QoE Tuning for Remote Access of Interactive Volume Visualization Applications

Sam Jonesi, Jerry Adams, Samaikya Valluripally, Prasad Calyam, Brad Hittle, Albert Lai
University of Missouri-Columbia, University of Hawaii-West Oahu, Washington University in St. Louis, Ohio Supercomputer Center, USA
Email: sgjkbd@mail.missouri.edu, jerrya6@hawaii.edu, svbqb@mail.missouri.edu, calyamp@missouri.edu, bhittle@osc.edu, amlai@wustl.edu

Abstract—Remote access of interactive volume visualizations such as e.g., remote MRI (magnetic resonance image) viewing and inspection with three-dimensional images is important in smart health care applications. However, due to the large scale of data sets involved in the computation and various network/system factors (i.e., network bandwidth, CPU/GPU), delivering satisfactory user Quality of Experience (QoE) for remote access is quite challenging. In this paper, we investigate tuning of user QoE based on controllable parameters such as data transmission rate i.e., the client-side encoding scheme selection, and the computation resource scale i.e., the GPU server hardware size/number. The novelty of our studies is in the joint use of a “network-aware encoding scheme” on the client-side along with an “encoding-aware server scaling” on the server-side to guide efficient tuning decisions within a remote access system. We also describe a ‘Remote Interactive Volume Visualization System’ (RIVVS) case study and analyze utility functions (e.g., bandwidth consumption, GPU utilization) that guide the design of a tournament scheme for subjective testing in an application-specific context. Our RIVVS tested results with human subjects show that our approach can help in efficient tuning of remote MRI access configurations with satisfactory user QoE for: (a) good-to-poor network health conditions, (b) low-to-high remote access user workloads involving a diverse set of thin clients such as personal computers, smart phones and tablets.

Keywords—QoE Tuning, Remote Desktop Applications, Encoding Selection, GPU Scalability

I. INTRODUCTION

Remote Desktop Access (RDA) in the class of Virtual Network Computing (VNC) [1] (e.g., UltraVNC, TurboVNC, RealVNC) allows remote viewing and control of computer resources through the Internet. It is particularly relevant in smart health applications in fields of medical imaging and interactive volume visualization that involve storage and computational analysis of large data sets (i.e., gigabyte to petabyte sized data sets). Users of such smart health applications require virtual desktop and high-performance computing (HPC) environments involving e.g., high-speed networking, GPU resources and mass storage. They also demand the remote access to have high levels of Quality of Experience (QoE) during visual interpretation and visualization steering of the large data sets. Moreover, with multiple users simultaneously accessing the data sets with a variety of client devices (e.g., laptops, handhelds), additional challenges emerge to manage the Quality of Service (QoS) of the virtual desktop and HPC resources.

This material is based upon work supported by the National Science Foundation under Award Numbers: CNS-1347889 and CNS-1359125. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Fig. 1: Illustration showing how QoE is affected by effects of QoA and QoS

Figure 1 shows the impact of QoS at the RDA application back end that impacts the Quality of Application (QoA), which in turn affects user QoE. Bottlenecks arise when QoS factors such as insufficient bandwidth between the user and the back end, or inadequate compute resources (e.g., GPUs) at the back end impair the application frame rate, image quality and responsiveness in the visualization steering. These impairments ultimately affect the user’s subjective experience of the application that can be measured in terms of the commonly used Mean Opinion Score (MOS) user rankings [2] for a given application setting. The important characteristic in this interplay of QoS, QoA and QoE is that it is ‘application dependent’, and hence deriving utility functions can identify bottleneck cases and guide the resource control adaptation of user side and back end configurations to improve satisfaction of user QoE. The common controllable parameters are the data transmission rate at the user side i.e., the client-side encoding scheme selection, and the computation resource scale at the back end i.e., the GPU server hardware size/number.

Encoding refers to the compression of image data pixels by the schemes within RDA protocols (e.g., Remote Frame Buffer protocol in VNC [3]). The encoding schemes differ in their tradeoffs between image quality and the bandwidth consumption that affects QoA [4] e.g., an encoding scheme may sacrifice 10% color density for 50% less data needed to be transmitted on a network with poor QoS. Thus, the challenge to tune user QoE in non-ideal network conditions will be to select the most suitable encoding scheme amongst the available encoding schemes in a given RDA protocol. Similarly, non-ideal system conditions arise if the GPU resources to support the visualization tasks are incorrectly sized or if the number of end-users simultaneously connecting overloads the number of GPU server resources. Moreover, depending upon the devices used at the user ends, the GPU utilization/server can differ. In such cases, the satisfactory user QoE delivery challenge is to limit the number of users and combination of devices that should be scheduled for a given configuration of GPU resources being shared for remote access by multiple users.
In this paper, we investigate tuning of user QoE of a smart health application for remote MRI viewing and manipulation (e.g., view rotation or two-dimensional sliced views) using a RDA infrastructure. The tuning of user QoE is based on controllable parameters such as data transmission rate i.e., the client-side encoding scheme selection, and the computation resource scale i.e., the GPU server hardware size/number. The novelty of our studies is in the joint use of a “network-aware encoding scheme” on the client-side along with an “encoding-aware server scaling” on the server-side to guide efficient tuning decisions within a remote access system.

The “network-aware encoding scheme” uses objective measurements (e.g., bandwidth consumption, GPU utilization) and their corresponding utility functions for selecting the most suitable encoding scheme. Given a suitable encoding scheme, the “encoding-aware server scaling” limits the number of users and combination of devices to be scheduled on a per-server basis. The QoE decision process involves a tournament scheme that takes tradeoff choices’ guidance from the utility functions in order to reduce the number of testing iterations necessary in subjective QoE measurements data collection and analysis. Our tournament scheme approach builds upon concepts used in genetic algorithms [5] and is also broadly applicable i.e., its effectiveness is independent of the specific RDA protocol or GPU server hardware configuration for any given application context. We demonstrate the QoE tuning results for subjective testing tournaments with human subjects in a ‘Remote Interactive Volume Visualization System’ (RIVVS) case study with testbed configurations featuring: (a) good-to-poor network health conditions, and (b) low-to-high remote access user workloads involving a diverse set of thin clients such as personal computers, smart phones and tablets.

The remainder of the paper is organized as follows: Section II presents a background and lists related work. Section III describes the RIVVS case study and objective testing results to derive utility functions that guide the encoding selection and GPU server sizing adaptations for QoE tuning. Section IV details the performance evaluation results for subjective testing to efficiently tune user QoE under various network health and user workload conditions. Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

Remote access techniques for MRI viewing and manipulation have been studied since the 1990s [6], and there has been a need to revisit the approaches as the Internet has evolved with high-speed network connections as well as improvements in codecs. Authors in [7] used objective measurements to determine impact of choosing different gradient encoding schemes for MRIs. Virtual Network Computing with software packages such as UltraVNC [1] that support a variety of encoding schemes is often used in many medical applications, and has utility for purposes such as MRI viewing and manipulation. Particularly, the Tight encoding scheme as mentioned in [8] aims specifically on improving the remote access performance on low bandwidth networks with variable compression levels.

Alternately, if significant CPU resources are available, the Zlib encoding scheme which uses zlib library is preferable to compress raw pixel data [9]. There have also been efforts in works such as [10] and [11] to extend the basic virtual network computing architectures with custom-built infrastructure and web-portals for collaborative visualization of image data sets for both research and training purposes. The goal of our work is to build upon these earlier efforts to broadly characterize the interplay of the QoS, QoA and QoE within objective measurements for utility functions, and subjective measurements for user satisfaction for interactive volume visualization applications in smart healthcare.

Assessment approaches to compare various encoding schemes have been proposed in works such as [12], where objective techniques (no subjective measurements) are used to differentiate performance based on the correlation between the encoded input data and the corresponding output. The novelty of our work is in extending objective measurements data to apply the tournament scheme approach that can allow resource providers and end-users to efficiently tune their infrastructure configurations, both encoding selection as well as resource sizing, under non-ideal network and heavily-loaded system settings. Particularly, if there are multiple adaptation options in terms of encoding schemes or resource sizes, we show that our tournament scheme approach can be applied for rapid elimination of irrelevant adaptations and selection of the winner i.e., the best/optimal adaptation (see Section IV). We remark that our goal of this work does not involve modeling QoE for prediction purposes, however our data sets and analysis can be used in frameworks such as [13] that predict user QoE and adaption methods by accounting for variations in QoS factors.

III. CASE STUDY OF A SMART HEALTH APPLICATION

A. Smart Health Application

Figure 2 shows the RIVVS application that supports MRI scans’ visual inspection needs of a Small Animal Imaging Center. Typically, researchers from remote locations access the massive MRI data sets with typical sizes of $\approx 0.5$ GB.
and conduct interactive visualization tasks over the internet with low powered thin clients or mobile devices. The current RIVVS application thus allows for virtual collaborations without expensive data downloads/uploads, and pervasive access to high-performance computing resources such as GPU servers needed for the visual inspection and manipulation tasks. For the purpose of our study, we have setup a network emulator that can throttle the end-to-end bandwidth available between the back end and the user sides, when users are using Ultra VNC from a variety of devices (i.e., laptop, smartphone, tablet) on a remote desktop instance.

B. QoE Tuning Decision Approach

For the QoE tuning in the RIVVS application, we present an approach that leverages the interplay between QoS-QoA-QoE (3Q) metrics as shown in Figure 3. The scheme begins by checking whether the QoE is satisfactory. In cases where tuning is necessary, the tuning is first pursued through client-side adaptation mechanisms such as changing the encoding selection scheme that has the best-fit utility function and performs most satisfactorily. The chosen scheme wins a tournament amongst all candidate schemes in terms of image quality and responsiveness for good (Network 1), acceptable (Network 2) and poor (Network 3) network health conditions listed in Table I. We remark that Network 1 serves as a baseline for a wired network access at the remote user; Network 2 serves as a wireless access network profile at the remote user with near proximity to the access point; Network 3 serves as a wireless access network profile at the remote user with far proximity (e.g., >10 meters) from the access point. We term this step as the “network-aware encoding selection”, which is relatively less disruptive in the end-to-end RIVVS impact. In the event the client-side adaptation does not achieve the satisfactory QoE levels, server-side adaptation is pursued via an “encoding-aware server scaling” approach. The server-side adaptation mechanisms are relatively more disruptive in terms of the end-to-end RIVVS impact, and could involve changing the GPU hardware configuration (i.e., scale up or scale out) based on the ideal configuration for a given workload of diverse thin client types.

![Fig. 3: QoE tuning model based on client or server side adaptations](image)

C. Network-aware Encoding Selection

The Ultra VNC thin client used on the user side in this study supports selection of the following ten encoding types: Tight, ZRLE, Zlib, ZlibHex, Ultra, Hextile, ZYWRLE, CoRRE, 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. During the testing, for an encoding scheme selected, the default settings maintained were: Full Colors, CopyRect Encoding enabled, Cache Encoding enabled, Zip/Tight Compression of 6, and JPEG (Tight) Quality of 6.

![Fig. 4: Objective Trial-1 results on the Network-1 condition](image)

![Fig. 5: Objective Trial-2 results on the Network-2 condition](image)

![Fig. 6: Objective Trial-3 results on the Network-3 condition](image)

We modeled several candidate encoding schemes as part of the selection of the encoding schemes with best-fit utility functions under different network health conditions. Particularly, we modeled the bandwidth consumption metric for a given network connection’s available bandwidth and loss levels, and chose a winning best-fit utility function using bootstrap regression analysis. For instance, the encoding scheme ZRLE’s best-fit utility function is:

$$ utility = (0.364) \times (0.899)^L \times (1 - 41.636^{-1.0764 \times BW}) $$

Due to space constraints, we do not show the modeling details for other candidate encoding schemes. The objective testing tasks for these network conditions involved – measuring the amount of data transmitted across the

<table>
<thead>
<tr>
<th>Network</th>
<th>Latency</th>
<th>Average Download</th>
<th>Average Upload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network 1</td>
<td>40.00ms</td>
<td>38.63Mbps</td>
<td>5.07Mbps</td>
</tr>
<tr>
<td>Network 2</td>
<td>40.07ms</td>
<td>20.09Mbps</td>
<td>5.06Mbps</td>
</tr>
<tr>
<td>Network 3</td>
<td>45.67ms</td>
<td>0.73Mbps</td>
<td>2.12Mbps</td>
</tr>
</tbody>
</table>
connection while loading and displaying a rotating MRI for a 60 second period. The data transmitted to load the desktop view when opening the VNC connection was noted and then the MRI was loaded and a timer was started. At the end of the 60 seconds, the final amount of data transmitted across the connection was noted and the difference of the final amount and initial amount was the data needed to load and maintain the rotating MRI scan. We can see that the Tight encoding consumes the least bandwidth, and there are other schemes e.g., ZYWRLE, Ultra, ZLib, ZRLE that are comparable in terms of their low bandwidth consumption.

D. Encoding-aware Server Scaling

Assuming that the network-aware encoding selection scheme does not result in satisfactory user QoE as discussed in Figure 2, we apply the encoding-aware server scaling to adapt the server-side configuration using GPU Virtualization. GPU Virtualization allows for the sharing of a single physical GPU with multiple users. GPU Virtualization in the RIVVS is accomplished by virtualizing or brokering the physical GPUs using a 3D X Server as shown in Figure 2. The 3D X Server acts as the hypervisor, providing a layer of abstraction between the physical GPU and virtual desktops, or the X Proxy Displays as it is known in this environment. The X Proxy Displays and their applications behave as if they have their own dedicated GPU and the server’s physical GPU and driver think they are responding to one master host. 3D X Server intercepts the applications API calls and translates commands through the X protocol to the X Proxy display. The VNC Server then transfers the display to user and receives remote client keyboard and mouse events.

To create the utility functions for the GPU server sizing adaptations that deliver satisfactory user QoE, we measured the GPU memory utilization values obtained using the NVIDIA-SMI command input into the Command Line interface on the back end for a combination of workloads. Basically, we measured the GPU utilization for various combinations of thin clients to determine the scalability i.e., how many resources were still available for a given workload combination. To gather this data using objective testing, the GPU utilization was measured every second for 15 seconds and the resulting average was used. Table II shows the different workloads used for creating the utility function shown in Figure 7 for the GPU scalability. Note that each data point corresponds to a workload combination involving a unique set of thin client devices. The set of workloads considered for this experiment $W_L_1,...,W_L_{11}$ were configured based on the increasing order of GPU utilization %, with each workload representing a single device or a combination of laptop, smartphone and tablet devices. The configurations of these devices were as follows: $laptop_1$ (Windows 7/4GB RAM), $laptop_2$ (Windows7/8GB RAM), $laptop_3$ (OS X 10.9.1/4GB RAM), $smartphone$ is smartphone (Android/2GB RAM), $tablet$ (iOS 5.1.1/256MB RAM).

IV. PERFORMANCE EVALUATION

A. Evaluation for Encoding Scheme Selection

Although the encoding scheme utility functions in Section III-C can provide guidance on the amount of data transmitted under ideal and non-ideal network conditions, they do not indicate which encoding scheme selection actually provides the most satisfactory user QoE i.e., in terms of image rendering start time, responsiveness in manipulation and image quality. In addition, the objective testing results do not provide guidance on how should the encoding scheme selection change to tune the user QoE depending on the good, acceptable or poor network health conditions. Moreover, comparing every encoding scheme against each other results in $n^2$ comparisons ($n$ being the number of encoding schemes) that are impractical to assess in terms of user QoE. To circumvent these problems, we utilize the utility functions and propose a tournament scheme based subjective testing that efficiently: (a) reduces the comparisons to $\log(n)$ complexity by implicit comparisons, and (b) allows us to select a ‘winning’ encoding scheme that delivers most satisfactory user QoE for a given network health condition.

In this work, Four human subjects were recruited to participate in the subjective tournament trials to inspect and manipulate a mouse model MRI scan. We chose four subjects using guidance from [15] that at least four data points in subjective rankings are a minimum number to establish statistical significance. The Mean Opinion Score (MOS) ranking on a scale of 1 (Poor) to 5 (Excellent) was used to pick the winning encoding scheme in a tournament i.e., the scheme with higher MOS ranking was used to eliminate the inferior schemes in terms of user QoE satisfaction. To minimize user bias in selecting the winners, we provided MOS ranking training and randomized the encoding scheme match-ups. Under ideal network conditions with no network emulation, we found that ZRLE emerged as the ‘root’, which suggests it was the winner as shown in Figure 8. However, Figure 9 shows the result for Network-1 condition, and the winner was the Tight encoding scheme because it produced high responsiveness.

<table>
<thead>
<tr>
<th>Workload</th>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_{L_1}$</td>
<td>$laptop_3$</td>
</tr>
<tr>
<td>$W_{L_2}$</td>
<td>$smartphone$</td>
</tr>
<tr>
<td>$W_{L_3}$</td>
<td>$laptop_1$</td>
</tr>
<tr>
<td>$W_{L_4}$</td>
<td>$tablet$</td>
</tr>
<tr>
<td>$W_{L_5}$</td>
<td>$laptop_2$</td>
</tr>
<tr>
<td>$W_{L_6}$</td>
<td>$laptops_2,laptops_3,tablet,smartphone$</td>
</tr>
<tr>
<td>$W_{L_7}$</td>
<td>$laptops_1,laptops_2,laptops_3,tablet,smartphone$</td>
</tr>
<tr>
<td>$W_{L_8}$</td>
<td>$laptops_1,laptops_2,laptops_3,smartphone$</td>
</tr>
<tr>
<td>$W_{L_9}$</td>
<td>$laptops_1,laptops_2,tablet$</td>
</tr>
<tr>
<td>$W_{L_{10}}$</td>
<td>$laptops_1,laptops_2,tablet,smartphone$</td>
</tr>
<tr>
<td>$W_{L_{11}}$</td>
<td>$laptops_1,laptops_2,tablet,smartphone$</td>
</tr>
</tbody>
</table>

Fig. 7: GPU Utilization with multiple devices to determine the scalability of the system.
and low load time along with satisfactory image quality. Similarly, Figures 10 and 11 show the results for Network-2 and Network-3 conditions, and the winner again in both these tournaments was the Tight encoding scheme.

These results through the reduced testing iterations owing to the tournament scheme further confirm the evidence from the objective testing that – the Tight encoding scheme is relatively more suitable under non-ideal network conditions, because of its low bandwidth consumption rate and relatively highest user QoE delivery. Under exceptionally poor network conditions, even the Tight encoding scheme might not produce satisfactory user QoE and the QoE tuning might involve using a less complex color scheme to improve performance. The human subjects remarked that the impaired user QoE in remote MRI viewing occurs when the MRI data loses incredible amounts of detail to the point where significant features of the image are not recognizable, or even worse, when entire portions of the MRI essentially vanish and appear blank due to the impaired effects of the color scheme.

B. Evaluation for GPU Resource Sizing

The GPU scalability of the system was measured in two ways. First, a subjective QoE baseline for each client machine was established individually using the winner of the encoding scheme trials (which happened to be the Tight encoding scheme). Then, as more clients were added to the server, each client was tested to check whether there was a noticeable QoE degradation. Second, the GPU utilization for various combinations of thin clients was objectively measured to determine how many resources were still available. To gather this data the GPU utilization was measured every second for 15 seconds and the resulting average was used.

The utility functions for the GPU utilization % in Section III-D can provide guidance on the combination of thin client devices that are feasible to be supported on the back end in a scalable manner under low-to-high workloads. However, they do not indicate which workload combinations actually provide the most satisfactory user QoE i.e., in terms of image rendering start time, responsiveness in manipulation and image quality. In addition, the objective testing results do not provide guidance on how should GPU resource size or even the number of GPU resources be changed in order to tune the user QoE depending on the workload conditions. Given a desired user QoE level expectation, and assuming the expectations correspond to a goal of a desired GPU utilization % goal. To circumvent these problems, we leverage the utility functions and propose a tournament scheme based subjective testing that: (a) selects the most feasible workload combination that satisfies user QoE expectation and produces the highest scalability for a desired GPU utilization % goal, and (b) allows for selecting the right GPU resource size or the number of GPU resources that delivers most satisfactory user QoE for a given workload condition.

TABLE III: Feasible workload combinations involving thin client devices for a given set of GPU Utilization goals

<table>
<thead>
<tr>
<th>GPU Utilization Goal</th>
<th>Feasible Workloads</th>
<th>Winner (No. of devices)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;40%</td>
<td>W L1, W L2</td>
<td>W L2 (single device)</td>
</tr>
<tr>
<td>&lt;60%</td>
<td>W L3, W L4, W L5</td>
<td>W L5 (single device)</td>
</tr>
<tr>
<td>&lt;80%</td>
<td>W L6, W L10, W L11</td>
<td>W L11 (five devices)</td>
</tr>
</tbody>
</table>
Table III shows the multiple workload combinations that are feasible for different GPU utilization % goals of <40% (low), <60% (medium) and <80% (high) that meet corresponding levels of user QoE expectations. These RIVVS goals have been set based on empirical observations on the user QoE levels desired by the users, however the goals can be customized for different deployments depending upon the application-specific user QoE preferences. By using the tournament scheme in subjective testing based on these utility functions, we obtained the winner workload combination as shown in Figure 12 for the 80% GPU utilization goal. The winning workload combination is one that satisfies user QoE expectations and produces the highest scalability. Thus, owing to the tournament scheme, GPU resource sizing adaptations can be selected with limited physical testing of all the feasible workload combinations. Additionally, the tournament scheme allows for efficient selection of the most feasible workloads that can be scheduled on the server-side GPU resources in order to maximize scalability of the RIVVS, while also delivering satisfactory user QoE.

V. CONCLUSION AND FUTURE WORK

In this paper, we conducted objective and subjective performance characterization studies to enhance the user QoE in smart healthcare related interactive volume visualization applications over the Internet. Our purpose was to develop utility functions for remote desktop access with objective testing of controllable parameters such as: (a) thin client encoding scheme selection, and (b) GPU computation resource scale. We found that the utility functions inherently do not provide the needed guidance for QoE tuning, but create a large number of feasible options that need to be verified with excessive subjective testing with actual users.

Given that excessive subjective testing is prohibitively expensive in both cost and time considerations even for a modest number of feasible options in encoding scheme selection (i.e., could produce $n^2$ tests) or for low-to-high scalability goals, we proposed a novel scheme that leverages ‘network-aware encoding selection’ along with a ‘encoding-aware server scaling’ that feature a tournament scheme to select winning configurations amongst the large number of feasible options. Results from our tournament scheme for a ‘Remote Interactive Volume Visualization System’ (RIVVS) case study showed that it is efficient in providing concrete user QoE tuning guidance. Further, we found that the tournament scheme is generally applicable for cross-validation testing in other remote desktop application cases, where objective and subjective testing are synergistically used to cope with network health and workload variations that can impact user QoE.

Our future work is to enhance the utility functions and create models that allow for comprehensive modeling of the QoS, QoA and QoE factors. Such an effort in turn can guide more sophisticated QoE tuning adaptations in the cloud-scale delivery of interactive volume visualization applications.

REFERENCES