LLGrid: On-Demand Grid Computing with gridMatlab and pMatlab

Albert Reuther

MIT Lincoln Laboratory

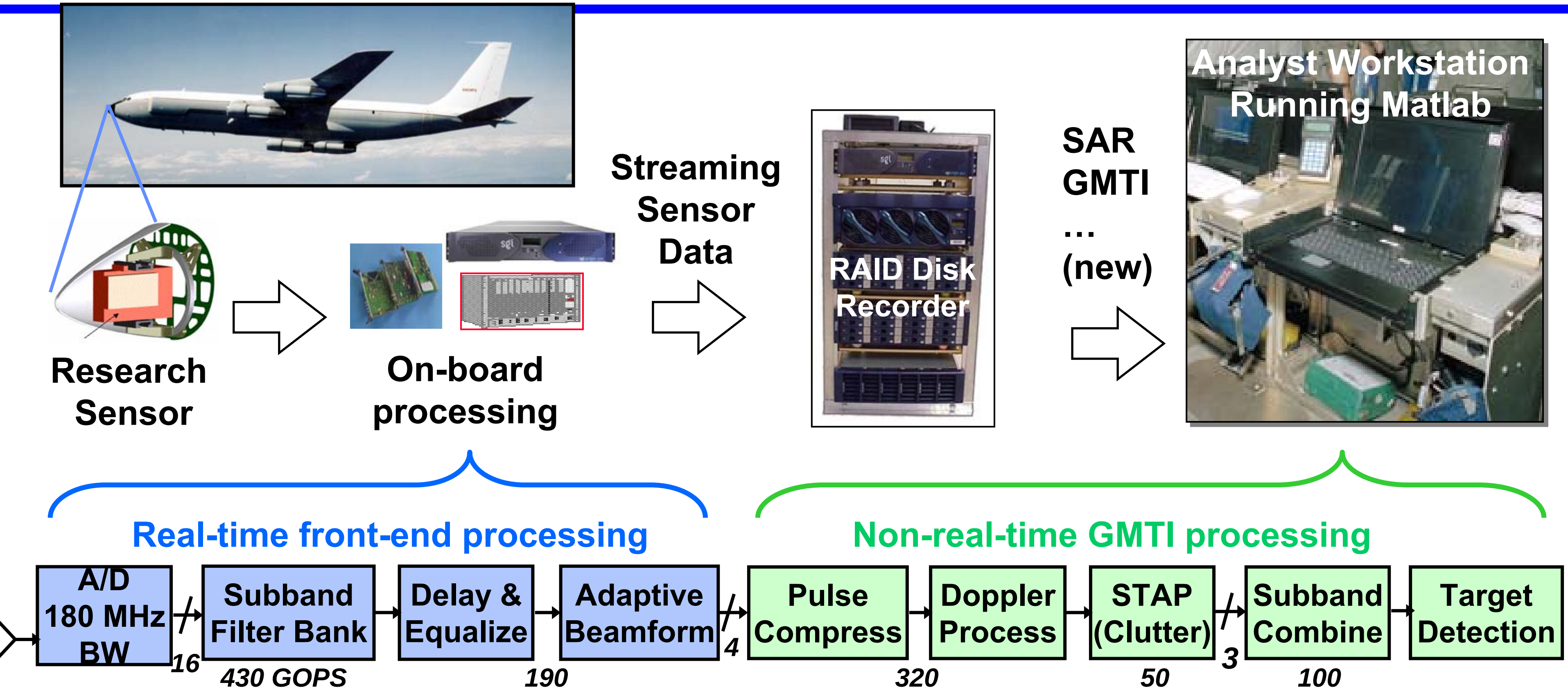
29 September 2004

This work is sponsored by the Department of the Air Force under Air Force contract F19628-00-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

LLGrid On-Demand Grid Computing System
Agenda

- **Introduction**
  - ***Example Application***
  - ***LLGrid Vision***
  - ***User Survey***
  - ***System Requirements***
- **LLGrid System**
- **Performance Results**
- **LLGrid Productivity Analysis**
- **Summary**

Example App: Prototype GMTI & SAR Signal Processing



Research Sensor

On-board processing

Streaming Sensor Data

RAID Disk Recorder

SAR
GMTI
…
(new)

Analyst Workstation Running Matlab

Real-time front-end processing

Non-real-time GMTI processing

A/D 180 MHz BW → 16 → Subband Filter Bank → Delay & Equalize → Adaptive Beamform → 4 → Pulse Compress → Doppler Process → STAP (Clutter) → 3 → Subband Combine → Target Detection

430 GOPS

190

320

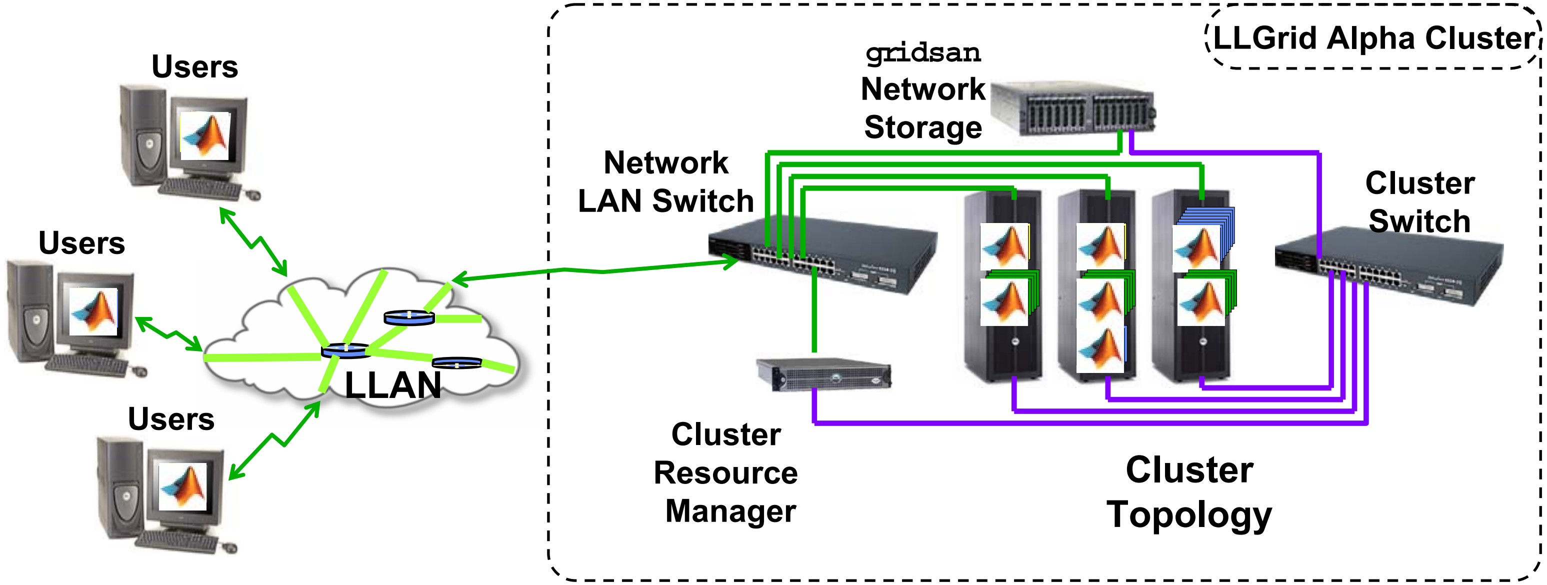
50

100

- Airborne research sensor data collected
- Research analysts develop signal processing algorithms in MATLAB® using collected sensor data
- Individual runs can last hours or days on single workstation

LLGrid

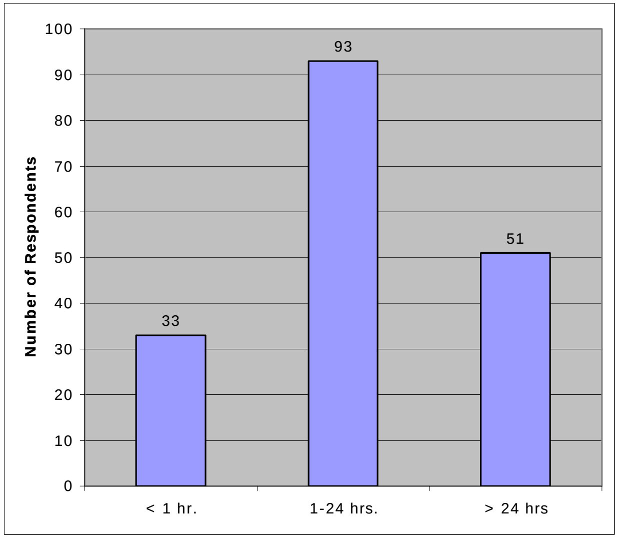
<u>Goal</u>: ***To develop a grid computing capability that makes it as easy to run parallel Matlab programs on grid as it is to run Matlab on own workstation.***

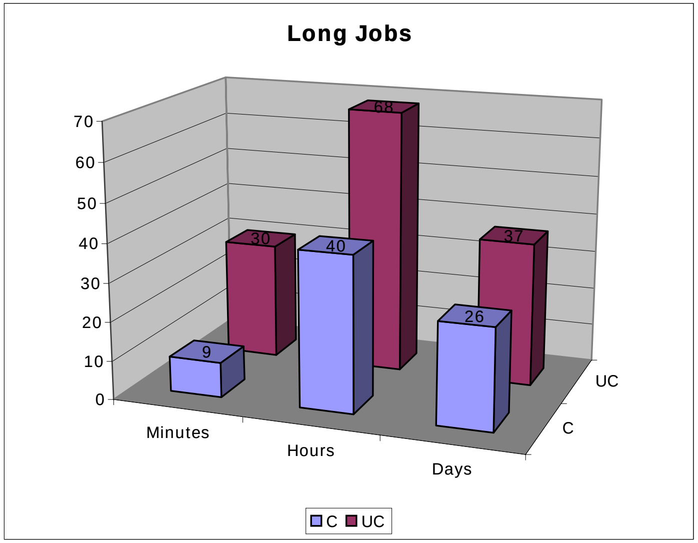


<u>Lab Grid Computing Components</u>

- Enterprise access to high throughput Grid computing
- Enterprise distributed storage

MATLAB® Users Survey

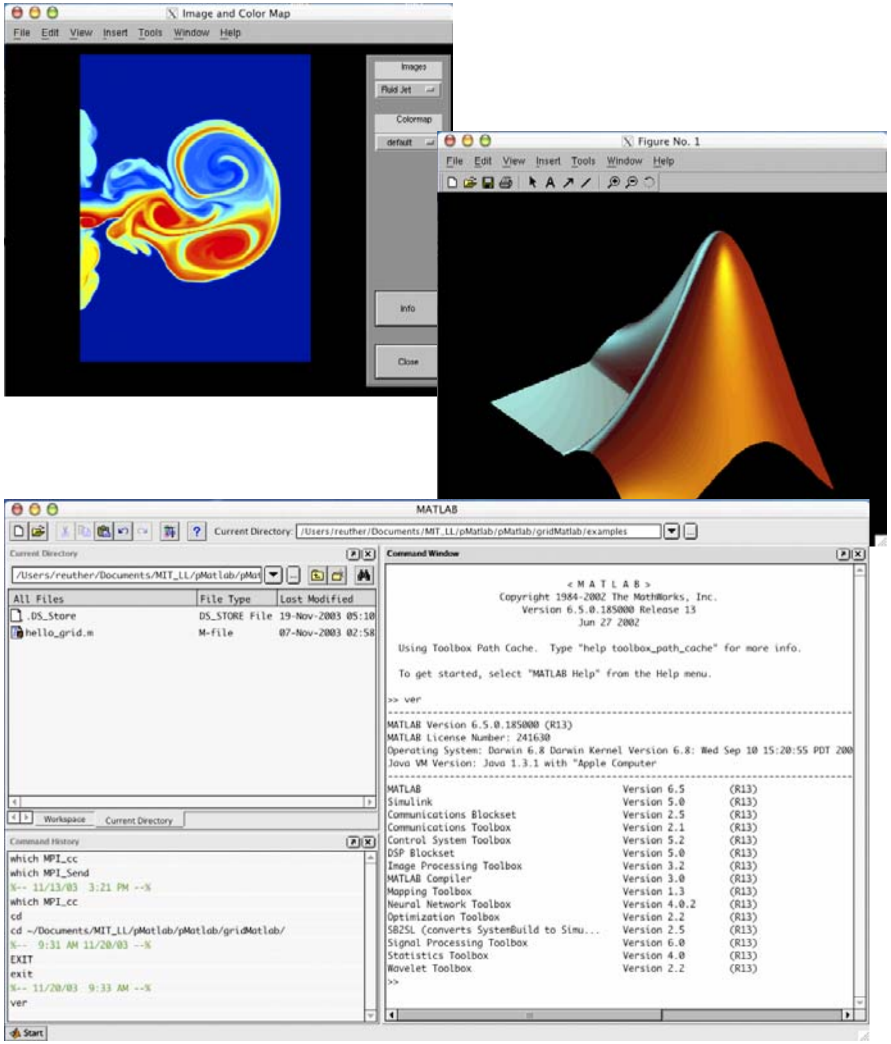




- Conducted survey of Lab staff
  - Do you run long MATLAB jobs?
  - How long do those jobs run (minutes, hours, or days)?
  - Are these jobs unclassified, classified, or both?
- Survey results:
  - 464 respondents
  - 177 answered "Yes" to question on whether they run long jobs
- Lincoln MATLAB users:
  - Engineers and scientists, generally not computer scientists
  - Little experience with batch queues, clusters, or mainframes
  - Solution must be easy to use

LLGrid User Requirements

- Easy to set up
  - First time user setup should be automated and take less than 10 minutes
- Easy to use
  - Using LLGrid should be the same as running a MATLAB job on user’s computer
- Compatible
  - Windows, Linux, Solaris, and MacOS X
- High Availability
- High Throughput for Medium and Large Jobs



Current Directory: /Users/reuther/Documents/MIT_LL/pMatlab/pMatlab/gridMatlab/examples

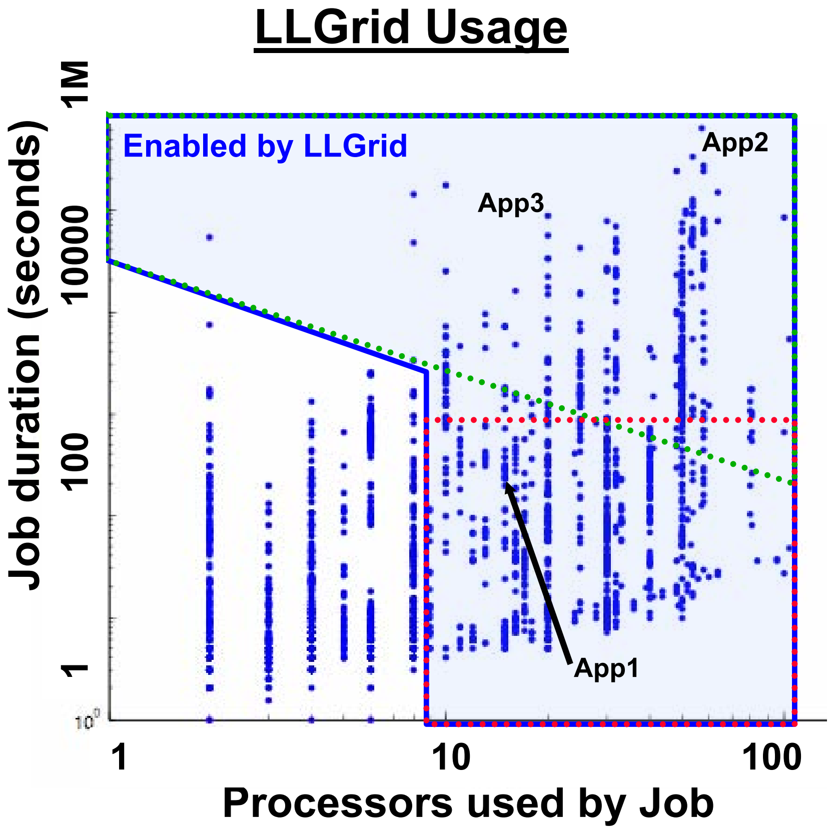
/Users/reuther/Documents/MIT_LL/pMatlab/pMat

All Files	File Type	Last Modified
.DS_Store	DS_STORE File	19-Nov-2003 05:10
hello_grid.m	M-file	07-Nov-2003 02:58

```
                         < M A T L A B >
             Copyright 1984-2002 The MathWorks, Inc.
               Version 6.5.0.185000 Release 13
                         Jun 27 2002

  Using Toolbox Path Cache.  Type "help toolbox_path_cache" for more info.

  To get started, select "MATLAB Help" from the Help menu.

>> ver
-------------------------------------------------------------------------
MATLAB Version 6.5.0.185000 (R13)
MATLAB License Number: 241630
Operating System: Darwin 6.8 Darwin Kernel Version 6.8: Wed Sep 10 15:20:55 PDT 200
Java VM Version: Java 1.3.1 with "Apple Computer
-------------------------------------------------------------------------
MATLAB                                    Version 6.5       (R13)
Simulink                                  Version 5.0       (R13)
Communications Blockset                   Version 2.5       (R13)
Communications Toolbox                    Version 2.1       (R13)
Control System Toolbox                    Version 5.2       (R13)
DSP Blockset                              Version 5.0       (R13)
Image Processing Toolbox                  Version 3.2       (R13)
MATLAB Compiler                           Version 3.0       (R13)
Mapping Toolbox                           Version 1.3       (R13)
Neural Network Toolbox                    Version 4.0.2     (R13)
Optimization Toolbox                      Version 2.2       (R13)
SB2SL (converts SystemBuild to Simu...    Version 2.5       (R13)
Signal Processing Toolbox                 Version 6.0       (R13)
Statistics Toolbox                        Version 4.0       (R13)
Wavelet Toolbox                           Version 2.2       (R13)
>>
```

```
which MPI_cc
which MPI_Send
%-- 11/13/03  3:21 PM --%
which MPI_cc
cd
cd ~/Documents/MIT_LL/pMatlab/pMatlab/gridMatlab/
%--  9:31 AM 11/20/03 --%
EXIT
exit
%-- 11/20/03  9:33 AM --%
ver
```

LLgrid Usage



>8 CPU hours - Infeasible on Desktop

>8 CPUs - Requires On-Demand Parallel Computing

3500 jobs, 3600 CPU Days
December 03 – June 04

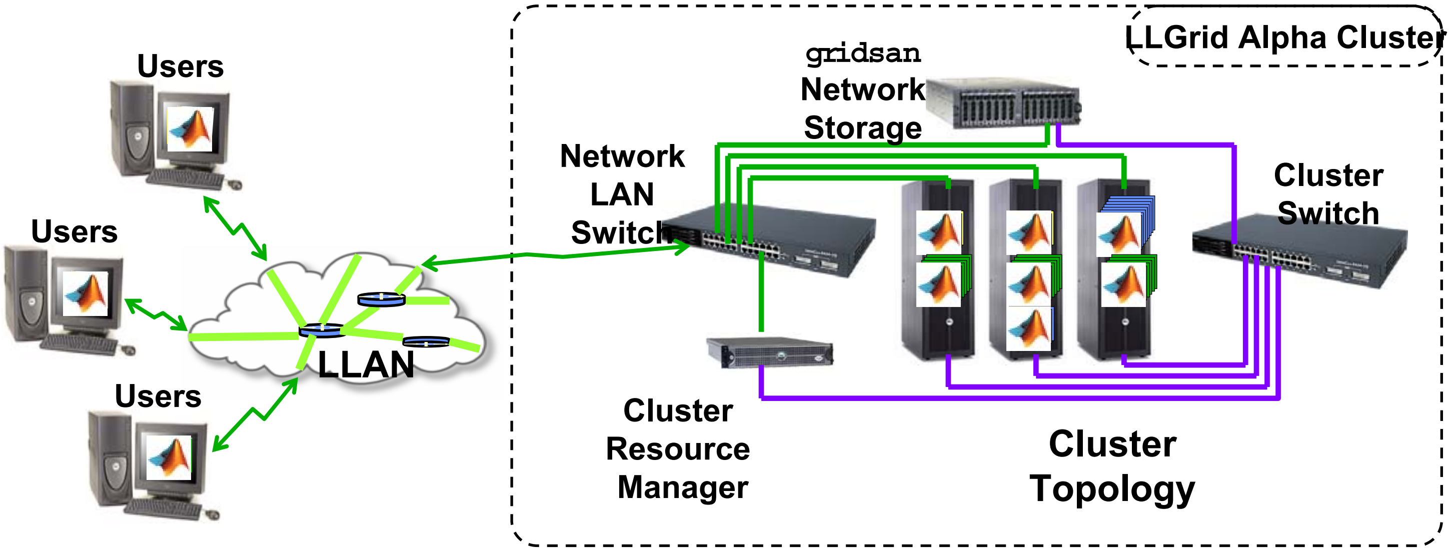
- Allowing Lincoln staff to effectively use parallel computing daily from their desktop
  - Interactive parallel computing
  - 160 CPUs, 25 Users, 11 Groups
- Extending the current space of data analysis and simulations that Lincoln staff can perform
  - Jobs requiring rapid turnaround
    App1: Weather Radar Signal Processing Algorithms
  - Jobs requiring many CPU hours
    App2: Hyperspectral Image Analysis
    App3: Laser Propagation Simulation

LLGrid On-Demand Grid Computing System
Agenda

- Introduction
- **LLGrid System**
- Performance Results
- LLGrid Productivity Analysis
- Summary

- *Overview*
- *Hardware*
- *Management Scripts*
- *MatlabMPI*
- *pMatlab*
- *gridMatlab*

LLGrid Alpha Cluster



Key Innovations:

pMatlab - Global array semantics for parallel MATLAB

gridMatlab - User's computer is transparently included into LLGrid

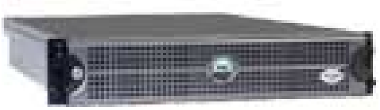
- User never logs into LLGrid (only mounts file system)

Alpha Grid Hardware

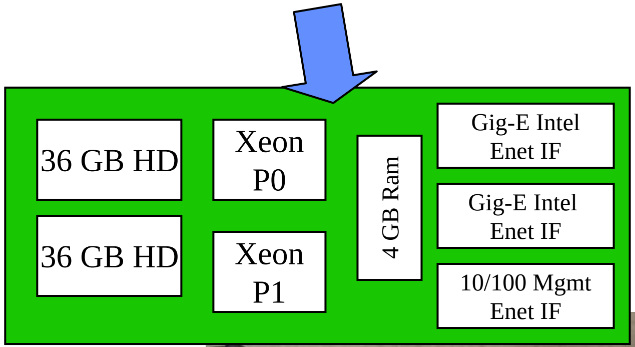
80 Nodes + Head Node - 160+2 Processors, 320 GB RAM

Nodes:

DELL 2650 &



DELL 1750



36 GB HD

36 GB HD

Xeon P0

Xeon P1

4 GB Ram

Gig-E Intel Enet IF

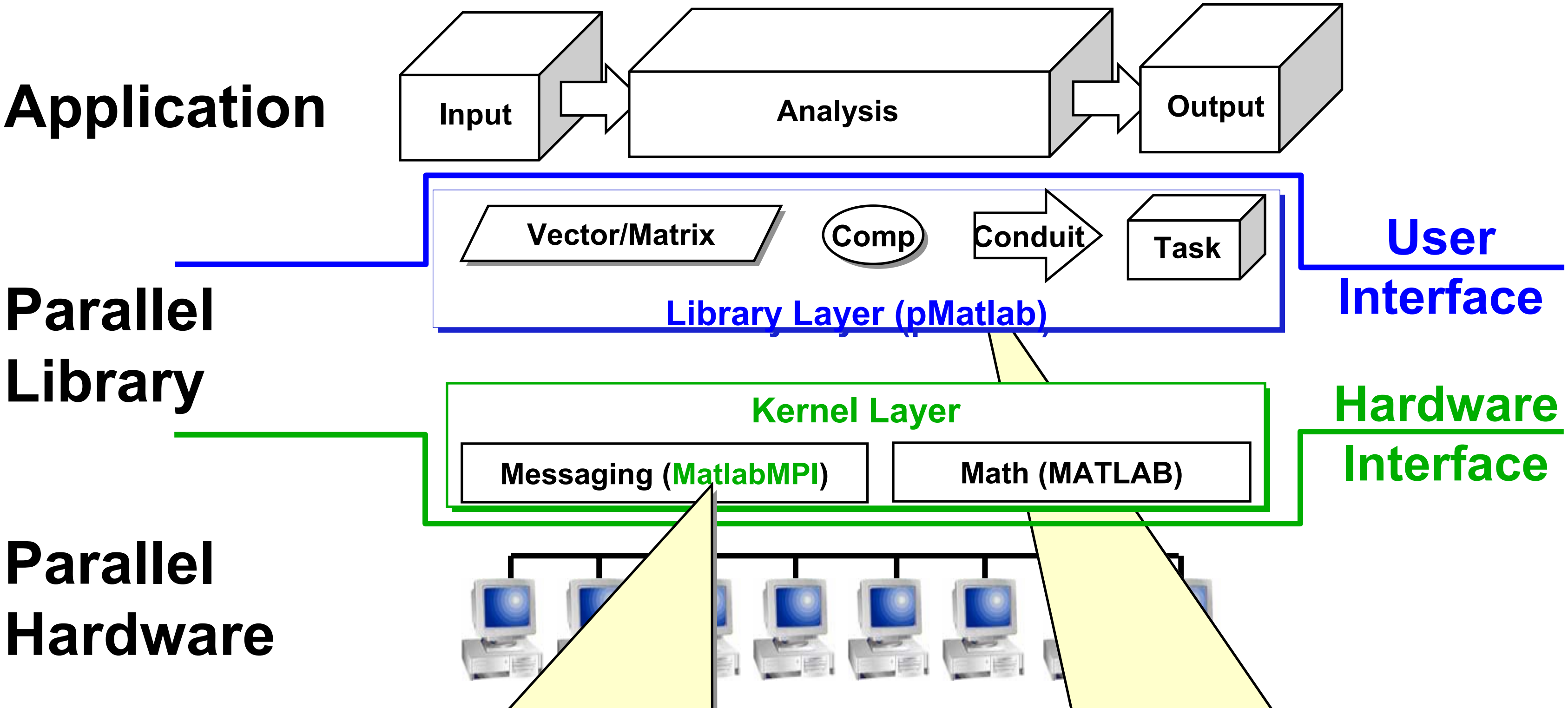
Gig-E Intel Enet IF

10/100 Mgmt Enet IF

- Dual 2.8 & 3.06 GHz Xeon (P4)
- 400 & 533 MHz front-side bus
- 4 GB RAM memory
- Two 36 GB SCSI hard drives
- 10/100 Mgmt Ethernet interface
- Two Gig-E Intel interfaces
- Running Red Hat Linux

- Commodity Hardware
- Commodity OS
- High Availablity

pMatlab Software Layers



- Can build a parallel library with a few messaging primitives
- MatlabMPI provides this messaging capability:

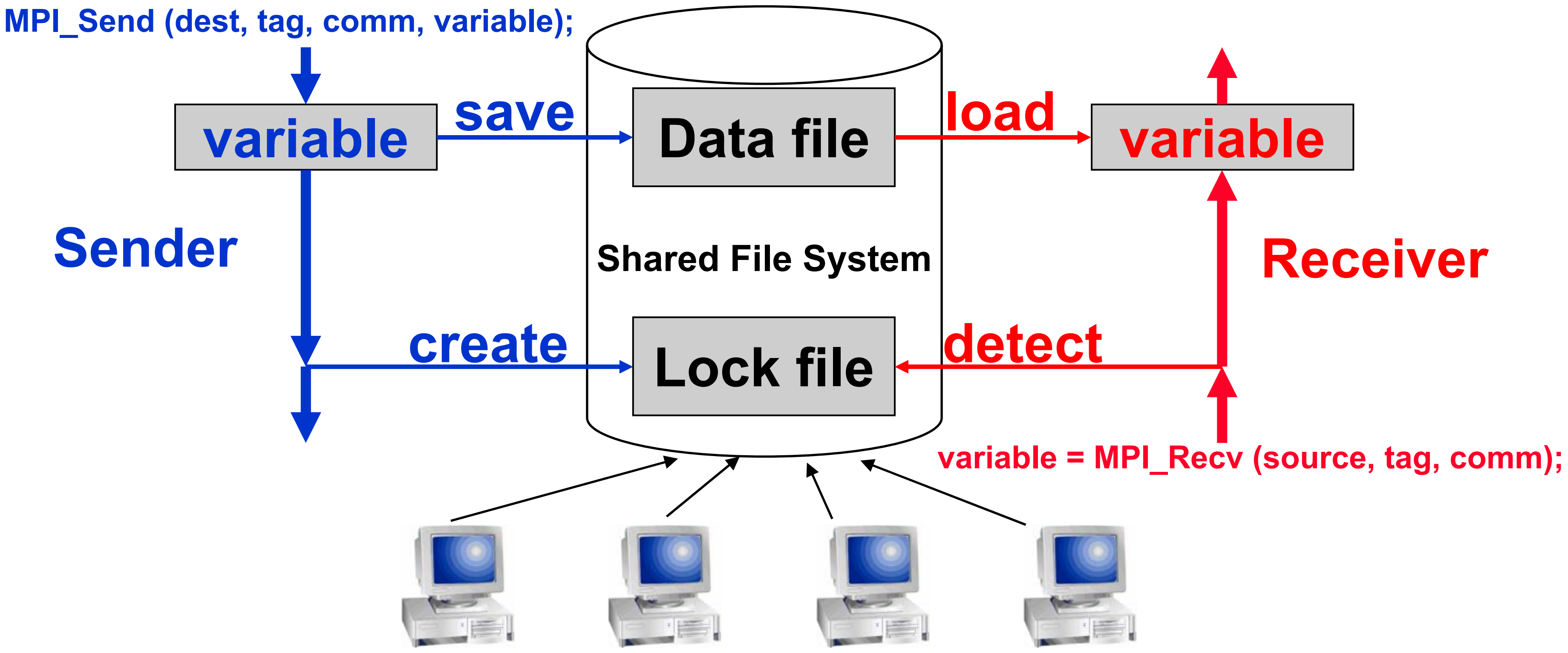
```
MPI_Send(dest,comm,tag,X);
X = MPI_Recv(source,comm,tag);
```

- Can build a application with a few parallel structures and functions
- pMatlab provides Global Array Semantic via parallel arrays and functions

```
X = ones(n,mapX);
Y = zeros(n,mapY);
Y(:,:) = fft(X);
```

MatlabMPI:
Point-to-point Communication

- Any messaging system can be implemented using file I/O
- File I/O provided by MATLAB via load and save functions
  - Takes care of complicated buffer packing/unpacking problem
  - Allows basic functions to be implemented in ~250 lines of MATLAB code

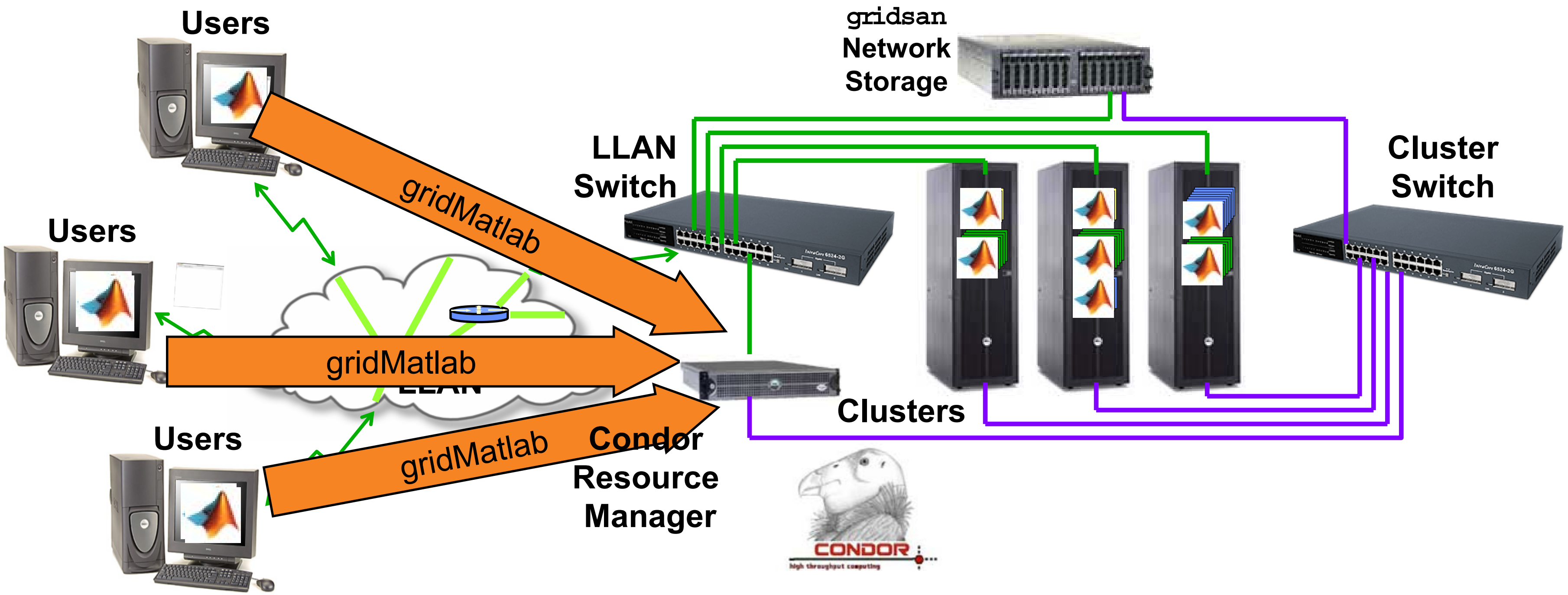


MPI_Send (dest, tag, comm, variable);

variable = MPI_Recv (source, tag, comm);

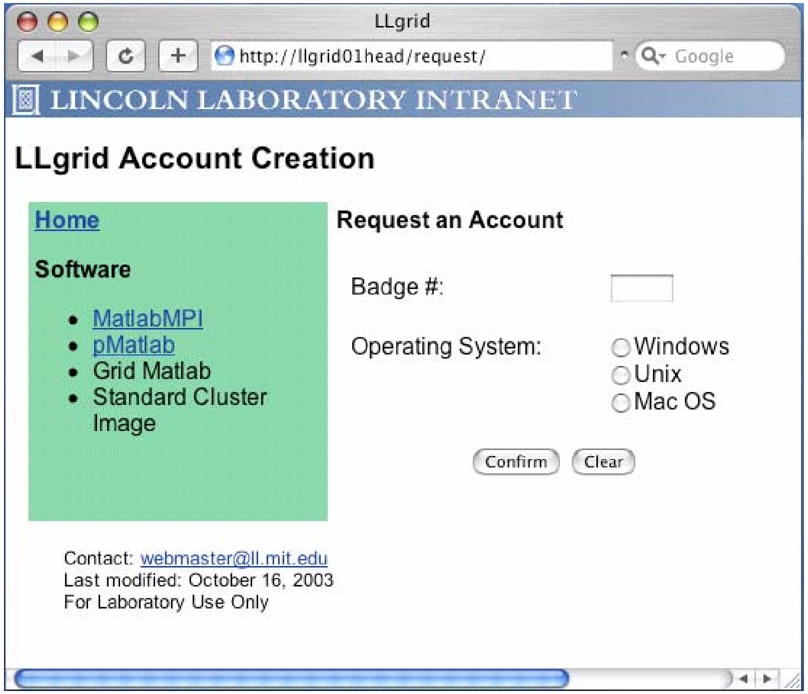
- **Sender** saves variable in Data file, then creates Lock file
- **Receiver** detects Lock file, then loads Data file

gridMatlab: Enable Grid Computing



- Transparent interface between pMatlab (MatlabMPI) and resource mngr.
- User’s computer is included in LLGrid for own job only
- Amasses requested resources for on-demand, interactive job computation
- Handles all communication with the Condor resource manager (including submission file, launch scripts, and job aborts)
- User never interacts with queue system directly

LLGrid Account Creation



LLgrid Account Creation

Home

Software

- MatlabMPI
- pMatlab
- Grid Matlab
- Standard Cluster Image

Request an Account

Badge #:

Operating System: ○ Windows ○ Unix ○ Mac OS

Confirm Clear

Contact: webmaster@ll.mit.edu
Last modified: October 16, 2003
For Laboratory Use Only

LLGrid Account Setup

- Go to Account Request web page; Type Badge #, Click "Create Account"
- Account is created and mounted on user's computer
- Get User Setup Script
- Run User Setup Script
- User runs sample job

• **Account Creation Script** (Run on LLGrid)	– Creates account on gridsan – Creates NFS & SaMBa mount points – Creates cross-mount communication directories
• **User Setup Script** (Run on User's Computer)	– Mounts gridsan – Creates SSH keys for grid resource access – Links to MatlabMPI, pMatlab, & gridMatlab source toolboxes – Links to MatlabMPI, pMatlab, & gridMatlab example scripts

Account Setup Steps

Typical Supercomputing Site Setup

- Account application/ renewal *[months]*
- Resource discovery *[hours]*
- Resource allocation application/renewal *[months]*
- Explicit file upload/download (usually ftp) *[minutes]*
- Batch queue configuration *[hours]*
- Batch queue scripting *[hours]*
- Differences between control vs. compute nodes *[hours]*
- Secondary storage configuration *[minutes]*
- Secondary storage scripting *[minutes]*
- Interactive requesting mechanism *[days]*
- Debugging of example programs *[days]*
- Documentation system *[hours]*
- Machine node names *[hours]*
- GUI launch mechanism *[minutes]*
- Avoiding user contention *[years]*

LLGrid Account Setup *[minutes]*

- Go to Account Request web page; Type Badge #, Click "Create Account"
- Account is created and mounted on user's computer
- Get User Setup Script
- Run User Setup Script
- User runs sample job

MIT Lincoln Laboratory

MathWorks Distributed MATLAB (DML)

“Dear MATLAB user,

“This is an invitation to participate in an upcoming Beta Test for the Distributed MATLAB product. This will be available on the following platforms, Win 2000, Win NT, WIN XP, and Linux.

“The goal of this first release of Distributed MATLAB is to address the requirements of coarse-grain applications, in which the same MATLAB algorithm is executed in remote MATLAB sessions on different data sets without communication or data exchange between sessions.”

– From DML beta email

Lincoln has installed DML and is testing it to determine how it integrates with LLGrid technologies.

LLGrid On-Demand Grid Computing System
Agenda

- Introduction
- LLGrid System
- **Performance Results**
- LLGrid Productivity Analysis
- Summary

Performance: Time to Parallelize

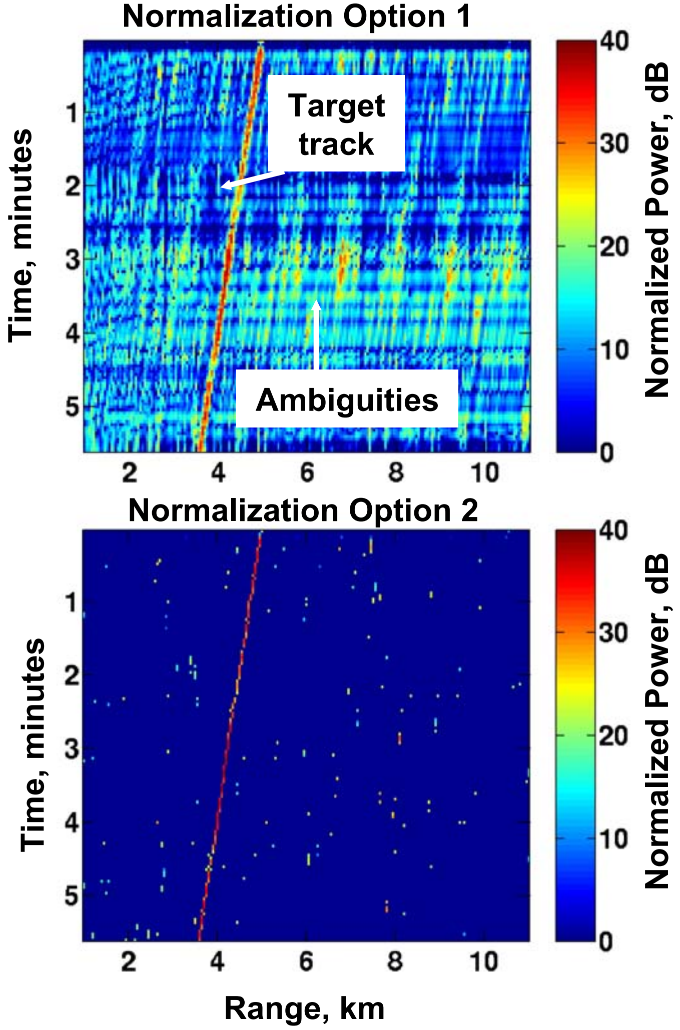
Important Considerations

Description	Serial Code Time	Time to Parallelize	Applications that Parallelization Enables
Missile & Sensor BMD Sim. (BMD)	2000 hours	8 hours	Discrimination simulations Higher fidelity radar simulations
First-principles LADAR Sim. (Ladar)	1300 hours	1 hour	Speckle image simulations Aimpoint and discrimination studies
Analytic TOM Leakage Calc. (Leak)	40 hours	0.4 hours	More complete parameter space sim.
Hercules Metric TOM Code (Herc)	900 hours	0.75 hours	Monte carlo simulations
Coherent laser propagation sim. (Laser)	40 hours	1 hour	Reduce simulation run time
Polynomial coefficient approx. (Coeff)	700 hours	8 hours	Reduced run-time of algorithm training
Ground motion tracker indicator computation simulator (GMTI)	600 hours	3 hours	Reduce evaluation time of larger data sets
Automatic target recognition (ATR)	650 hours	40 hours	Ability to consider more target classes Ability to generate more scenarios
Normal Compositional Model for Hyper-spectral Image Analysis (HSI)	960 hours	6 hours	Larger datasets of images

pMatlab Application to 3D Spatial Normalization

- A Lincoln group is developing normalization algorithms for 3D matched-field (MFP) beamformers
- Sponsored by DARPA-ATO under Robust Passive Sonar program
- Large search space (O(1e7) cells) makes normalizer evaluation on processed data difficult
- pMatlab code enabled rapid algorithm development and parameter selection
  - > 20x speedup by exploiting parallelism across frequency on nodes of Linux cluster
  - Development time was ~1 day

Simulated data:
Matched field output at target depth, bearing



Agenda

- Introduction
- LLGrid System
- Performance Results
- LLGrid Productivity Analysis
- Summary

LLGrid Productivity Analysis for ROI*

$$\begin{array}{c}\text{productivity} \\ \text{(ROI)}\end{array} = \frac{\left(\text{Utility}\right)}{\left(\begin{array}{c}\text{Software} \\ \text{Cost}\end{array}\right) + \left(\begin{array}{c}\text{Maintenance} \\ \text{Cost}\end{array}\right) + \left(\begin{array}{c}\text{System} \\ \text{Cost}\end{array}\right)}$$

*In development in DARPA HPCS program

LLGrid Productivity Analysis for ROI*

$$\begin{array}{c}\text{productivity}\\ \text{(ROI)}\end{array} = \frac{\left(\text{time saved by users on system}\right)}{\left(\begin{array}{c}\text{time to}\\ \text{parallelize}\end{array}\right) + \left(\begin{array}{c}\text{time to}\\ \text{train}\end{array}\right) + \left(\begin{array}{c}\text{time to}\\ \text{launch}\end{array}\right) + \left(\begin{array}{c}\text{time to}\\ \text{admin.}\end{array}\right) + \left(\begin{array}{c}\text{system}\\ \text{cost}\end{array}\right)}$$

*In development in DARPA HPCS program

LLGrid Productivity Analysis for ROI

$$\begin{array}{c}\text{productivity}\\ \text{(ROI)}\end{array} = \frac{\begin{pmatrix}\text{time saved by users on system}\end{pmatrix}}{\begin{pmatrix}\text{time to}\\ \text{parallelize}\end{pmatrix} + \begin{pmatrix}\text{time to}\\ \text{train}\end{pmatrix} + \begin{pmatrix}\text{time to}\\ \text{launch}\end{pmatrix} + \begin{pmatrix}\text{time to}\\ \text{admin.}\end{pmatrix} + \begin{pmatrix}\text{system}\\ \text{cost}\end{pmatrix}}$$

Production LLGrid model assumptions

- 200 users Lab-wide
- 20 simultaneous jobs
- Average 10 CPUs per job
- 2 SLOCs per hour
- 1000 SLOCs per simulation * Lab-wide users
- 1.0% time-to-parallelize overhead
- Training time - 4 hours * Lab-wide users
- 10,000 parallel job launches
- 10 seconds to launch
- One sys-admin ≈ 2000 hours
- 200 CPUs @ $5k per node ≈ 5000 hours

$$\begin{pmatrix}\text{time saved}\\ \text{by users on}\\ \text{system}\end{pmatrix} = \begin{pmatrix}\text{User}\\ \text{salary}\end{pmatrix} * \begin{pmatrix}\text{Total time}\\ \text{system is}\\ \text{in use}\end{pmatrix} * \begin{pmatrix}\text{Average}\\ \text{number of}\\ \text{users}\end{pmatrix} * \left(1 - \frac{1}{\begin{pmatrix}\text{Average \# of}\\ \text{CPUs per job}\end{pmatrix}}\right)$$

$$\begin{pmatrix}\text{time to}\\ \text{parallelize}\end{pmatrix} = \begin{pmatrix}\text{User}\\ \text{salary}\end{pmatrix} * \begin{pmatrix}\text{Total}\\ \text{\# of}\\ \text{users}\end{pmatrix} * \begin{pmatrix}\text{Prog.}\\ \text{rate}\end{pmatrix} * \begin{pmatrix}\text{Average}\\ \text{lines of}\\ \text{code}\end{pmatrix} * \left(\frac{1}{\begin{pmatrix}\text{Cost for}\\ \text{parallel}\end{pmatrix}} - 1\right)$$

$$\begin{pmatrix}\text{time to}\\ \text{train}\end{pmatrix} = \begin{pmatrix}\text{User}\\ \text{salary}\end{pmatrix} * \begin{pmatrix}\text{Total}\\ \text{\# of}\\ \text{users}\end{pmatrix} * \begin{pmatrix}\text{Time to}\\ \text{train a}\\ \text{user}\end{pmatrix}$$

$$\begin{pmatrix}\text{time to}\\ \text{launch}\end{pmatrix} = \begin{pmatrix}\text{User}\\ \text{salary}\end{pmatrix} * \begin{pmatrix}\text{Number of}\\ \text{launches}\end{pmatrix} * \begin{pmatrix}\text{Time to}\\ \text{launch}\end{pmatrix}$$

$$\begin{pmatrix}\text{time to}\\ \text{admin.}\end{pmatrix} = \begin{pmatrix}\text{Admin.}\\ \text{salary}\end{pmatrix} * \begin{pmatrix}\text{Number}\\ \text{of admins}\end{pmatrix} * \begin{pmatrix}\text{Admin}\\ \text{time}\end{pmatrix}$$

$$\begin{pmatrix}\text{system}\\ \text{cost}\end{pmatrix} = \begin{pmatrix}\text{User}\\ \text{salary}\end{pmatrix} * \begin{pmatrix}\text{Time-value}\\ \text{of system}\end{pmatrix}$$

LLGrid Productivity Analysis for ROI

$$\text{productivity (ROI)} = \frac{\left(\text{time saved by users on system}\right)}{\left(\text{time to parallelize}\right) + \left(\text{time to train}\right) + \left(\text{time to launch}\right) + \left(\text{time to admin.}\right) + \left(\text{system cost}\right)}$$

Production LLGrid model assumptions

- 200 users Lab-wide
- 20 simultaneous jobs
- Average 10 CPUs per job
- *2 SLOCs per hour*
- 1000 SLOCs per simulation * Lab-wide users
- *1.0% time-to-parallelize overhead*
- Training time - 4 hours * Lab-wide users
- 10,000 parallel job launches
- 10 seconds to launch
- One sys-admin ≈ 2000 hours
- 200 CPUs @ $5k per node ≈ 5000 hours

$$ROI_{expected} = \frac{36,000}{1000+27.8+2000+5000}$$

$$ROI_{expected}^{*} \approx 4.5$$

Steady state with
full LLGrid

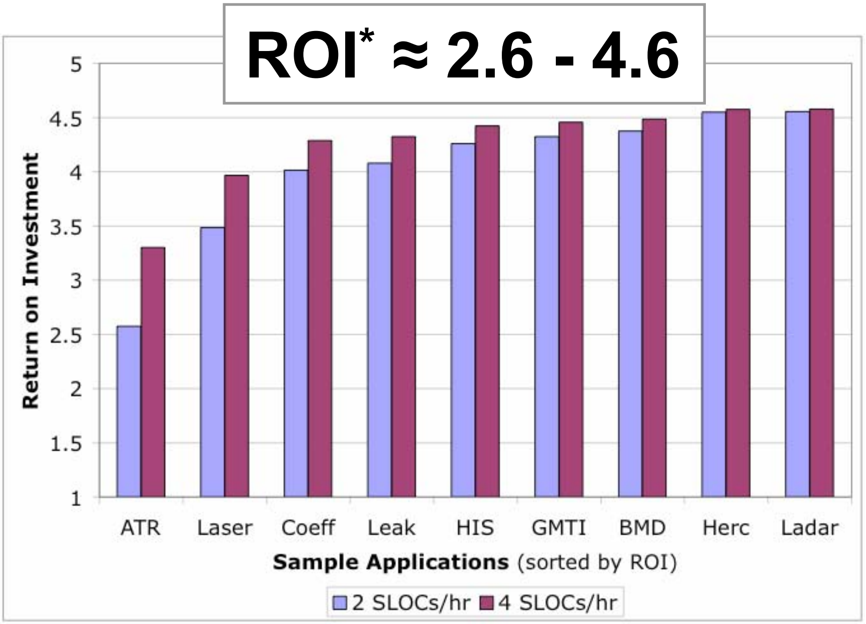
* Mileage may vary

LLGrid Productivity Analysis for ROI

$$\text{productivity (ROI)} = \frac{\left(\text{time saved by users on system}\right)}{\left(\text{time to parallelize}\right) + \left(\text{time to train}\right) + \left(\text{time to launch}\right) + \left(\text{time to admin.}\right) + \left(\text{system cost}\right)}$$

Production LLGrid model assumptions

- 200 users Lab-wide
- 20 simultaneous jobs
- Average 10 CPUs per job
- ***2-4 SLOCs per hour***
- 1000 SLOCs per simulation * Lab-wide users
- ***Measured time-to-parallelize overhead***
- Training time - 4 hours * Lab-wide users
- 10,000 parallel job launches
- 10 seconds to launch
- One sys-admin ≈ 2000 hours
- 200 CPUs @ $5k per node ≈ 5000 hours



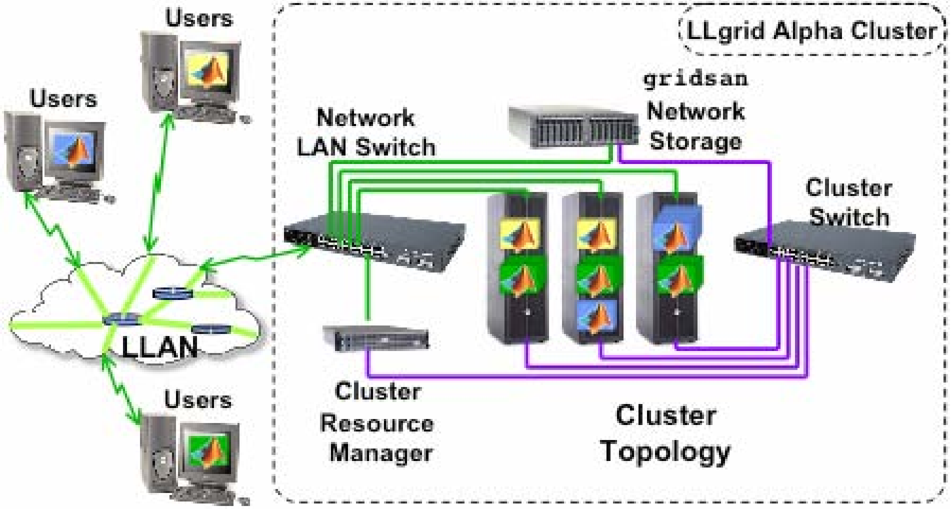
ROI* ≈ 2.6 - 4.6

* Varying Mileage

LLGrid On-Demand Grid Computing System
Agenda

- Introduction
- LLGrid System
- Performance Results
- LLGrid Productivity Analysis
- **Summary**

Summary



- Easy to set up
- Easy to use
- User’s computer transparently becomes part of LLGrid
- High throughput computation system
- 25 alpha users, expecting 200 users Lab-wide
- Computing jobs they could not do before
- 3600 CPU days of computer time in 8 months
- LLGrid Productivity Analysis - ROI ≈ 4.5