Telepath: Cluster-based Visualization Session Services

Randall Frank
Lawrence Livermore National Laboratory
VIEWS Visualization Project Lead

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

UCRL-PRES-204201

Telepath: Goals

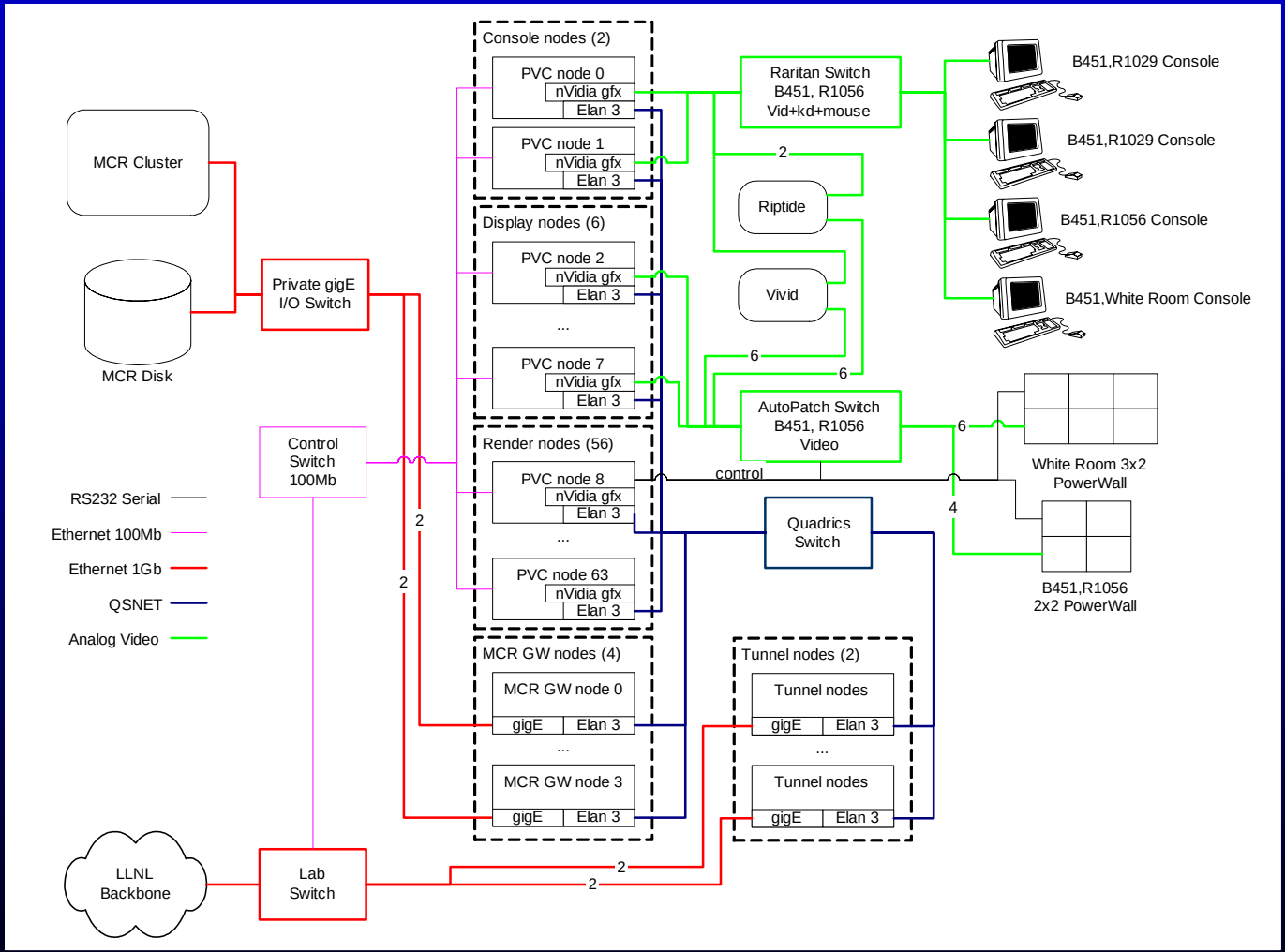
Provide a simple interface to the scheduling of resources and realization of multiple, simultaneous, dynamic, end-to-end visualization environments on a cluster.

- Schedule and allocate the hardware resources
- Set up the resources (displays, switches, etc)
- Launch application service layers
- Tear-down the session (session management)

- All in a secure/reliable fashion…

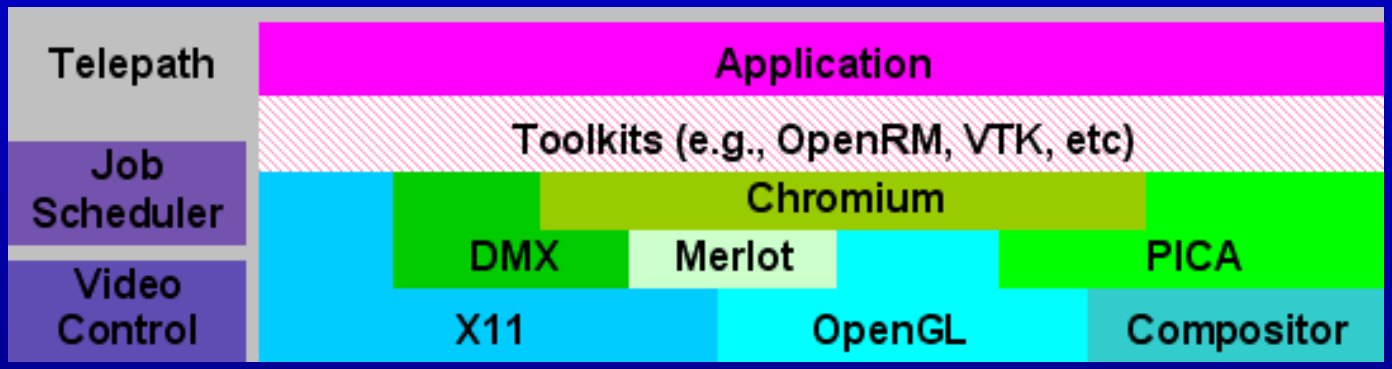
Telepath: Hardware Environment

“Classic” COTS clusters



Telepath: Software Environment

VIEWS scalable rendering stack



Services

- XFree86/OpenGL, DMX, Chromium, MIDAS, etc

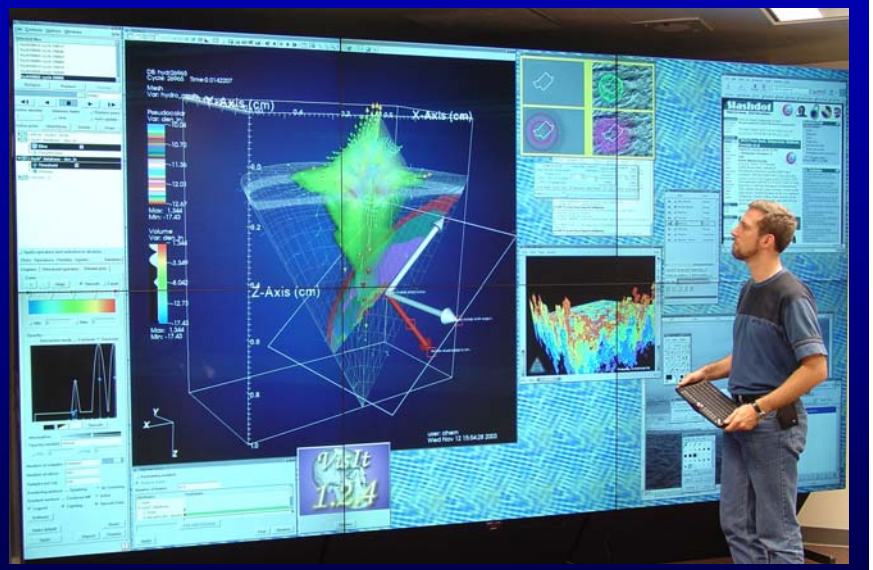
Applications

- VisIt, EnSight, TeraScale Browser, Blockbuster

Telepath: Requirements

User requirements

- Multi-user, simultaneous, dynamic
- Reliability, fault tolerance (“session repair”)
- SIMPLE!!!



System requirements

- Scheduler w/node “attributes”
- Co-exist with non-viz jobs

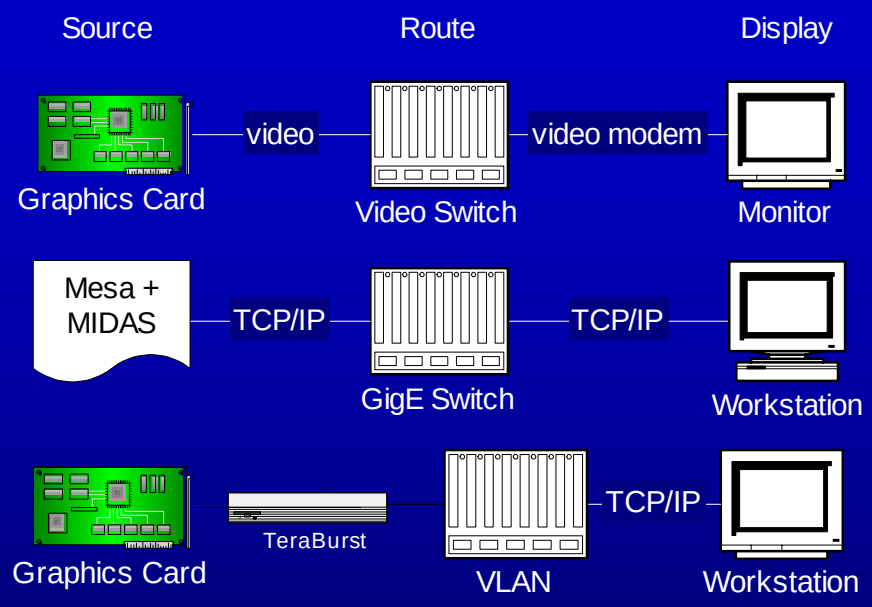
Security requirements

- Known, “rest” state
- Secure transport, authorization, authentication
- Wherever possible, user space services

Telepath: Session Components

Abstract components:

- Compute resources
  - Nodes to run applications
- "Source": Produces imagery
- "Route": Transports imagery
  - Source to display
  - Switches, networks, etc
- "Display": Presents imagery to a user
  - Includes "nulldisplay"
- "Services": User requested session capabilities
  - X11, OpenGL, DMX,…; Explicit or implicit
- "Stream": Defines imagery characteristics



Telepath: Python Modules

A python module exists for each type of component in a given site's deployment.

Stream definition

- Resolution, refresh rate, color depth, etc
- Handshake between components

Standardized module interfaces

- Reset, Query, Setup, Resolve functions
- Communication to devices hidden from Telepath

External site configuration tables

- Layouts, devices, dependencies, etc

Telepath: Execution

Register phase

- Verify availability of resources
- Identify dependencies & build pipeline via recursive table parse

Resolve phase

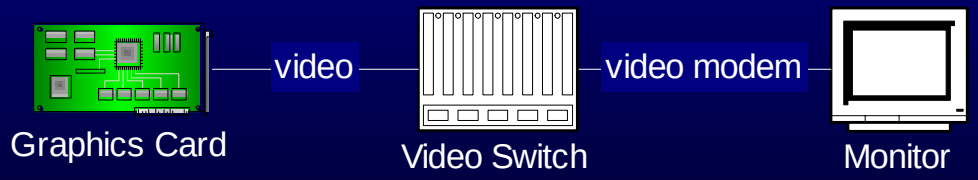
- Two pass – validate & constrain the stream.

Setup phase

- Initiate the components and enable the devices
- Unwind on error and "reset"

User session phase

- Users run applications

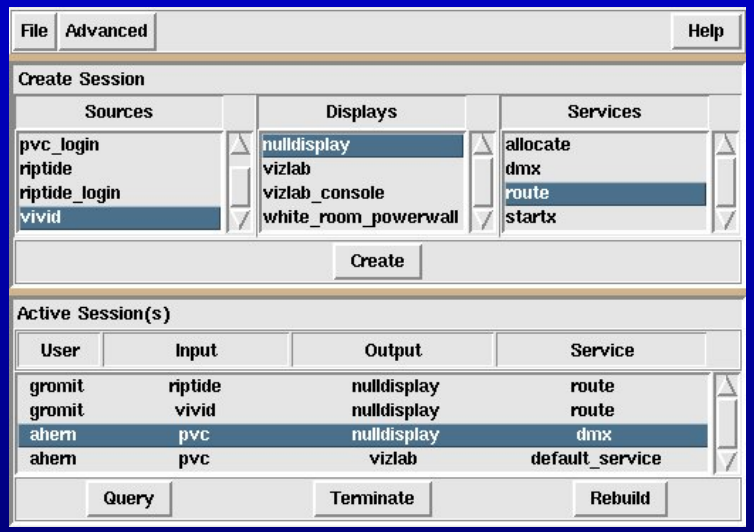


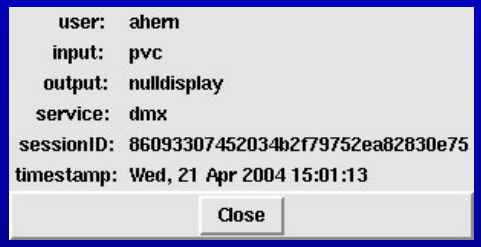
Reset phase

- Unwind the setup operation and bring to "rest" state

Examples

GUI interface





Command line

```
telepath --output=white_room_powerwall --service=dmx
telepath --output=white_room_powerwall \
     --service='dmx;inputfrom=pvc9:0;chromium‘
telepath --input='pvc;nodes=8' --output=nulldisplay \
     --service=startx
telepath --query
```

Telepath: Technologies

Python and related software

- Python, Pyro, Pexpect, TkInter

Other software layers

- Conman, OpenSSL, LCRM (DPCS)

Hardware

- Cyclades terminal server
- Lightwave, Autopatch switches
- Clarity, Barco … projectors/displays

Status and Futures

Complete version (1.0) running in the OCF on PVC and vivid clusters (some support for SGI)

- Currently driving various walls
- Vivid testbed and PVC production

SCF deployment underway on gViz cluster

- Initial version will not control the displays/switches
- Full version deployed as SGI system retired

Future: many, many ideas…

- Additional services, improved application launching, better use of node attributes, more devices, scheduled reservations…

A Cast of Thousands…

Required input from and participation by a large number of contributors: devices, architecture, coding, interoperability and security (and gentle prodding ☺).

- Sean Ahern
- Michael Atzet
- Randy Frank
- Ross Gaunt
- Bob Howe
- Holger Jones
- Liam Krauss
- Scott Miller
- Ramon Newton
- Dale Southard