizing the resources locally. Due to this RDA applications need to have exceptional Quality of Experience to provide users with the best performance. The QoE in this case is interplay of the Quality of Application (QoA) and Quality of Service (QoS). The QoA refers to the performance of the application itself such as in this case the image compression, color density, image response time. QoS refers to the quality of the network connection, such as bandwidth, packet loss, delay and jitter.

In previous work [1] a decision tree model has been proposed which analyzes the three types of quality, QoE, QoA and QoS. This "3Q Model" makes decisions on the QoA which includes the client end, then on the QoS which includes both the network and server side. This paper expands upon the 3Q decision tree model and explores more techniques for optimizing the QoE.

In this paper the QoE improving techniques are dynamic Virtual Network Computing (VNC) protocol [2] encoding selection, affecting the QoA. Shared Graphics Processing Unit (GPU) Virtualization [3] and implementation of the OpenFlow protocol [4] are both ways to im affecting the QoS.

2. Related Work

Previous work has been done in regards to improving the QoA and QoS of applications. One such paper [5] uses a human-and-network-aware encoding selection scheme that considers both the condition of the network and the perceived performance of the application by the end user. The best encoding type is automatically selected based on this feedback. This paper uses the Remote Instrument Collaboration Environment (RICE) application as a case study to test their selection scheme and it shows that human-and-network-aware encoding selection scheme outperforms network-aware only schemes.

Another paper [6] describes application-driven improvements with a network-as-a-service overlay which include traffic engineering, traffic shaping and transit scheduling. These QoS improvements ensure predictable network performance. The results were validated in case studies that used four different data-intensive applications being accessed across a wide area network and in a network emulator.

As mentioned earlier a decision tree model has been proposed in a previous paper [1]. The decision tree analyzes the QoA and QoS in order to optimize the overall QoE. The model views the many variables of a RDA application and improves it in three sections; client, network and server. First improvements would be made are at the client side, where improvements are easiest to implement. Such improvements at the client are image compression, pixel encoding and prefetching images. Second area for improvements is the network. Improvements done to the network such as bandwidth allocation, patch switching and shortest cost routing. Using software-define networking technologies such as OpenFlow controllers allows for easier manipulation and configuration of the network. The last area for improvements in this model is the server. Server side improvements are the most difficult to improve upon as the resources are "static" in the sense that they least able to be changed. Some improvements here are resource scheduling or sharing, hybridization of the client-server resources and even physical hardware upgrades. The decision tree model also provides feedback in that the user or human perceived feedback determines the configuration of the application in order to improve QoE. Likewise the application feedback is used to modify and control the QoS variables to meet the QoA requirements. If QoA adequately meets the needs of the user, then their QoE is positively impacted. If the QoA does not meet their needs, then the QoE is negatively impacted. In summary it takes any context awareness and prescribes actions to maximize user QoE.

3. Problem

Though the 3Q decision tree model has already been shown to be effective in other data-intensive applications, there are still areas of improvement especially regarding the client and network parts. One way to improve the client part is through dynamic VNC encoding selection schemes. The VNC protocol is a popular choice for RDA applications as it is open source and being based on Remote Frame Buffer (RFB) protocol [7], it has many pixel encoding schemes to choose from. RDA applications need to select suitable encoding types to provide the best performance. Being able to dynamically selecting the best encoding scheme without any user input is important for providing the best QoE.

Another area for improving the model is at the network. Path switching, bandwidth allocation or rerouting is possible 'on the fly' through software-defined networking. Using the OpenFlow protocol with OpenFlow controller could provide this in RDA applications.

On the server side, Shared GPU Virtualization is a unique way to improve the QoE for RDA applications, especially ones that require powerful graphical performance. Effectively sharing the GPU resources to the end client would greatly improve performance of RDA applications.

Being able to improve the QoA and QoS through improvements like these are important to having an effective decision tree model and improving the overall QoS.

4. Methods

4.1. Remote Interactive Volume Visualization Infrastructure for Researchers (RIVVIR)

The University of Missouri (MU) in cooperation with Ohio State University (OSU) has developed a 'Remote Interactive Volume Visualization Infrastructure for Researchers' (RIVVIR) application that allows "health care" professionals to access virtual desktops (VDs) that host computationally intensive applications in a cloud platform. RIVVIR is a RDA application that allows interactive visualization of massive data sets such as MRI's that can be accessed over the internet with low powered thin-clients or mobile devices. This application is used as a case study to verify the 3Q decision tree model.

4.2. Dynamic VNC encoding selection scheme

The RIVVIR application uses the VNC protocol to connect an end user to the VNC server running at OSU. VNC uses the RFB protocol which allows for pixel encoding. Pixel encoding this case refers to the encoding of pixels that can be used to transport the image data in which VNC generates.

This paper tests eight encoding types: Tight, ZRLE, Zlib, ZlibHex, Ultra, Hextile, RRE and Raw. Each encoding type has the option to be encoded in four different color densities: Full Colors, 256 Colors, 64 Colors and 8 Colors. Every encoding type along with each of the four color densities are tested against different arbitrary network conditions such as available bandwidth, packet loss or delay. End users have many VNC clients to choose from, this paper uses one of the more popular clients, Ultra VNC [8]. Ultra VNC allows for easy selection of encoding schemes along with color density through a GUI or command line.

4.2. Shared GPU Virtualization

Shared GPU Virtualization is one of the more effective ways of sharing GPU resources remotely to multiple end users. Shared GPU virtualization works by allowing the VDI hypervisor to provide a layer of abstraction and lets the application behave as if it has its own physical dedicated GPU while the server's actual GPU and driver think it's responding to one master host. The hypervisor intercepts the API calls and translates commands from the application, passing them along to the graphics driver [9]. This allows for multiple virtual machines (VM) on the hypervisor to use a single GPU concurrently. This method of GPU virtualization is more effective than other methods such as GPU pass-through which passes the physical GPU to a single VM on the hypervisor. Obviously GPU pass-through is not effective for RDA applications that support multiple VM's such as the RIVVIR application.

GPU Virtualization on the RIVVIR application is unique in that it uses the VNCServer to act like a hypervisor in terms of creating an X display, which translates all graphics calls through the X protocol to the root X display and VNC comes in to transfer the display to user [10]. RIVVIR uses OpenGL and the VirtualGL library

4.3. OpenFlow Controller

Implementing the OpenFlow controller on the network between the remote client and the RDA application would allow for software-defining techniques such as path switching, bandwidth allocation and packet re-routing. There are several different OpenFlow controllers, the one that is used in this paper is Open Floodlight. Open Floodlight allows for modules that can be implemented to improve upon the controller. One such module is QoS which provides Quality of Service modules. This QoS is ideal for bandwidth allocation on end-to-end connections.

5. Implementation

Ultra VNC has the ability to automatically select encoding types but it only takes bandwidth into account not packet loss or the application thus it is not as accurate. This paper proposes to be able to likewise dynamically select encoding types but have the ability to take other aspects of the network into account such as packet loss or application. Preliminary decision rules have been made regarding the selection of encoding types. If the available bandwidth is greater than 1 Mbps then Hextile full colors can be used. Greater than 128 Kbps but less than 256 Kbps ZRLE 256 colors are used. Greater than 19 Kbps but less than 128 Kbps, Tight 64 colors can be used. And finally less than 5 Kbps but greater than 19 Kbps, Tight 8 colors is used. A generic algorithm has been developed to determine the best encoding type. It is similar to a tournament of elimination, encoding types competing against each other.

The implementation of the Shared GPU Virtualization involves performing tests on the hypervisor and capturing metrics. The tests will be done on the actual RIVVIR application infrastructure. The RIVVIR hypervisor server's configuration is as follows: 16 core Xeon E5530 2.4GHz, 72GB RAM, a Tesla C1060 which has 4 cards with 4GB video RAM each and Red Hat OS. The performance metrics to capture are VM consolidation, the number of VMs that can be supported concurrently on a server until CPU utilization is over 80% which is the threshold for optimal performance. Response time or the measurement of the VM's response time in seconds from when an image is fully loaded on the remote client. Remote frames per seconds (RFPS) number of remotely delivered frames per second that correspond to frame updates generated by 3D applications and videos.

Implementation of the Open Floodlight controller involves using network emulation, in this paper NISTnet emulation is used to simulate a network connection between a remote client and the RIVVIR application. The Open Floodlight controller is applied to the simulated network using the QoS module to allocate bandwidth arbitrarily.

6. Experimental Results

7. References

[1] Calyam, Prasad, Chris Dopuch, Sam Jonesi, Albert Lai, Brad Hittle, Alex Berryman, and Ashok Krishnamurthy. "Remote Desktop QoE Optimization Through Context Awareness". University of Missouri, 2013.

[2] T. Richardson, Q. Sta_ord-Fraser, et. al., Virtual Network Computing, IEEE Internet Computing, 1998.

[3] Spracklen, Lawrence. "VMware Horizon View 5.2 and Hardware Accelerated 3D Graphics." VMWare.com. VMware, inc. http://www.vmware.com/files/pdf/vmware-horizon-view-hardware-accelerated-3Dgraphics-performance-study.pdf. 2013.

[4] OpenFlow Protocol - https://www.opennetworking.org/sdn-resources/onf-specifications/openflow

[5] Calyam, Prasad, Abdul Kalash, Ashok Krishnamurthy, Gordon Renkes, and Chris Dopuch. Human-context and Packet Loss Awareness for Improving Encoding Selection in Virtual Network Computing. Ohio State University, 2013.

[6] Calyam, Prasad, Sripriya Seetharam, and Tsegereda Beyene. *ADON: Application-Driven Overlay Network-as-a-*

Service for Data-Intensive Science. University of Missouri, 2012.

[7] T. Richardson, The Remote Frame Buffer (RFB) Protocol, http://www.realvnc.com/docs/rfbproto.pdf, 2010.

[8] Ultra VNC - http://www.uvnc.com/

[9] Herrera, Alex. "NVIDIA GRID: Graphics Accelerated VDI with the Visual Performance of a Workstation." Nvidia Corp, 2014. Web.

[10] Brad Hittle. “Running Visualization Applications Remotely.” Ohio State University. PowerPoint. 2012