



VDBench: A Benchmarking Toolkit for Thin-client based Virtual Desktop Environments

Alex Berryman, berryman@oar.net

In collaboration with: **Dr. Prasad Calyam** (OSC/OARnet), **Prof. Albert Lai** (OSUMC), **Matt Honigford** (VMware)

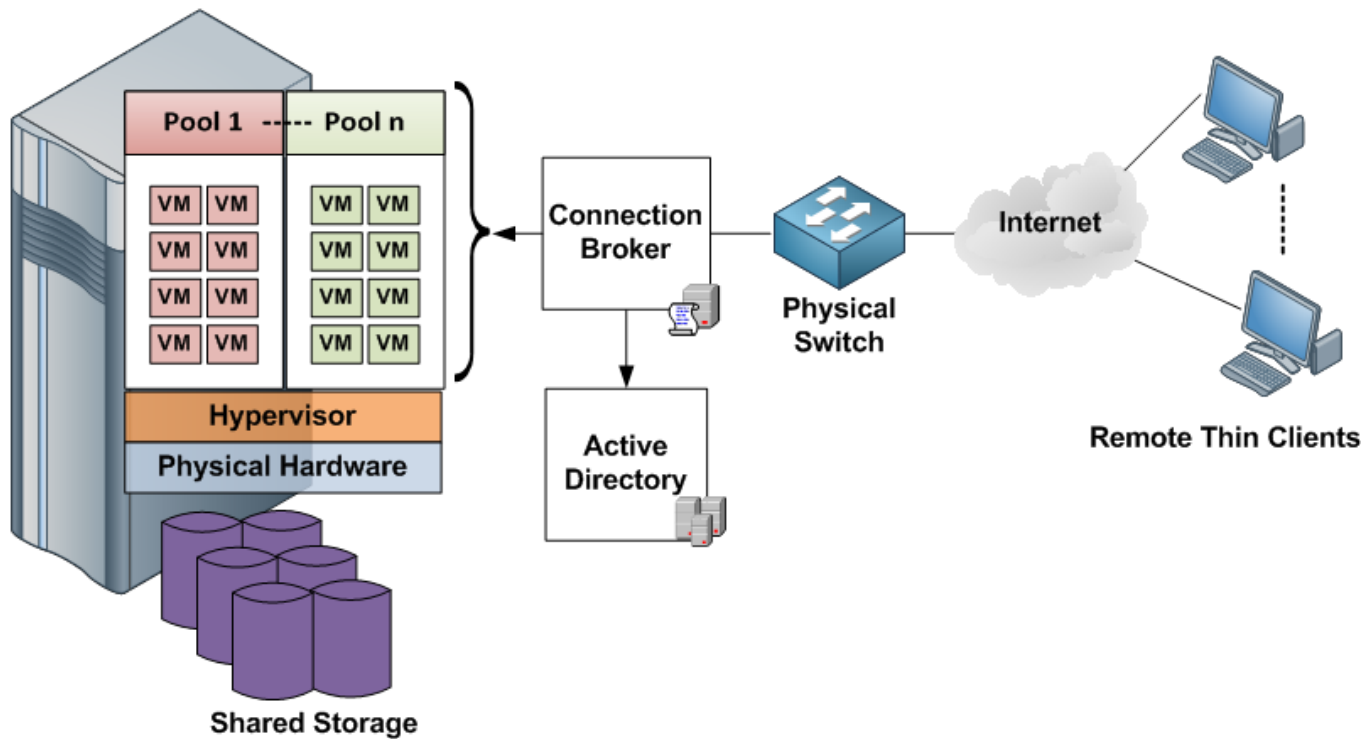
*IEEE CloudCom, Indianapolis, IN
December 2nd, 2010*

Topics of Discussion

- Background and Motivation
- VDBench Components and Data Flows
 - Architecture, Metrics, Techniques
- VDBench Experiment Results
 - User Load Simulation based Benchmarking
 - Slow-motion Application Interaction based Benchmarking
- VDBench Use Cases
 - Performance Mapping to User Application and Group Profiles
 - Resource Location Selection for Performance Balancing

Virtual Desktop Infrastructure

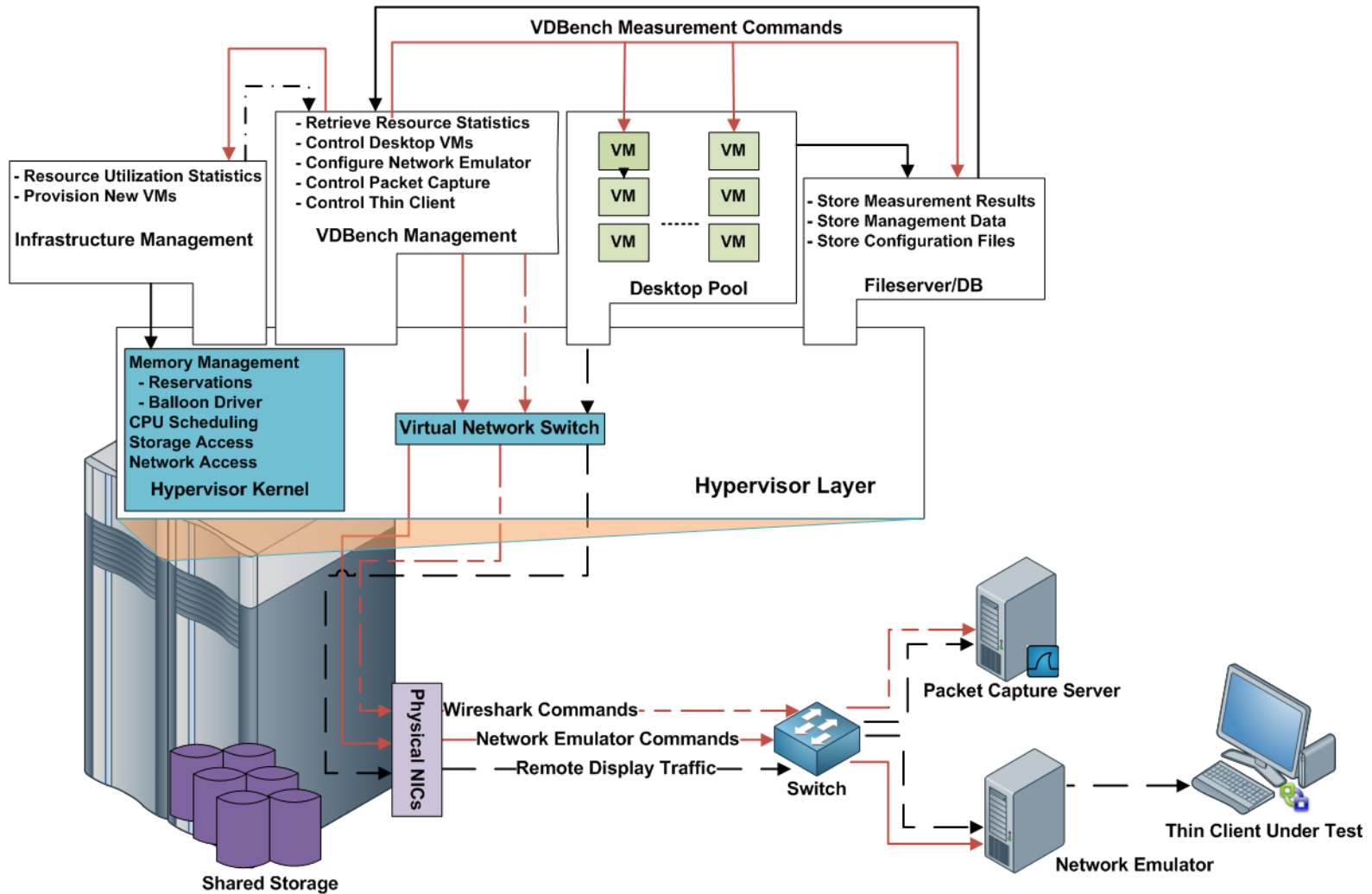
- VMware View is one of the popularly used VDI solutions
 - Personalized user desktops, applications and data access while maintaining centralized control and security



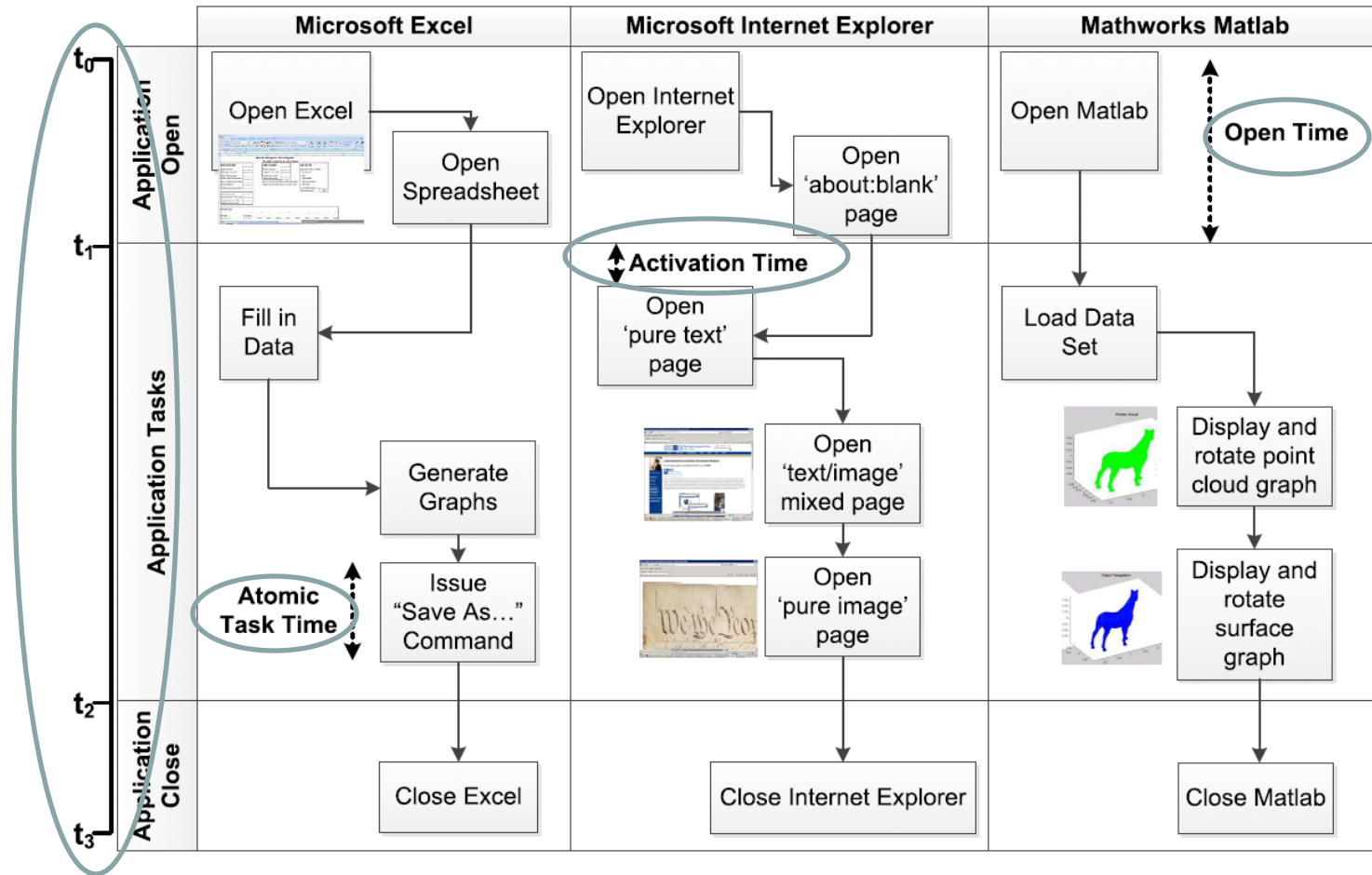
Research Context

- Recent advances in thin clients and the numerous benefits in transitioning user desktops to cloud environments
 - Convenience, Cost savings, Green IT, Security, ...
- *Need for “system-aware”, “network-aware”, “human-aware” frameworks and tools to deploy virtual desktop clouds*
 - Existing work focuses mainly upon system (i.e., CPU and memory) measurements for server-side resource adaptation
 - Our focus is to couple client-and-server resource adaptation with measurements of network health and user experience
 - Minimize cloud resource over-commitment
 - Avoid guesswork in configuring thin client protocols
 - Deliver optimum user experience of virtual applications

VDBench Components and Data Flows



Application Tasks Progression



User-Load Simulation Metrics

- Controlled Variables:
 - Number of VMs concurrently running to simulate user-load
 - “Ceiling Response Time” is set to 30% increase from the ideal “Application Response Time”
- Metrics:
 - *Application Response Time*
 - Application Open Time
 - Aggregate Inter-application task
 - Matlab surface visualization
 - Atomic tasks
 - Alt+tab, ‘Save As’, web-page loads
 - *Memory Availability*
 - $(\text{Memory Allocated} - \text{Memory Balloon}) / \text{Memory Allocated}$
 - *Available Memory utilization*
 - $\text{Memory Used} / (\text{Memory Allocated} - \text{Memory Balloon})$

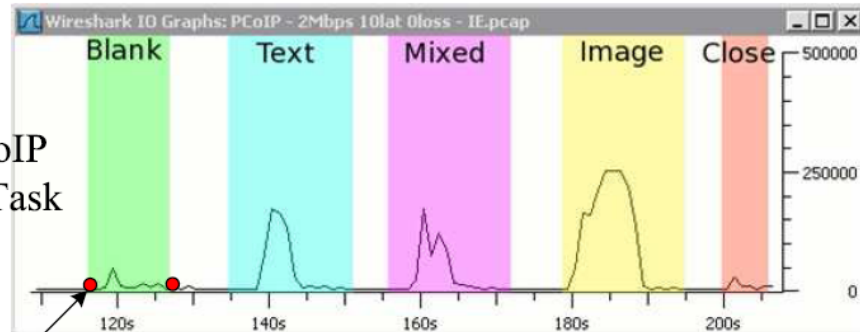
Slow-Motion Benchmarking Metrics

- Controlled Variables:
 - Network Health:
 - Latency, Loss, Available Bandwidth
 - Codec choice:
 - RDP, RGS, PCoIP
- Metrics:
 - *Render Time*: time taken for a screen update to complete
 - Assume that the time between the first and last packet of a screen update is equal to the time taken to display the update
 - *Coding Efficiency*: amount of data to transfer screen information
 - Atomic: Single screen update
 - Aggregate: Many screen updates (e.g., Video playback)
 - *Video Quality*: ratio of actual video playback with ideal playback
 - Measures the information loss when a codec transmits a video file

$$Video\ Quality = \frac{\frac{Data\ Transferred\ (aggregate\ fps)}{Render\ Time\ (aggregate\ fps)}}{\frac{Data\ Transferred\ (atomic\ fps)}{Render\ Time\ (atomic\ fps)}} \cdot \frac{IdealTransfer(atomic\ fps)}{IdealTransfer(atomic\ fps)}$$

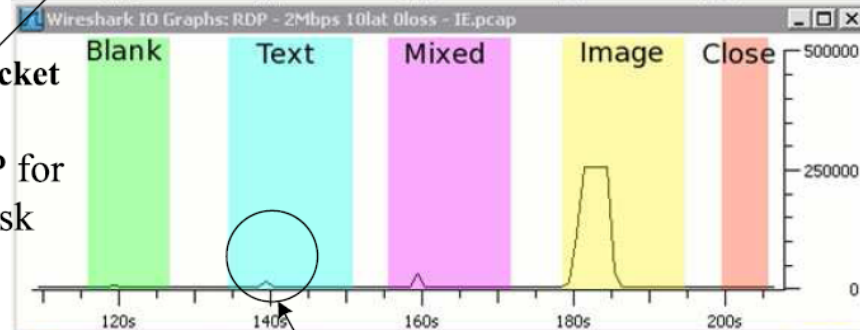
Slow-Motion Benchmarking Technique

(a) Bandwidth Consumption of PCoIP for an Aggregate Internet Explorer Task



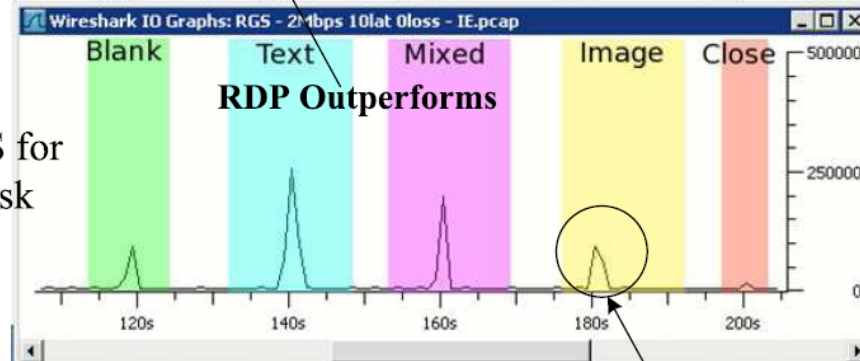
Marker Packet

(b) Bandwidth Consumption of RDP for an Aggregate Internet Explorer Task



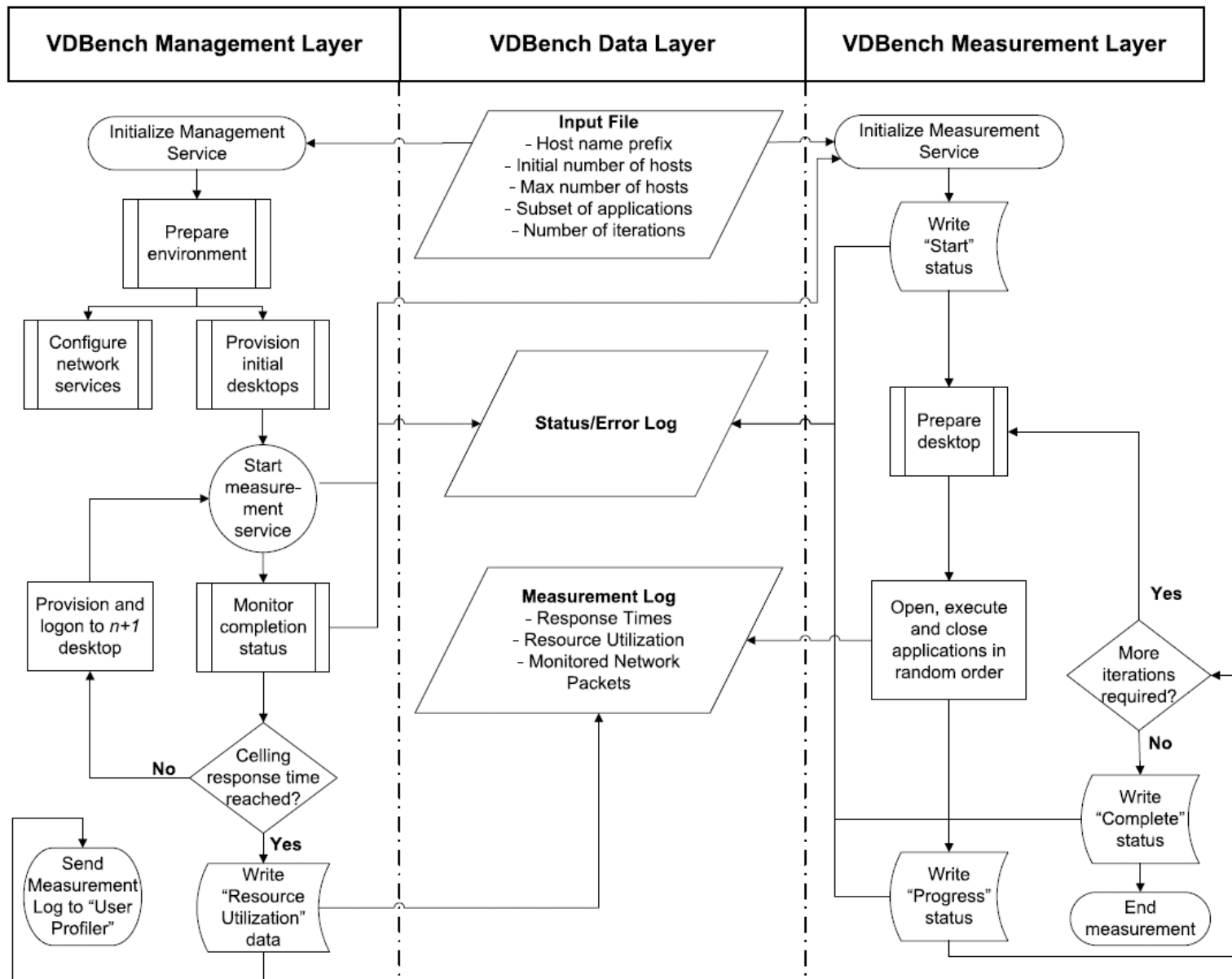
RDP Outperforms

(c) Bandwidth Consumption of RGS for an Aggregate Internet Explorer Task



RGS Outperforms

VDBench Control Logic



Research Goals and Contributions

- Develop VDBench techniques to simulate realistic user workflows under synthetic system loads and network health impairments
 - Useful to measure corresponding user-perceived ‘interactive response times’ (e.g., application launch time, web-page download time, video quality)
- Correlate thin-client user events with server-side resource performance events and develop novel performance metrics
 - Proposed the use of ‘marker packets’ that leverage and extend earlier research on slow-motion benchmarking of thin-client performance
- Generate resource utilization profiles of different applications and user groups based on VDBench measurements
 - Useful to intelligently map pools of desktops to resources such that user satisfaction is ensured with minimal resource over-provisioning

Related Work

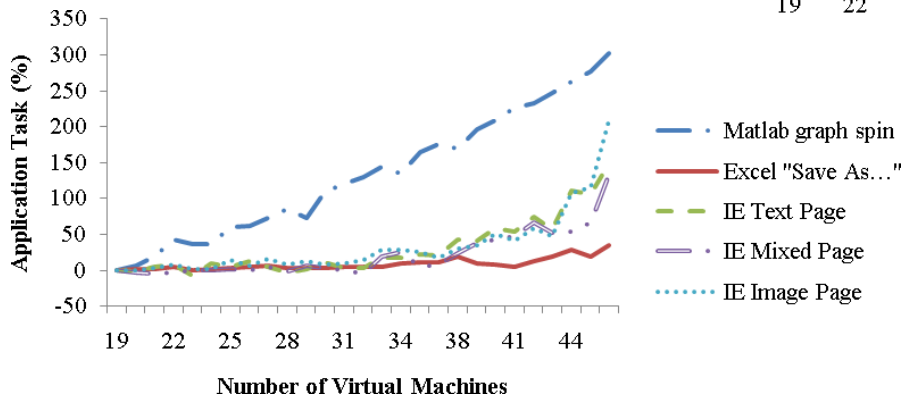
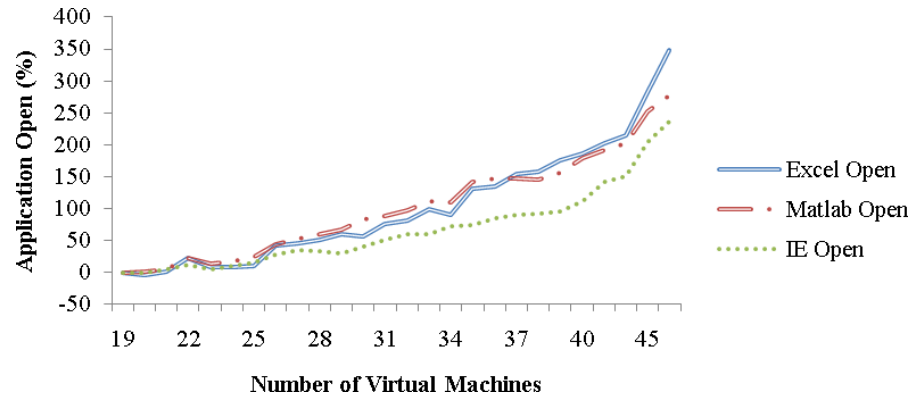
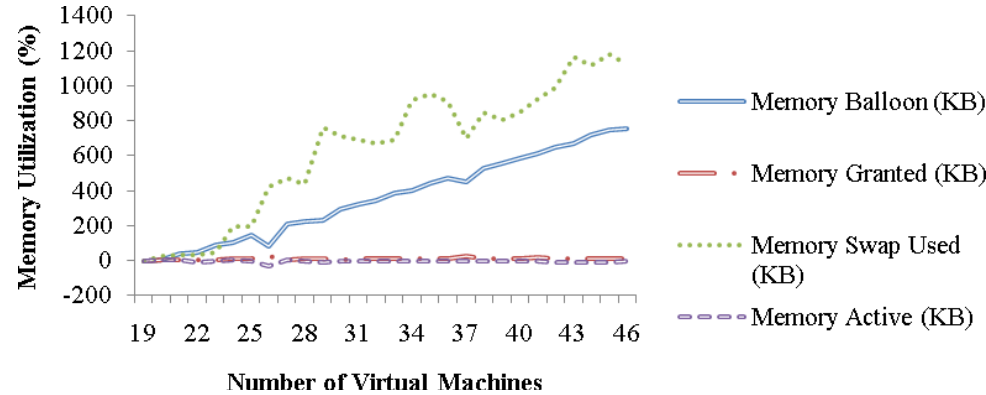
- Slow-motion benchmarking of thin-client display protocols was developed by [Nieh, et. al.]
 - Focus is on measuring user perceived performance by monitoring client-side of a remote desktop session
 - Lai, et. al. used slow-motion benchmarking to investigate thin-client display protocol characteristics over WAN connections
- Existing virtual desktop benchmarking tools such as “Login VSI” developed by [Sprujit, et. al.] simulate realistic user workflows
 - They neglect the distinction between client-side rendering and server-side processing and hence cannot measure thin-client user experience
- VNCPlay [Zeldovich et. al.] and Deskbench [Rhee et. al.] focus on thin-client workflow replay
 - They do not address event-synchronization with server side performance measurements

VDBench Experiments Overview

- VDI Scalability
 - Stress test VMs under various user application work loads
 - User Applications: MS Excel, IE Browser, Windows Media Player, Matlab
 - We used “Autoit” user application work loads (scripts for repeatable and automated GUI interactions with key presses, mouse movements)
- VDI Reliability
 - Evaluate performance of remote desktop protocols
 - Protocols Evaluated: Microsoft Remote Desktop Protocol (RDP), HP Remote Graphics Software (RGS), Teradici PC-over-IP (PCoIP)
 - We used Netem for network (i.e., bandwidth, delay, loss) emulation

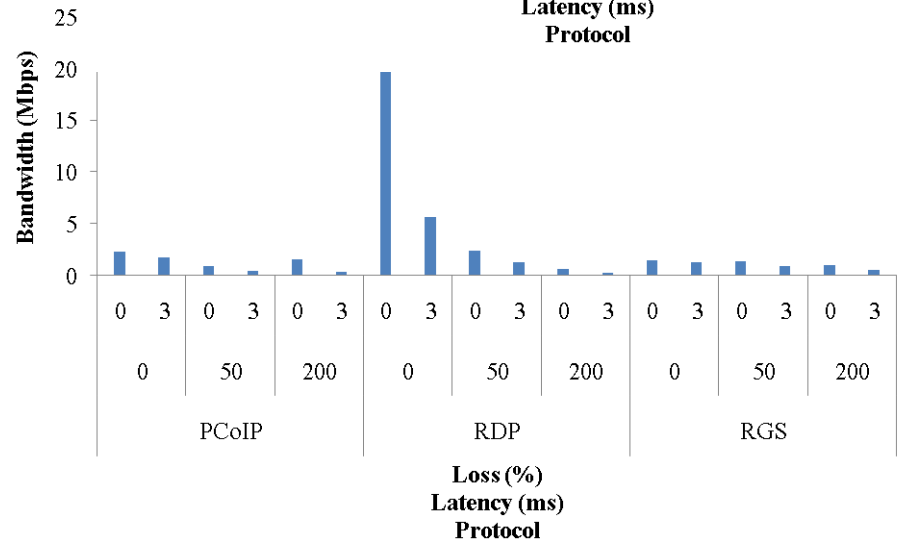
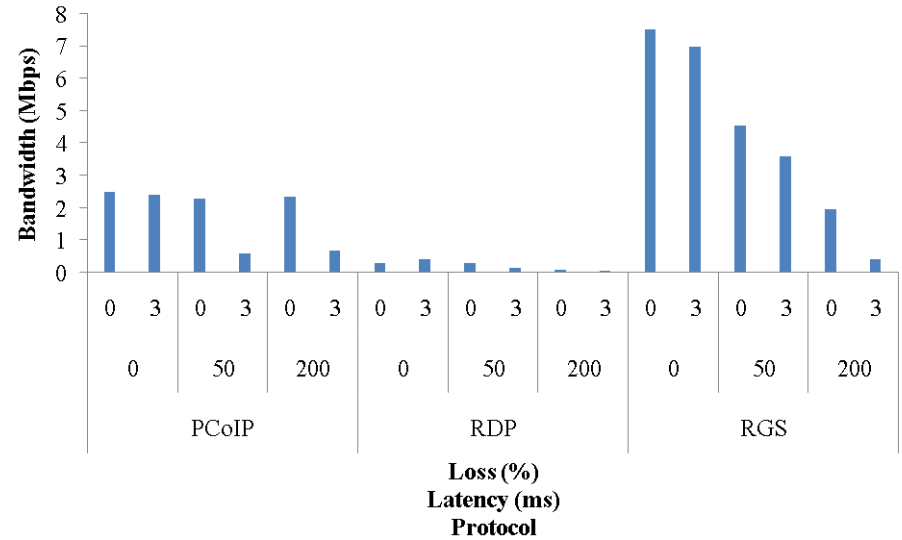
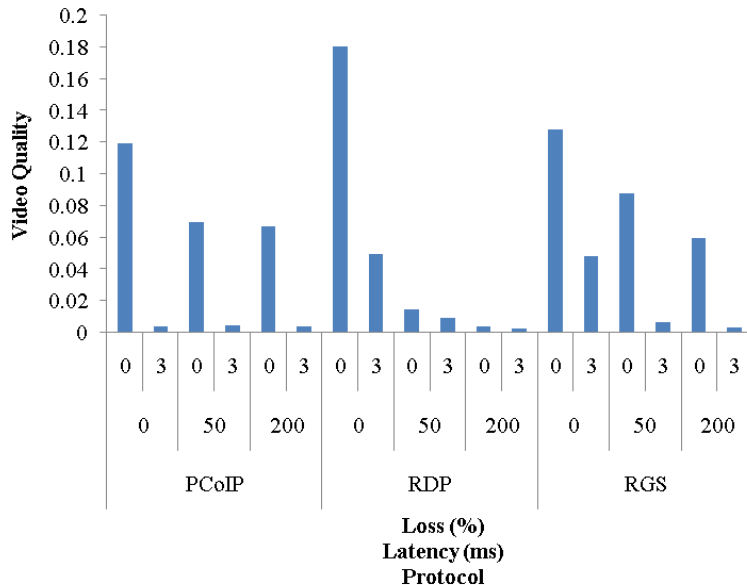
User Load Simulation based Benchmarking

- Memory utilization shows a linear increase in Memory Balloon size as the number of VMs increase
- Increasing Memory Balloon size on each VM leads to an increase of user-perceived application open and task times



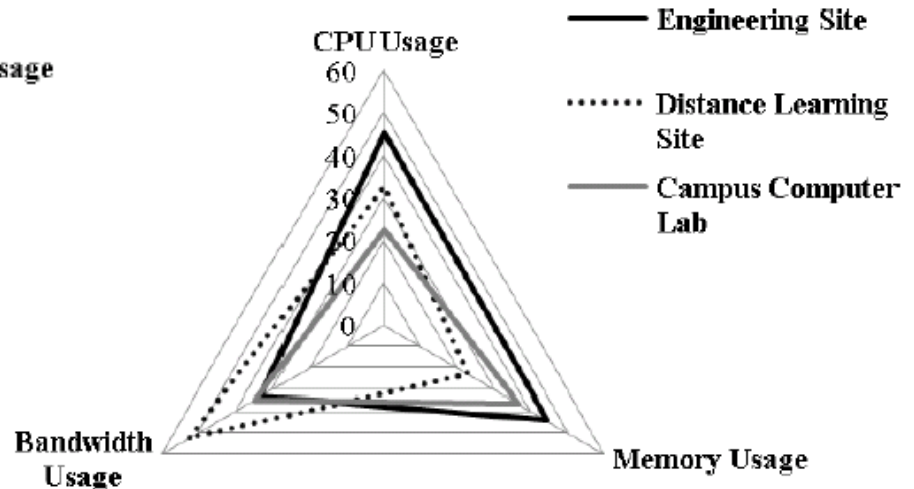
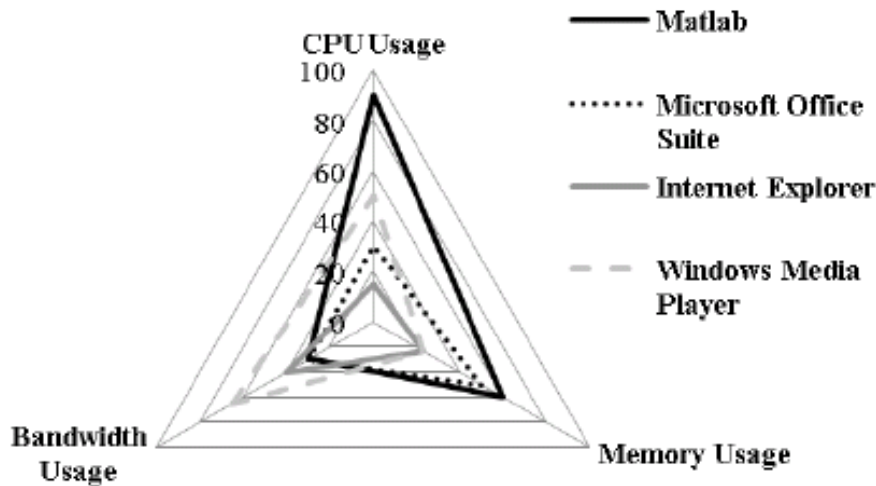
Slow-motion Application based Benchmarking

- Slow-motion benchmarking results give the bandwidth consumption of each protocol for a specific type of screen content and network health condition



VDBench Use Cases

- Performance Mapping to User Application and Group Profiles
- Resource Location Selection for Performance Balancing



Thank you for your attention!

