Accelerating Remote Display Performance for GUI-based Applications

DOE Computer Graphics Forum

April 28, 2008

NERSC is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.

Motivation

- Complaints in 2006 NERSC User Survey about poor performance of remote visualization applications
- Poor X11 network performance …
  - … makes it tedious for users to use NERSC resources
  - … prevents some users from using NERSC resources

- Help existing users!
- Make NERSC analytics resources more attractive to new users!
- Prepare for increasing data size that prevents users from moving data off-site!

Problems

- X11 is a verbose protocol
- Frequent blocking operations
- Many network round-trips necessary, even when drawing a simple menu

- NERSC users often at the end of a high-latency network link
  - High latency likely to be the main cause of performance problems
  - Limited bandwidth may contribute, but likely to have less impact

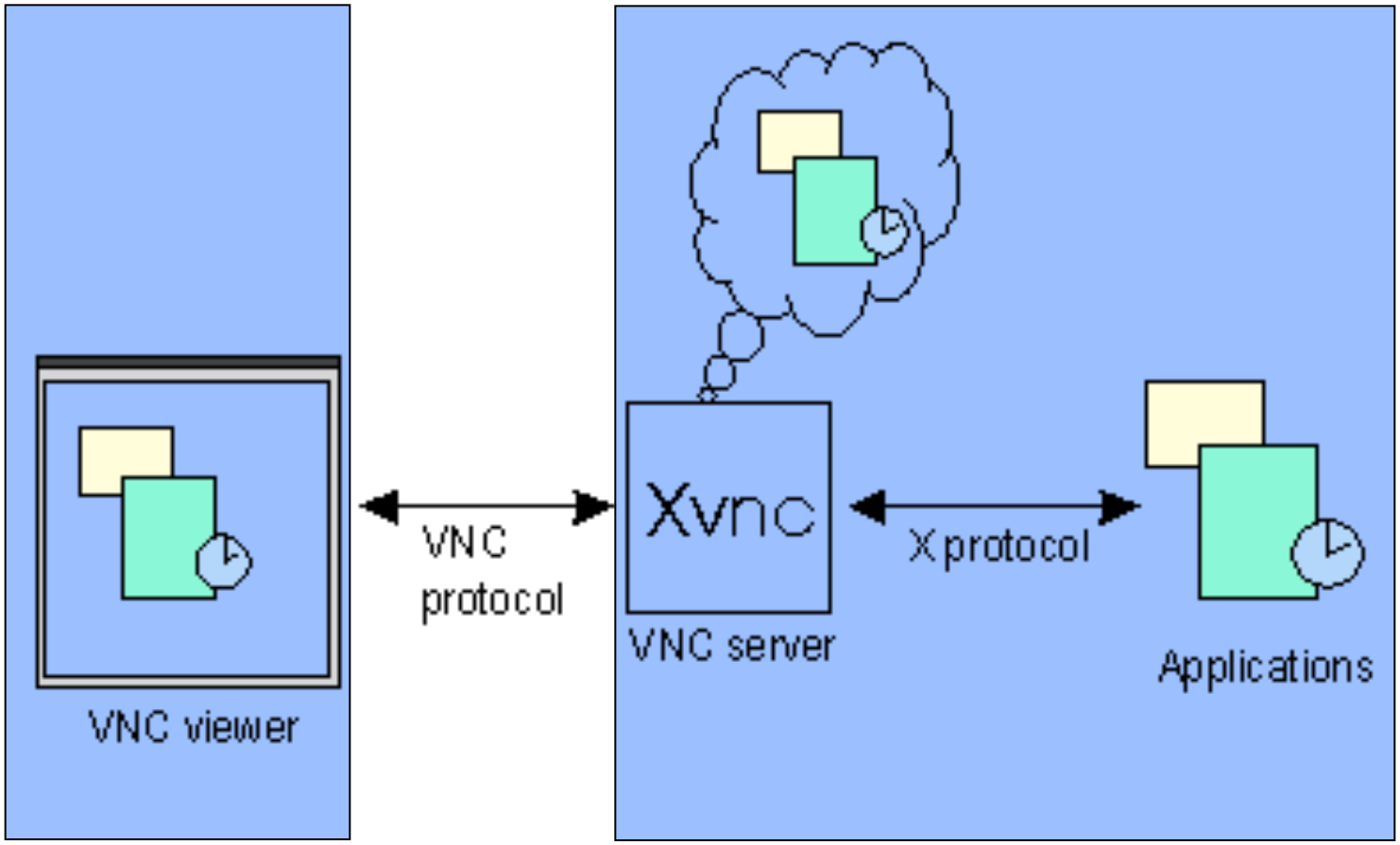
X11 Alternatives – Overview

- Two major “players” under consideration:
  - Virtual Network Computing (VNC)
    - RealVNC, TightVNC, TurboVNC
  - (Free)NX

X11 Alternatives – VNC

- **Use Remote Framebuffer Protocol (RFB) transmitting frame buffer updates instead of "draw commands"**
- **Local X Server (Xvnc) replaces remote X11 connection with RFB connection**
- **Common versions:**
  - **RealVNC** (Enhancement of original prototype created at Olivetti & Oracle Research Lab)
  - **TightVNC** (Improved image compression)
- **TurboVNC** (Enhancements relevant to visualization: double buffering, compression speed)
- **Currently available on DaVinci**

VNC Diagram



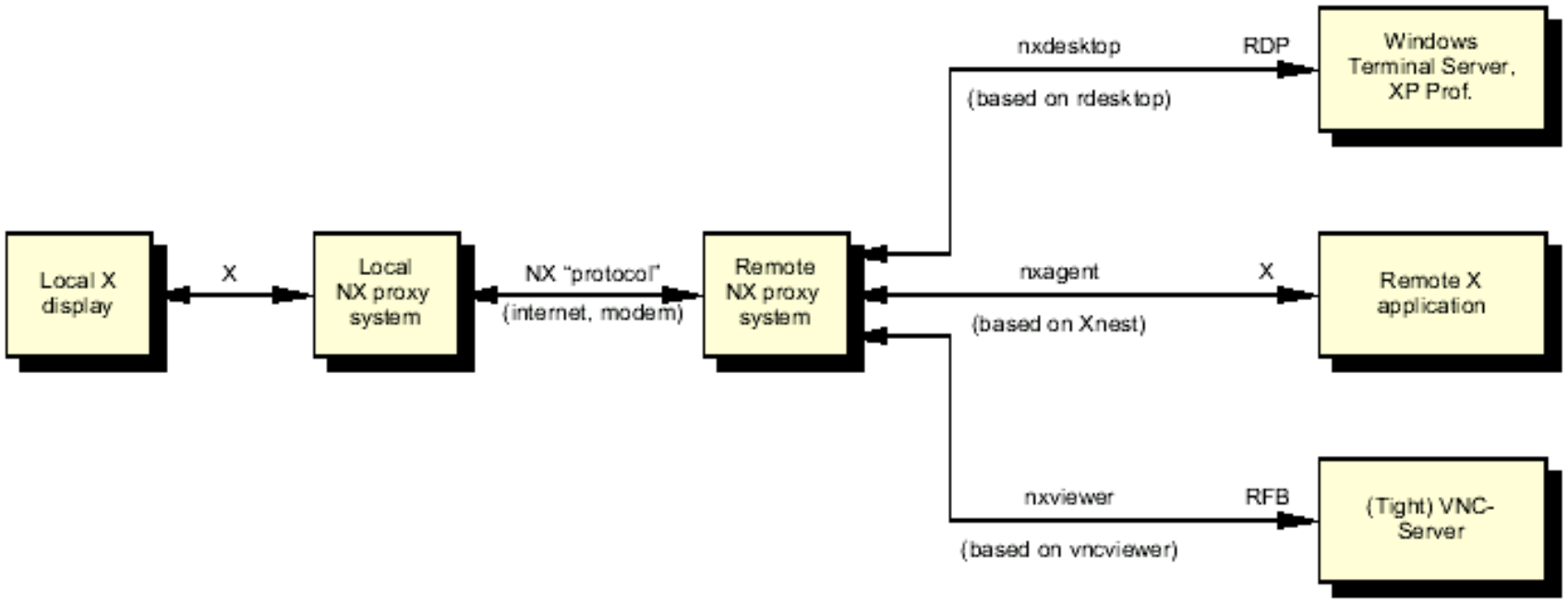
Remote User

NERSC (“DaVinci”)

X11 Alternatives – (Free)NX

- Based on differential X protocol compressor
- Compresses X11 traffic
- Adds proxy/agent that caches images etc. and "shortcuts" many operations requiring handshake
- Commercial (NoMachine, Inc.) and open source version (FreeNX) available
- "Floating" window session (without desktop) possible
- Out-of-box GLX support via software rendering

NX System Architecture



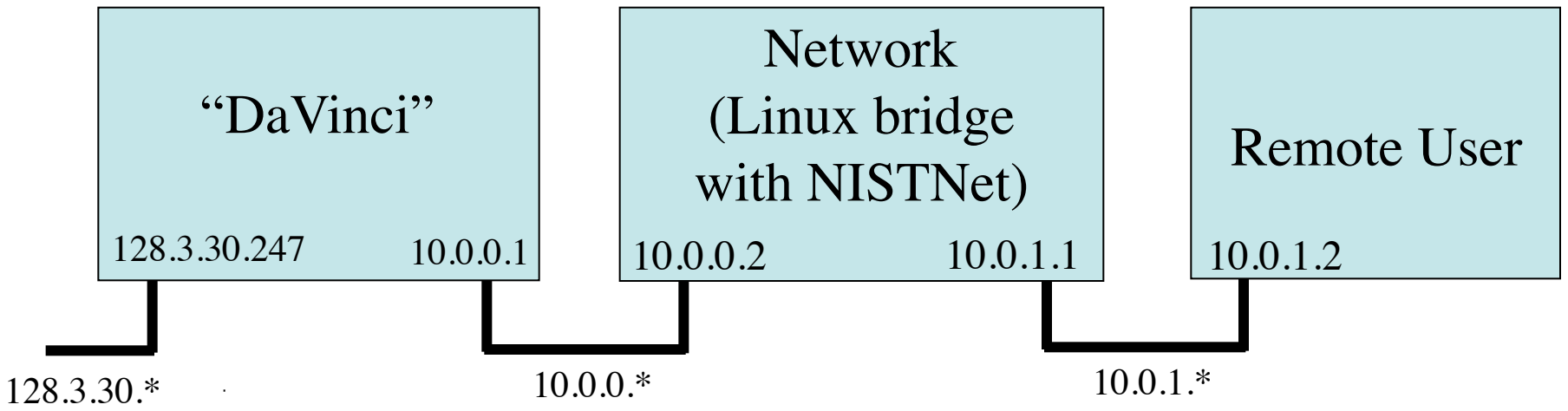
Evaluation Criteria

- Speed / Interactivity
  - How “usable” over high-latency link?
  - Measurements:
    - Time to finish update
- Deployability
  - Complexity of remote setup?
    - Both solutions have easy to use clients
    - Other considerations?
- Security
  - Any new security concerns by using new service?

Benchmarking X11 Alternatives

- Simulate remote network conditions
  - Obtain measurements of network characteristics to remote users' sites
    - Mainly latency and bandwidth / transfer rate
  - Derive realistic range of conditions
  - Simulate network
- Measure performance
  - First subjective evaluation
  - Exact timings / measurements
    - Use of common applications
    - Definition of metrics that are "fair" despite differences in protocol/implementation

Simulating the Network



- **Run most test applications locally on fake "DaVinci"**
- **LBNL/NERSC network for connection to NERSC license server (for Matlab, …)**

Measuring Performance

- Test with existing applications (Matlab, Maple, IDL, …)
  - Subjective speed experience
  - Time between “mouse click” and “last screen update” (using XTrap and XDamage extensions)

Initial Performance Test Results – "UCLA"

Action	SSH	VNC	FreeNX
Establish connection	n/a	~11s	~16s
Start Matlab (-nosplash)	9.6s	4.9s	5s
Open edit window	2.9s	1.3s	1.2s
Activate File menu	0.6s	0.1s	0.1s
Activate Edit menu	0.6s	0.1s	0.1s
Activate Text menu	0.5s	0.2s	0.1s
Close edit window, redraw main window	1.5s	0.4s	0.3s
Close matlab	0.5s	0.6s	0.6s

Latency: 5ms each way, 10ms RTT, No bandwidth limitation via NISTNet
Measured bandwidth: 324 Mbits/sec (iperf)

Initial Performance Test Results – "PPPL"

Action	SSH	VNC	FreeNX
Establish connection	n/a	~5.7s	~10.2s
Start Matlab (-nosplash)	39.5s	4.6s	5.4s
Open edit window	14.9s	1.3s	1.2s
Activate File menu	3.7s	0.3s	0.2s
Activate Edit menu	7.6s	0.4s	0.2s
Activate Text menu	5.1s	0.4s	0.3s
Close edit window, redraw main window	7.3s	1.4s	1.8s
Close matlab	2.1s	1.54s	1.1s

Latency: 40ms each way, 80ms RTT, No bandwidth limitation via NISTNet
Measured bandwidth: 38.9 Mbits/sec (iperf)

Security Considerations

- VNC
  - Security strength of RFB authentication (Hash + armoring of data stream)
    - Simple encryption, probably more secure than X
  - Requires users to choose yet another password
    - Not automagically like X MAGIC-COOKIE
  - Tunneling via SSH possible
- NX
  - Uses existing X11 and SSH protocols
  - Requires a separate user “nx” to initiate session
  - Clients connect with public key without pass phrase
  - Use regular user account passwords to initiate session

Anticipating Future Requirements

- Any solution needs to provide OpenGL / GLX functionality
  - Required by VisIt, some Matlab modules, IDL, …
- Current NERSC machines: Software GL

- Current (and future) data set sizes warrant hardware rendering
- Evaluate feasibility to use graphics hardware remotely via Chromium Render Server and VirtualGL

NX <-> VNC Advantages

- Speed measurements almost a tie
- NX
  - Moving windows “snappier”
  - Probably less network traffic and more tolerance of low-bandwidth situations
  - Users completely “shielded” from having to start server etc.
  - Less protocol divergence and “viewer confusion”
  - Desktopless “floating window” mode – though somewhat “flaky”
  - SSH-based
- VNC
  - Better hardware OpenGL/VirtualGL support
  - Existing Chromium Render server support
  - Java servers available
  - Pure web browser-based AJAX solutions in development

Current State of Affairs

- Completed:
  - Network simulation setup
  - Simple user interaction timer using XTrap and XDamage extensions
  - Set up VNC version and FreeNX on test network
  - Initial tests
- In Progress
  - Evaluation with NERSC Networking and Security Group (NAST)
  - Perform further tests
  - Initial recommendation
- To do:
  - ???Implement frame-rate measurement tool and GL Tests???
  - Evaluation with select NERSC users
  - HOWTOs and deployment

Questions?

NERSC is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.