Enabling Grids for E-science in Europe

MPI and Grid

Brian Coghlan, John Walsh, Stephen Childs and Kathryn Cassidy
Trinity College Dublin

Acknowledgements

- **Initial slides derived from slides by:**
  - Vered Kunik, Israeli Grid NA3 Team,

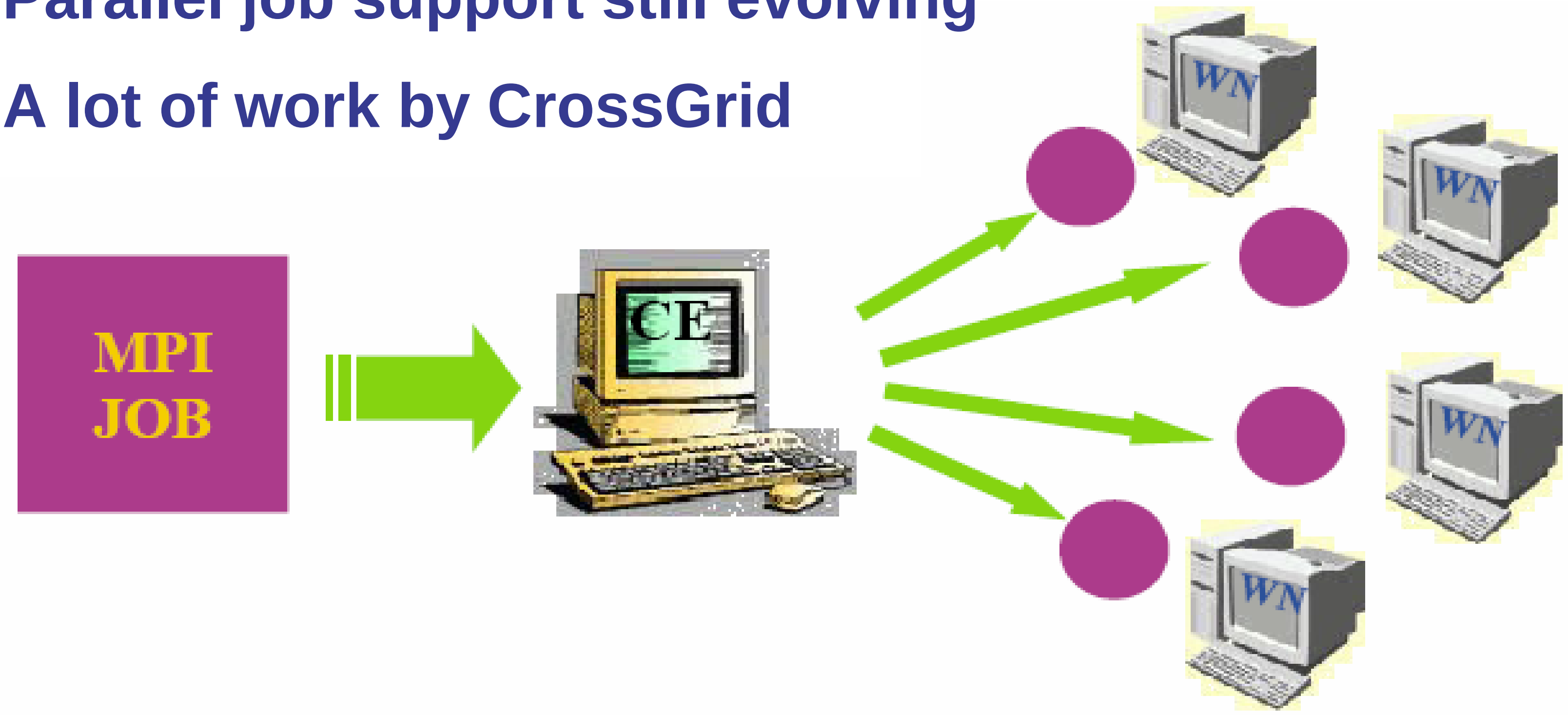
    for the Israeli Grid Workshop, Ra’anana, Israel, Sep-2005
  - Miroslav Ruda, Masaryk University and CESNET,

    for Grid for Complex Problems, Slovakia, 29-NOV-2005
- **Extended by:**
  - Brian Coghlan, John Walsh, Stephen Childs and Kathryn Cassidy, TCD,

    for the Grid User’s Course, Trinity College Dublin, 14/14-MAR-2006

Using MPI on the Grid

- The MPI job is run in parallel on several CPUs
- Libraries supported for parallel jobs: only MPICH so far
- Parallel job support still evolving
- A lot of work by CrossGrid



Using MPI on the Grid

- **MPICH is a public domain version of the MPI library**
- **Some Grid-Ireland sites are modified to handle MPI**
- **Procedure:**
  - write your code to use MPI
  - have MPI installed on the worker node
  - specify JobType=”MPICH” in JDL file
  - specify NodeNumber=”<number_of_MPI_processes>” in JDL file
  - specify a MPI wrapper script that takes the MPI job as argument
  - the script should run your code using mpiexec or mpirun

Example: use the Grid to approximate π

MPI Example 1

```
]
Type = "Job";
JobType = "MPICH";

Executable = “MPI_setup.sh";
Arguments = "cpi 2";
NodeNumber = 2;

StdOutput = “cpi.out";
StdError = “cpi.err";

InputSandbox = {“MPI_setup.sh","cpi"};
OutputSandbox = {“cpi.err",“cpi.out","executable.out"};
]
```

forces MPI

MPI wrapper script

# of required CPUs

executable

input files

output files

MPI Example 1

- **Submit the MPI job to the Grid:**
  - edg-job-submit cpi.jdl
- **The Broker will automatically match the queue to the JDL**
  - JobType=”MPICH”
    - means that a MPI-capable queue will be chosen
- **The UI will automatically add the following to your JDL**
  - Member(other.RunTimeEnvironment, “MPICH”);
    - specifies that the queue must be for WNs with MPICH software installed
  - other.TotalCPUs >= NodeNumber;
    - specifies the minimum number of CPUs on the queue
  - Rank = other.FreeCPUs;
    - ranks the queues by number of free CPUs
    - queue with largest no.free CPUs matching all other requirements is chosen

MPI Example 1

- **Unfortunately automatic site adaptation doesn’t yet work**
  - Site-specific MPI setup scripts aren’t yet automatically run
  - So the MPI wrapper script must do the site-specific setup too
    - So automatic queue selection is not yet supported
  - Everything else works fine

- **So must write site-specific wrapper scripts!**
  - Should be fixed very soon

MPI Example 2

```
]
Type = "Job";
JobType = "MPICH";

Executable = “tcdMPI_setup.sh";
Arguments = "cpi 2";
NodeNumber = 2;

StdOutput = “cpi.out";
StdError = “cpi.err";

InputSandbox = {“tcdMPI_setup.sh","cpi"};
OutputSandbox = {“cpi.err",“cpi.out","executable.out"};
]
```

forces MPI

TCD-specific MPI wrapper script

# of required CPUs

executable

input files

output files

MPI Example 2

- Now you must act as Broker
- First discover the queues that support MPI:
  - edg-job-list match cpi.jdl

```
The following CE(s) matching your requirements have been found:
    gridgate.cs.tcd.ie:2119/jobmanager-lcgpbs-cosmo
    gridgate.cp.dias.ie:2119/jobmanager-lcgpbs-cosmo
    gridgate.mp.ucd.ie:2119/jobmanager-lcgpbs-cosmo
```

- Then select a queue and submit the MPI job:
  - edg-job-list match –lrms pbs cpi.jdl \
    -r gridgate.cs.tcd.ie:2119/jobmanager-lcgpbs-cosmo cpi.jdl

Using MPI on the Grid

- **The JDL Requirements attribute can be set to:**
  - Member(“MPICH”, other.GlueHostApplicationSoftwareRunTimeEnvironment)

  ➡ indicates that the MPICH must be installed on the WNs

  - other.GlueCEInfoTotalCPUs >= NodeNumber

  ➡ number of CPUs must be at least equal to NodeNumber
- **The JDL Rank attribute can also be set to:**
  - other.GlueCEStateFreeCPUs

  ➡ the queue with the largest number of free CPUs is chosen

MPI Example 3

- **Let’s do another example**
- **Sample JDL file**
  - JDL file takes the application to run as an argument
  - executes a MPI wrapper script with the application as an argument

```
[
 Type="Job";
 JobType="MPICH";
 NodeNumber=10;
 Executable="MPI-wrapper.sh";
 Arguments="helloworld";
 StdOutput="std.out";
 StdError="std.err";
 InputSandbox={"MPI-wrapper.sh","helloworld.c“};
 OutputSandbox={"std.err","std.out“};
]
```

MPI Example 3

- **Sample wrapper script**
  - compiles the application that was passed in as argument
  - then runs application using mpiexec
  - works at TCD

```
#!/bin/bash -x
# the binary to execute
EXE=$1
# compile source
mpicc -o $(EXE) $(EXE).c
# then execute
mpiexec -mpich-p4-no-shmem `pwd`/$EXE > std.out 2> std.err
```

MPI Example 3

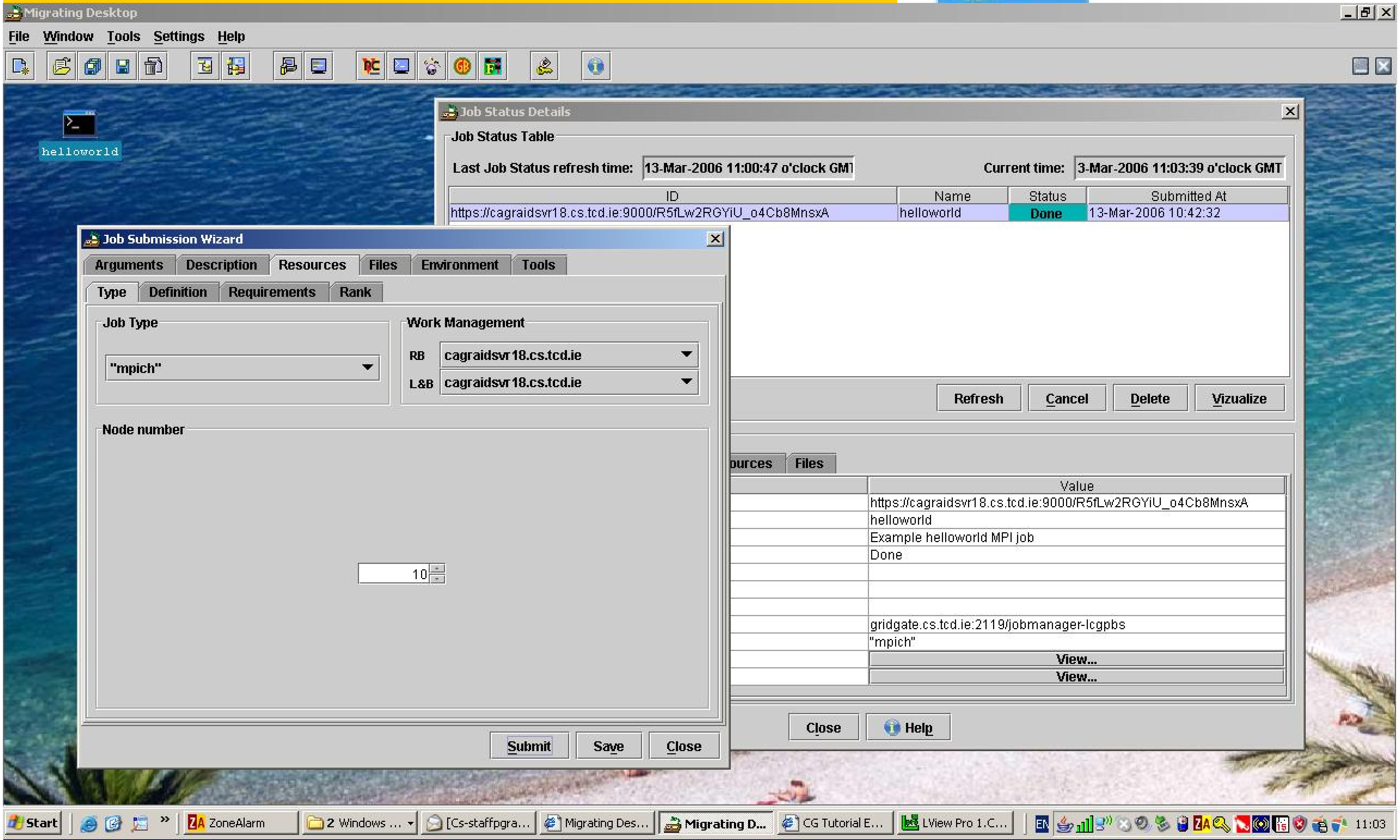
- **Sample C application to run**
  - simple MPI hello world application
  - prints the hostname – can see what nodes are used

```
#include <stdio.h>
#include <mpi.h>
int main (int argc, char *argv[]) {
   int myrank, size;
   MPI_Init(&argc,&argv);                              /* initialize MPI */
   MPI_Comm_rank(MPI_COMM_WORLD,&myrank); /* get my rank
     */
   MPI_Comm_rank(MPI_COMM_WORLD,&size);     /* get total
     no.CPUs */
   printf("Processor %d of %d: Hello World!\n",myrank,size);
   system("/bin/hostname");
   MPI_Finalize();                                     /* terminate MPI */
}
```

Using MPI on the Grid

- No need to change your MPI code
- Simple MPI wrapper script handles compiling and running your code
- JDL file handles running the application on multiple nodes, finding suitable nodes, etc.
- You can run your existing MPI applications with minimal change
- Will run at TCD, but not on DIAS Leda or UCD Rowan
- Can submit on command-line using **edg-job-submit**
- Or you can cheat with the Migrating Desktop

Using MPI on the Grid


Migrating Desktop
File Window Tools Settings Help
helloworld
Job Status Details
Job Submission Wizard
Arguments Description Resources Files Environment Tools
Type Definition Requirements Rank
Refresh Cancel Delete Vizualize
ources Files
Close Help
Submit Save Close
Start ZoneAlarm 2 Windows ... [Cs-staffpgra... Migrating Des... Migrating D... CG Tutorial E... LView Pro 1.C... 11:03

Job Status Table

Last Job Status refresh time: 13-Mar-2006 11:00:47 o'clock GMT

Current time: 3-Mar-2006 11:03:39 o'clock GMT

ID	Name	Status	Submitted At
https://cagraidsvr18.cs.tcd.ie:9000/R5fLw2RGYiU_o4Cb8MnsxA	helloworld	Done	13-Mar-2006 10:42:32

Job Type: "mpich"

Work Management

RB: cagraidsvr18.cs.tcd.ie

L&B: cagraidsvr18.cs.tcd.ie

Node number: 10

	Value
	https://cagraidsvr18.cs.tcd.ie:9000/R5fLw2RGYiU_o4Cb8MnsxA
	helloworld
	Example helloworld MPI job
	Done
	gridgate.cs.tcd.ie:2119/jobmanager-lcgpbs
	"mpich"
	View...
	View...

A real MPI example

- **Gareth Murphy of DIAS has a CFD application to model astrophysical jets flowing into molecular clouds**
  - processes input files
  - outputs a number of data files in HDF5 format
- **Consists of:**
  - a JDL file
  - a MPI wrapper script
  - a tgz file containing required libraries
  - a tgz file containing the executable source and data files

A real MPI example

- **JDL file**
  - Specifies the MPI wrapper script as the executable
  - Specifies the library and code tarballs in the input sandbox
  - Specifies the tarred output files in the output sandbox

```
Type = "Job";
JobType = "MPICH";
NodeNumber = 10;
Executable = "mpi-application.sh";
StdOutput = "std.out";
StdError = "std.err";
InputSandbox = {"mpi-application.sh", "code.tgz", "libraries.tgz"};
OutputSandbox = {"std.out","std.err", "mpi-output.tgz"};
Arguments = "";
RetryCount = 1;
```

A real MPI example

- **MPI wrapper script**
  - untars the libraries and code
  - compiles the code
  - runs the MPI executable
  - tars the output files

```
#!/bin/bash
tar xzvf libraries.tgz
tar xzvf code.tgz
cp lib/* code/lib/
cd code/src/
make
cd ../bin/
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$HOME/code/lib"
mpiexec ./mpi-executable
tar czvf ../../mpi-output.tgz outputfiles*
```

Using MPI on the Grid