



**Interactive European Grid**

# **MPI-START**

**Sven Stork**  
**HRLS, Stuttgart**

*Grids y e-Ciencia 2007, Santander*

- Motivation
  - ▶ Parallel job support in Grids
  - ▶ Problems with the current approach
  - ▶ Future Parallel Job Support in Grid
- mpi-start
  - ▶ Design Goals
  - ▶ Architecture
- Usage
- Advanced Usage

## □ Parallel job support in GRID

- ▶ The current Grid middle ware has only support for “Normal” jobs and “MPICH” jobs.
- ▶ “Normal” JobType
  - Execution of a sequential program
  - Max 1 process allocation.
- ▶ “MPICH” JobType
  - The name is program, only supports MPICH
  - Hard coded into the WMS/RB
  - The grid middle ware produce a wrapper script that executes the binary with (mpich) mpirun

- Problems with hard coded approach
  - ▶ For every new implementation that needs to be supported the middle ware needs to be modified
  - ▶ The modified middle ware need to be compiled again (compiling about 1-2 hours, setting up a proper build environment about a few days)
  - ▶ The change middle ware needs to go through the whole release cycle:
    - Test + Validation
    - Testbed
    - Release
    - takes about 8 month in the EGEE project
  - ▶ Combinations of schedulers and MPI implementations don't work together
    - how to find the hostfile
    - e.g. format of the SGE machinefile is not supported by mpich2.

## □ More Grid specific problem

- ▶ The cluster where the MPI job is supposed to run doesn't have a shared file system.
  - How to distribute the binary and input files ?
  - How to gather the output ?
- ▶ How to compile MPI program ?
  - How can a physicist working on Windows workstation compile his code for/with an Itanium MPI implementation ?
  - License issues when giving people access to compiler ?

## □ Future Parallel Job Support in Grid

### ▶ Short/Medium Term

- Support of MPI in a single cluster
- Support of different MPI implementations simultaneously
- Remove all MPI implementation specific features from the middle ware
  - Int.EU.Grid : Still provide for each MPI implementation a special JobType, but this information is only passed through to a generic wrapper script. The RB will automatically select proper sites.
  - EGEE : MPI Job == Normal Job that is allowed to allocate more than 1 process. The user has to take care about the handling all the complexity of the MPI implementations. The user has to specify the requirements for a suitable site.
- Have an abstraction layer that simplifies/abstracts from the low-level handling

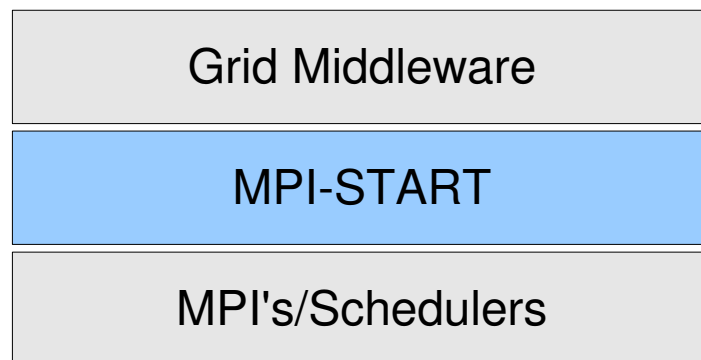
## □ Future Parallel Job Support in Grid

### ▶ Long term

- Have support for the MPI jobs between multiple clusters.
- Have support for other parallel programming models (e.g. shared memory/OpenMP).
  - Problem with allocation of N processes on the same physical node..

## □ Goals of mpi-start

- ▶ Specifies a unique interface to the upper layer to specify an MPI job
- ▶ Support of a new MPI implementation doesn't require any change in the Grid middle ware
- ▶ Support of “simple” 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

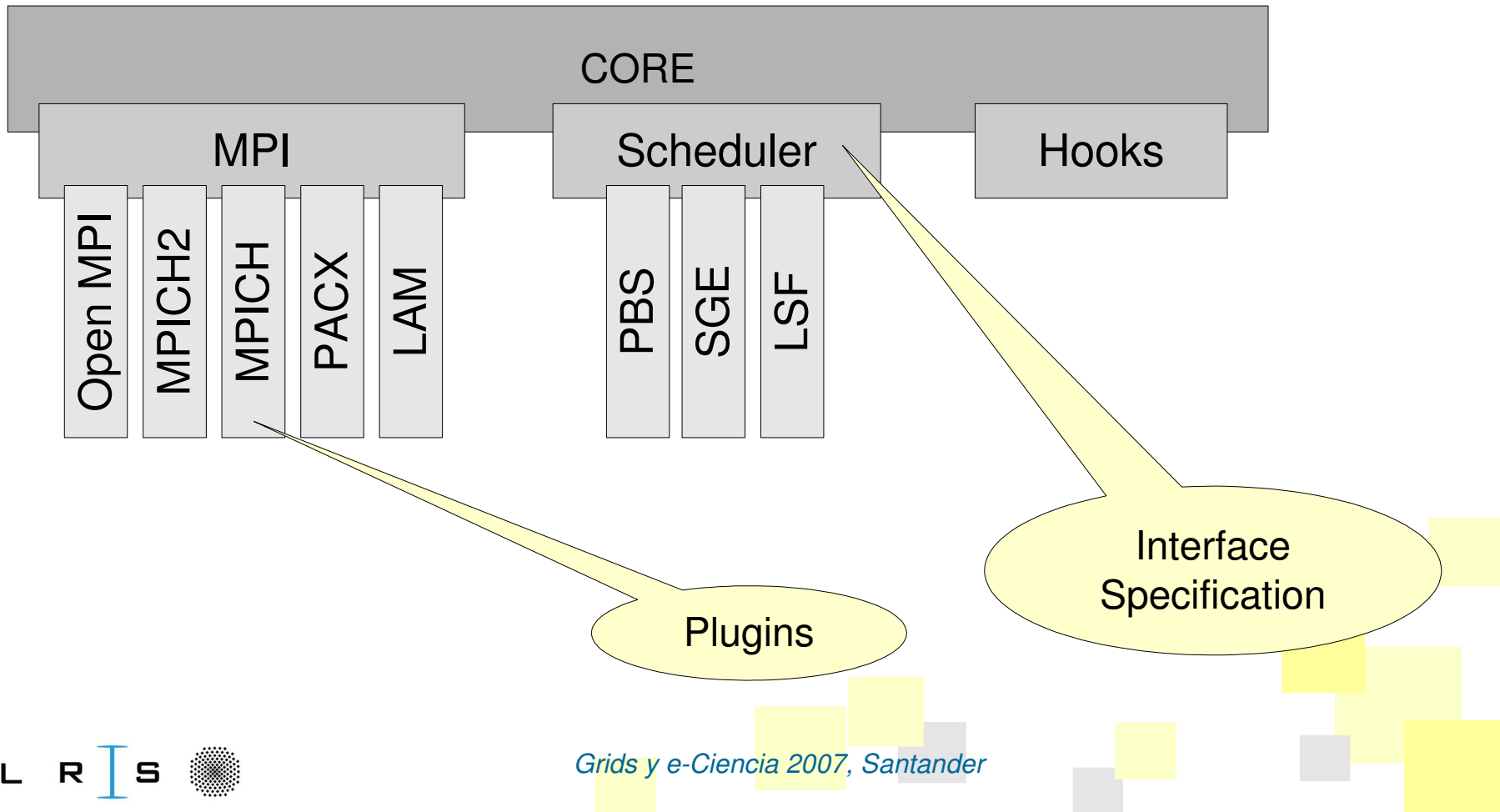
- Plugin/Component architecture

### ▶ Relocatable

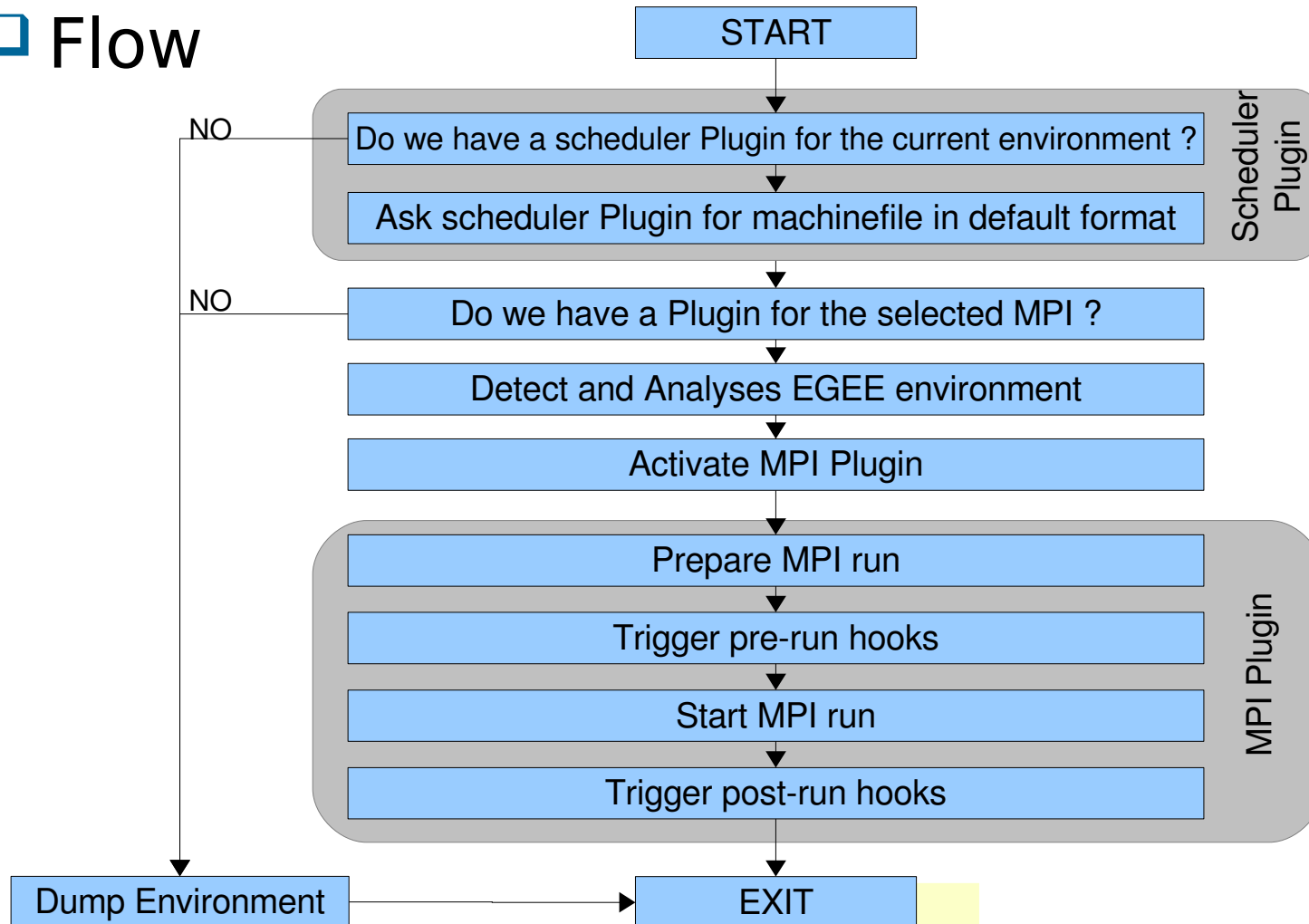
- The program must be independent of absolute path, to adapt to different site configurations.
- Remote “*injection*” of mpi-start along with the job

### ▶ Very good “remote” debugging features

## Architecture



## Flow



## □ 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.

## □ 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 (usage: compilation of the

### ▶ 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 (usage: upload of results).

## □ 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

## □ Example of start script

```
#!/bin/sh
# IMPORTANT : This example script execute a
#             non-mpi  program
#
export I2G_MPI_APPLICATION=/bin/hostname
export I2G_MPI_APPLICATION_ARGS=
export I2G_MPI_NP=2
export I2G_MPI_TYPE=openmpi
export I2G_MPI_FLAVOUR=openmpi
export I2G_MPI_JOB_NUMBER=0
export I2G_MPI_STARTUP_INFO=
export I2G_MPI_PRECOMMAND=
export I2G_MPI_RELAY=

$I2G_MPI_START
```

must be set to  
the absolute path  
of mpi-start

## □ User specified hooks

```
cat > pre_run_hook.sh << EOF
pre_run_hook () {
    echo "pre run hook called "
    # - compile program
    # - fetch input data
    return 0
}
EOF
```

```
cat > post_run_hook.sh << EOF
pre_run_hook () {
    echo "post run hook called "
    # - cleanup
    # - upload results
    return 0
}
EOF
```

```
export I2G_MPI_PRE_RUN_HOOK=./pre_run_hook.sh
export I2G_MPI_POST_RUN_HOOK=./post_run_hook.sh
```



## □ Debugging Support

- ▶ The debugging support is controllable via environment variables
- ▶ The default is **not** to produce any additional output
- ▶ I2G\_MPI\_START\_VERBOSE
  - If set to 1 only very basic information are produced
- ▶ I2G\_MPI\_START\_DEBUG
  - If set to 1 information about the internal flow are outputted
- ▶ I2G\_MPI\_START\_TRACE
  - If set to 1 that “set -x” is enabled at the beginning.

## □ Debugging output example (I2G\_MPI\_START\_VERBOSE=1)

```
*****
UID      = iman003
HOST     = iwra54.fzk.de
DATE     = Mon Dec 18 16:25:31 CET 2006
VERSION  = 0.0.26
*****
mpi-start [INFO   ]: search for scheduler
mpi-start [INFO   ]: activate support for pbs
mpi-start [INFO   ]: activate support for openmpi
mpi-start [INFO   ]: call backend MPI implementation
mpi-start [INFO   ]: start program with mpirun
=[START]=====

<<OUTPUT>>

=[FINISHED]=====
```

- ❑ Simple Job
  - ▶ hostname
  - ▶ IMB (aka Pallas)
- ❑ Jobs with pre-/post-run hook
  - ▶ O3

## □ hostname.jdl

```
Executable      = "/bin/hostname";  
JobType         = "Parallel";  
SubJobType      = "openmpi";  
NodeNumber      = 4;  
StdOutput       = "std.out";  
StdError        = "std.err";  
OutputSandbox   = {"std.out", "std.err"};
```

## stdout.txt

```
wn12-ieg.bifi.unizar.es  
wn12-ieg.bifi.unizar.es  
wn11-ieg.bifi.unizar.es  
wn11-ieg.bifi.unizar.es
```

### ▣ imb.jdl

```
Executable      = "IMB-MPI1";  
Arguments       = "barrier";  
JobType         = "openmpi";  
NodeNumber      = 4;  
StdOutput       = "std.out";  
StdError        = "std.err";  
OutputSandbox  = {"std.out", "std.err"};  
InputSandbox    = {"IMB-MPI1"};
```

## □ stdout.txt

```
#-----  
# Intel (R) MPI Benchmark Suite V2.3, MPI-1 part  
#-----  
# Date      : Mon Aug 13 13:05:03 2007  
# Machine   : i686# System      : Linux  
# Release   : 2.4.21-47.0.1.EL.cernsmp  
# Version   : #1 SMP Thu Oct 19 16:35:52 CEST 2006  
  
...  
  
#-----  
# Benchmarking Barrier  
# #processes = 2  
# ( 2 additional processes waiting in MPI_Barrier)  
#-----  
#repetitions  t_min[usec]  t_max[usec]  t_avg[usec]  
           1000           11.43           11.43           11.43  
  
#-----  
# Benchmarking Barrier  
# #processes = 4  
#-----  
#repetitions  t_min[usec]  t_max[usec]  t_avg[usec]  
           1000           187.78           187.88           187.83
```

## □ o3.jdl

```
JobType           = "openmpi";
NodeNumber        = 8;
VirtualOrganisation = "imain";
Executable        = "o3sg_8";
StdOutput         = "std.out";
StdError          = "std.err";
InputSandbox      = {"o3sg_8", "o3_hooks.sh", "input.8"};
OutputSandbox     = {"std.out", "std.err"};
Environment       = {"I2G_MPI_PRE_RUN_HOOK=./o3_hooks.sh",
                    "I2G_MPI_POST_RUN_HOOK=./o3_hooks.sh"};
```

### o3\_hooks.sh (1/2)

```
#!/bin/sh
export OUTPUT_PATTERN=L8
export OUTPUT_ARCHIVE=output.tar.gz
export OUTPUT_HOST=iwrse2.fzk.de
export OUTPUT_SE=lfn:/grid/imain/sven
export OUTPUT_VO=imain

pre_run_hook () {
}

# the first paramter is the name of a host in the
copy_from_remote_node() {

    if [[ $1 == `hostname` || $1 == 'hostname -f' || $1 == "localhost" ]]; then
        echo "skip local host"
        return 1
    fi

    # pack data
    CMD="scp -r $1:\">$PWD/$OUTPUT_PATTERN\" ."
    echo $CMD
    $CMD
}

...
```



## □ o3\_hooks.sh (2/2)

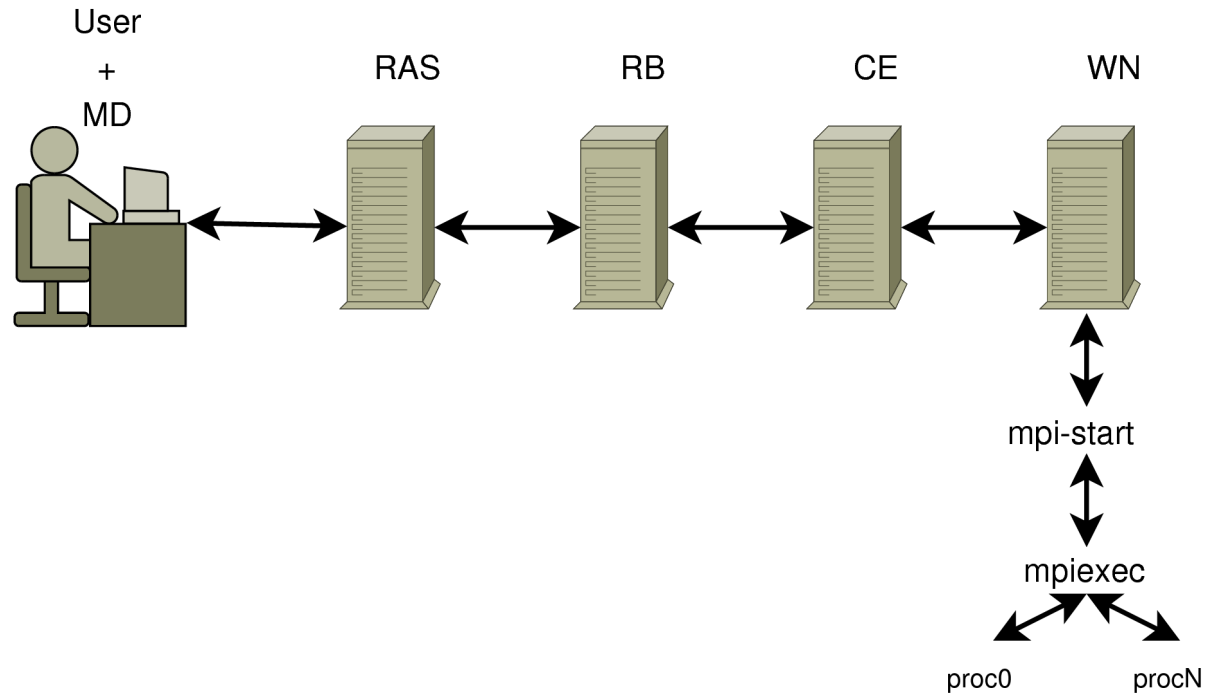
```
...  
post_run_hook () {  
    echo "post_run_hook called"  
  
    if [ "x$MPI_START_SHARED_FS" == "x0" ] ; then  
        echo "gather output from remote hosts"  
        mpi_start_foreach_host copy_from_remote_node  
    fi  
  
    ls -al  
  
    echo "pack the data"  
    tar cvzf $OUTPUT_ARCHIVE $OUTPUT_PATTERN  
    echo "upload the data"  
    lcg-cr --vo $OUTPUT_VO -d $OUTPUT_HOST -l $OUTPUT_SE/$OUTPUT_ARCHIVE  
file://$PWD/$OUTPUT_ARCHIVE  
  
    return 0  
}
```

- Advanced Features
  - ▶ Interactivity
  - ▶ Remote Injection
  - ▶ “Remote Debugging”

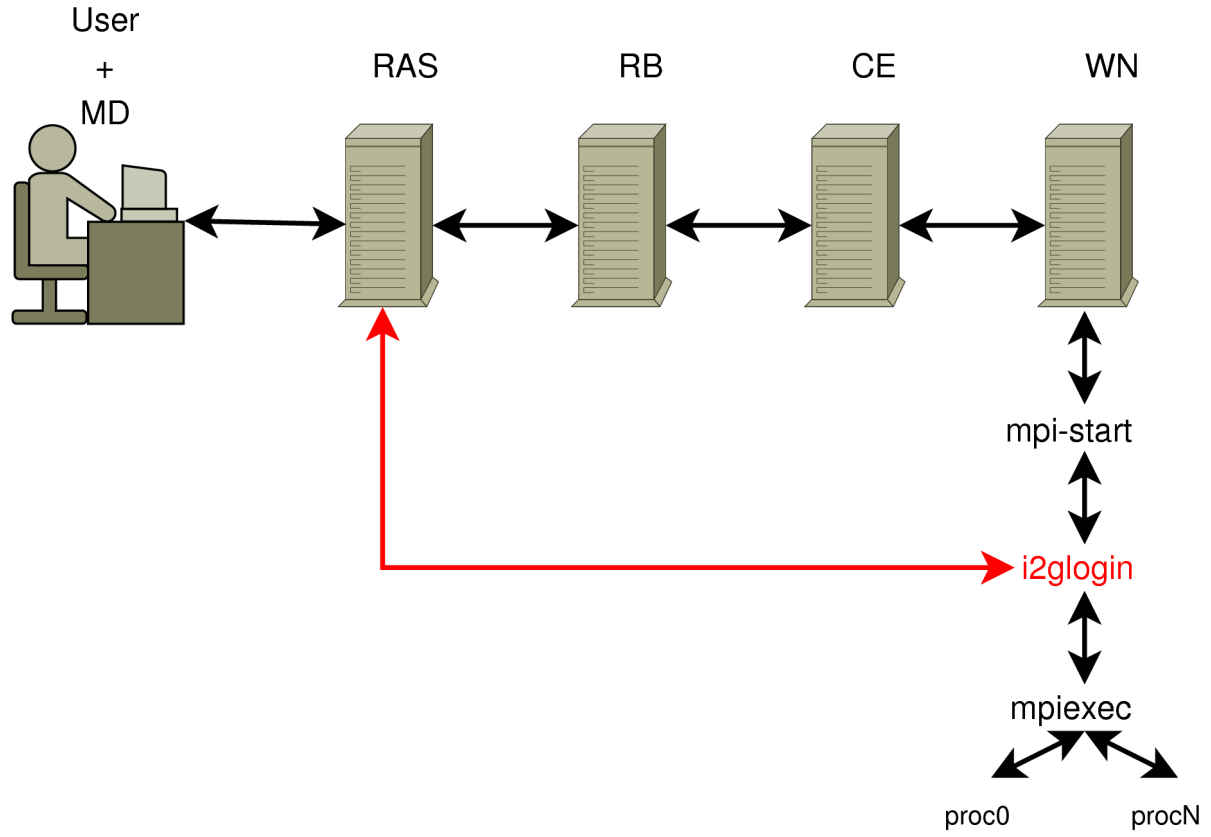
- ❑ mpi-start supports interactivity through special pre-command
  - ▶ \$I2G\_MPI\_PRECOMMAND mpirun ...
- ❑ I2G\_MPI\_PRECOMMAND can be used to let interactive agent “control” the MPI run
  - ▶ redirect I/O
  - ▶ redirect network traffic
  - ▶ ...

# Advanced Features

## Interactivity



# Advanced Features Interactivity



# Advanced Features

## Remote Injection

- mpi-start can send along with the job
  - ▶ just unpack mpi-start
  - ▶ setup environment
  - ▶ go

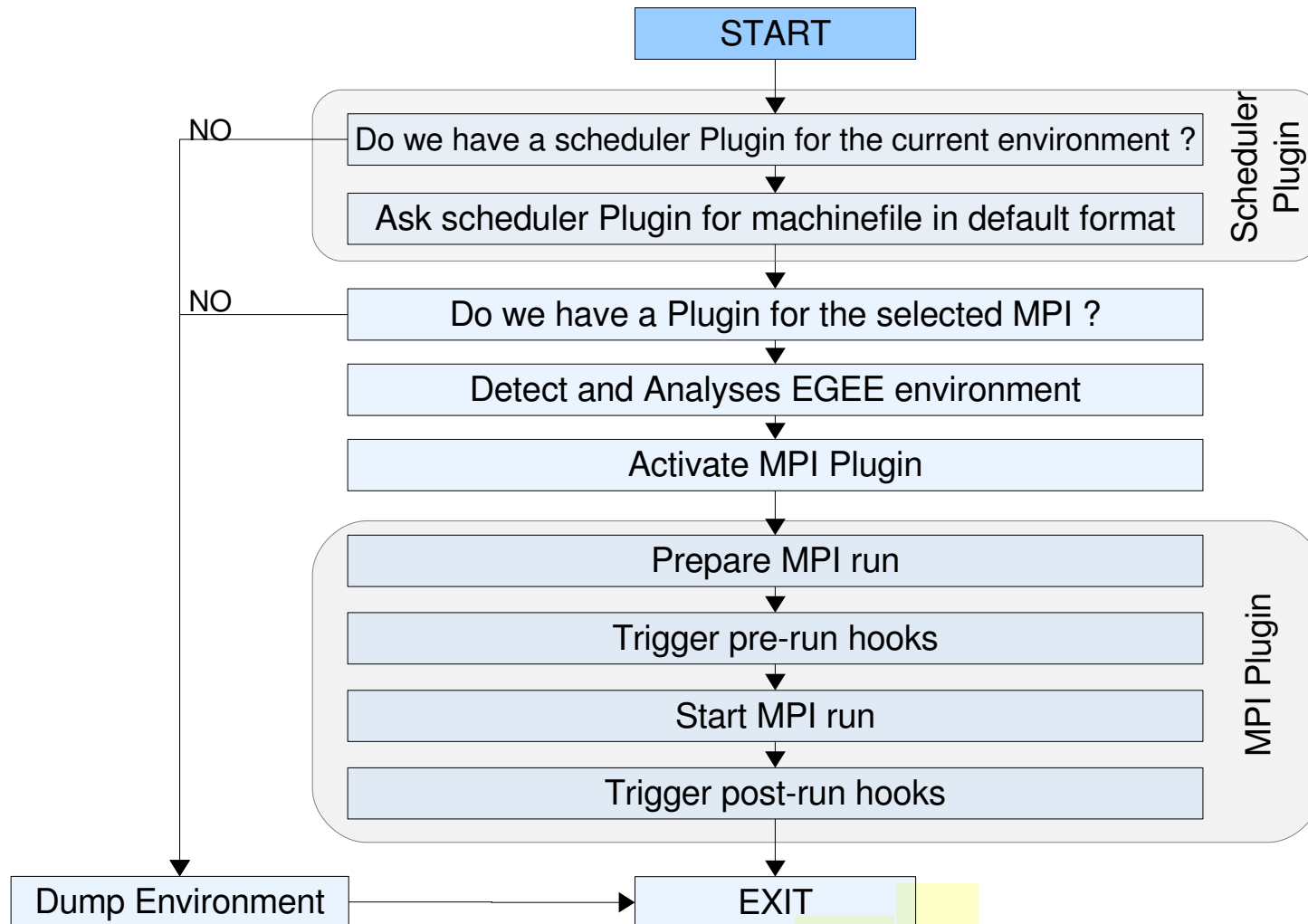
### remote\_injection.jdl

```
Executable      = "my-starter.sh";
JobType         = "Parallel";
SubJobType      = "Plain";
NodeNumber      = 4;
StdOutput       = "std.out";
StdError        = "std.err";
OutputSandbox  = {"std.out", "std.err"};
InputSandbox    = {"my-mpi-start.tar.gz",
                  "my-starter.sh"}
```

just allocate  
nodes

tarball with  
mpi-start

# Advanced Features “Remote Debugging”





## □ header and configuration

```
*****
UID      =   imain003
HOST     =   wn12-ieg.bifi.unizar.es
DATE     =   Mon Aug 13 16:47:31 CEST 2007
VERSION  =   0.0.43
*****

mpi-start [DEBUG ]: dump configuration
mpi-start [DEBUG ]: => I2G_MPI_APPLICATION=/bin/hostname
mpi-start [DEBUG ]: => I2G_MPI_APPLICATION_ARGS=
mpi-start [DEBUG ]: => I2G_MPI_TYPE=openmpi
mpi-start [DEBUG ]: => I2G_MPI_VERSION=
mpi-start [DEBUG ]: => I2G_MPI_PRE_RUN_HOOK=
mpi-start [DEBUG ]: => I2G_MPI_POST_RUN_HOOK=
mpi-start [DEBUG ]: => I2G_MPI_PRECOMMAND=
mpi-start [DEBUG ]: => I2G_MPI_FLAVOUR=openmpi
mpi-start [DEBUG ]: => I2G_MPI_JOB_NUMBER=4
mpi-start [DEBUG ]: => I2G_MPI_STARTUP_INFO=
mpi-start [DEBUG ]: => I2G_MPI_RELAY=ce-ieg.bifi.unizar.es
...
```

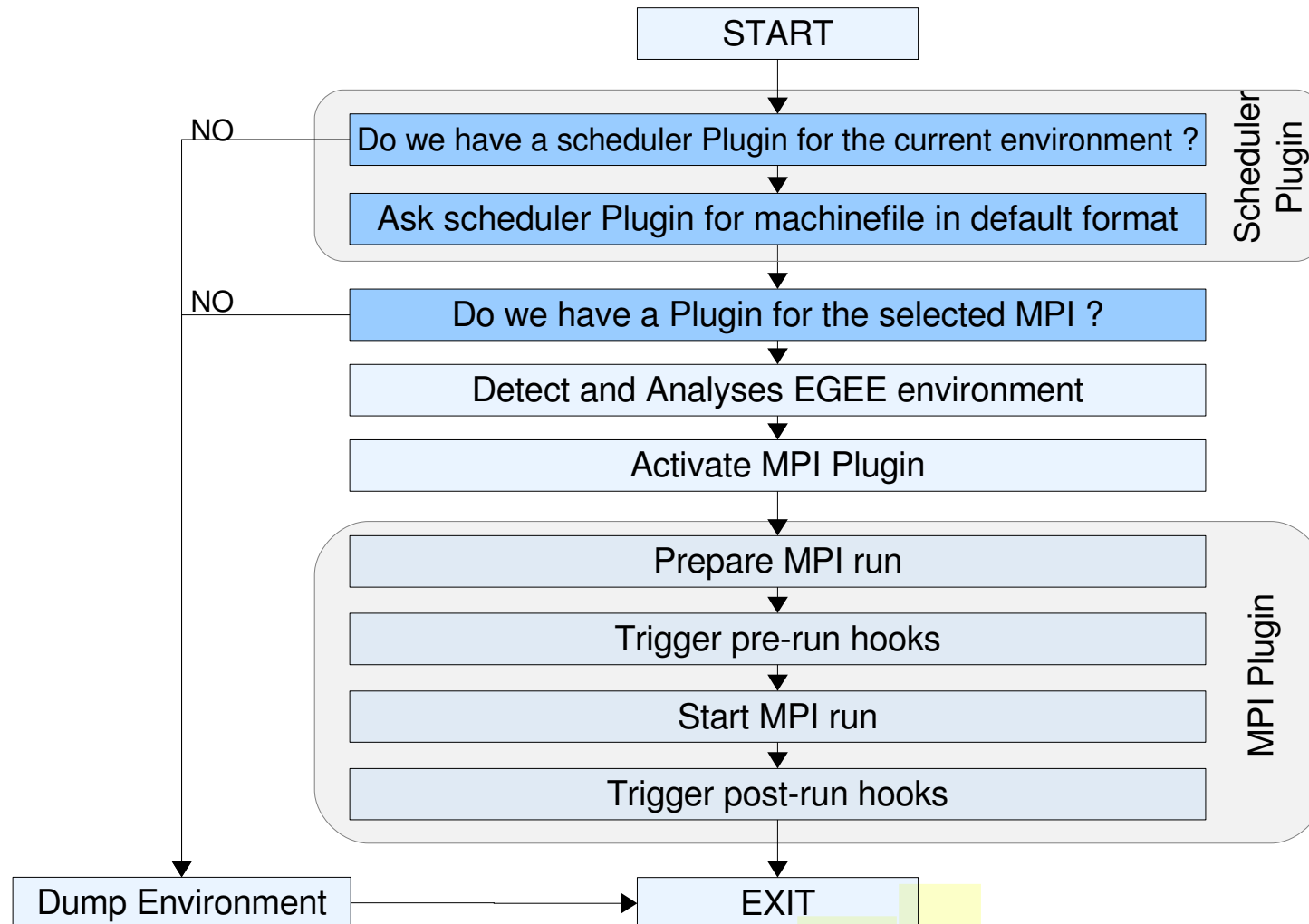
## □ header and configuration

```
*****
UID      =   imain003
HOST     =   wn12-ieg.bifi.unizar.es
DATE     =   Mon Aug 13 16:47:31 CEST 2007
VERSION  =   0.0.43
*****

mpi-start [DEBUG ]: dump configuration
mpi-start [DEBUG ]: => I2G_MPI_APPLICATION=/bin/hostname
mpi-start [DEBUG ]: => I2G_MPI_APPLICATION_ARGS=
mpi-start [DEBUG ]: => I2G_MPI_TYPE=openmpi
mpi-start [DEBUG ]: => I2G_MPI_VERSION=
mpi-start [DEBUG ]: => I2G_MPI_PRE_RUN_HOOK=
mpi-start [DEBUG ]: => I2G_MPI_POST_RUN_HOOK=
mpi-start [DEBUG ]: => I2G_MPI_PRECOMMAND=
mpi-start [DEBUG ]: => I2G_MPI_FLAVOUR=openmpi
mpi-start [DEBUG ]: => I2G_MPI_JOB_NUMBER=4
mpi-start [DEBUG ]: => I2G_MPI_STARTUP_INFO=
mpi-start [DEBUG ]: => I2G_MPI_RELAY=ce-ieg.bifi.unizar.es
...

```

# Advanced Features “Remote Debugging”



# Advanced Features “Remote Debugging”

## □ search for matching scheduler plugin

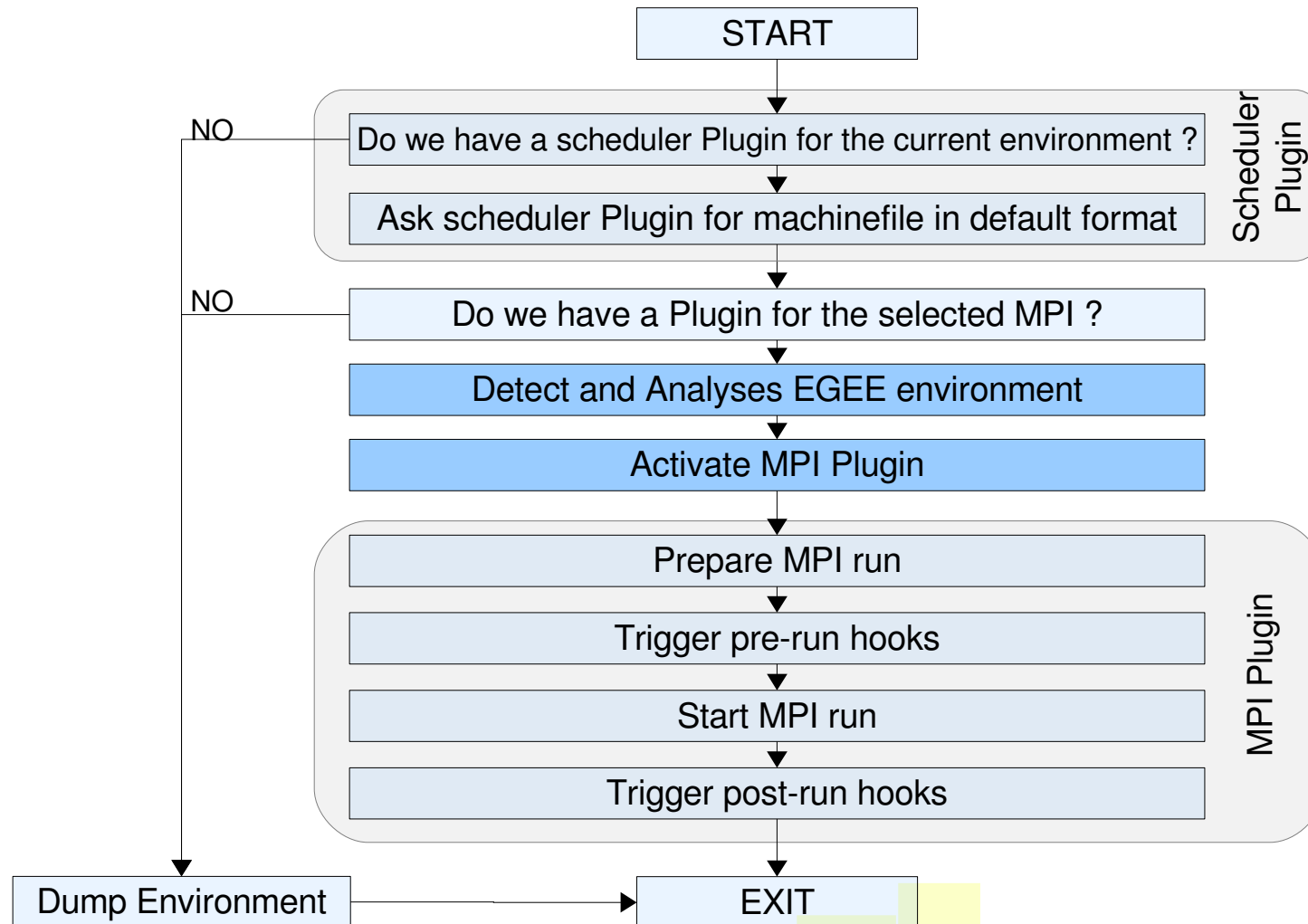
```
mpi-start [INFO ]: search for scheduler
mpi-start [DEBUG ]: source /opt/i2g/bin/../etc/mpi-start/lsf.scheduler
mpi-start [DEBUG ]: checking for scheduler support : lsf
mpi-start [DEBUG ]: checking for $LSF_HOSTS
mpi-start [DEBUG ]: source /opt/i2g/bin/../etc/mpi-start/pbs.scheduler
mpi-start [DEBUG ]: 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
mpi-start [DEBUG ]: Dump machinefile:
mpi-start [DEBUG ]: => wn12-ieg.bifi.unizar.es
mpi-start [DEBUG ]: => wn12-ieg.bifi.unizar.es
mpi-start [DEBUG ]: => wn11-ieg.bifi.unizar.es
mpi-start [DEBUG ]: => wn11-ieg.bifi.unizar.es
mpi-start [DEBUG ]: starting with 4 processes.
```

# Advanced Features “Remote Debugging”

- ❑ dump machine file
- ❑ set MPI processes to number of found CPUs
  - ▶ Scheduler misconfiguration may cause the allocation of less CPUs than request (e.g. 1 CPU)

```
mpi-start [INFO ]: search for scheduler
mpi-start [DEBUG ]: source /opt/i2g/bin/./etc/mpi-start/lsf.scheduler
mpi-start [DEBUG ]: checking for scheduler support : lsf
mpi-start [DEBUG ]: checking for $LSF_HOSTS
mpi-start [DEBUG ]: source /opt/i2g/bin/./etc/mpi-start/pbs.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
mpi-start [DEBUG ]: Dump machinefile:
mpi-start [DEBUG ]: => wn12-ieg.bifi.unizar.es
mpi-start [DEBUG ]: => wn12-ieg.bifi.unizar.es
mpi-start [DEBUG ]: => wn11-ieg.bifi.unizar.es
mpi-start [DEBUG ]: => wn11-ieg.bifi.unizar.es
mpi-start [DEBUG ]: starting with 4 processes.
```

# Advanced Features “Remote Debugging”

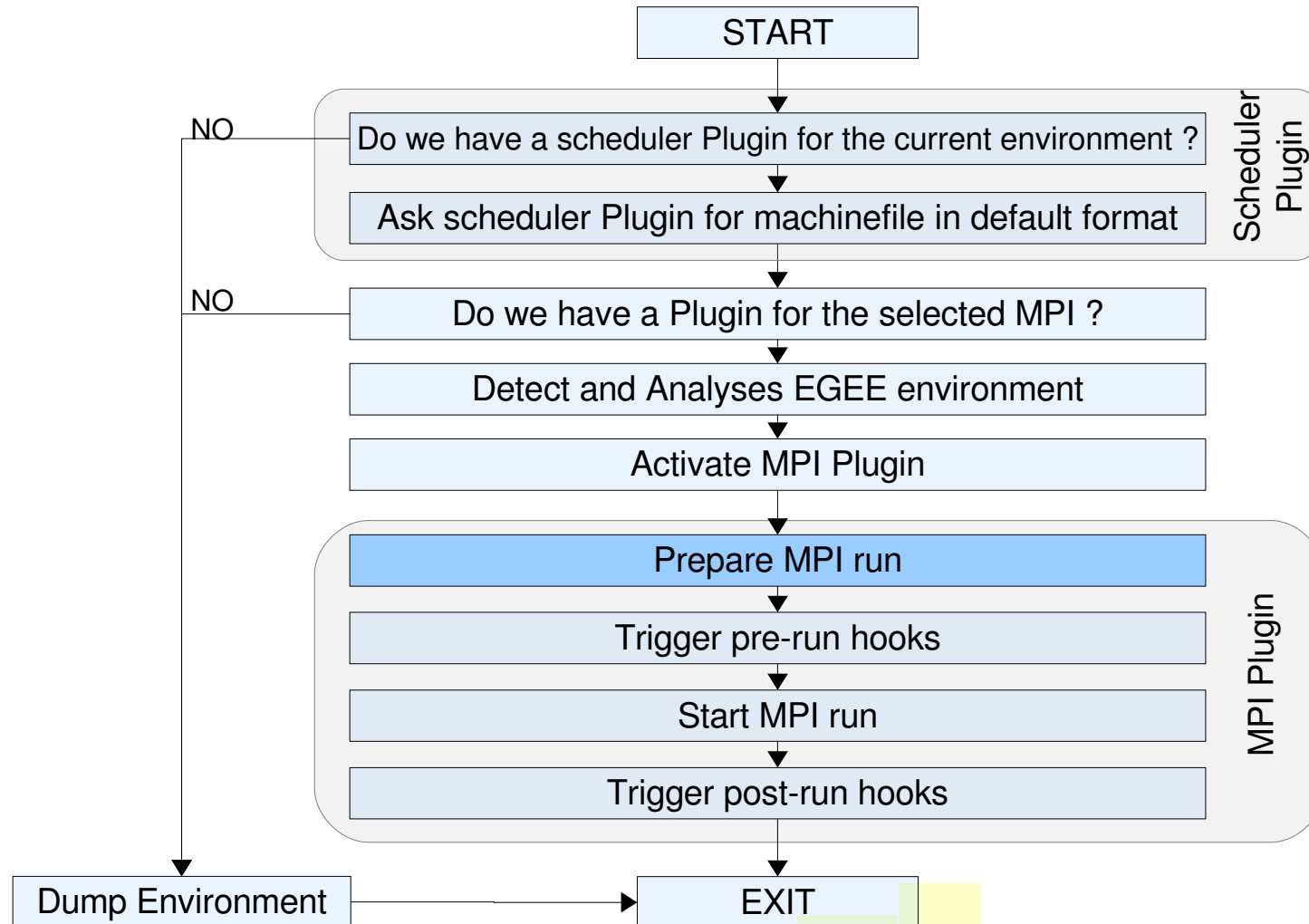


# Advanced Features “Remote Debugging”

- analyze and check EGEE environment
- load MPI plugin

```
mpi-start [DEBUG ]: check for EGEE environment
mpi-start [DEBUG ]: using user requested MPI flavour
mpi-start [DEBUG ]: check for default MPI version
mpi-start [DEBUG ]: couldn't find EGEE environment
mpi-start [INFO ]: activate support for openmpi
mpi-start [DEBUG ]: source : /opt/i2g/bin/./etc/mpi-start/openmpi.mpi
```

# Advanced Features “Remote Debugging”



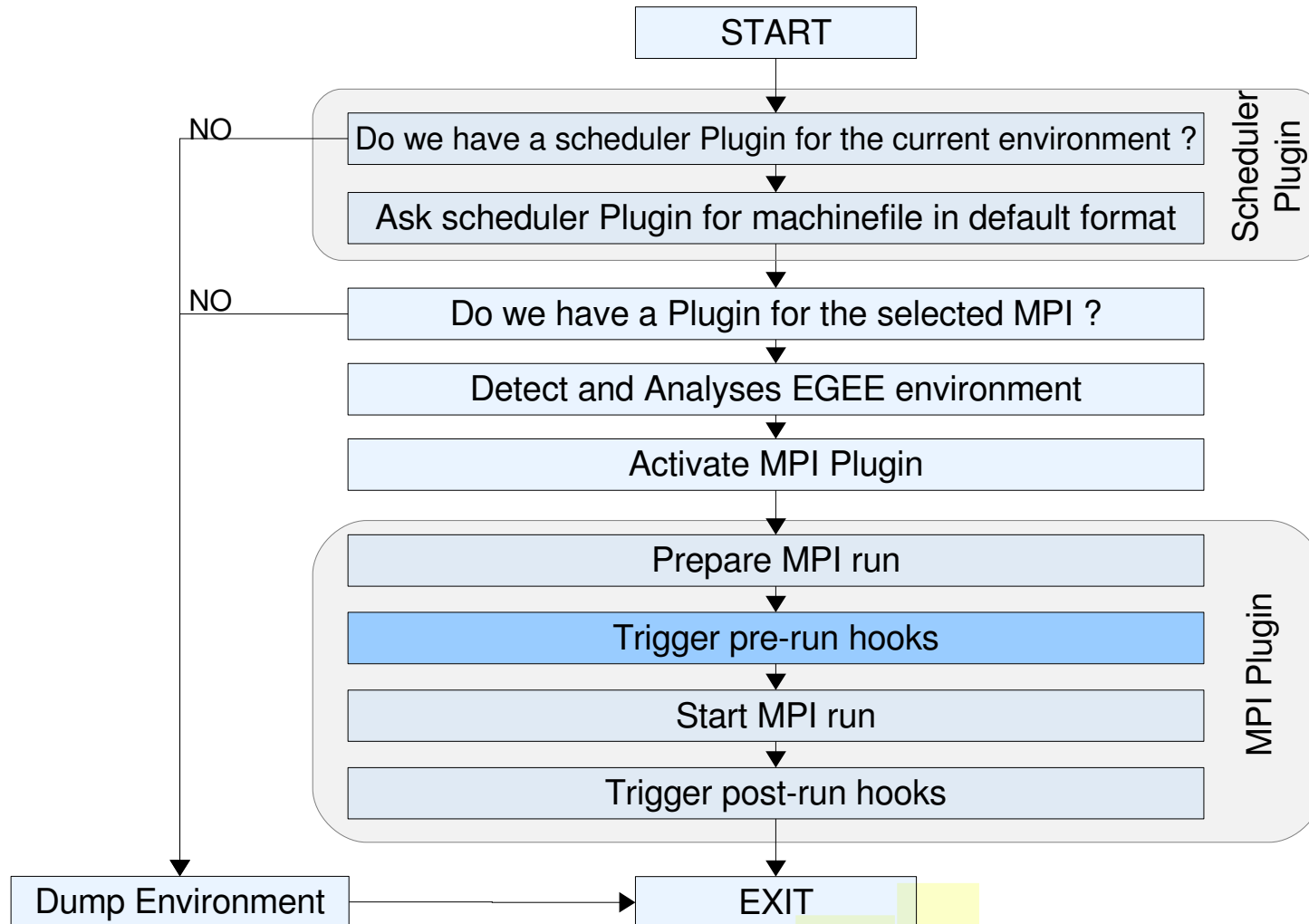


# Advanced Features “Remote Debugging”

- setup the environment for the selected MPI implementation
  - ▶ setup PATH and LD\_LIBRARY\_PATH
    - using modules
    - manual update

```
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
mpi-start [INFO  ]: start program with mpirun
```

# Advanced Features “Remote Debugging”



# Advanced Features “Remote Debugging”

- sophisticated shared filesystem detection
  - ▶ checking for mounted filesystems
  - ▶ correct handling of symlinks

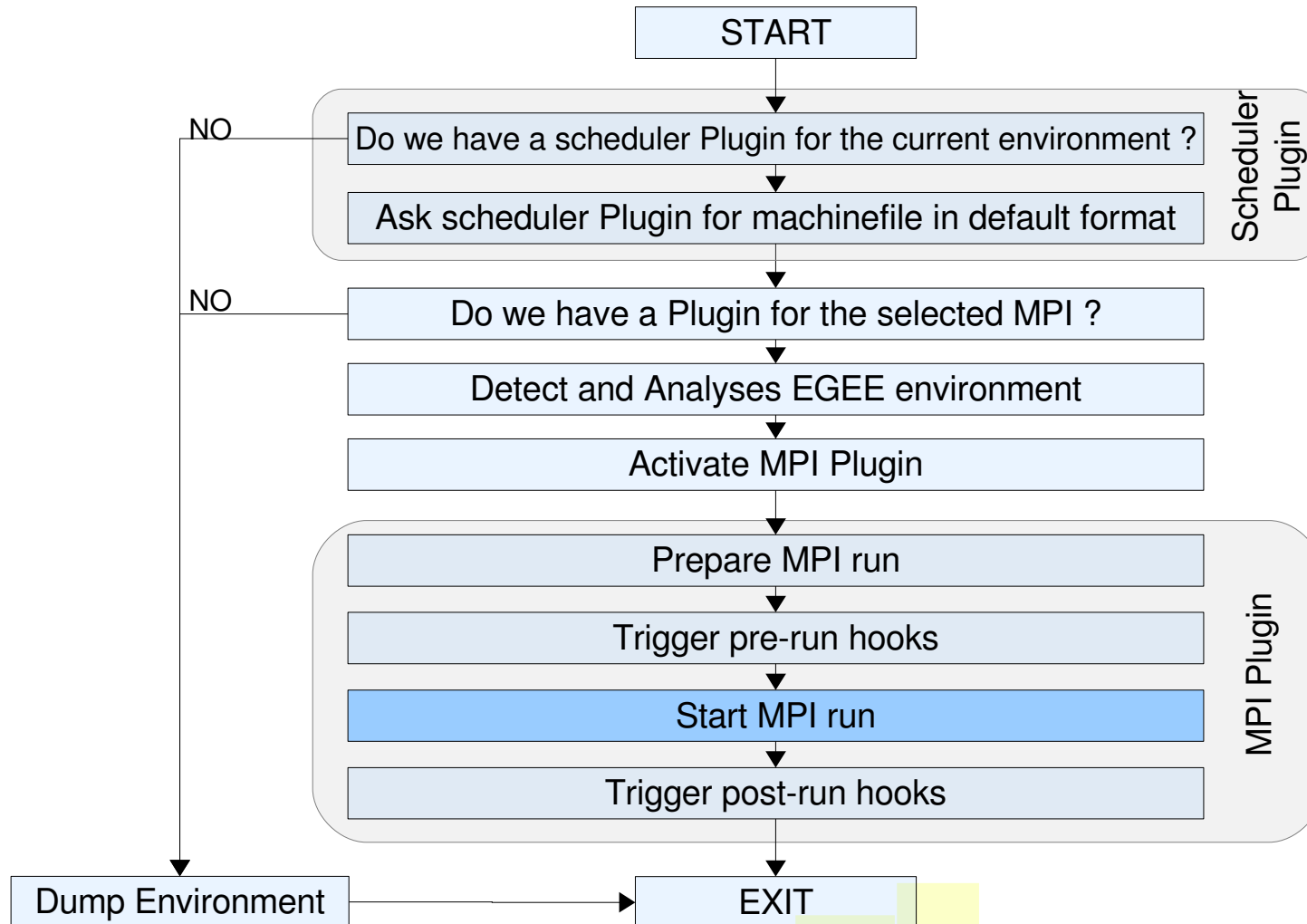
```
mpi-start [DEBUG ]: mpi_start_pre_run_hook
mpi-start [DEBUG ]: mpi_start_pre_run_hook_generic
mpi-start [DEBUG ]: detect shared filesystem
mpi-start [DEBUG ]: dump mount point information:
mpi-start [DEBUG ]: => / = ext3
mpi-start [DEBUG ]: => /proc = proc
mpi-start [DEBUG ]: => /dev/pts = devpts
mpi-start [DEBUG ]: => /proc/bus/usb = usbdevfs
mpi-start [DEBUG ]: => /dev/shm = tmpfs
mpi-start [DEBUG ]: => /opt/exp_soft = nfs
mpi-start [DEBUG ]: => /root/conf_ieg = nfs
mpi-start [DEBUG ]: current working directory : /home/imain003/globus-
tmp.wn12-ieg.650.0/https_3a_2f_2fi2g-
rb01.lip.pt_3a9000_2f72Bm6Y433WNkuIA9eTTH8A_0
mpi-start [DEBUG ]: found local fs : ext3
```

# Advanced Features “Remote Debugging”

- Distribute binary if required (non-shared file system)

```
mpi-start [DEBUG ]: mpi_start_post_run_hook_copy_ssh
mpi-start [DEBUG ]: fs not shared -> distribute binary
mpi-start [DEBUG ]: distribute "/bin/hostname" to remote node : wn11-
ieg.bifi.unizar.es
Scientific Linux CERN Release 3.0.8 (SL)
Scientific Linux CERN Release 3.0.8 (SL)
mpi-start [DEBUG ]: distribute "/bin/hostname" to remote node : wn12-
ieg.bifi.unizar.es
mpi-start [DEBUG ]: skip local machine
```

# Advanced Features “Remote Debugging”

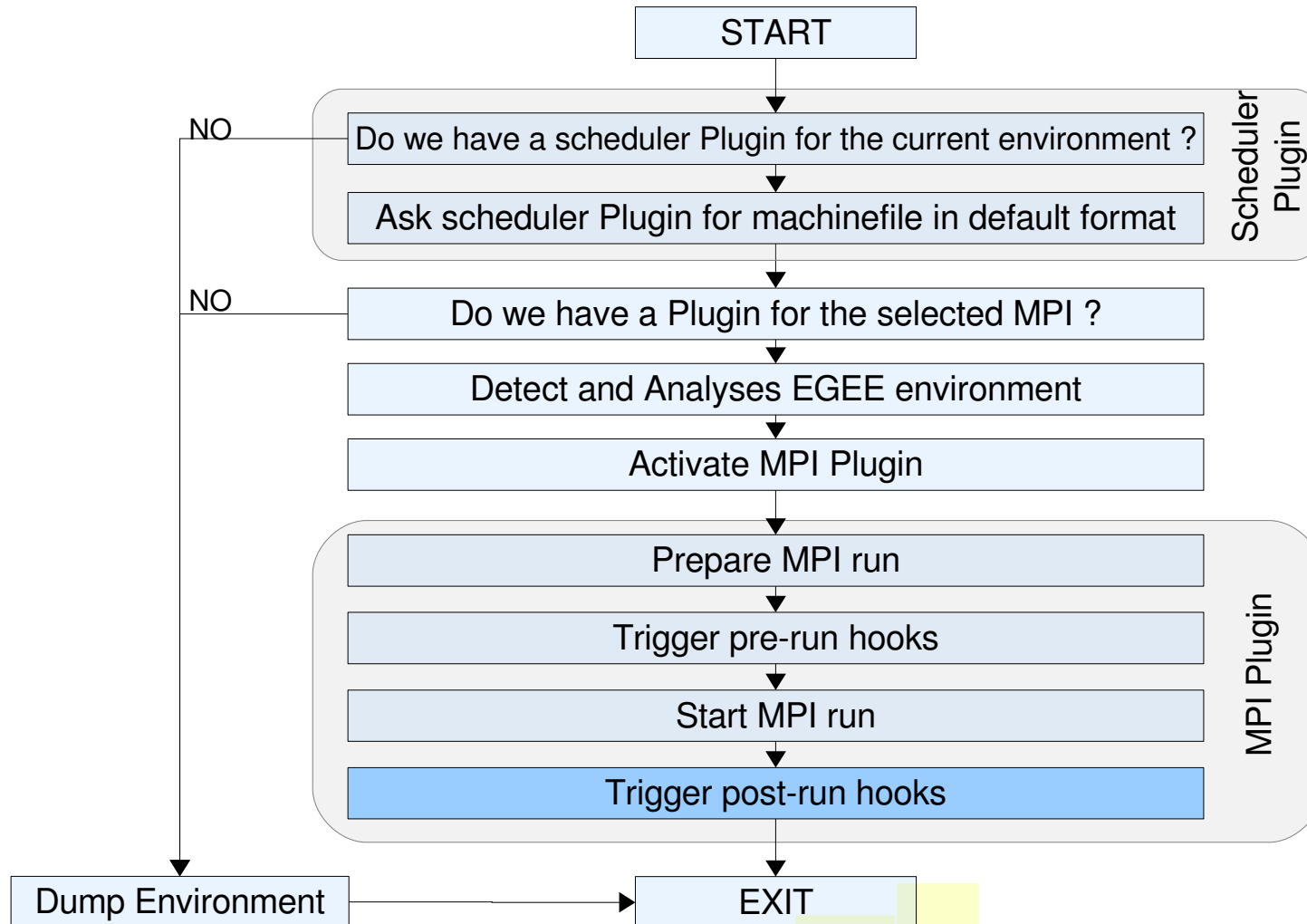


# Advanced Features “Remote Debugging”

## □ execute application

```
=[ START ]=====
mpi-start [DEBUG ]: /opt/i2g/openmpi/bin/mpixec -x X509_USER_PROXY --
prefix /opt/i2g/openmpi -machinefile /var/spool/pbs/aux//13646.ce-
ieg.bifi.unizar.es -np      4 /bin/hostname
wn11-ieg.bifi.unizar.es
wn12-ieg.bifi.unizar.es
wn11-ieg.bifi.unizar.es
wn12-ieg.bifi.unizar.es
=[ FINISHED ]=====
```

# Advanced Features “Remote Debugging”



# Advanced Features “Remote Debugging”

## □ trigger post-run hooks

```
mpi-start [DEBUG ]: mpi_start_post_run_hook
mpi-start [DEBUG ]: mpi_start_post_run_hook_generic
mpi-start [DEBUG ]: mpi_start_post_run_hook_generic
mpi-start [DEBUG ]: fs not shared -> cleanup binary
mpi-start [DEBUG ]: cleanup "/bin/hostname" to remote node : wn11-
ieg.bifi.unizar.es
Scientific Linux CERN Release 3.0.8 (SL)
mpi-start [DEBUG ]: cleanup "/bin/hostname" to remote node : wn12-
ieg.bifi.unizar.es
mpi-start [DEBUG ]: skip local machine
```