Contribution ID: **64** Type: **Oral contribution**

Massive Ray Tracing in Fusion Plasmas on EGEE

Wednesday, 1 March 2006 15:30 (30 minutes)

Plasma heating in magnetic confinement fusion devices can be performed by launching a microwave beam with frequency in the range of the cyclotron frequency of either ions or electrons, or close to one of their harmonics. The Electron Cyclotron Resonance Heating (ECRH) is characterized by the small size of the wavelength that allows one to study the wave properties using the geometrical optics approximations. This means that the microwave beam can be simulated by a large amount of rays. If there is no critical plasma layer (like cut off or resonance) close to the beam waist, it is possible to use the far field approximation and the beam can be simulated by a bunch of one or two hundred rays, which can be performed in a cluster. However, if the beam waist is closed to the critical layer and the heating method uses Electron Bernstein Waves (EBW), the number of rays needed is much larger. Being all the ray computations independent, this problem is well suited to be solved in the grid relying on the EGEE infrastructure [1].

We have developed a MRT (Massive Ray Tracing) framework using the lcg2.1.69 User Interface C++ API. It sends over the grid the single ray tracing application (called Truba [2]) which performs the tracing of a single ray. This framework works in the following way: First of all, a launcher script generates the JDL files needed. Then, the MRT framework launches all the single ray tracing jobs simultaneously, periodically querying each job's state. And finally, it retrieves the job's output.

We performed several experiments in the SWETEST VO with a development version of Truba, whose average execution time on a Pentium 4 3.20 GHz is 9 minutes. Truba's executable file size is 1.8 MB, input file size is 70 KB, and output file size is about 549 KB. In the SWETEST VO, there were resources from the following sites: LIP (16 nodes, Intel Xeon CPU 2.80 GHz), IFIC (117 nodes, AMD Athlon 1.2 Ghz), PIC (69 nodes, Intel Pentium 4 2.80 GHz), USC (100 nodes, Intel Pentium III 1133 MHz), IFAE (11 nodes, Intel Pentium 4 2.80 GHz) and UPV (24 nodes, Pentium III). All Spanish sites are connected by RedIRIS, the Spanish Research and Academic Network. The minimum link bandwidth is 622 Mbps and the maximum, 2.5 Gbps.

The MRT framework traced 50 rays and it took an overall time of 88 minutes. In this case, we analyzed the following parameters: execution time (how much time took Truba to be executed in the remote resource not including queue time), transfer time, overhead (how much overhead is introduced by the Grid and the framework itself due to all the inner nodes and stages the job passes through) and productivity (number of jobs per time unit). The average execution time was 10.09 minutes and its standard deviation was 2.97 minutes (this is due to the resource heterogeneity). The average transfer time was 0.5 minutes and its standard deviation was 0.12 minutes (this is due to dynamic network bandwidth). The average overhead was 29.38 minutes. Finally, the productivity was 34.09 rays/hour.

Nevertheless, we found the lack of opportunistic migration (some jobs remained "Scheduled"for too long) and fault tolerance mechanisms (specially during submission using Job Collections, retrieving output and some "Ready"status that were really "Failed"and took too long to be rescheduled) as limitations of the LCG-2

infrastructure (some of the nodes marked by the GOC as "OK"were not). Even, problems handling Job Collections and submitting more than 80 jobs were found.

In order to bypass these problems, we used GridWay, a light-weight framework. It works on top of Globus services, performing job execution management and resource brokering, allowing unattended, reliable, and efficient execution of jobs, array jobs, or complex jobs on heterogeneous, dynamic and loosely-coupled Grids. GridWay performs all the job scheduling and submission steps transparently to the end user and adapts job execution to changing Grid conditions by providing fault recovery mechanisms, dynamic scheduling, migration on-request and opportunistic migration [3]. This scheduling is performed using the data gathered from the Information System (GLUE schema) that is part of the LCG-2 infrastructure.

GridWay performs the job execution in three simple steps: Prolog, which prepares the remote system by creating an experiment directory and transferring the needed files. Wrapper, which executes the actual job and obtains its exit status code. And Epilog, which finalizes the remote system by transferring the output back and cleaning up the experiment directory.

After performing different experiments in similar conditions, we obtained the following results. The overall time was 65.33 minutes. The average execution time was 10.06 minutes and its standard deviation was 4.32 minutes (this was almost the same with the pilot application). The average transfer time was 0.92 minutes and its standard deviation was 0.68 minutes (this was higher because of the submission of the Prolog and Epilog scripts). The average overhead was 22.32 minutes (this was lower as less elements were taking part in the scheduling process). And finally, the productivity was 45.92 rays/hour.

The reason for this higher productivity is that GridWay reduces the number of nodes and stages the job passes through. Also, this productivity is the result of GridWay's opportunistic migration and fault tolerance mechanisms.

As a key improvement needed to better exploit this technique on EGEE we can find that the data contained in the Information System should be updated more frequently and should represent the real situation of the remote resource when trying to submit a job to it. This is a commitment between the resource administrator and the rest of the EGEE community.

The last aspect we would like to notice is the difference between the LCG-2 API and DRMAA. While the LCG-2 API relays on a specific middleware, DRMAA (which is a GGF standard) doesn't. The scope of this user API specification is all the high level functionality which is necessary for an application to consign a job to a DRM system, including common operations on jobs like synchronization, termination or suspension. In case this abstract is accepted, we would like to perform an on line demonstration.

REFERENCES:

[1] Massive Ray Tracing in Fusion Plasmas: Memorandum of Understanding. Francisco Castejón. CIEMAT. Spain.

[2] Electron Bernstein Wave Heating Calculations for TJ-II Plasmas. Francisco Castejón, Maxim A. Tereshchenko, et al. American Nuclear Society. Volume 46, Number 2, Pages 327-334, September 2004.

[3] A Framework for Adaptive Execution on Grids. E. Huedo, R. S. Montero and I. M. Llorente. Software - Practice & Experience 34 (7): 631-651, June 2004.

Primary authors: Dr HUEDO, Eduardo (Centro de Astrobiología (Spain)); Dr LLORENTE, Ignacio M. (Universidad Complutense de Madrid (Spain)); Mr VAZQUEZ-POLETTI, Jose Luis (Universidad Complutense de Madrid (Spain)); Dr MONTERO, Ruben S. (Universidad Complutense de Madrid (Spain))

Presenter: Mr VAZQUEZ-POLETTI, Jose Luis (Universidad Complutense de Madrid (Spain))

Session Classification: 1b: Astrophysics/Astroparticle physics - Fusion - High-Energy physics

Track Classification: Astroparticle physics - Fusion - High-Energy physics