

# Improvements on the EGEE code FAFNER2

*M. Rodríguez<sup>1</sup>, J. Guasp<sup>1,2</sup>, F. Castejón<sup>1,2</sup>,  
I.M. Llorente<sup>3</sup>, A.J. Rubio-Montero<sup>1</sup>, R. Mayo<sup>1</sup>*

**1. CIEMAT**

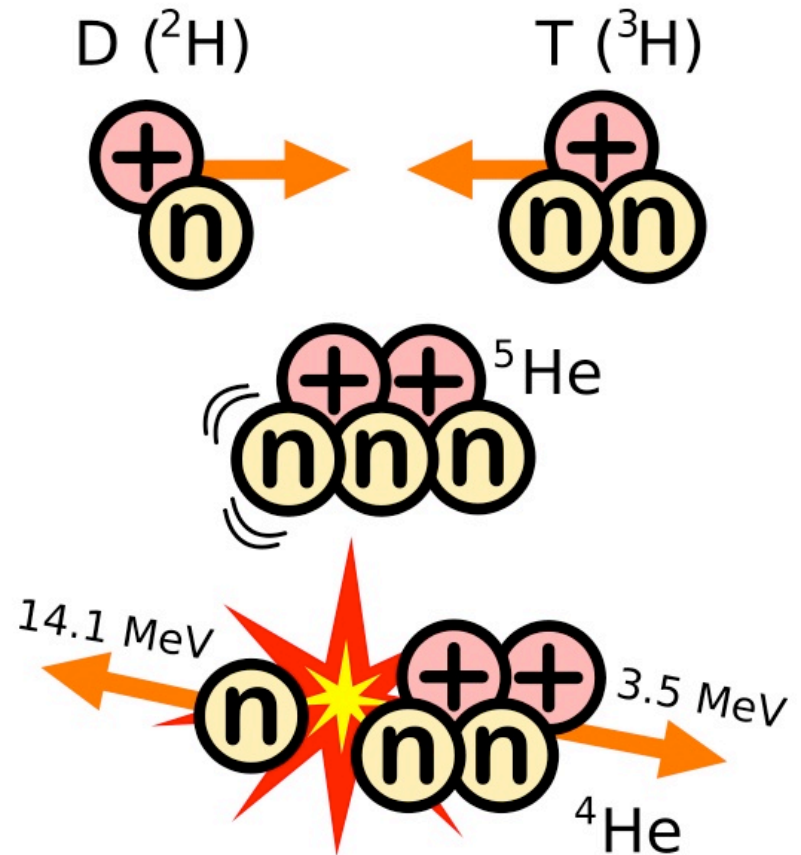
**2. Laboratorio Nacional de Fusión**

**3. DSA-Research Group**

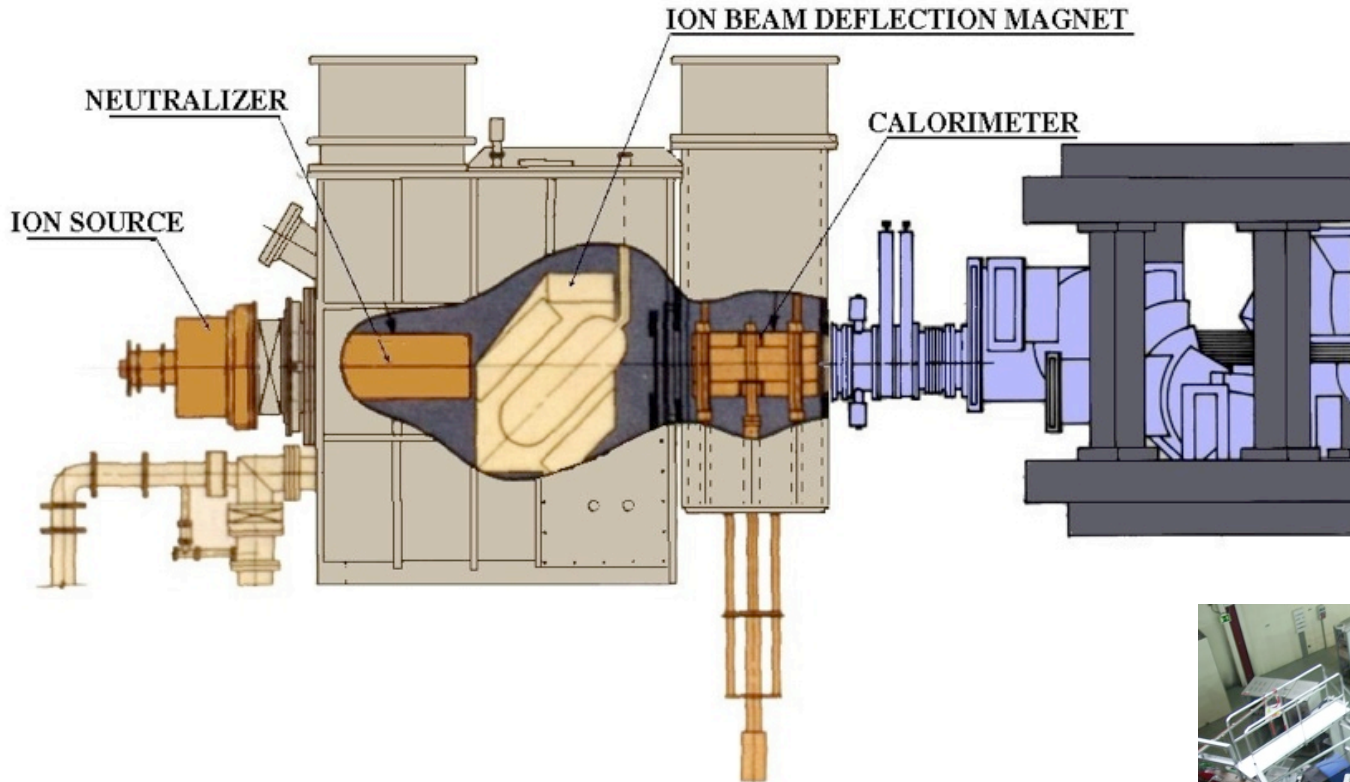
- **Nuclear Fusion**
- **The code**
- **From MPI to DRMAA**
  - High level description
  - Low level description
- **Performance**

- **Nuclear Fusion**
- **The code**
- **From MPI to DRMAA**
  - High level description
  - Low level description
- **Performance**

- $^2\text{H} + ^3\text{H} \rightarrow ^4\text{He} + \text{n}$
- $T > 10^6$  Kelvin
- How? With a magnetic field



- Employed at TJ-II and ITER



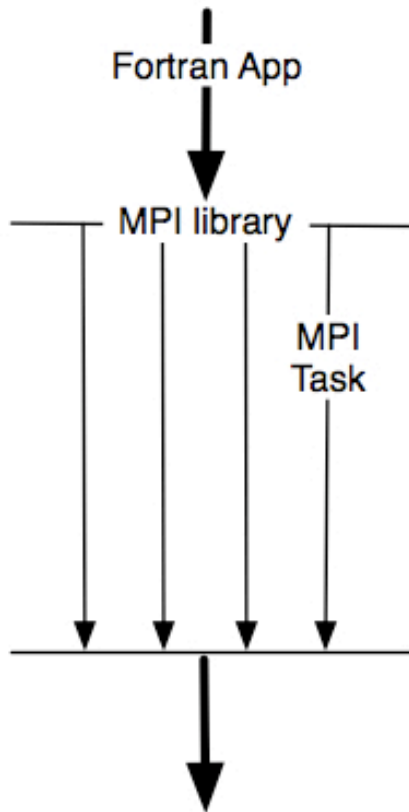
- **Nuclear Fusion**
- **The code**
- **From MPI to DRMAA**
  - High level description
  - Low level description
- **Performance**

- **Monte Carlo code**
- **Original version** <sup>1</sup>
  - Modified to include TJ-II geometry <sup>2</sup>
  - updated
    - Irix O.S. over MIPS
    - Parallel library: SHMEM
- **MPI version** <sup>3</sup>
  - Linux over X86
  - Parallel library: MPI
  - Grid Enabled
- **References**
  - 1. F.G. Lister. FAFNER: A Fully 3-D Neutral Beam Injection Code Using Monte Carlo Methods. Max-Planck-Institut für Plasmaphysik, 1985
  - 2. J.A. Teubel. Monte Carlo simulations of NBI into the TJ-II helical axis stellarator. Report IPP 4/268, 264(4), 1994.
  - 3. EGEE'08: FAFNER-2: adaptation of a code for estimating NBI heating of fusion plasmas on the Grid

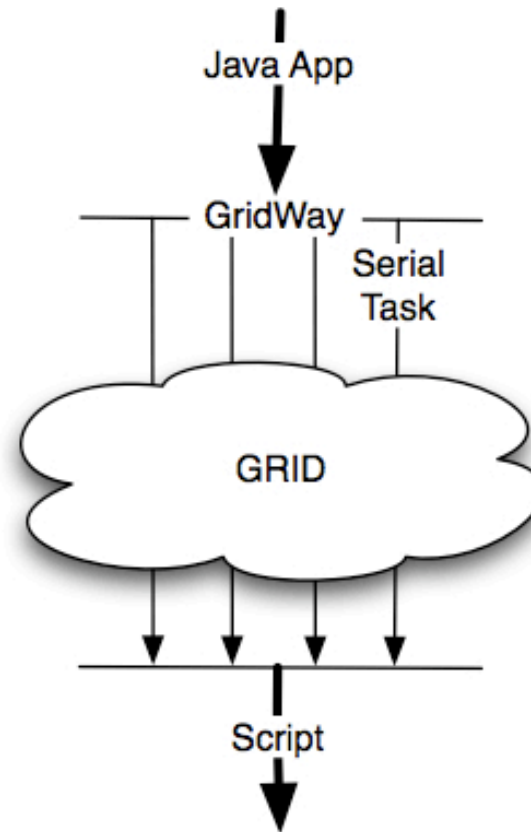
- **Nuclear Fusion**
- **The code**
- **From MPI to DRMAA**
  - High level description
  - Low level description
- **Performance**



## MPI



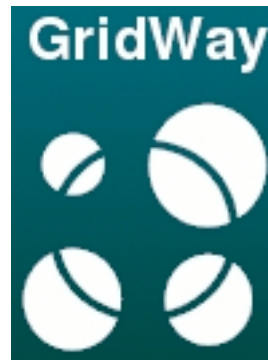
## DRMAA



- **Benefits**
  - portability
    - when we started working, MPI not in every site
  - scalability
  - fault tolerance
  - simpler is better
- **Drawbacks**
  - increased network traffic
- **We are maintaining both versions, MPI and DRMAA**

- **Analyze the code**
  - input data & division of the problem
  - output data & partial results
- **Remove MPI from code**
  - serial application with same behavior than a 1-thread MPI
- **Create java DRMAA application**
- **Create script that operates with partial results**
- **Metascheduler: GridWay**

- **Forget about grid!**
  - scheduling
  - job submission & monitoring
  - fault detection & recovery
- **Straightforward connection with Java DRMAA application**



<http://www.gridway.org>

- **Nuclear Fusion**
- **The code**
- **From MPI to DRMAA**
  - High level description
  - Low level description
- **Performance**

- **Nobody likes code, so this will be short...**

## Create job template...

```
jt.setRemoteCommand(myScript); SIMPLE SCRIPT
```

```
jt.setArgs(DRMAA_GW_TASK_ID); INPUT PARAM
```

```
jt.setInputFiles("bin.tar.gz"); EXECUTABLE & DATA
```

```
jt.setOutputFiles("results" + DRMAA_GW_TASK_ID +  
".tar.gz"); RESULTS
```

```
jt.setRescheduleOnFailure("yes"); FAULT TOLERANT
```

## ... and execute it

```
session.runBulkJobs(jt, number); RUN
```

- **compilation**

- Include the DRMAA library

- `import org.ggf.drmaa.*;`

- Compiling and linking options:

- `-cp $GW_LOCATION/lib/drmaa.jar:$GW_LOCATION/include/`

- **Execution**

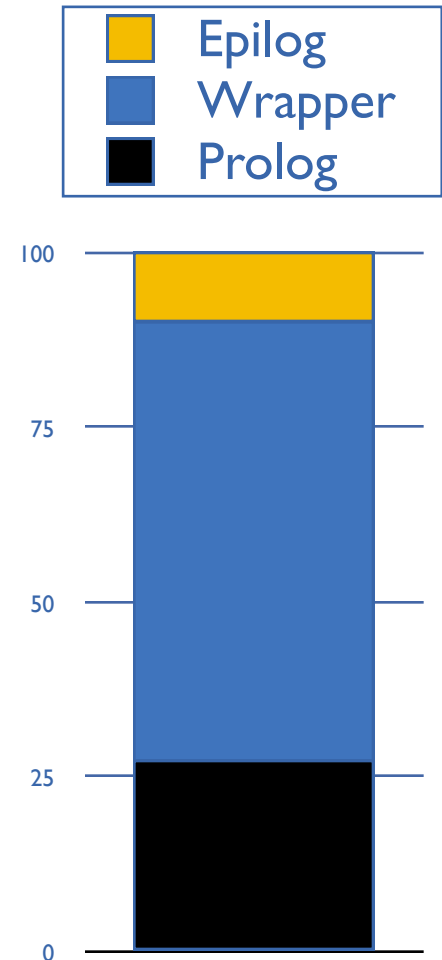
- include GW library

- `-classpath $GW_LOCATION/lib/drmaa.jar`



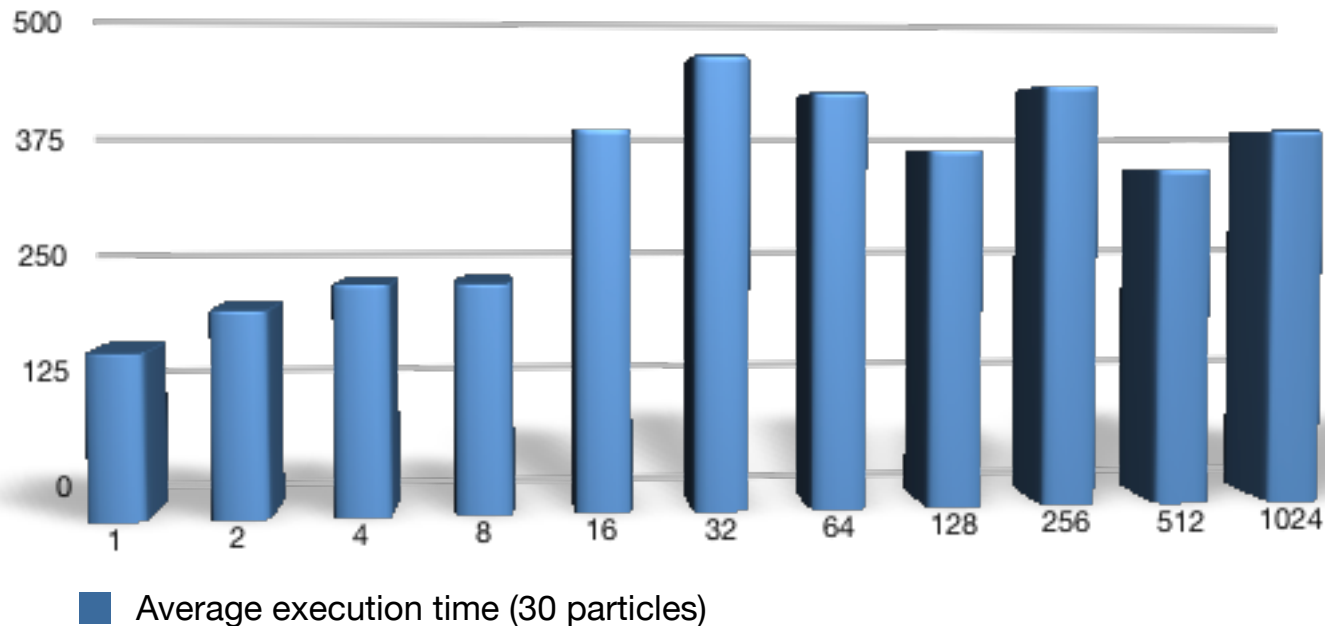
- **Nuclear Fusion**
- **The code**
- **From MPI to DRMAA**
  - High level description
  - Low level description
- **Performance**

- **Local**
  - Java DRMAA application
  - post process script
- **On the Grid**
  - Prolog:
    - *create remote directory*
    - *transfer input files*
  - Wrapper
    - *execute the job*
    - *obtain exit status code*
  - Epilog
    - *transfer output files*
    - *clean remote directory*



- **Local**
  - Java DRMAA
    - ~0
  - Post process script: lineal in time
    - output files must be uncompressed and read
    - ~2 seconds/task
- **On the Grid**
  - Prolog: lineal in network traffic
    - input data (~34 MB) must be transfered for each task
  - Epilog: lineal in network traffic
    - output data (~10 KB) must be transfered for each task
  - Wrapper?

- **Wrapper**
  - time/task, with different number of tasks
  - ~constant: scales correctly



**Thanks for your attention**