Contribution ID: **50** Type: **poster**

prolog-mpi: a System for Explicit Predicate Distribution in Prolog Systems

Wednesday 9 May 2007 17:30 (20 minutes)

Describe the scientific/technical community and the scientific/technical activity using (planning to use) the EGEE infrastructure. A high-level description is needed (neither a detailed specialist report nor a list of references).

prolog-mpi provides an abstract distributed-computing interface to operators of the Prolog computer language. Prolog is a popular language in a variety of fields, especially in the mathematical fields of provability, artificial intelligence, and optimisation. The prolog-mpi primary executable, pl-mpi, is an extension of the familiar SWI-Prolog interpreter that operates upon custom predicates in order to distribute instructions and data to a network of compute nodes.

Report on the experience (or the proposed activity). It would be very important to mention key services which are essential for the success of your activity on the EGEE infrastructure.

The University of Latvia began developing prolog-mpi as a means to provide preexisting Prolog scripts with the ability of explicit distribution. To date, prolog-mpi has been successfully integrated into a significant NLP code-base: parallel regions are explicitly labelled and, when executed by pl-mpi, these regions are properly distributed and recombined. The prolog-mpi system is also bundled with a utility for automated testing, producing a range of run-time profiles for the purposes of correctness testing and benchmarks. In order to operate prolog-mpi, computing environments must provideMPI. Further, each system must have a functional SWI-Prolog dynamic library available for linking. With these requirements met, prolog-mpi may run both interactively or non-interactively. The system has unbounded scale in terms in terms of input size and number of participating nodes.

With a forward look to future evolution, discuss the issues you have encountered (or that you expect) in using the EGEE infrastructure. Wherever possible, point out the experience limitations (both in terms of existing services or

missing functionality)

When approaches were considered for parallelisation, choice of parallel interface was
a matter of significant concern. MPI was chosen because of its support on local grids
available to the IMCS, not due to an informed endorsement by an authoritative text.
Unfortunately, current deployments of MPI on the BalticGrid lack the ability to span
multiple grid sites, which artificially constrains processor availability.

Describe the added value of the Grid for the scientific/technical activity you (plan to) do on the Grid. This should include the scale of the activity and of the potential user community and the relevance for other scientific or business applications

The complexity of most academic research systems is considerable. Execution on
conventional computer hardware is often unreasonable; thus, in order to execute such
systems within a practical amount of time, grid architectures offer an environment of
massive, scalable sequence parallelisation. Unfortunately, multicomputer environments
are not available to all computer languages. The University of Latvia IMCS developed
prolog-mpi in order to allow preexisting prolog systems to take advantage of explicit
parallelism. The system provides a number of custom predicates that transparently
handle both blocking and non-blocking distribution strategies. prolog-mpi's
precursor, dnlp, was featured at the BalticGrid 2AHM, October 2006. It demonstrated
considerable polynomial speed-up of Prolog systems previously constrained to
uniprocessor environments or overly-fine-grained implicit distribution. prolog-mpi
surpasses this system with a general interface for explicit parallelisation.

Author: Mr DZONSONS, Kristaps (Researcher)

Co-author: Mr BARZDINS, Guntis (Professor)

Presenter: Mr DZONSONS, Kristaps (Researcher)

Session Classification: Poster and Demo Session