Large Scale Visualization on the Cray XT3 Using ParaView

Cray User's Group 2008

May 8, 2008

Kenneth Moreland
David Rogers
John Greenfield
Sandia National Laboratories

Berk Geveci
Patrick Marion
Kitware, Inc.

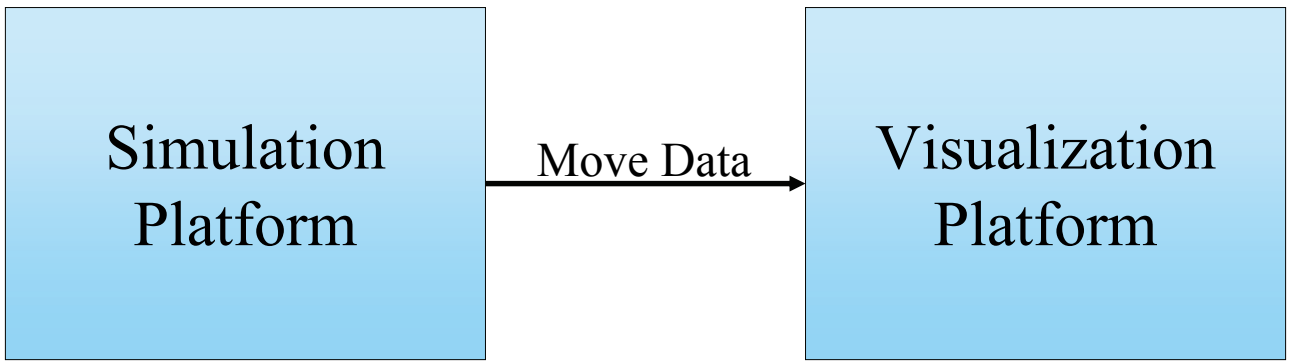
Alexander Neundorf
Technical University of Kaiserslautern

Kent Eschenberg
Pittsburgh Supercomputing Center

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

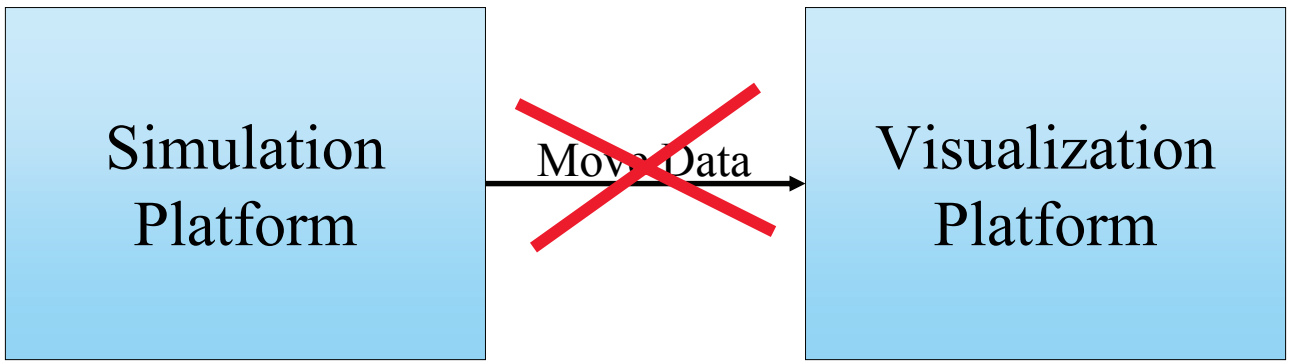
Motivation

- We've spent 20 years developing specialized hardware for graphics/visualization applications.
- Originally, it was customary to move data from the simulation platform to the visualization platform.



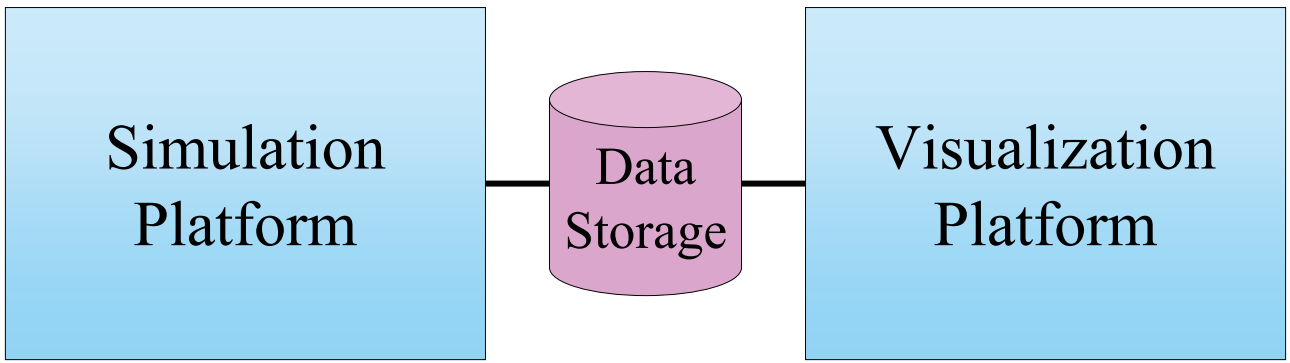
Motivation

- **Moving data from large simulations is prohibitive.**
  - **It can take from hours to weeks depending on the data and the connection.**



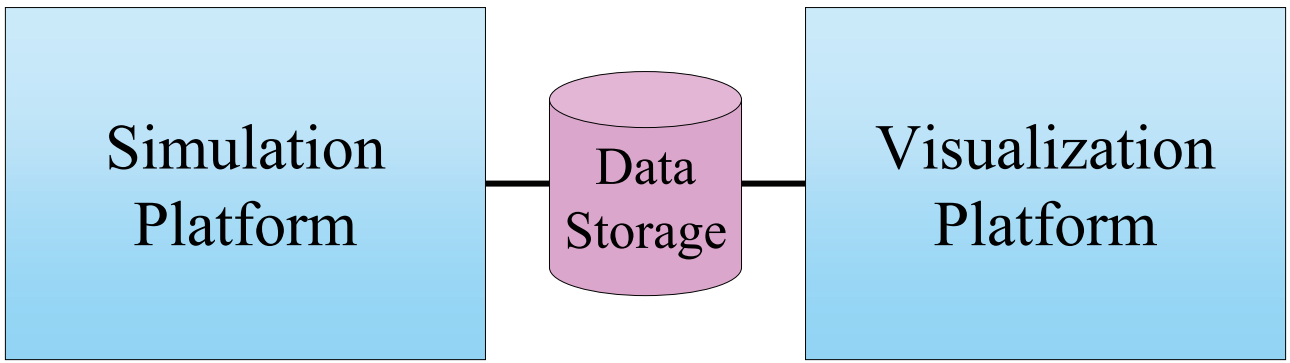
Motivation

- **Moving data from large simulations is prohibitive.**
  - **It can take from hours to weeks depending on the data and the connection.**
- **We've learned to co-locate the visualization platform with the data.**
  - **Or at least have a dedicated high speed network.**



Motivation

- **Bottlenecks are (in order):**
  - **File data management (moving/reading).**
  - **Data Processing (isosurfaces, mathematics, computational geometry, etc.).**
  - **Rendering.**
- **In fact, we've had great success deploying on clusters with no graphics hardware.**



Ideal Visualization Computer

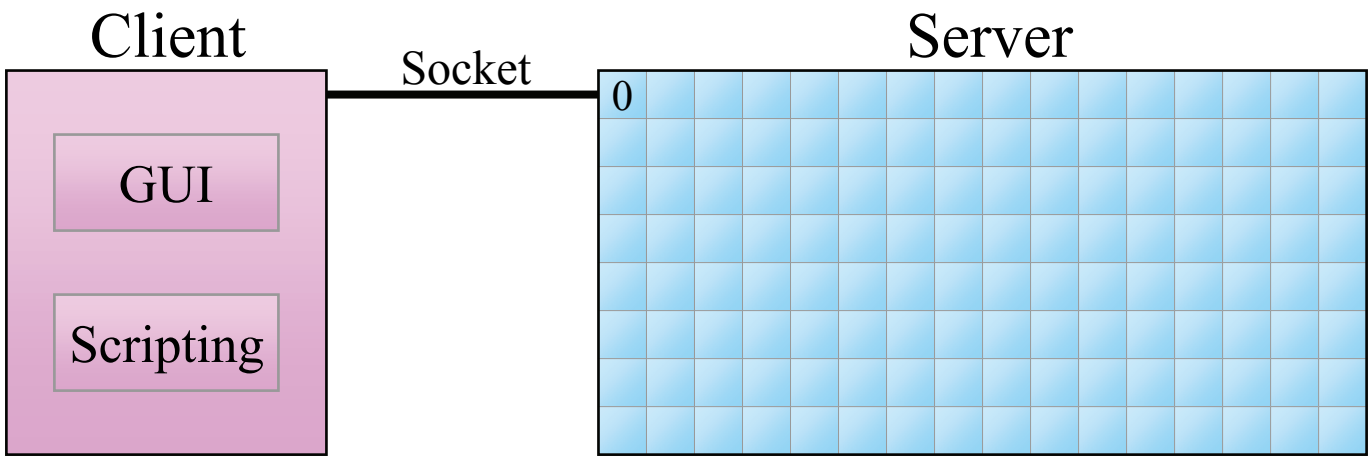
- **Cloud 9: A large parallel computer with direct access to data files and dedicated graphics hardware.**

Ideal Visualization Computer

- Cloud 9: A large parallel computer with direct access to data files and dedicated graphics hardware.
- Cloud 8.5: A large parallel computer with direct access to data files.
  - Hey, that could be the Cray thingy we ran the simulation on.

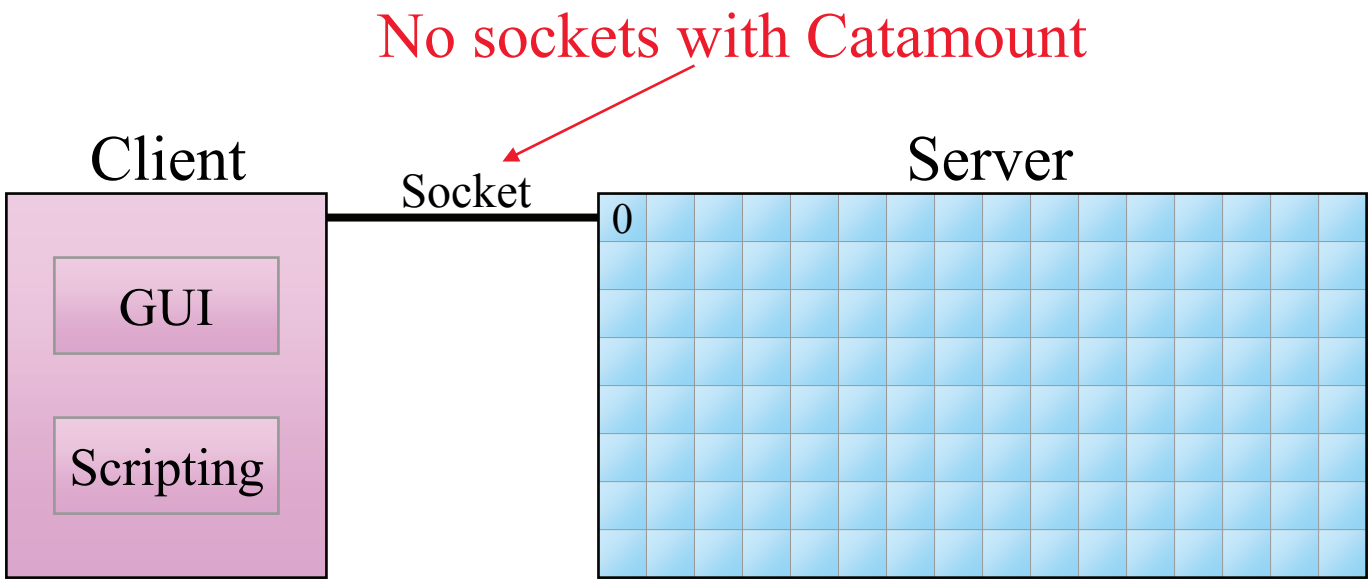
Problem with Interactive Visualization

- **For real time interaction, you need a remote connection to a GUI (usually through a socket).**



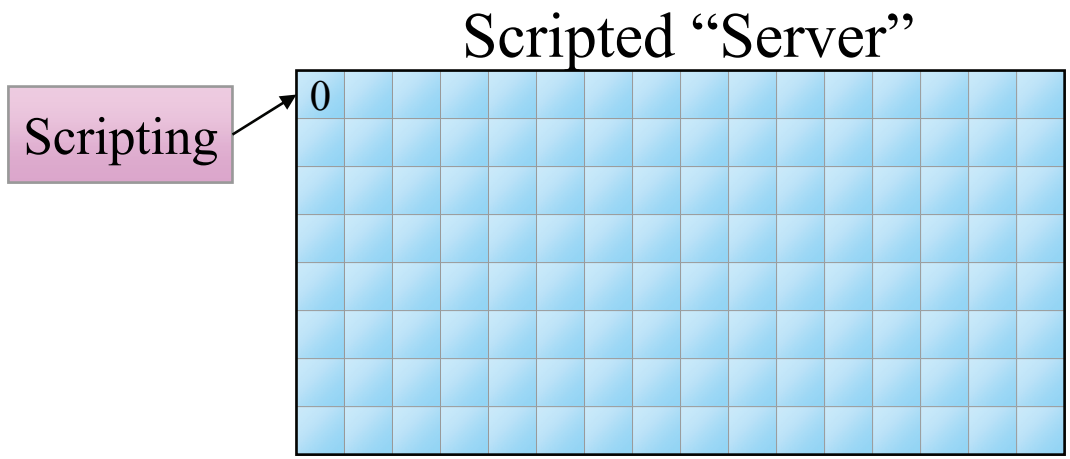
Problem with Interactive Visualization

- **For real time interaction, you need a remote connection to a GUI (usually through a socket).**



Problem with Interactive Visualization

- For real time interaction, you need a remote connection to a GUI (usually through a socket).
- We get around this problem by removing the client altogether and move the scripting over to the parallel "server," which is doing the heavy lifting.



Ideal Visualization Computer

- Cloud 9: A large parallel computer with direct access to data files and dedicated graphics hardware and on-demand interactive visualization.
- Cloud 8.5: A large parallel computer with direct access to data files and on-demand interactive visualization.
- Cloud 6.2: A large parallel computer with direct access to data files and scriptable visualization.

Why Not Portals?

- Previous work has solved the same problem using VisIt and portals.
- Even if we implemented this, we may not be able to use it.
  - Use of portals this way concerns administrators.
    - Extra network traffic.
    - Security issues (limit comm in/out compute nodes).
  - Allocated nodes sitting idle waiting for user input (common during interactive) is frowned upon.
    - Compute time is expensive.
  - Job queues cannot quickly start interactive jobs.
    - Compute time is expensive, nodes constantly busy.
- We could pursue this, but are unmotivated.

Implementation Details

- Python for Catamount.
- OpenGL for Catamount with no rendering hardware.
- Cross Compiling ParaView source.

Python for Catamount

- Initial port completed last year.
  - Reported implementation at CUG 2007.
- No dynamic libraries: compile modules statically.
  - Must know modules *a-priori*.
- Cannot directly execute cross-compiled executables.
  - Used yod to get return codes for the configure/build process.
- Other Catamount-specific build configuration.

Python for Catamount
Improvement for 2008: CMake build scripts

- Created CMake build scripts to use in place of Python's autoconf scripts.
  - Leverages cross-compile support implemented for ParaView source (discussed later).
- "Toolchain" files (small system-specific scripts) set up cross-compile build parameters.
  - Set up Catamount-specific configuration.
- Can pass configure/build runtime checks.
  - Don't need to use yod during build.
- Makes specifying modules to statically link easier to select.

OpenGL for Catamount

- Use Mesa 3D: the *de facto* software implementation of OpenGL.
  - Also contains code for using OpenGL without an X11 host.
- Mesa 3D build comes with cross-compile support.
  - We added cross-compile configuration for Catamount.
  - Our configuration is now included with Mesa 3D version 7.0.2 and later.

Cross Compiling ParaView

- ParaView uses the CMake build tool.
  - 12 months ago CMake had no special support for cross-compilation, and it was difficult.
- Added explicit cross-compilation controls.
  - Toolchain files make specifying target system parameters simple.
  - “Try run” queries are skipped.
    - CMake variables simplify hand-coding the info.
    - CMake creates an editable script that can be edited to the target system’s behavior to automate filling these variables.
      - A completed script is packaged with the ParaView source code.

Cross Compiling ParaView

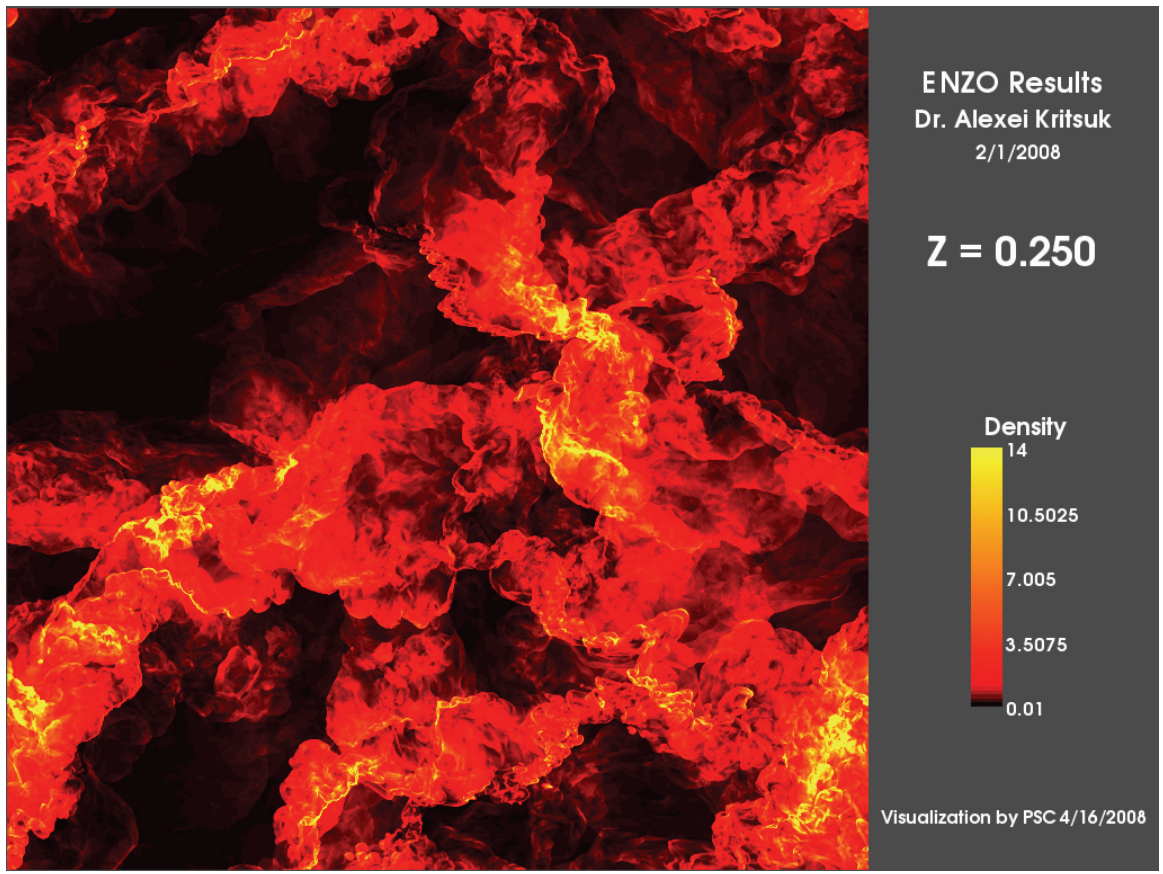
- ParaView build creates programs that generate new source code to compile.
  - Example: ParaView builds a program that parses VTK header files and generates C++ code for Python bindings.
  - These programs cannot be run locally when cross-compiling.
- Solution: build ParaView twice, once for the local machine and once for the target machine.
  - Target machine uses programs from the local build.
  - CMake options to build only these intermediate applications trivializes the local build time.

Example Usage

- ParaView deployed on Bigben Cray XT3 at Pittsburgh Supercomputing Center.
- Bigben used by institutions around the country.
  - Many users have no direct access to visualization resources local to PSC.
- Moving simulation results time consuming.
- Instead, perform visualization on Bigben.
- Visualization results typically much smaller than simulation results.

Hydrodynamic Turbulence and Star Formation in Molecular Clouds

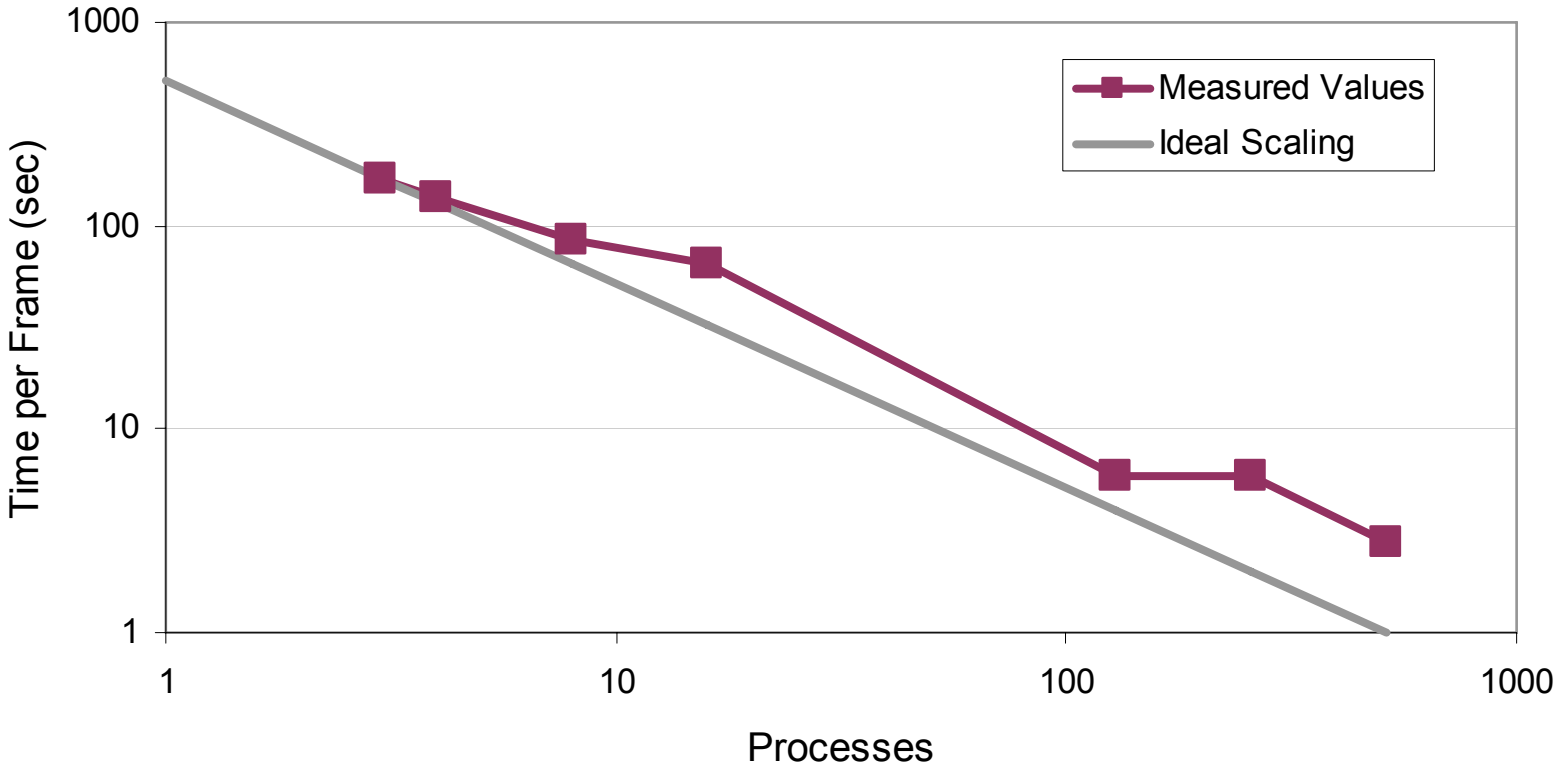
- Alexei Kritsuk, University of California, San Diego.
- Density on 2048^3 grid.



Sandia National Laboratories

Scaling

Time to Render and Save a Frame with Respect to Job Size



Sandia National Laboratories

Conclusions

- The port of ParaView to Cray XT3 ready for anyone wishing to use it.
- Scripted visualization is straightforward and scalable.
- Interactive visualization on the Cray XT3 is unavailable due to the lack of sockets.
  - Anyone interested could implement communication in and out of compute nodes if they have a use case (any takers?).
  - As Cray is moving to Compute Node Linux, which supports sockets, the point is probably moot.