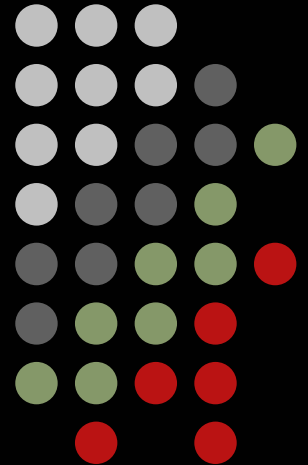


Event Distribution and Image Delivery for Cluster-based Remote Visualization

Dale Beermann
University of Virginia
Oct. 19 2003

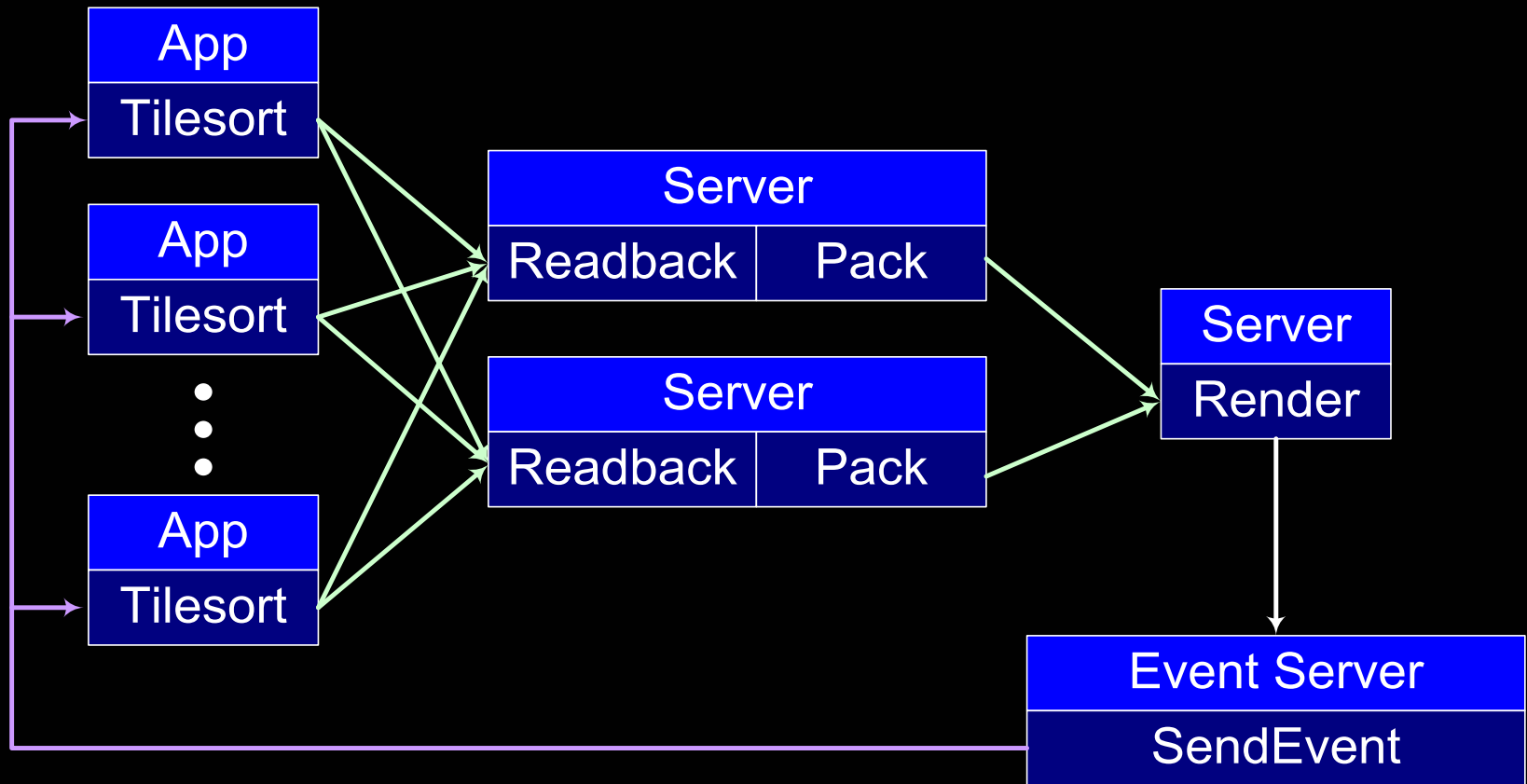




Motivations

- Graphics as a Remote Service
 - Enabling better use of available resources
 - Providing graphics to a variety of users
 - Transparent to the end user
 - Ability to adapt to the user's capabilities

Event Distribution





Event Distribution

- CRUT: An API for sending events
 - Completely abstracted from the rendering context
 - Responsible for configuration and sending events
- Client Library
 - Callback Registration
 - Main Loop / Event retrieval



Event Distribution

- Server side flexibility
 - Can be written using any toolkit
 - API takes care of distributing events
- Client side flexibility
 - Don't bind the application to using a main loop



Event Distribution

- Simple sampling technique to process only the most important events.





Image Delivery

- Allows use over low-bandwidth connections

200x200 Image

117 KB



1.4 KB





Image Delivery

- Determining maximum RMS Error

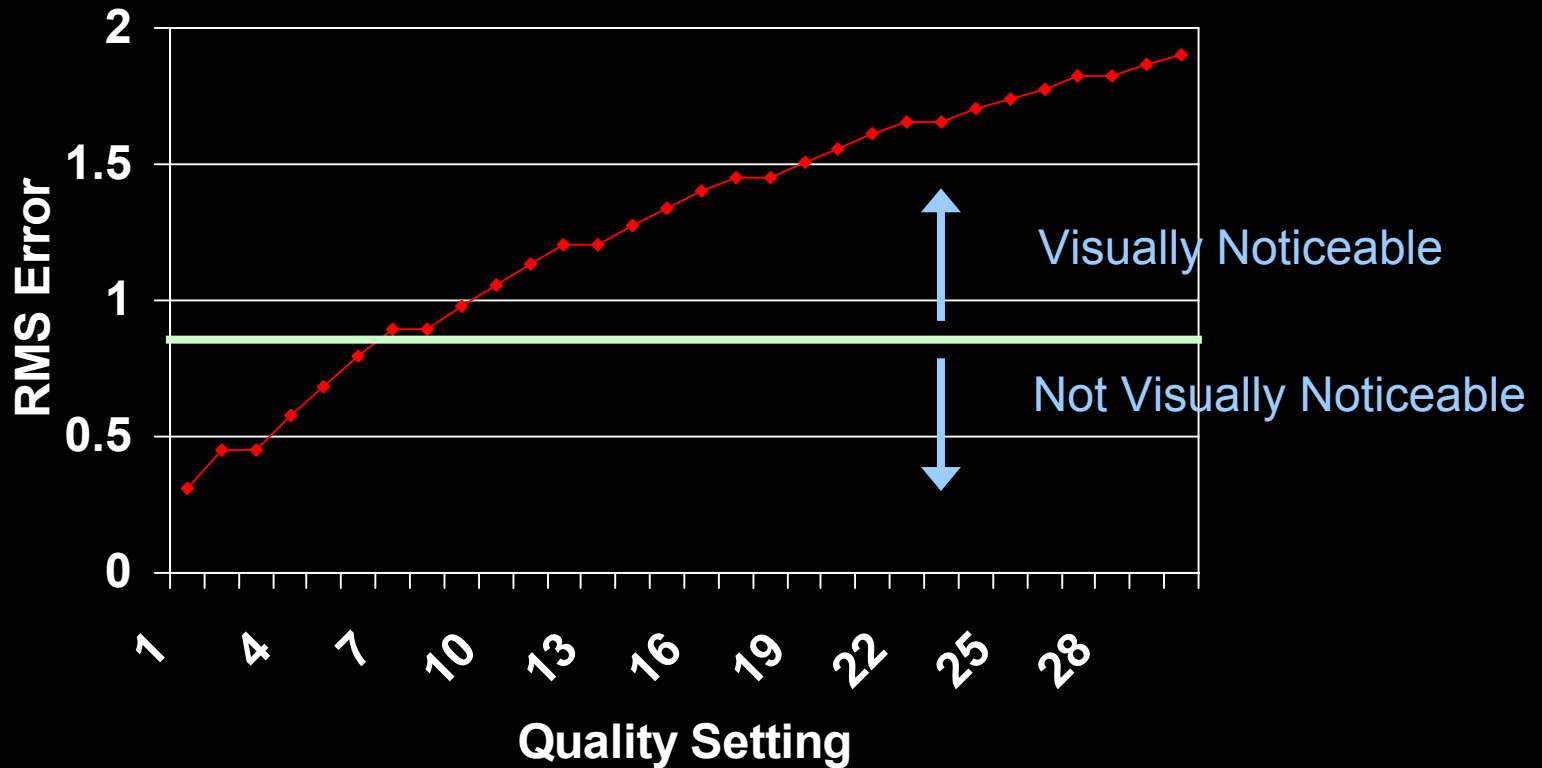


Image Delivery



- Choosing quality and GOP size

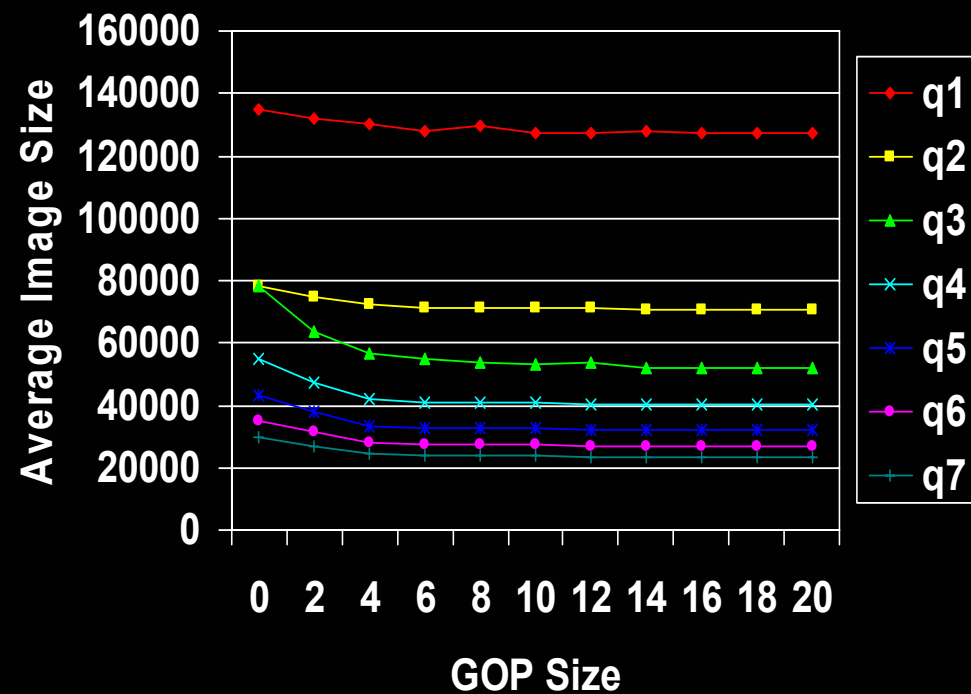
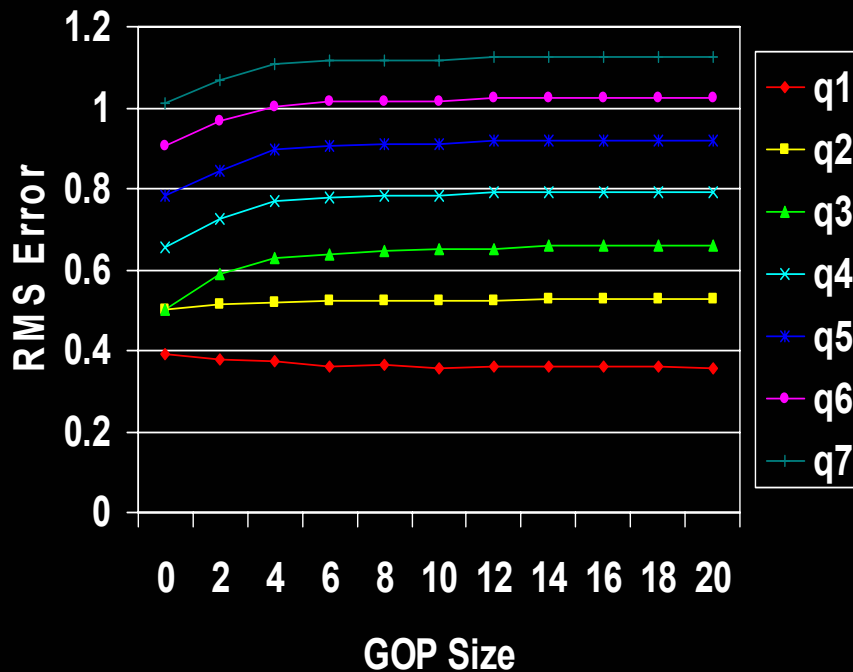
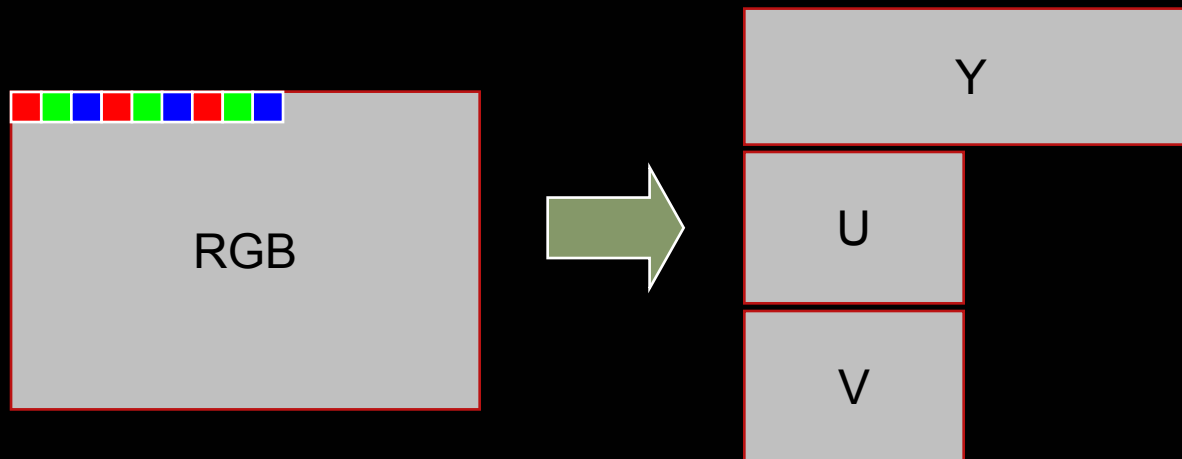




Image Delivery - Issues

- Framebuffer readback
- Conversion from RGB interleaved to YUV planar

$$\begin{aligned} Y &= ((66 * R + 129 * G + 25 * B + 128) \gg 8) + 16 \\ U &= ((-38 * R - 74 * G + 112 * B + 128) \gg 8) + 128 \\ V &= ((112 * R - 94 * G - 18 * B + 128) \gg 8) + 128 \end{aligned}$$





Other Compression Options

- Compression within the packer/unpacker
 - gzip/zlib or other compression library
 - Deering's geometry compression (SIGGRAPH '95)



The Missing Pieces

- Resource Sharing and Allocation
 - Allowing for multiple users with varying needs
- An Adaptable Graphics System
 - Adjust system parameters to best suit the needs of the users

Demo



Questions?

