Interactive European Grid

MPI Support in Int.EU.Grid: Open MPI, PACX-MPI, MPI-Start, Marmot

Kiril Dichev

HLRS, Stuttgart

3rd EGEE User Forum, 11.-14. February 2008

Clermont-Ferrand, France

MPI Support

Overview

- Open MPI
  - HIgh-performance MPI-2 implementation
- PACX-MPI
  - MPI between clusters
- MPI-Start
  - A layer for starting MPI processes (EGEE and I2G)
- Marmot
  - MPI application checking tool

int.eu.grid

Open MPI

H L R S

Open MPI

Who is MPI?

- Founders
  - High Performance Computing Center, Stuttgart
  - Indiana University
  - Los Alamos National Laboratory
  - The University of Tennessee
- Current status
  - 15 Members
  - 9 Contributors
  - 1 Partner

Open MPI

Project Goals

- Stat- of-the-art MPI implementation
  - Full support of the MPI-2 standard
  - Avoidance of old legacy code
  - Profit from long experience in MPI implementations
  - Avoiding the “forking” problem
  - Community / 3rd party involvement
  - Production-quality research platform
  - Rapid deployment for new platforms

Open MPI

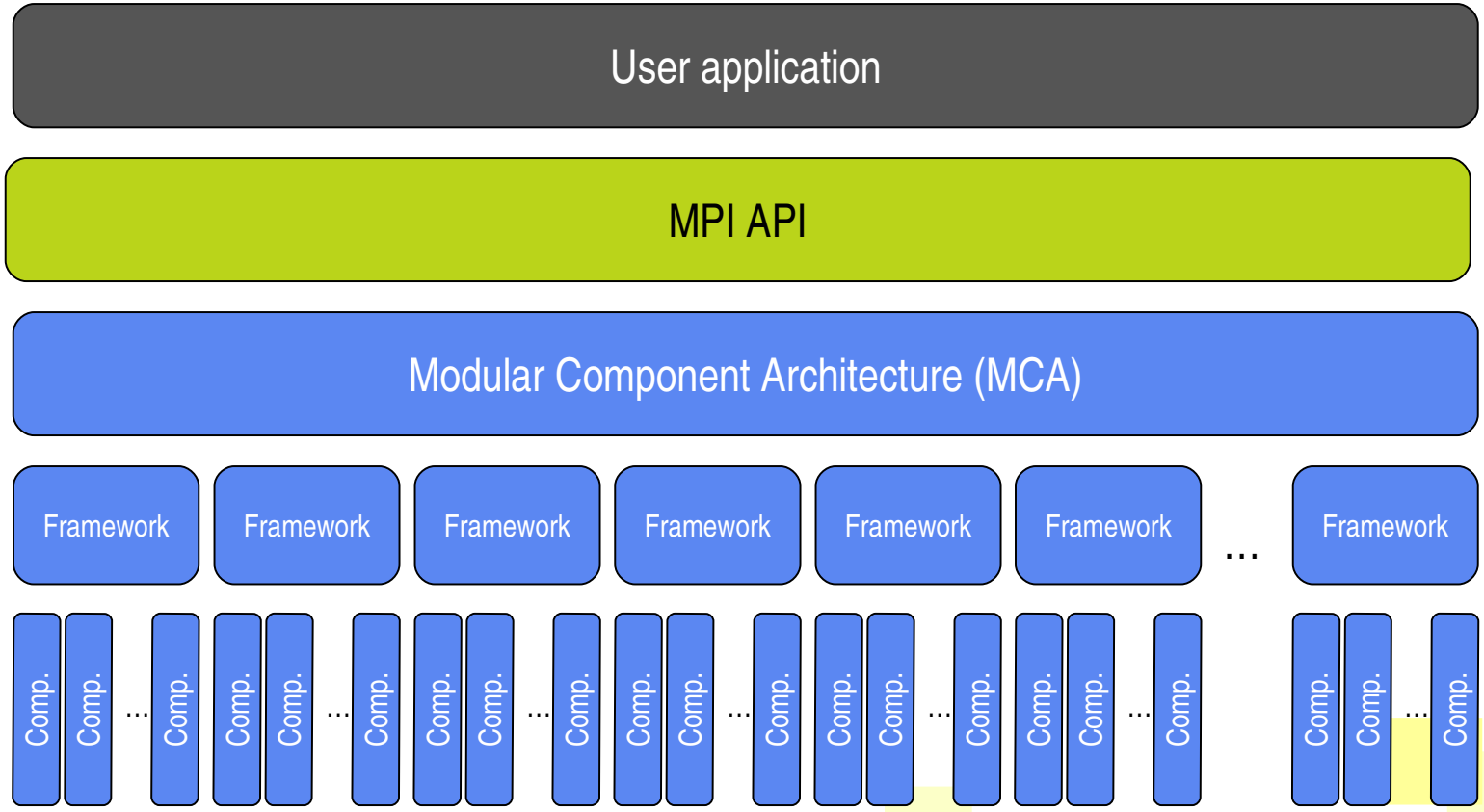
Design Goals

- Component architecture
- Design for heterogeneous environments
- Multiple networks (run-time selection)
- Support for automatic error detection / retransmission
- Portable and performant
  - Small cluster
  - “Big iron” hardware
  - Grids

Open MPI

Modular Component Architecture

- MCA top level view



Open MPI

Requirements

- The orte daemons need an open TCP/IP Port for incoming connections
- Different requirements for the different PLS
- ssh requires login without password (e.g. public keys)
- software installation
  - Open MPI needs to be installed on the
    - WN
    - CE (for PACX-MPI runs)

int.eu.grid

PACX-MPI

H L R S

PACX-MPI

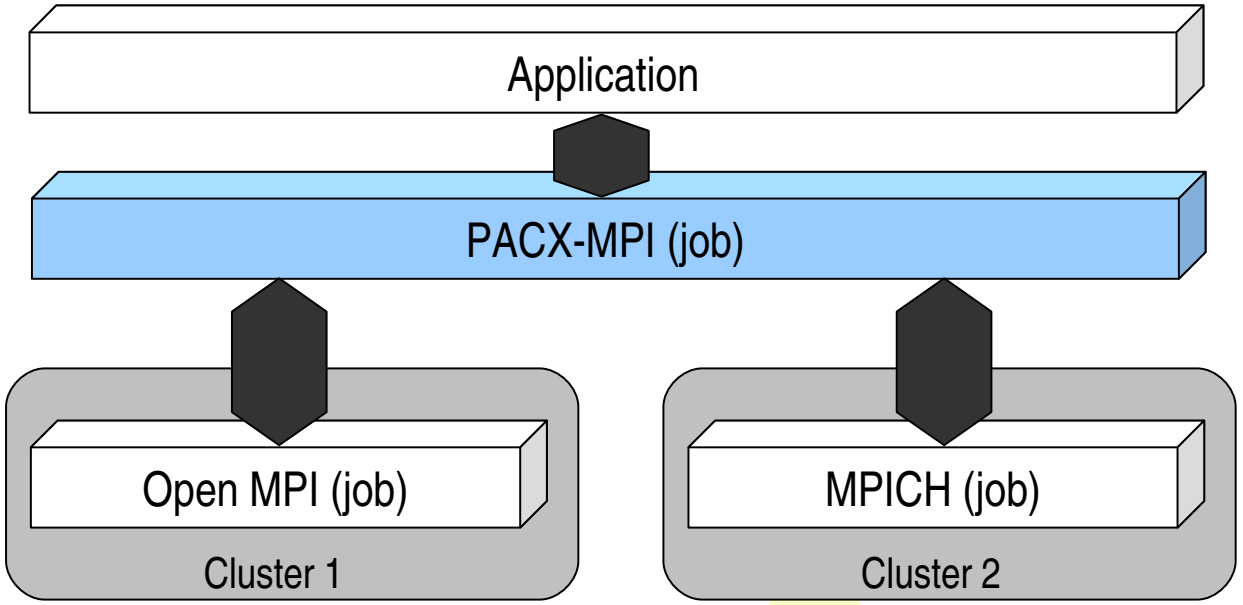
Overview

- A middleware to seamlessly run MPI applications on a network of parallel computers (originally dev. in 1995 to connect Vector+MPP).
- PACX-MPI is an optimized standard-conforming MPI-implementation, applications just need **recompilation**(!)
- For C: pre-processor renaming: MPI_Send becomes PACX_Send.
- For Fortran: Function replacement @ link-step.

PACX-MPI

Overview

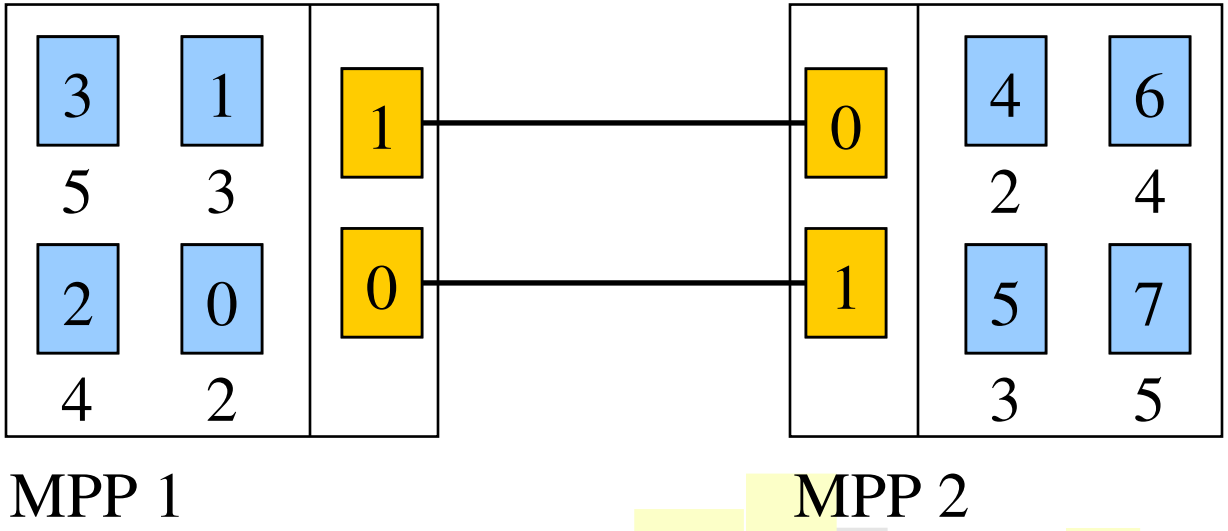
- PACX-MPI starts an MPI job in each cluster
- PACX-MPI "merges/manages" these MPI jobs internally and emulate transparently a bigger MPI job to the application



PACX-MPI

Usage

- Compiling with PACX
  - pacxcc -c hello.c
  - pacxcc -o **hello** hello.o
- Running requires 2 additional processes:



int.eu.grid

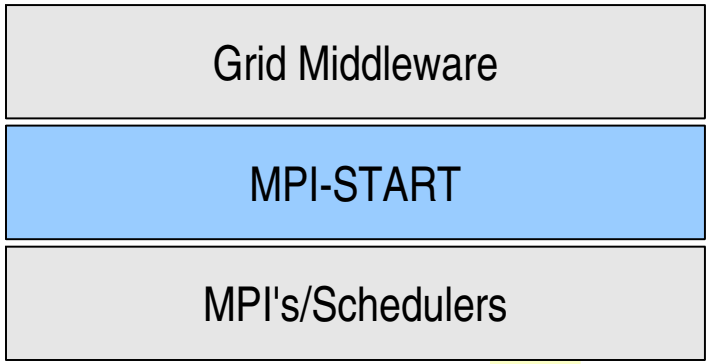
MPI-Start

H L R S

MPI-Start

Goals

- Goals of mpi-start:
  - Define a unique interface to the upper layer for MPI jobs
  - Support of a new MPI implementation doesn't require any change in the Grid middleware
  - Support of file distribution
  - Provide some support for the user to help manage his data.

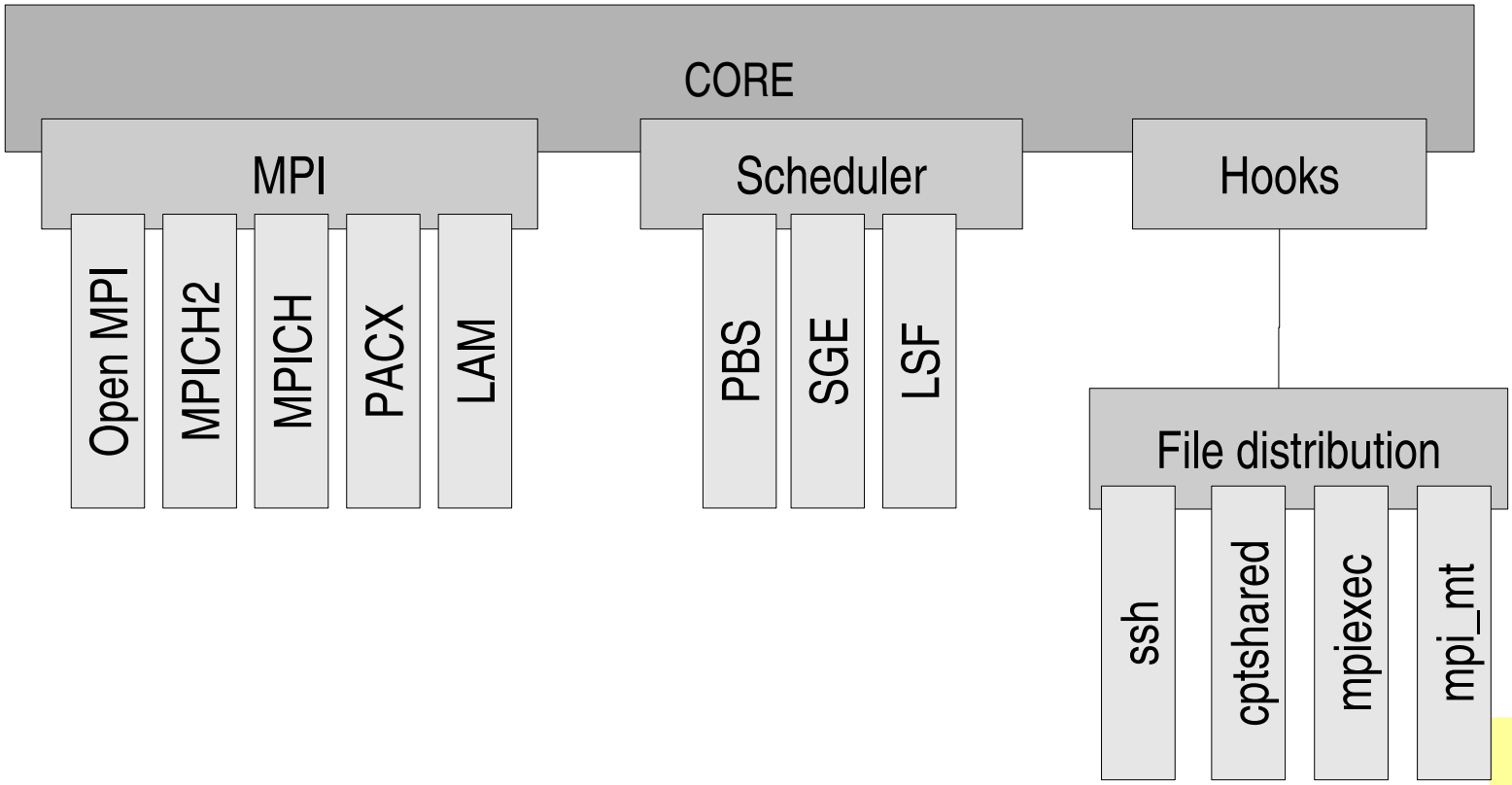


Design Goals

- 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
- Very good "remote" debugging features

MPI-Start

Architecture



MPI-Start

int.EU.grid status

- Support of
  - MPI on a single cluster
  - MPI on multiple clusters (with PACX-MPI)
  - different schedulers
  - different file distribution mechanisms
  - different MPI tools (Marmot)
- Plugins being developed together with EGEE

int.eu.grid

MPI-Start in I2G and EGEE

- Current version 0.0.58
  - tested and verified in I2G
  - Also available in ETICS for EGEE

int.eu.grid

Marmot Integration in Grids

HLRS

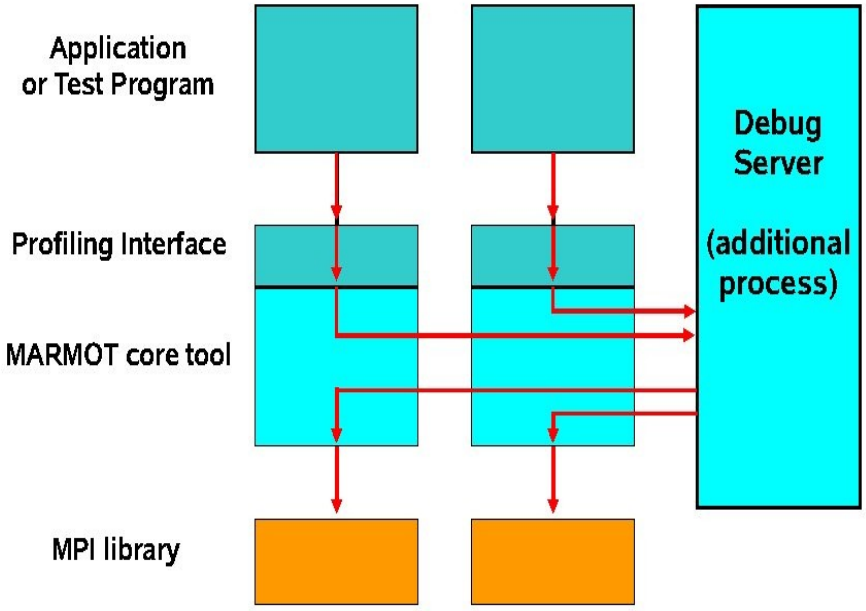
Marmot

Overview

- MPI checking tool for MPI errors at runtime
- Developed in the frame of CrossGrid
- No source code modification required (only recompilation **or** dynamic reordering of search paths)
- One additional process working as debug server
- Implementation of C and Fortran language binding of MPI-1.2 standard

Marmot

Architecture



int.eu.grid

Marmot in I2G

- Marmot currently includes static and dynamic libraries
- The usage of dynamic libraries allows the user to immediately(!) use Marmot by:
  - specifying an extra flag
  - adding an extra process
  - specifying the output log file
- Technical details are done by MPI-Start:
  - Environment variables are set for intercepting MPI calls
  - Marmot output file information is set as expected

Marmot in I2G

- 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
```

Marmot Log File (1)

```
75: Error from rank 9(Thread: 0) with  Text:
ERROR: MPI_Type_struct: datatype [0] is Fortran-Type!
```

```
array_of_types[0] = MPI_INTEGER;
array_of_types[1] = MPI_LONG_LONG_INT;

MPI_Type_struct(COUNT, array_of_blocklengths,
        array_of_displacements, array_of_types, &sendtype);
```

Marmot Log File (2)

```
64: Warning from rank 5(Thread: 0) with Text:
 WARNING: MPI_Type_struct: blocklength[0] = 0!
```

```
array_of_blocklengths[0] = 0;
```

Marmot Log File (3)

```
103: Note from rank 7(Thread: 0) with  Text:
NOTE: MPI_Type_commit: Datatype already committed!
```

```
MPI_Type_commit(&sendtype);
MPI_Type_commit(&sendtype);
```

int.eu.grid

Questions?

H L R S