Remote Display

Ricardo A. Baratto
PhD Candidacy Exam
Columbia University

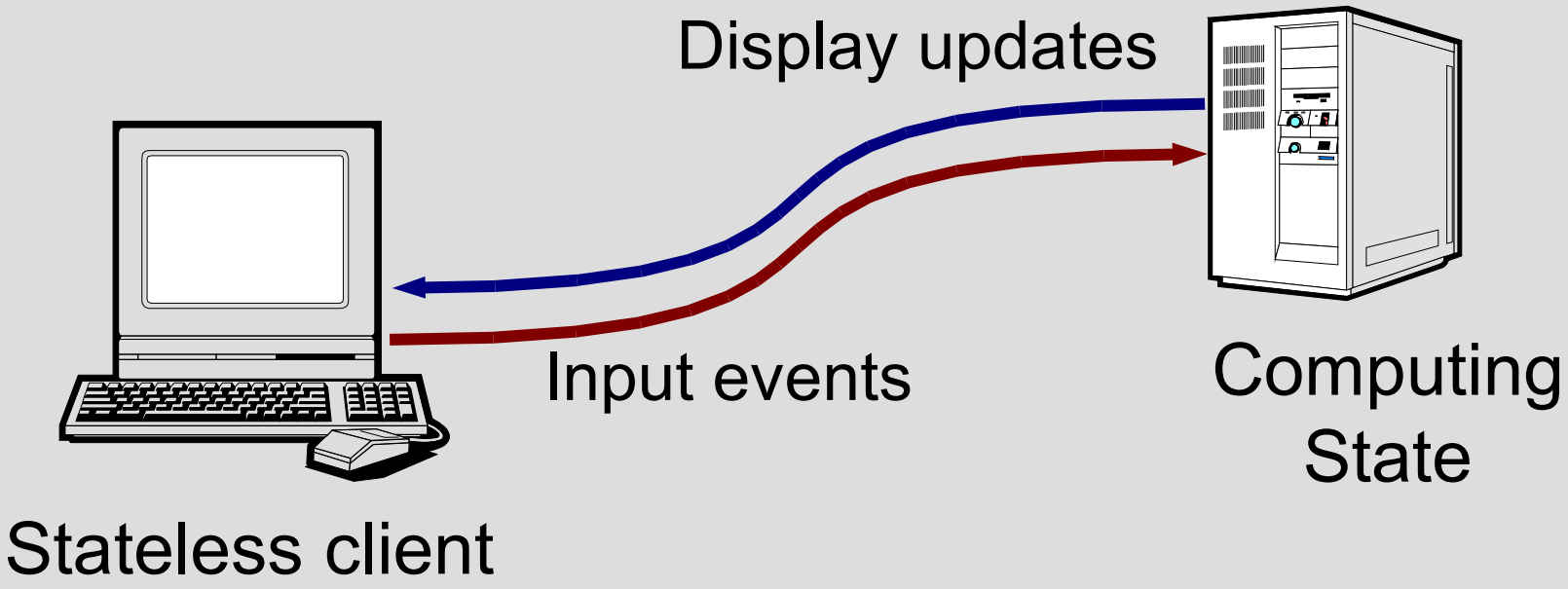
September 20, 2004

Outline

- Remote Display Systems
- Compression
- Delivery Optimizations
- Remote 3D Display
- Measurement Techniques
- Measurement Results
- Conclusions

What is remote display?

- Applications decoupled from display
- Thin-clients:



Remote Display and Thin-client Systems

Characteristics:

- Division of roles and state
  - Client mobility
- Type of protocol updates
  - High-level, low-level, pixel-level
- Delivery of updates
  - Server-driven, client-driven, user-driven
- Adaptive?
- Application support:
  - Tailored to general or specific applications
  - Transparency to applications

X [Scheifler-Gettys 86]

- Client has all state
  - Inversion of client-server roles
- High-level protocol

Problems:

- No mobility [Richardson 94]
- No compression support [Danskin 94]
- Synchronization

VNC [Richardson 98]

- Stateless client
- One pixel-level primitive
  - Multiple encodings
- Client-driven updates

Problems:

- Poor Interactivity
- No perfect encoding exists

Thin-client to the limit

The client is just an I/O interface to the underlying infrastructure
[Truman 98, Schmidt 99]

- Specialized clients
- Stateless
- No support for application execution

InfoPad [Narayanaswamy 96, Truman 98]

- Client as access and communications device
  - Wireless
  - Multimedia
- Decentralized hardware
  - “Collection of peripherals”
- Specialized interface
  - Speech and handwriting input

Problems:

- Specialized solution

SLIM/SunRay [Schmidt 99]

- Hardware-only access console
- Low-level protocol
  - Mimic client hardware
- Relaxed delivery of updates
  - UDP and own error recovery mechanisms
  - Dedicated interconnection fabric

Problems:

- Bandwidth-intensive
- Not suitable for shared networks

Rajicon [Su 02]

- Cellphones as access devices
  - Ubiquitous connectivity
- What kind of user interface?
  - Driven by constrained environment

Compression

- Must balance speed and bandwidth usage
- Tailored to characteristics of display contents
- Must be lossless

Approach

- Exploit characteristics of desktop content
  - Sharp edges
  - Solid/Patterned background elements
- Exploit repetitions in desktop content
  - Icons, window decorations, text

- Updates: HBX [Danskin 94], FABD [Gillbert 98], PWC [Ausbeck 00], TCC [Christiansen 00, 02]
- Framebuffer: TCVQ [Gillbert 00]

TCC [Christiansen 00, 02]

- Separate the details: Marks
  - Small, few colors
- Underlying components are more uniform
  - e.g. solid background regions
- Used by GoToMyPC

Delivery Optimizations

- How to improve the transmission of display data?

- Asymptotic reliable delivery [Han 96]
- Localization [Aksoy 00]
- Update dependency tracking and squashing [Gilbert 00]

Asymptotically Reliable Delivery
[Han-Messerschmitt 96]

- Too much overhead from error correction and retransmissions
- Corrupted data can be useful
  - Graphics are resilient to errors
  - Improve response time by not delaying delivery

Problems

- At odds with compression
- Applications must be aware of mechanism

Localization [Aksoy-Helal 00]

- Move functionality to the client
- Used by Citrix MetaFrame

Remote 3D Display

How to balance the thin-ness of the client with the requirements of the application?

- High resource requirements
  - Shared environment
- Approach: Partition the 3D pipeline

Dedicated rendering server [Stegmaier 02]

- Generic solution
- Could possibly off-load application server

Problem

- How to deliver the content?

Push functionality to the client [Levoy 95]

- Render high-quality and low-quality images
- Transfer only difference image
  - Improved delivery

Problem

- Not generic
- Server still doing all the work

Stream of rendering components [Humphreys 01,02]

- Divide pipeline for scalability
  - Does not address delivery issues
- Framework for balancing rendering work
  - Possibly dynamically?

Measurement Techniques

- Traditional application benchmarks not suitable
  - Only measure server performance
  - Many are throughput-based
- Cannot instrument proprietary systems

Capture application traces
[Danskin 94, Schmidt 99]

- Capture protocol messages generated in a user session

Benefits
- Realistic
- Repeatable
- Flexible

Problem
- Need open protocol

Slow-motion benchmarking
[Nieh-Yang-Novik 03]

- Use network monitoring
  - Systems are just blackboxes
  - Measure of client-perceived performance
- Introduce delays
  - Avoid merging of display updates
  - Plus: Mimics real user behavior

Problems

- Client processing not fully accounted for
- Cheating

Measurement Results

- User-perceived latency is key [Wong 00, Schmidt 99]
- Network latency is key [Lai 02]
- Thin-clients ideal for constrained environments (e.g. PDAs) [Lai 04]
- User-perceived latency not driven by data transfer [Lai 02, 04]

Conclusions

- Many systems and many approaches
  - Not one perfect system
  - Is this even possible?
- Complex systems
  - Plenty of room for optimizations
- System response time is key
  - More important than bandwidth usage
- Open problem: Remote 3D display