

IV Grid Plugtests: composing dedicated tools to run an application efficiently on Grid'5000

Tuesday, February 12, 2008 11:50 AM (20 minutes)

Exploiting efficiently the resources of whole Grid'5000 with the same application requires to solve several issues:

- 1) resources reservation;
- 2) application's processes deployment;
- 3) application's tasks scheduling.

For the IV Grid Plugtests, we used a dedicated tool for each issue to solve.

The N-Queens contest rules imposed ProActive for the resources reservations (issue 1).

Issue 2 was solved using TakTuk which allows to deploy a large set of remote nodes. Deployed nodes take part in the deployment using an adaptive algorithm that makes it very efficient.

For the 3rd issue, we wrote our application with Athapascan API whose model is based on the concepts of tasks and shared data. The application is described as a data-flow graph using the Shared and Fork keywords. This high level abstraction of hardware gives us an efficient execution with the Kaapi runtime engine using a work-stealing scheduling algorithm to balance the workload between all the distributed processes.

3. Impact

To run our N-Queens application on the grid, we composed three tools : ProActive, TakTuk and Kaapi. The grid's architecture was provided by Plugtests organizers through a deployment descriptor file which contains required information to reserve and contact nodes (gateways, resources managers). ProActive was in charge of reserving all the nodes and creating a tunnel to each cluster of the grid. Then Taktuk just used these tunnels to connect all the nodes of all the clusters and started the Kaapi processes.

Our N-Queens application ran successfully during this Plugtests. We deployed our Kaapi processes on 1364 nodes of Grid5000 (one process by node) in less than 3 minutes. The computation used 3654 cores (each Kaapi process creates one computation thread by core). Using this deployment during the one-hour slot, we computed all the solutions of one 23-Queens (35min 7s) and of six 22-Queens (about 2min 21s each). These results gave us the first place of the contest.

URL for further information:

http://www-id.imag.fr/Laboratoire/Membres/Besseron_Xavier/IV_Grid_Plugtests/

4. Conclusions / Future plans

We learnt two main lessons from these experiences:

- Kaapi middleware allows us to scale up to thousands of heterogeneous cores while the efficiency is preserved. On going work is to increase the scalability on highly heterogeneous networks.
- Fault tolerance is essential to run application at such a scale. Many times during the contest, our application crashed because some nodes in the grid failed. Two fault tolerance protocols are currently in development for Kaapi.

Provide a set of generic keywords that define your contribution (e.g. Data Management, Workflows, High Energy Physics)

Grid, Deployment, Work-stealing scheduling, Tools for the grids

1. Short overview

This year, the IV Grid Plugtests took place in Beijing, China from October the 29th to November the 1st, 2007. Organized by ETSI and INRIA, it proposed a contest on the N-Queens problem in order to test grid technologies.

We offer a feed-back about our experience of running efficiently our N-Queens application on a whole computing grid like Grid'5000, composing tools from reservation and deployment to tasks scheduling.

Primary authors: WAGNER, Frédéric (LIG - MOAIS project); HUARD, Guillaume (LIG - MOAIS project); GUELTON, Serge (LIG - MOAIS project); GAUTIER, Thierry (LIG - MOAIS project); DANJEAN, Vincent (LIG - MOAIS project); BESSERON, Xavier (LIG - MOAIS project)

Presenters: WAGNER, Frédéric (LIG - MOAIS project); HUARD, Guillaume (LIG - MOAIS project); GUELTON, Serge (LIG - MOAIS project); GAUTIER, Thierry (LIG - MOAIS project); DANJEAN, Vincent (LIG - MOAIS project); BESSERON, Xavier (LIG - MOAIS project)

Session Classification: From research to production grids: interaction with the Grid'5000 initiative

Track Classification: Application Porting and Deployment