grid*Mathematica*$^{®}$ on Apple Xserve

White Paper
October 2003

What Is It?

■ General Overview

grid*Mathematica* implements many parallel programming primitives and includes high-level commands for parallel execution of matrix operations, plotting, and other functions. It comes with sample applications of many popular new programming approaches, such as parallel Monte Carlo simulation, visualization, searching, and optimization. The implementations for all high-level parallel processing commands are provided in *Mathematica* source form, so they can also serve as templates for users to build their own parallel programs.

A typical installation of grid*Mathematica* involves a master kernel, a license manager, and one *Mathematica* kernel per available node. *MathLM*, the license manager, makes sure that each machine on the cluster is properly licensed and provides the necessary passwords needed by the *Mathematica* nodes. The master kernel handles all input, output, and scheduling. It can be controlled from any *Mathematica* front end or via batch files, either locally or via a remote connection. Users can launch remote kernels from the master kernel using devices such as RSH or SSH. Once the remote kernels are launched, they are ready to receive commands from the master machine.

■ Key Advantages

Special Pricing

grid*Mathematica* provides powerful computing capabilities at a price that will not hurt your organization's pocketbook. Wolfram Research is offering grid*Mathematica* at a cost per node that is far less than what users would have to pay for an equivalent *Network Mathematica* installation.

Computational Ability

grid*Mathematica* gives immediate access to the world's leading collection of algorithms and mathematical knowledge. It offers all of the same features and programming capabilities as *Mathematica*, including thousands of functions covering areas such as numerical computation, symbolic computation, graphics, and general programming.

Ease of Development

grid*Mathematica* introduces only a small number of new parallel computing constructs, and users familiar with *Mathematica* can transition to grid*Mathematica* without difficulty. Furthermore, programs written in *Mathematica* can be easily modified to run on a grid. Even users who are new to *Mathematica* can use its high-level programming capabilities, thousands of built-in functions, and just a few simple commands to solve grid-computing problems that used to require thousands of lines of code in C or Fortran.

Simple Code Example

Setup

This loads *Parallel Computing Toolkit*.

```
<< Parallel`
```

Set Up the Remote *Mathematica* Startup Commands

Define how you start a remote process on this grid.

```
$RemoteCommand =
  "ssh `1` /usr/local/bin/math5 –mathlink";
```

Depending on your network setup you may have different commands for different machines. Here is an example showing how to specify which machines to launch on a particular grid.

```
$AvailableMachines = Flatten[Table[
    Table[ RemoteMachine["xserve" <> ToString[i],
      $RemoteCommand], {i, 8} ], {2}]];
```

Launch All Remote *Mathematica* Processors

Launch all slaves.

```
Scan[LaunchSlave[#[[1]], #[[2]]] &,
 $AvailableMachines]
```

Query the slaves for some information.

```
RemoteEvaluate[{$MachineName, $Version}]
```

```
{{xserve1, 5.0 for Mac OS X (June 6, 2003)},
 {xserve2, 5.0 for Mac OS X (June 6, 2003)},
 {xserve3, 5.0 for Mac OS X (June 6, 2003)},
 {xserve4, 5.0 for Mac OS X (June 6, 2003)},
 {xserve5, 5.0 for Mac OS X (June 6, 2003)},
 {xserve6, 5.0 for Mac OS X (June 6, 2003)},
 {xserve7, 5.0 for Mac OS X (June 6, 2003)},
 {xserve8, 5.0 for Mac OS X (June 6, 2003)},
 {xserve1, 5.0 for Mac OS X (June 6, 2003)},
 {xserve2, 5.0 for Mac OS X (June 6, 2003)},
 {xserve3, 5.0 for Mac OS X (June 6, 2003)},
 {xserve4, 5.0 for Mac OS X (June 6, 2003)},
 {xserve5, 5.0 for Mac OS X (June 6, 2003)},
 {xserve6, 5.0 for Mac OS X (June 6, 2003)},
 {xserve7, 5.0 for Mac OS X (June 6, 2003)},
 {xserve8, 5.0 for Mac OS X (June 6, 2003)}}
```

Close the connections to the slave kernels.

```
CloseSlaves[];
```

Who Uses It?

■ Users with Access to Departmental Clusters

grid*Mathematica* provides an affordable, easy-to-use way to take full advantage of grid-computing hardware, such as the multiprocessor machines and computing clusters that are now more accessible to many research groups, universities, and companies.

■ Users of Large-Scale Clusters

Although primarily targeted at department-sized clusters to accommodate their pricing restrictions, grid*Mathematica* is fully scalable, and so it can be used on large-scale clusters as well.

grid*Mathematica* on Xserve

Mac OS X Server and Xserve hardware combine together to make a powerful and easy-to-configure system that simplifies the setup and deployment of both small departmental clusters and large-scale clusters.

Rendezvous allows you to quickly access the nodes of your Xserve cluster by name without having to set up DNS entries or host files.

The Server Status Utility allows you to retrieve status information on each individual Xserve in your cluster, making it easy to monitor the performance of your cluster.

Installation of grid*Mathematica* on Xserve is easily accomplished with four simple steps.

■ Installing the License Manager

MathLM can be installed on any machine in the grid. We recommend that you install it on the master node of your cluster.

1. Insert the CD into the CD-ROM drive. The CD icon should appear on your desktop.

2. Open a Terminal window.

3. Change directory to the Macintosh subdirectory on the CD. The CD mount point on Mac OS X is `/Volumes` and the name of the CD is `MathLM_5_0`.

```
cd/Volumes/MathLM_5_0/Macintosh
```

Note: If you drag the CD icon into a terminal window, the pathname for the CD is automatically pasted into the command line.

4. Type the following command and press RET.

```
./MathLMInstaller
```

5. The following prompt will appear with the MathID for your machine. To proceed with the installation, you will need to enter a password. If you have already received a password, type the letter `c` and press RET to proceed with the installation.

```
#./MathLMInstaller
```

MathLM will require a password in order to use it. To register and get a password, you will need to supply the following information:

```
Machine name:gauss
```

```
MathID:6619-76357-73584
```

6. Enter the license ID printed on your license certificate when prompted. Your license ID should be of the form L*nnnn*-*nnnn*, where the *n* are digits.

7. Enter your *MathLM* password when prompted and then press RET.

8. The installer prompts you to specify the directory where *MathLM* should be installed. The default location is `/usr/local/Wolfram/MathLM`. Press RET to accept the default, or type in a new location and then press RET.

9. The installer prompts you to specify the directory where symbolic links will be created. The default location is `/usr/sbin`. Press RET to accept the default, or type in a new location and then press.

10. Once the installation is complete, you must start *MathLM* manually.

To start *MathLM*, type:

```
./mathlm
```

Installing *Mathematica*

You will need to install *Mathematica* on each node of your cluster, including the master node. Installing *Mathematica* on Mac OS X simply involves dragging the *Mathematica* icon from the CD to the location on your hard drive where you wish to install it. The most common and recommended location is `/Applications`. Utilizing the power of Unix built into Mac OS X, you can use command-line utilities such as SCP and SH to script the install of *Mathematica* to all of the machines in your cluster over the network. This saves you time because you do not have to log on to the local console of each individual machine in the cluster. In order to easily launch the MathKernel from the command line you will want to create and place a shell script into `/usr/local/bin` that execs `/Applications/Mathematica 5.0.app/Contents/MacOS/MathKernel`.

The actual script would look like the following:

```
#!/bin/sh

# Script for running the MathKernel in a terminal

exec "/Applications/Mathematica 5.0.app/Contents/MacOS/MathKernel" $@
```

Installing *Parallel Computing Toolkit*

You will need to install *Parallel Computing Toolkit* on the master node.

1) Quit *Mathematica* if it is running. Insert the application CD. Double-click the application installer icon.

2) A dialog box appears. Click Continue to start the installation.

3) A dialog box appears showing the default installation location. Click Select Folder to choose another installation location if necessary. Click Install when you are ready to begin the installation.

4) A dialog box appears when the installation is complete. Click OK to continue. The application package is now ready to use.

The next time you start *Mathematica*, from the Help menu choose Rebuild Help Index. The application package's documentation will then be available in the Help Browser's Add-ons & Links category.

■ Setting Up SSH/RSH

In order to make communication with grid*Mathematica* as secure as possible, communication can be initiated using standard remote shells. In essence, grid*Mathematica* needs to be able to login to a remote machine and launch other instances of *Mathematica* on that remote machine. Mac OS X ships with both RSH and SSH pre-installed so no additional installation is necessary. However, with today's tighter security models, we recommend using SSH rather then RSH for remote login to the nodes of your cluster. Passwordless key-authentication is recommended, which makes it easy to remote login to the other nodes without being prompted for a password. To setup SSH to work without requiring a password, do the following:

1) On the master node, launch terminal.app located in `/Applications/Utilities` and enter the following command:

```
ssh-keygen -t rsa
```

2) You will be prompted for the file name in which to save the key. Press return to save it in the default location of `$HOME/.ssh/id_rsa`.

3) You will then be prompted for a password, leave it blank, and press RET.

4) Copy `$HOME/.ssh/id_rsa.pub` to the `$HOME/.ssh/authorized_keys2` .

5) Using SCP or SFTP copy `$HOME/.ssh/authorized_keys2` to `$HOME/.ssh/authorized_keys2` on all of the other nodes of your cluster

6) Now when using SSH from the master node to access the other nodes of your cluster, you can log into the nodes without using a password by specifying the -i option when running SSH. For example:

```
ssh -i $HOME/.ssh/id_rsa xserve_node01.local
```