Rocks Cluster

www.rocksclusters.org

Rocks is an open-source Linux cluster distribution that enables end users to easily build computational clusters.

CAWSC installed Rocks to solve problems faster than would be on a single-processor, high-speed computer.

Rocks Cluster

1) CAWSC put unwanted, hardware together for a new life as a Linux cluster. In this case, we installed a Rocks Cluster, developed and supported by SDSU (http://www.rocksclusters.org/wordpress/)

2) A disparate mix of 32- and 64-bit systems, 1 to 2 GHz, and 1 to 2 CPUs each.

3) As more hardware became available, it was simple matter to add to the cluster.

4) Only the frontend (master) node is on the network. All other nodes are behind the Frontend.

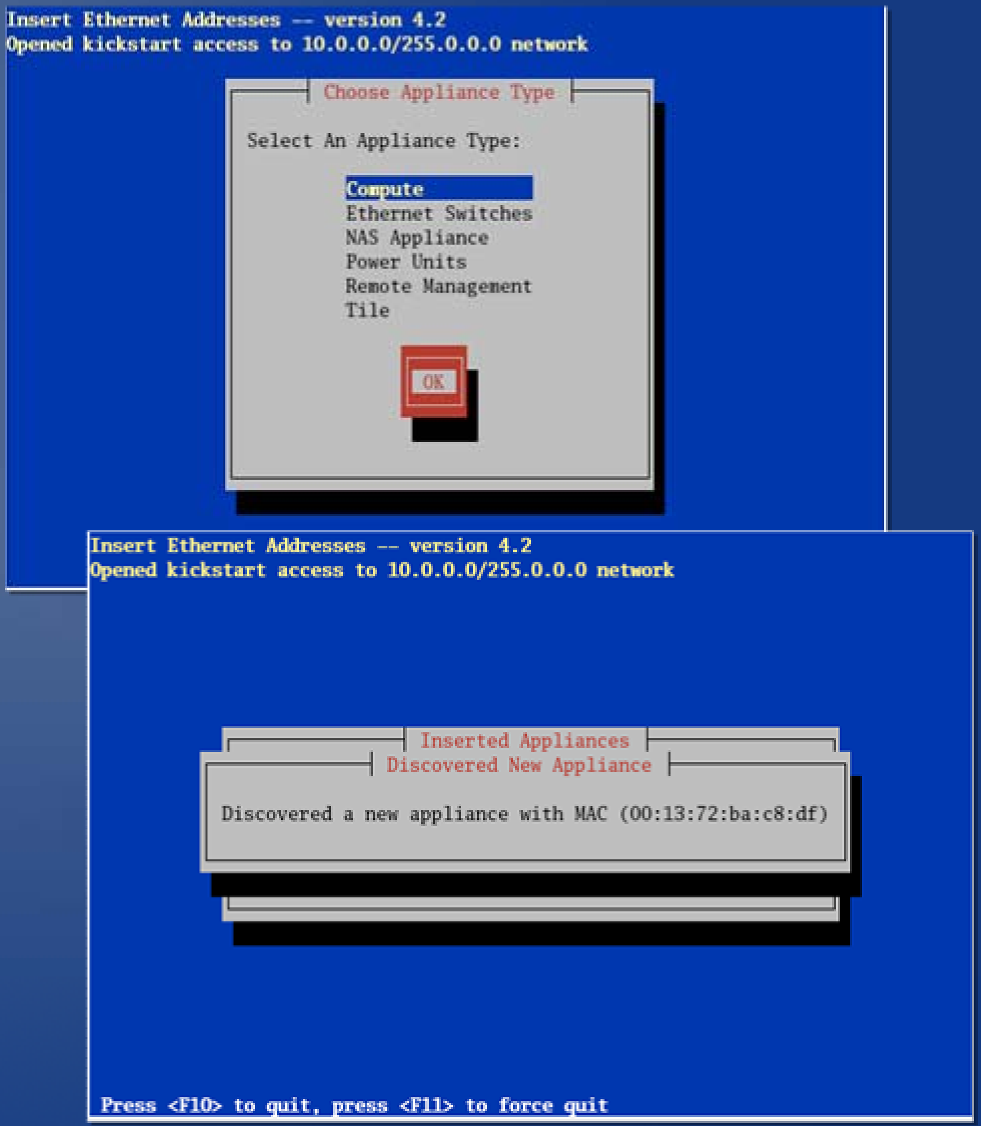
Adding Nodes is Quick and Simple

1) Plug in new node on cluster's private LAN,

2) Run 'insert-ethers‘ command on the Frontend system,

3) Insert Rocks Kernel (boot) CD into the newly added node and power it on.

4) The Frontend will detect the new client system, and give it a kickstart file. The kickstart will configure the client automatically.

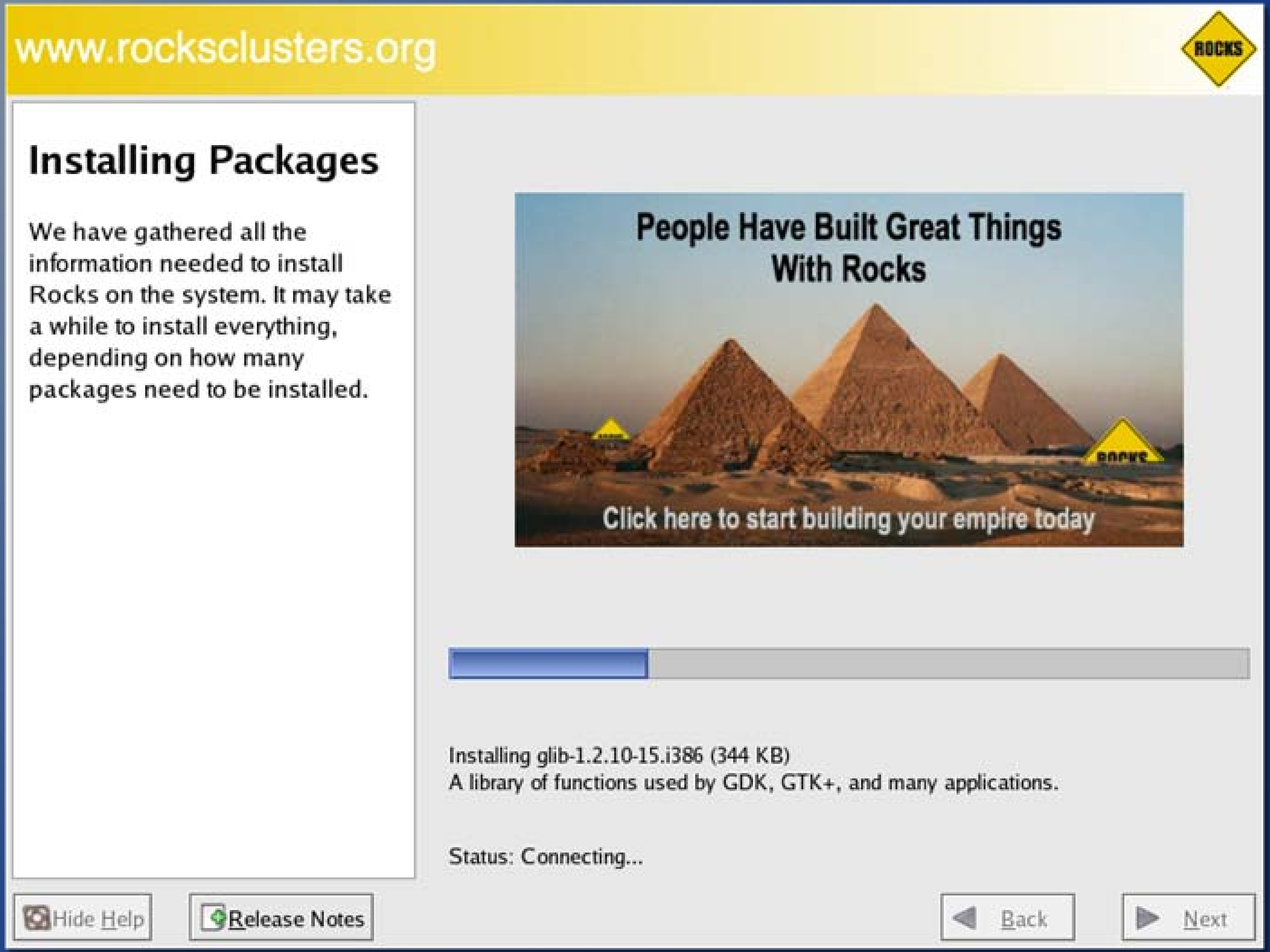


Adding Nodes is Quick and Simple
(cont'd)

When higher-end servers were retired from CAWSC field offices, they were added to the cluster in just a few hours.

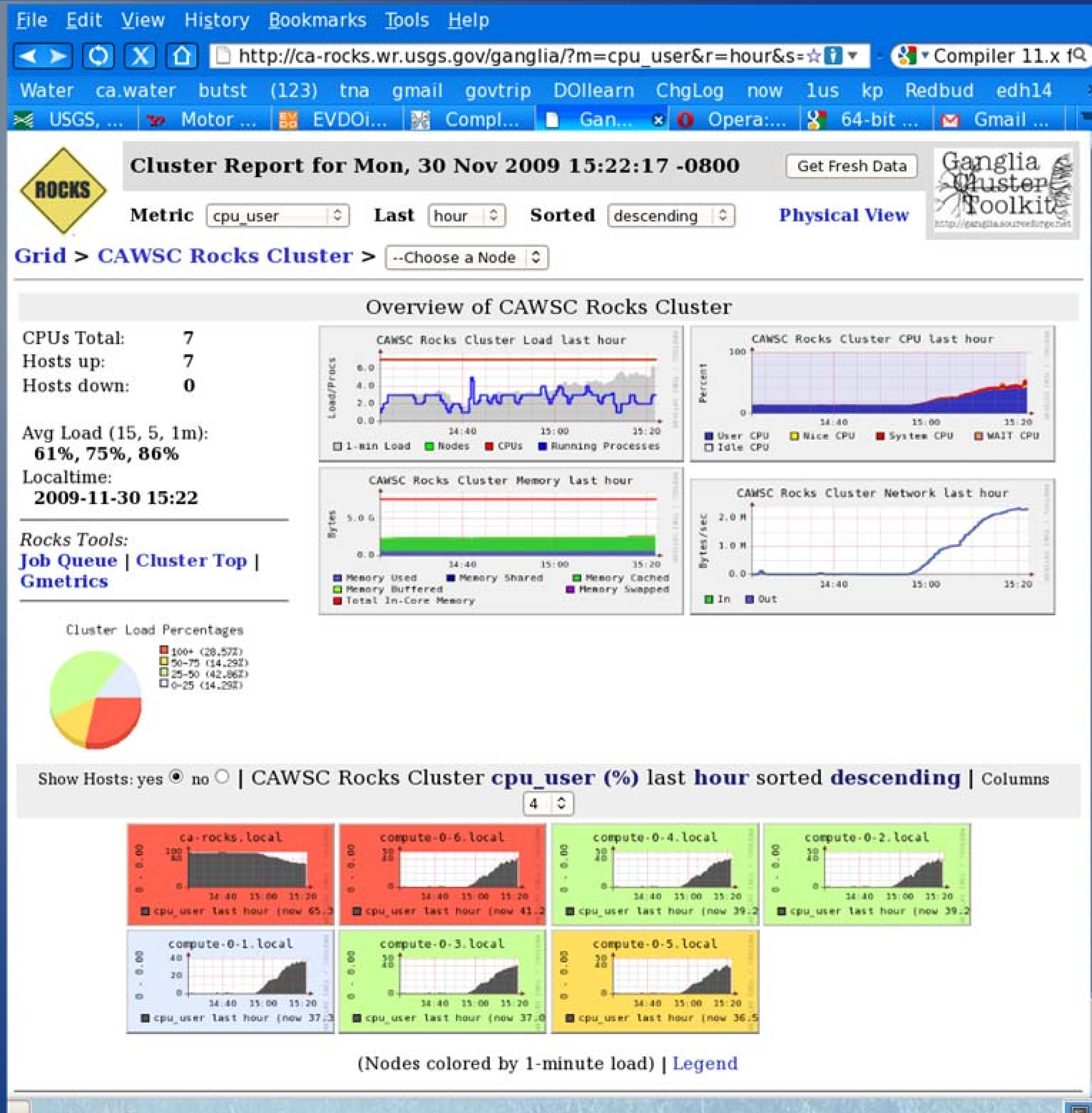
Rocks Cluster

New node installs automatically in 10-15 minutes.



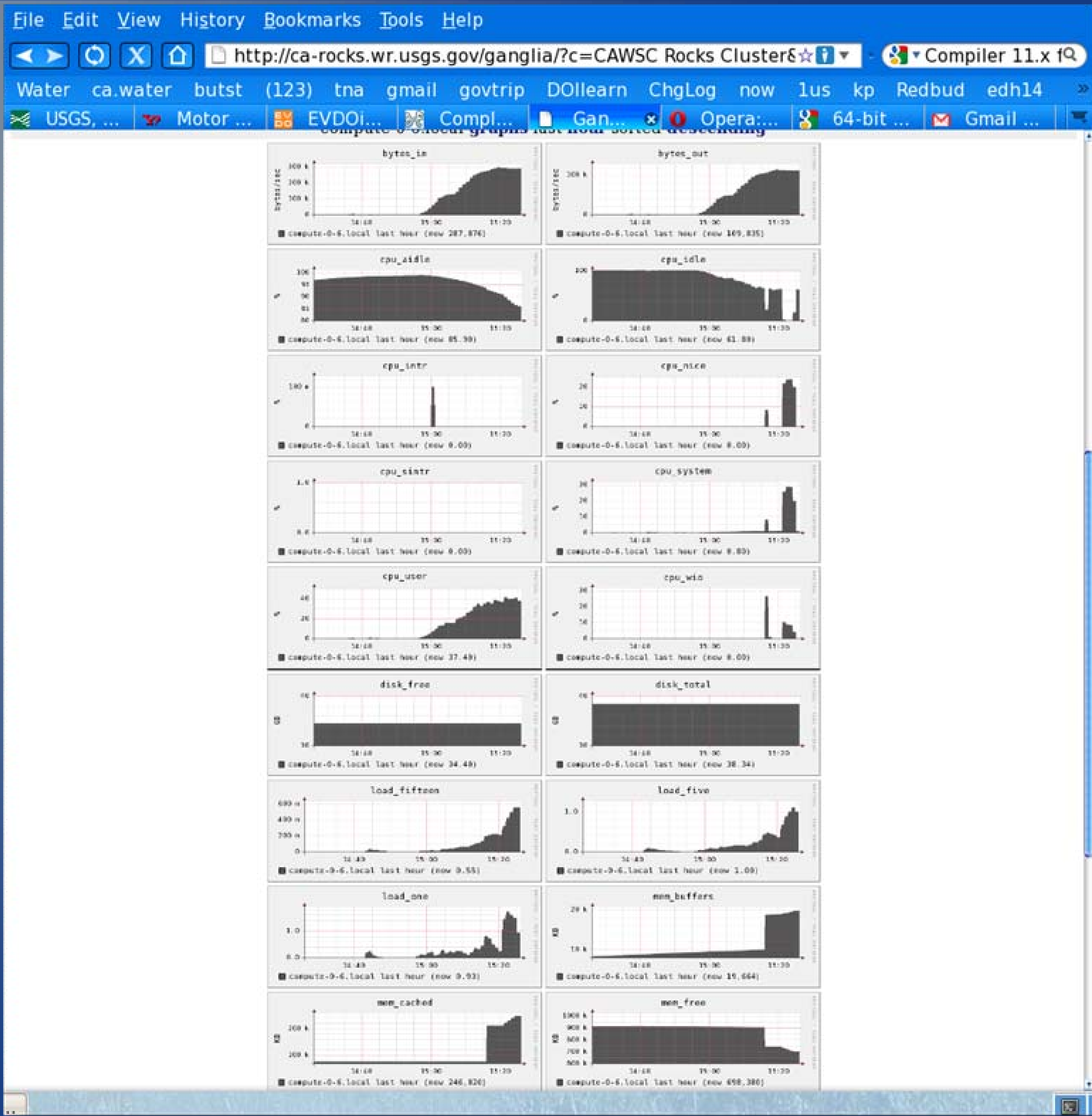
Web tool for process monitoring.

All CPU's.



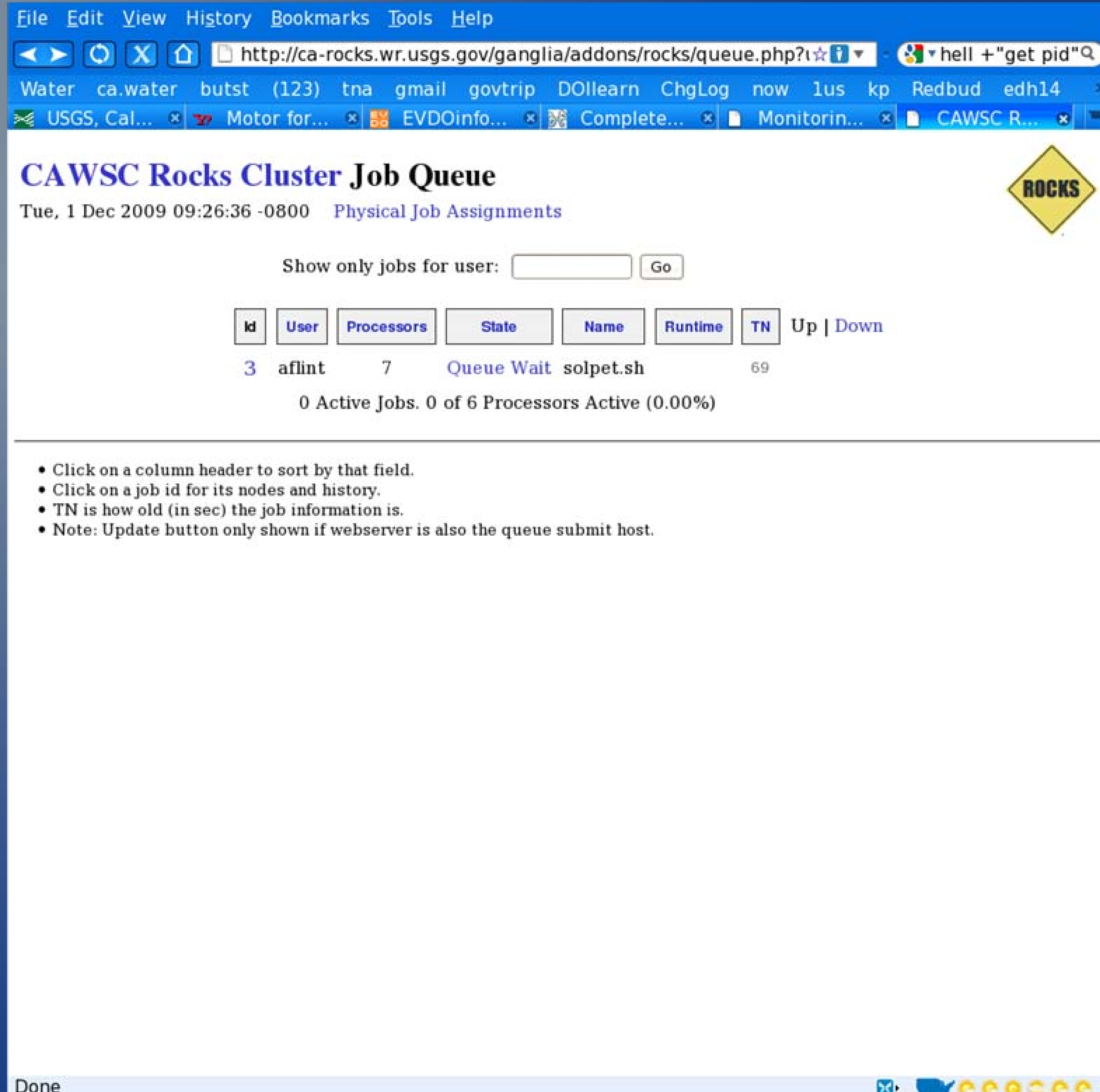
Web tool for process monitoring.

A particular node's statistics, top processes.



Web tool for process monitoring.

Job queue.



Programs on Cluster

Current programs running on the CAWSC Cluster

- MODFLOW – Three dimensional saturated zone flow model

- TOUGH2 – Three dimensional unsaturated zone flow model
  - iTOUGH2 – Inversion model for parameter estimation using TOUGH2

- SOLPET – Two dimensional solar radiation and evapotranspiration model used for regional recharge modeling

Symmetric vs. Asymmetric Processing

- Symmetric processing sends problem to multiple processors but does not move on to the next problem until all processors have finished (i.e. slowest node in the cluster limits overall processing speed),

- Asymmetric processing sends problem to multiple processors but individual processors can move on to their next problem without waiting for other processor to finish (i.e. slowest node in cluster does less total processing but does not itself significantly limit overall run time).

Some Compiler Specifics

- The INTEL Compiler can parallelize some existing codes to increase speed.
- Intel FORTRAN compiler comes with an optimization routine
- Intel VTune Performance Analyzer
  - Runs the application to determine:
    - Where the program spending its time
    - If there bottle necks than can be optimized
- Intel Cluster Tools
  - Analyzes the cluster performance

Some Specifics About Processing Speeds

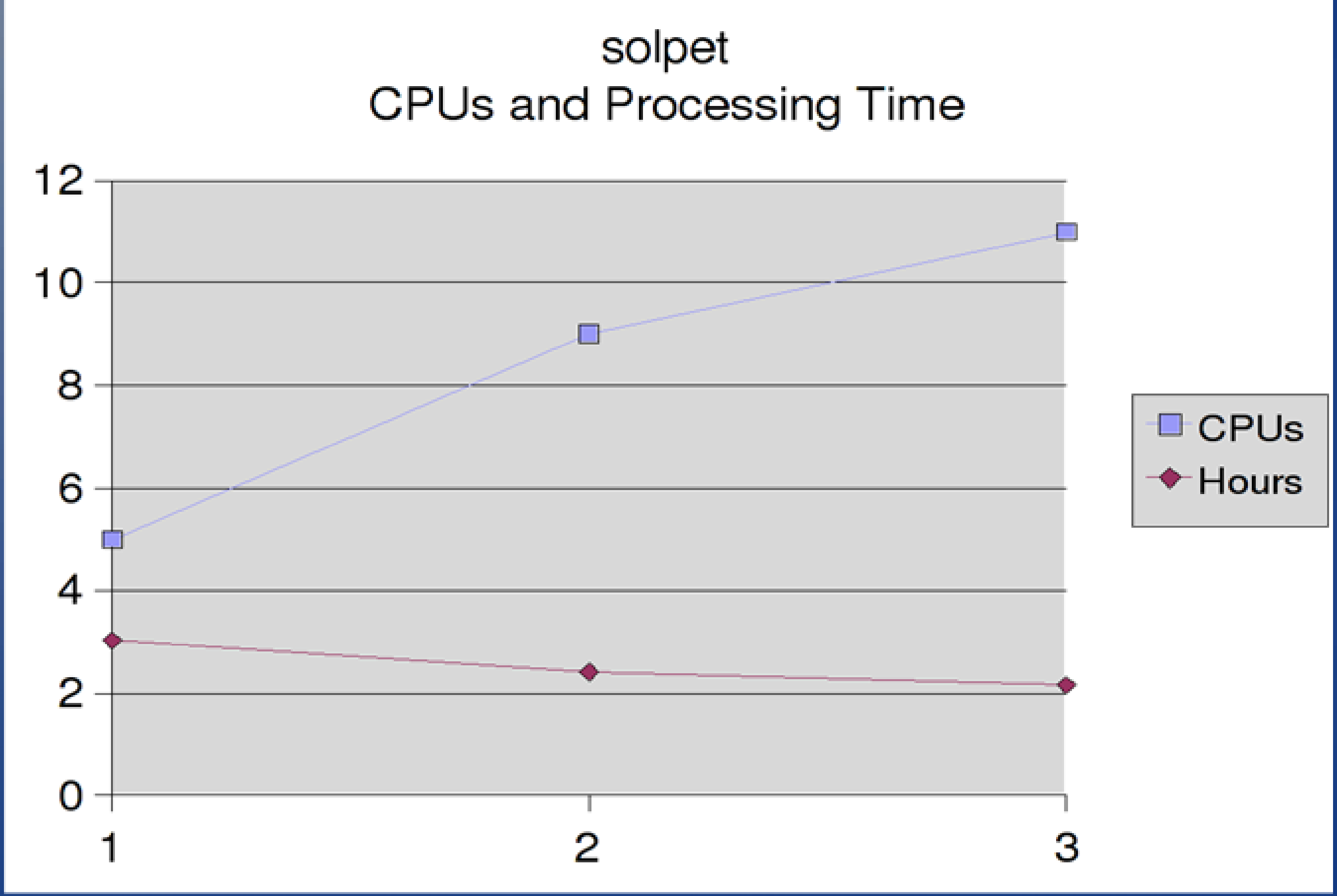
Dr. Alan Flint, Research Hydrologist, CAWSC, ran some performance tests, on both Linux and Windows clusters, in regard to the following cluster parameters:

- Number of CPUs, clock speed, and bus speed
- Symmetrical and Asymmetrical programming

CPUs & Processing Time

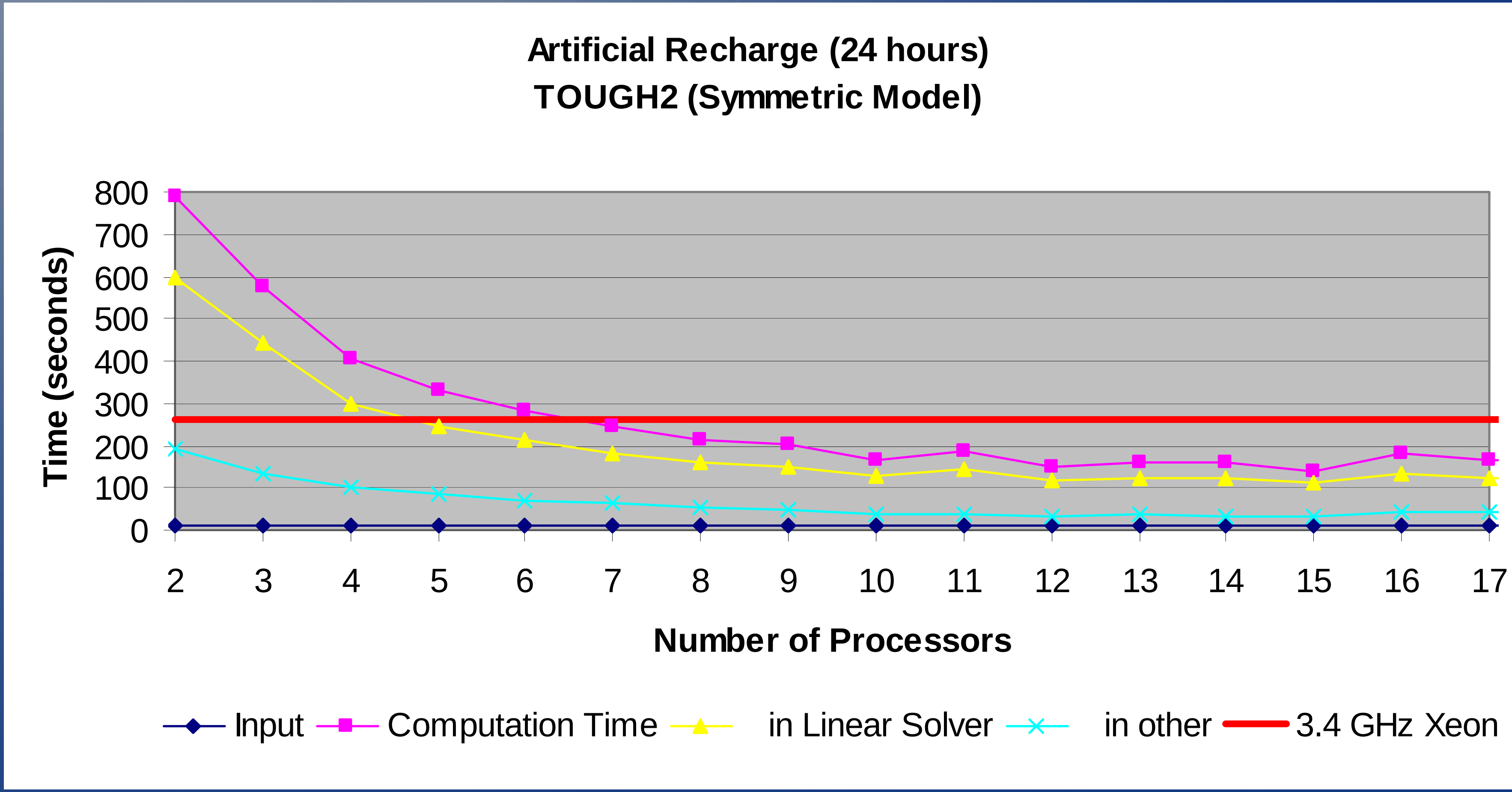
Though the number of CPUs is inversely proportional to processing time, it is not a linear relationship in either symmetric or asymmetric (parallel) programs.

Below are the results of a small test with an asymmetric model:



CPUs & Processing Time (cont'd)

Below is a more extensive test of a symmetric program by Dr. Flint, which shows diminishing returns.

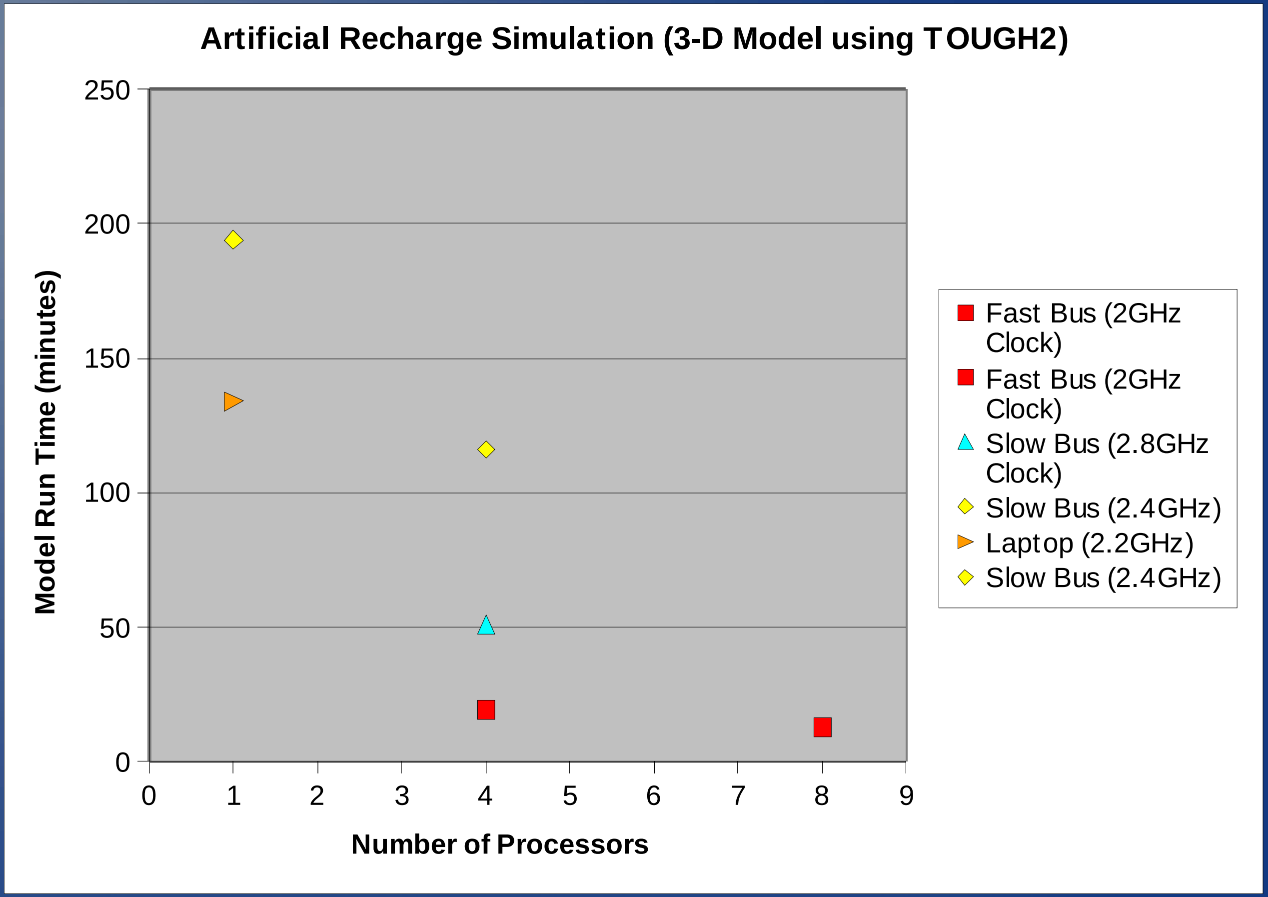


Bus vs. Clock Speed

Comparison:
1. Single, homogeneous system with fast bus speed: Windows system with dual quad-core (8 processors).

2. Multiple heterogenous systems: Windows systems made up of multiple dual-core, dual quad-core, or some combination.

Result: Bus speed makes a more significant difference over clock speed.



Examples of Need for Parallel Processing

Simulation of Artificial Recharge

Need to simulate 105 years of monthly potential evapotranspiration and 100 years of future potential evapotranspiration

- 12 hours on a single computer to run 1 month of simulation (26,493,613 model elements) or <u>6 days per year of</u> simulation
- 1 day <u>per year</u> on an 8-CPU Windows workstation using asymmetric processing

Examples of Need for Parallel Processing (cont'd)

Simulation of Artificial Recharge

Need to simulate 5 years of transient recharge and transport of nitrates and heat in 3-dimensional model

- 10 days of single computer to run 1 year of simulation (50,000 model elements)
- 5 days on cluster (15 really slow machines)

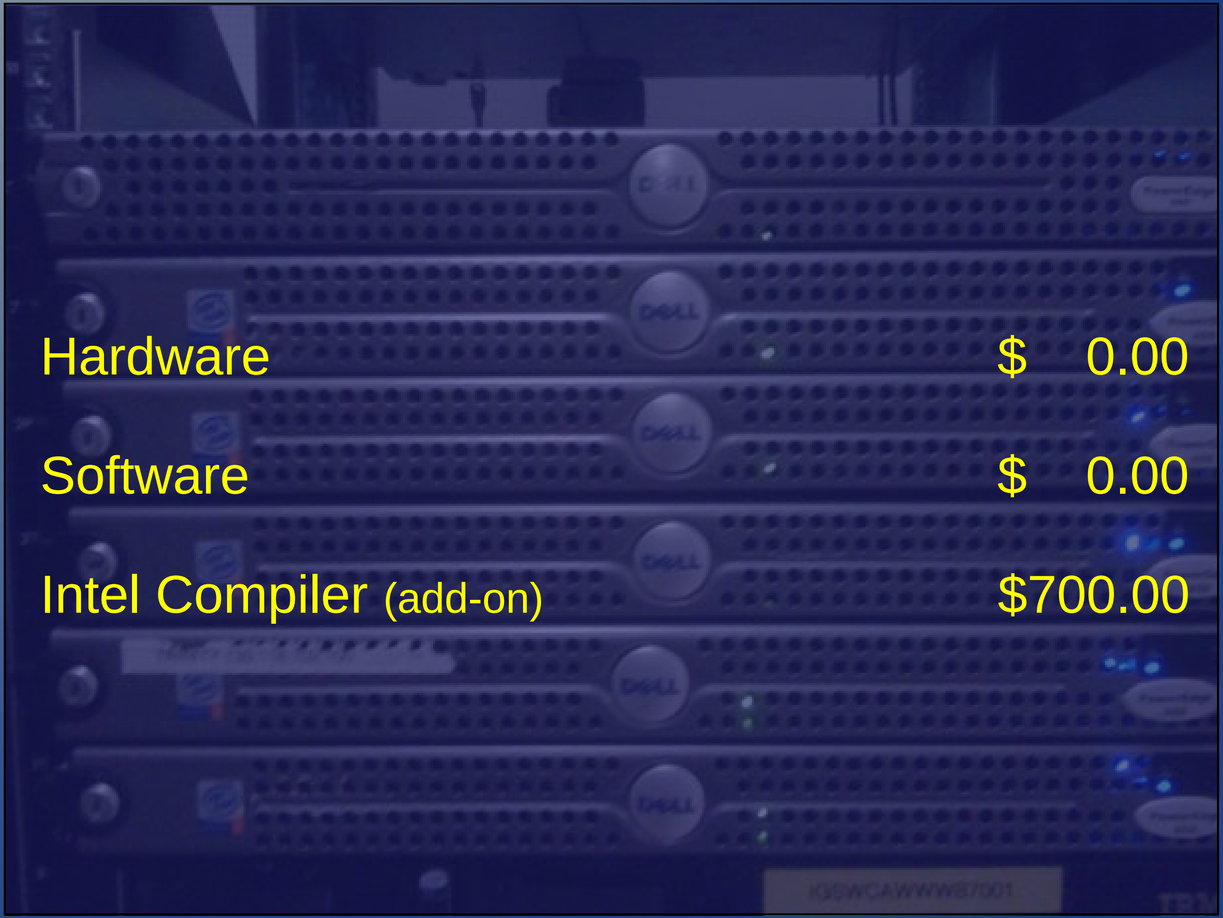
Simulation of Tributary Stream Temperatures

Simulate 4 years of daily minimum and maximum air temperature, cloudiness, radiation

- 60 days of single computer to run 4 years of simulation (45,006,610 model elements)
- Not yet implemented on the cluster

Rocks Cluster

COSTS



Hardware	$ 0.00
Software	$ 0.00
Intel Compiler (add-on)	$700.00