

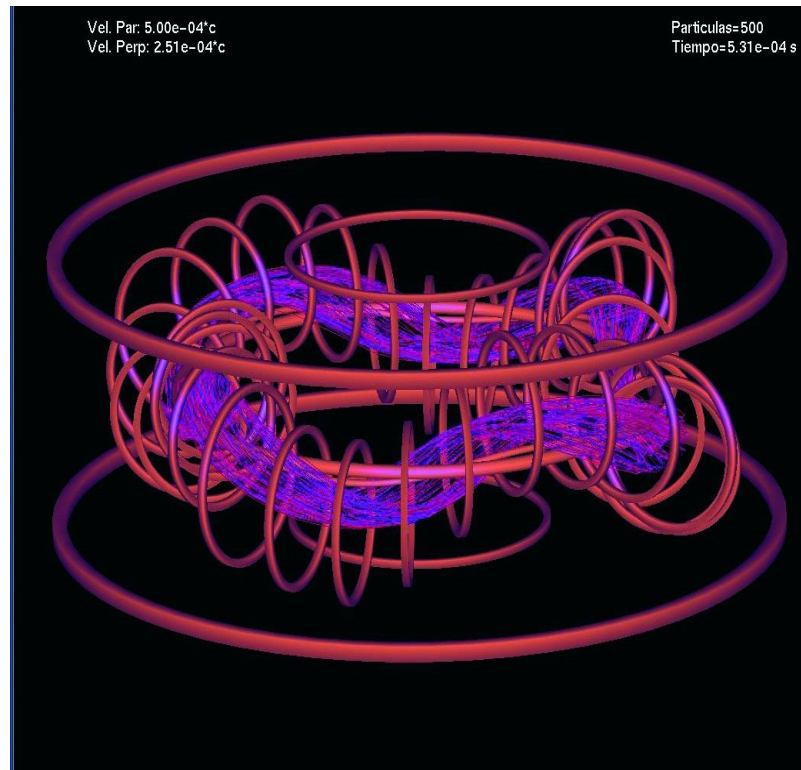


int.eu.grid

- *Fusion Plasma Application*
  - Done's
    - Description
    - Graphical Interface
    - Trajectories computing core
  - Do's
    - Description
    - Graphical Interface migration
      - Pros
      - Cons
    - Adaptation to int.eu.grid particularities
    - Migrating Desktop
- Suggestions are welcomed

### ■ *Description*

- The fusion application is based on a computing core which calculates the trajectories of particles inside a fusion device that are represented in a 3D environment in real time.

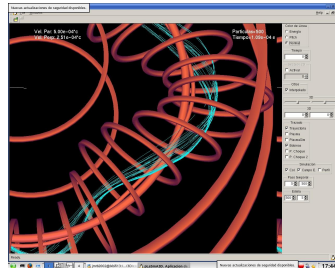


- Summarizing, the application depends on 4 main variables:

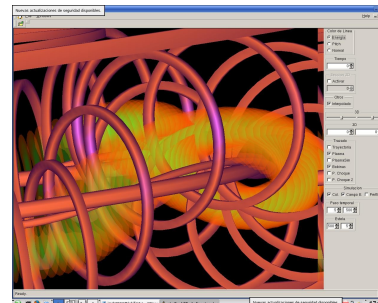
- TJ-II: Fusion Reactor



- Particles



- Plasma fusion



- Vacuum chamber

- The core of the application calculates the position of every particle in a determined time depending on some initial parameters (speed, position...) the definition of the vacuum chamber and the density of the particles inside the reactor, which we call plasma.
  
- With all these parameters, nowadays, the application has been tested on a single computer satisfactory with the order of  $10^3$  particles.
  - Pentium D
  - 2 GB RAM
  - 3,4 GHz dual core
  - EMT 64 extension
  - 2 Nvidia GTX 7800 256 MB RAM Graphic cards

- *Graphical Interface*

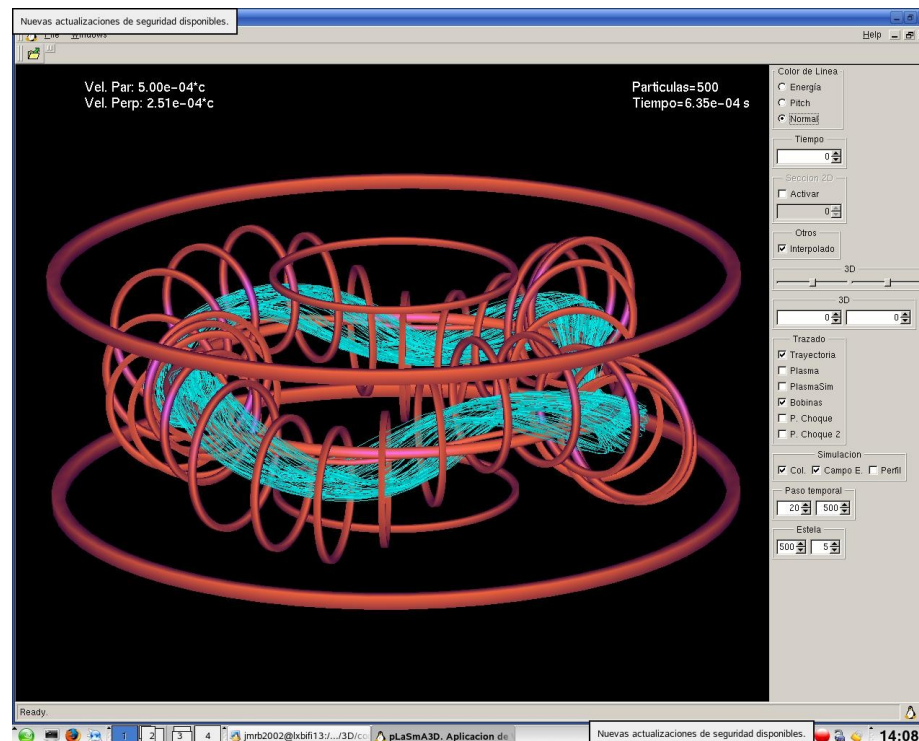
- **FoxToolkit:** FOX stands for Free Objects for X. It is a C++ based class library for building Graphical User Interfaces
- **C**
- **C++**
- **Mesa:** Mesa is a 3-D graphics library with an API which is very similar to that of OpenGL.
- **GLUT**

### ■ *Trajectories Computing Core*

- It is a C program which deals with the Stochastic Differential Equation that governs the evolution of plasma. It solves it by following a great number of particles trajectories, and obtaining averages of the relevant magnitudes
- It has already been successfully ported to the GRID.
- It has been tested on EGEE II fusion VO, and production runs are planned to start soon
- Using GRID technologies, we expect to achieve more precision in shorter periods of time
- In addition, we will be able to obtain relevant interactive graphical results, for example a "map" of the collision points of particles with the vacuum chamber of the TJ-II

### ■ Description

- The fusion application is planned to be ported to int.eu.grid submitting a great number of particles which positions can be calculated in different WNs and the parameters affecting them, changed interactively viewing the result in the graphical interface.



- *Graphical Interface migration*
  - Two main issues:
    - Porting to GRID
    - Interaction with a 3D visualization system
  - Possibilities:
    - Java --> JavaOSG
    - Java --> JOGL
    - Java --> Java3D
    - C++ wrapper to the GRID for the current application

### ■ Pros

- The current graphical interface already tested and running
- It uses Mesa for the 3D representation which has been demonstrated is good enough
- Support for 3D visualization system

### ■ Cons

- The lack of modularity
- Ignorance about the performance of Java 3D
- Integration with Migrating Desktop

- *Adaptation to int.eu.grid particularities*
  - Characteristics of the testbed
  - Services and Grid Middleware implementing interactivity
  - Interfaces and APIs to submit an interactive job
  - Performance of computing and bandwidth enough for interactivity?
  - Where is the graphical interface program run ?
    - In an intermediate specific node
    - At client side

- *Migrating Desktop*
  - Integration with the application
  - Plugin development
  - Is a must to have a Java application to integrate with it?

- *THANK YOU !!! Suggestions are welcomed*

