Interactive European Grid

MPI-START

Kiril Dichev

HLRS, Stuttgart

MPI Workshop, CNAF, Bologna, 19.-20. March 2008

int.eu.grid

Overview

- Motivation
- Design, Features, Architecture
- Environment Variables
- Workflow
- Use Case
- Tools Support

Motivation

- The support for parallel jobs in the EGEE initially supported only “MPICH” as job type
- The standard “MPICH” approach has many flaws:
  - It uses a hard-coded script
  - Modifications like new MPI implementations or schedulers not supported
  - Any modified middleware needs to go through a long release cycle

Motivation

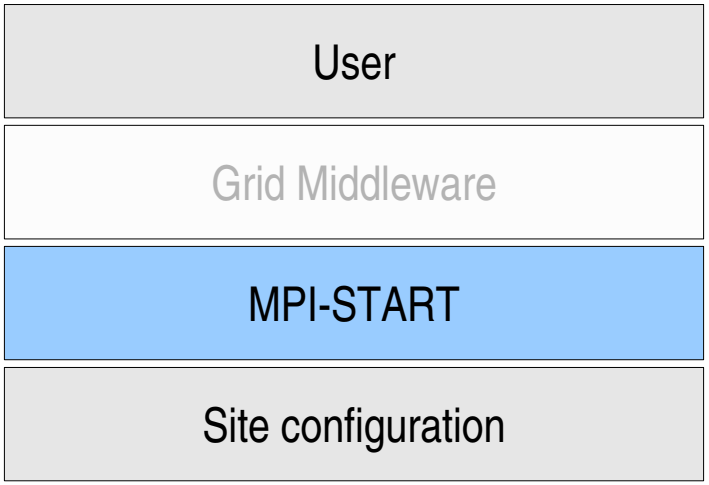
- The main idea of MPI-Start is to enable running MPI applications on different sites with different configurations
- A user can use custom scripts to specify the MPI version/ scheduler/file distribution system etc. to be used for a particular site
- Let the user:
  - Locally compile a program
  - Distribute binaries/input files and gather output files

Motivation

- Let MPI-Start identify some of the components like:
  - File system
  - Scheduler
- Optionally, the set of scripts can be installed into every cluster

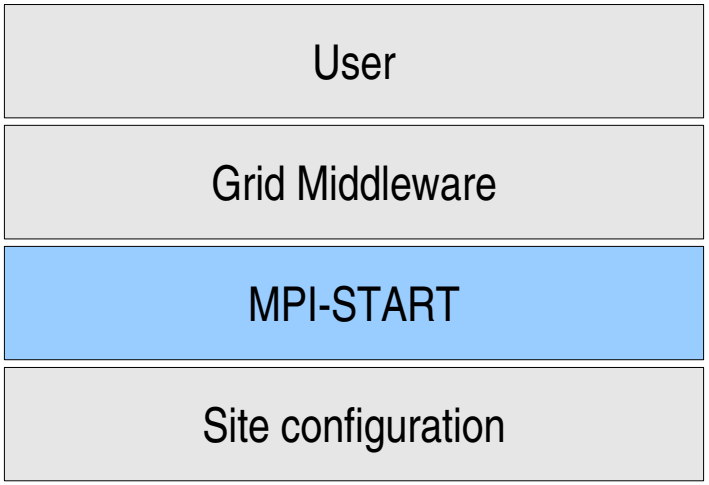
Design

- The user could directly specify environment variables to communicate with MPI-Start



Design

- Also, it is possible to have the middleware “translate” user input to MPI-Start and add additional information (like inter-cluster MPI information)



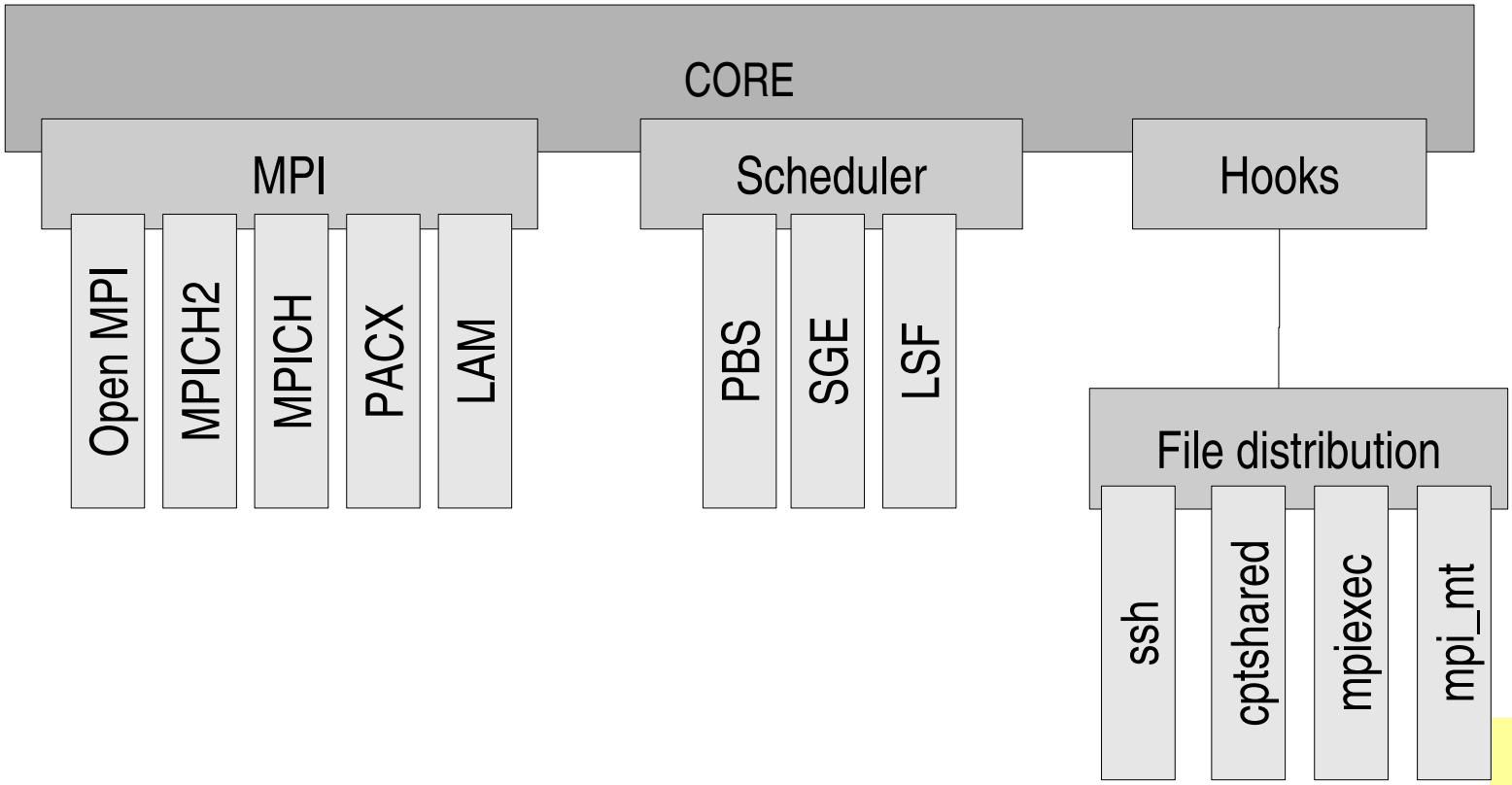
Features

- Portable
  - The program must be able to run under any supported operating system
- Modular and extensible architecture
  - Plugin/Component architecture
- Relocatable
  - The program must be independent of absolute path, to adapt to different site configurations.
  - Remote "*injection*" of MPI-Start along with the job

Features

- Hooks
  - enable immediate user script integration
- Easier to debug
  - verbose output can be activated
  - Important for easier script debugging

Architecture



Environment Variables

- In order for the user/middleware to communicate with the scripts, a clean interface is required
- Define a set of environment variables

Intra-Cluster Variables

- Interface **Intra** Cluster MPI
  - I2G_MPI_APPLICATION
    - This variable describes the executable.
  - I2G_MPI_APPLICATION_ARGS
    - This variable contains the parameters of the executable.
  - I2G_MPI_TYPE
    - Specifies the MPI implementation to use (e.g openmpi, ...).
  - I2G_MPI_VERSION
    - Specifies which version of the the MPI implementation to use. If not defined the default version will be used.

Intra-Cluster Variables

- Interface **Intra** Cluster MPI
  - I2G_MPI_PRECOMMAND
    - Specifies a command that is prepended to the mpirun (e.g. time).
  - I2G_MPI_PRE_RUN_HOOK
    - This variable can point to a shell script that must contain a “pre_run_hook” function. This function will be called before the parallel application is started (example: compilation of the binary)
  - I2G_MPI_POST_RUN_HOOK
    - Like I2G_MPI_PRE_RUN_HOOK, but the script must define a “post_run_hook” that is called after the parallel application finished (example: upload of results on SE)

Intra-Cluster Variables

- Interface **Intra** Cluster MPI
  - I2G_USE_MPITRACE
    - Flag for activating trace file generation, executable must be precompiled with MPITrace
  - I2G_USE_MARMOT
    - Flag for activating Marmot file generation, executable optionally precompiled with Marmot

Inter-Cluster Variables

- Interface **Inter** Cluster MPI
  - I2G_MPI_FLAVOUR
    - Specifies which local sub MPI implementation to use.
  - I2G_MPI_JOB_NUMBER
    - In the case of a multi cluster MPI job this variable indicate the sub-job id.
  - I2G_MPI_STARTUP_INFO
    - Special synchr. informations for a inter cluster MPI job.
  - I2G_MPI_RELAY
    - Specifies the host via which the MPI traffic will be routes

Workflow of MPI-Start

- Check for scheduler plugin
- Generate machinefile (in a generic format)
- Load preselected (or site-configured) MPI plugin
- Check if user specified MPI prefix/version
- Check some MPI-implementation- or scheduler-specific parameter options
- Check for MPI plugin

Workflow of MPI-Start

- Detect file system (based on checking /etc/mtab entries)
- Call user-defined pre-run hooks
- If not a shared file system, find best distribution method and copy files to all worker nodes
  - based on the MPI implementation or other settings
- Optional settings to activate MPI implementation
- All MPI-implementation specific parameters are set
- MPI-implementation dependent and generic tool setting

Workflow of MPI-Start

- Call generic mpirun/mpiexec with specified parameters
- Call user-defined post-run hook
- Optional tool post processing
- If not shared, use detected distribution method for cleaning files
- Return value of mpirun/mpiexec call

A Use Case

- Scheduler

```
mpi-start [DEBUG  ]: checking for scheduler support : pbs
mpi-start [DEBUG  ]:  checking for $PBS_NODEFILE
mpi-start [INFO   ]: activate support for pbs
mpi-start [DEBUG  ]:  return PBS_NODEFILE
```

A Use Case

- MPI plugin

```
mpi-start [DEBUG  ]:  using user requested MPI flavour
mpi-start [DEBUG  ]:  check for default MPI version
mpi-start [DEBUG  ]:  coulnd't find EGEE environment
mpi-start [DEBUG  ]: found openmpi and PBS, don't set machinefile
mpi-start [INFO   ]: activate support for openmpi
mpi-start [DEBUG  ]: source : /opt/i2g/bin/../etc/mpi-start/openmpi.mpi
mpi-start [DEBUG  ]: use user provided prefix : /opt/i2g/openmpi
mpi-start [DEBUG  ]: activate MPI via manually update
mpi-start [INFO   ]: call backend MPI implementation
```

A Use Case

File distribution

```
mpi-start [DEBUG  ]: current working directory : /home/imain031/globus-
tmp.ingrid01.24620.0/https_3a_2f_2fi2g-rb01.lip.pt_3a9000_2fyZysksQyZbvgo2IzciHTpQ_0
mpi-start [DEBUG  ]: found local fs : ext3
mpi-start [DEBUG  ]: mpi_start_pre_run_hook_copy
mpi-start [DEBUG  ]: fs not shared -> distribute binary
mpi-start [DEBUG  ]: I2G_MPI_FILE_DIST => ssh
mpi-start [DEBUG  ]: mpi_start_pre_run_hook_copy_ssh
mpi-start [DEBUG  ]: fs not shared -> distribute binary
```

A Use Case

- Running the application

mpi-start [DEBUG]: /opt/i2g/openmpi/bin/mpiexec -wdir /home/imain031/globus-tmp.ingrid01.24620.0/https_3a_2f_2fi2g-rb01.lip.pt_3a9000_2fyZysksQyZbvgo2IzciHTpQ_0 -x X509_USER_PROXY --prefix /opt/i2g/openmpi -np 12 /home/imain031/globus-tmp.ingrid01.24620.0/https_3a_2f_2fi2g-rb01.lip.pt_3a9000_2fyZysksQyZbvgo2IzciHTpQ_0/DD_filtre2-ifca-openmpi

.....

(application output)

Tools Support

- Tool support has been added for some tools:
  - Mpitrace (MPI trace file generator for the Paraver performance analyzer)
  - Marmot (MPI correctness checker)
- From the user perspective, just an extra environment variable is set

- An example JDL file for using Marmot:

```
JobType           = "openmpi";
NodeNumber        =  5;
VirtualOrganisation = "imain";
Executable        = "cg-tutorial-marmot-exercise";
StdOutput         = "cg-tutorial-marmot-exercise.out";
StdError          = "cg-tutorial-marmot-exercise.err";
InputSandbox      = {"cg-tutorial-marmot-exercise"};
OutputSandbox     = {"cg-tutorial-marmot-exercise.out","cg-tutorial-marmot-
exercise.err","MarmotLog.txt"};
Environment = {"I2G_USE_MARMOT=1"};
Requirements = other.GlueCEUniqueID == "ce-ieg.bifi.unizar.es:2119/jobmanager-lc
```

A Tool Framework

- Hard-code the tool support into MPI-Start
- Not realised as a framework yet, but a general approach seems possible
- Such a framework could work similarly to the hook approach:
  - Before execution, settings are made to activate the tool
  - After execution, post-processing for the tool is performed (e.g. output file generation)
- Without “Requirements” option assumptions are made that the tools are installed on all cluster nodes

A Tool Framework

- Some runtime checks need to be embedded:
  - Is the tool installed on the execution site (currently, this information is not published)
- A plugin only makes sense for tools that follow a straight-forward and not parametrized flow:
  - Example: generate a fixed extra file for the tool
- Not recommended for tools that are highly configurable
  - In this case, it is better to use pre/post run hooks to give the user flexibility to specify tool options

int.eu.grid

Questions?