Introduction

- Growth:
  - Networks – high bandwidth to the home
  - Thin clients – Remote Desktop, Google Desktop
  - Online games
- Opportunity:
  - Heavyweight, "fat" server hosting game
  - Stream game as interactive video over network
  - Played on a lightweight, thin client
- Motivation:
  - Rendering game that requires data and specialized hardware not at client
    - Sony Remote Play, OnLive
  - Augmented reality - physical world enhanced by thin, wearable computers (e.g., head-mounted displays)
  - Ease of implementation and maintenance

References


Outline

- Introduction (done)
- Games as a Service (next)
- Games as Video
- Game Video Performance
Why Games as a Service?

• Potential scalability
  – Overcome processing and storage limitations
• Cross-platform support
  – Can run games built for different platforms (e.g., Xbox and Playstation) on one device
• Piracy prevention
  – Since game code is stored in cloud, cannot be copied
• Click-to-play
  – Game can be run without installation

Cloud Game Modules (1 of 2)

• Input – receives control messages from players
• Game logic – manages game content
• Networking – exchanges data with server
• Rendering – renders game frames
• How do put in cloud?

Cloud Game Modules (2 of 2)

• Cuts
  1. All game logic on player, cloud only relay information (traditional network game)
  2. Player only gets input and displays frames (remote rendering)
  3. Player gets input and renders frames (local rendering)

Remote Rendering

• Cloud runs full, traditional game
• Captures video ("scrape" screen) and encode
• Client only needs capability to decode and play
  – Relatively minor requirements
• Bitrate requirements can be an issue

Remote Rendering

• e.g.,
  – Onlive (commercial)
  – Gaming Anywhere (research)
  – Cloud Saucer Shoot (teaching)
Local Rendering

• Instead of video frames, send display instructions
  – Potentially great bitrate savings
• Challenge for instruction set: able to represent all images for all games

Potential Distribution of Computing

• Partitioning coordinator if/when to migrate functionality (e.g., reduce cloud load and/or when terminal has greater capabilities)
  – Remote and Local rendering cases (above) are really just special cases
• Challenge: how to do so in general, how to synchronize if both cloud + terminal have module (e.g., “G”)

Outline

• Introduction (done)
• Games as a Service (done)
• Games as Video (next)
• Game Video Performance

Application Streams vs. Game Streams

• Traditional thin client applications (e.g., x-term, remote login shell):
  – Relatively casual interaction
    • e.g., typing or mouse clicking
  – Infrequent display updates
    • e.g., character updates or scrolling text
• Computer games:
  – Intense interaction
    • e.g., avatar movement and shooting
  – Frequently changing displays
    • e.g., 360 degree panning
Games as Streaming Video

- High bandwidth – push limits of graphics
  - Need efficient compression
- Adapting traditional video to network → motion and scene complexity crucial to maximize quality
  - High motion needs quality scaling
  - Low motion needs temporal scaling
  - Complex scenes limit ability to quality scale
  - Getting it “right” improves perceived quality by as much as 50%
- To more effectively stream games as video, need:
  1. Standard measures of motion and scene complexity
  2. Streaming game videos as benchmarks
  3. Understanding how current thin tech is limited

Game Perspectives

Motion

- 9 Videos varying motion/scene complexity
- Divide frame into 16 blocks
- User rated amount of motion (0, ¼, ½, ¾, 1)
- Results:
  - MPEG vector [12]:
  - PMES [9]:
  - Interpolated macroblocks [13]:
- Our measure:
  - Percentage of Forward/backward or Intracoded Macroblocks (PFIM) 0.95

Scene Complexity

- Same 9 Videos varying motion/scene complexity
- Divide frame into 16 blocks
- User rated complexity (0, ¼, ½, ¾, 1)
- Our measure:
  - Intracoded Block Size (IBS) 0.68
Select Games

<table>
<thead>
<tr>
<th>Perspective</th>
<th>Game</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>Battlefield 1942, Battlefield 2, Battlefield Vietnam, Doom 3, Medal of Honor, Allied Assault, Quake III Arena, Star Wars Battlefront</td>
</tr>
<tr>
<td>Third (Lin)</td>
<td>Fahrenheit, Guild Wars, Harry Potter Chamber of Secrets, The Incredibles, The Wonderful End of the World</td>
</tr>
<tr>
<td>Third (Isq)</td>
<td>Diablo II, Evil, Galactic Mule, Keilabiter, Lazarus, Pyramid Panic, Rainbow Reef, Wineman Sam</td>
</tr>
<tr>
<td>Omnipresent</td>
<td>Age of Empires 3, Age of Mythology, Battle for Middle Earth 2, Command and Conquer 3, Command and Conquer Generals, Company of Heroes, Star Wars Galactic Battlegrounds, Stronghold 2, Warcraft III</td>
</tr>
</tbody>
</table>

Capture Game Videos

- FRAPS (Direct X or OpenGL), 30 f/s
- PC Intel P4, 4.0 GHz, 512 MB RAM, nVidia Geforce 6800GT 256
  - After: MPEG compress using Berkeley MPEG Tools
- Resolution: 800x600 pixels
- Length: 30 seconds

Select Videos

- Widely used by multimedia community
- Range of motion and scene complexity
- Each 10 seconds long

Select Videos

<table>
<thead>
<tr>
<th>Video</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>Panning of a moving coastguard ship</td>
</tr>
<tr>
<td>Container</td>
<td>A container ship sailing slowly</td>
</tr>
<tr>
<td>Foreman</td>
<td>A close up of a talking head</td>
</tr>
<tr>
<td>Hall</td>
<td>An office hallway with some people</td>
</tr>
<tr>
<td>Mobile</td>
<td>Panning of moving toys</td>
</tr>
<tr>
<td>News</td>
<td>Two news reporters talking</td>
</tr>
<tr>
<td>Paris</td>
<td>Two people talking with gestures</td>
</tr>
<tr>
<td>Silent</td>
<td>A person demonstrating sign language</td>
</tr>
<tr>
<td>Vortex</td>
<td>Panning of a moving car</td>
</tr>
</tbody>
</table>

Motion and Scene Complexity

- MOTION
  - Games from .20 to .95
    - First (iso) highest (except side scroll)
    - Third (iso) lowest (except side scroll)
    - Omnipresent all medium
  - Videos all .70 to ~1

- SCENE COMPLEXITY
  - Games vary considerably across all genres
    - First (iso) least (may value responsiveness)
    - Omnipresent most (lots of detail for game play)
    - Third medium
  - Videos vary low to high but a bit less than highest omnipresent
Motion and Scene Complexity - Summary

Outline

- Introduction (done)
- Games as a Service (done)
- Games as Video (done)
- Game Video Performance (next)

What is OnLive and Why is it Important?

- Gaming in the cloud
- Thin client, no special hw requirements
  - PC, Mac, OnLive mini-console
- Game video streamed to client
- Importance:
  - Allows playing AAA games on simple devices
  - Provide access to legacy games on next-gen consoles without hardware compatibility

Goal of Study

- How does the magic of OnLive work?
  - “black box”
- Study network traffic turbulence of games on OnLive
  - Packet size
  - Inter-packet time
  - Overall bitrate up and down
  - Performance during loss & latency
- Controlled variation of network parameters
- Different genres of games
Unreal Tournament III (2007)  
First-person shooter

Batman: Arkham Asylum (2009)  
Third-person action-adventure

Grand Ages: Rome  
Real-time strategy, omnipresent (2009)

Experimental Set-Up
Design of Experiments

- All traffic measured UDP
- Varied capacity, loss and latency
- Parameters:
  - Capacity (down:up) 5:1 Mb/s, 10:2 Mb/s, and unrestricted
  - Latency (round-trip) 0, 40, and 70 ms
  - Loss (downstream) 0%, 1%, and 1.5%
  - Iterations: 2.5 minute game runs, 3 iterations for each experiment, following longer pilot studies

Downstream Bitrate
unrestricted

Upstream Bitrate
unrestricted

Downstream Bitrate
Capacity restriction

Upstream similar
Much less than downstream!

Only UT for subsequent analysis
Downstream Bitrate
Loss and Latency

Bitrate does not respond to loss and latency

TCP Friendly

\[ T \leq \frac{1.5 \sqrt{2/3} \times s}{R \times \sqrt{\beta}} \]

OnLive not TCP Friendly

Capacity restriction

Capacity affects frame rate
OnLive recommends 5 Mb/s, but accepts 2 Mb/s

Predicted Player Performance

(Model based on FPS data with restricted frame rates)
Capacity affects performance
Summary

- Games as service new model for cloud computing
  - Choices on distribution of rendering and computation
- Cloud games are like video, but different
  - Wider range of motion and scene complexity
- OnLive
  - Like video conference down, traditional games up
  - Bitrate responds to capacity, but not loss or latency
    - Not TCP-Friendly
  - Best for players above 5 Mb/s, with 2 Mb/s minimum
    - Lower capacities affect player performance

Turbulence Summary

<table>
<thead>
<tr>
<th>Application</th>
<th>Bitrate (kb/s)</th>
<th>Pkt size (bytes)</th>
<th>Inter-Pkt (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional game</td>
<td>67</td>
<td>75</td>
<td>45</td>
</tr>
<tr>
<td>Virtual environment</td>
<td>775</td>
<td>1,027</td>
<td>9</td>
</tr>
<tr>
<td>Live video</td>
<td>2,222</td>
<td>1,314</td>
<td>0.1</td>
</tr>
<tr>
<td>Thin Game</td>
<td>6,247</td>
<td>1,203</td>
<td>0.7</td>
</tr>
<tr>
<td>Pre-recorded Video</td>
<td>43,914</td>
<td>1,514</td>
<td>0.1</td>
</tr>
</tbody>
</table>