Grid Computing Programming Assignment

Grid Computing Module
2nd February 2007

Assignment

This assignment will explore various performance aspects of applying Grid technologies. The basis of the investigation is to:

1. Write a Java based application that will perform two main operations:

- search a text file containing space separated terms and retrieve the number of times a certain term occurs, i.e. you should develop a client that supplies a search term such as "forsooth" and a service that returns how many times this term occurs in the provided files;
- sort the text file so that the terms and number of times they occur is given in a decreasing rank ordering, i.e. most frequently occurring term is given first with the number of times it appears.

You may implement whatever search/sorting algorithm you wish. The text files to be used are of different sizes and provided in directory */home/gc5/assignment/inputFiles/.*

2. Perform benchmarking on the speed of the application on a single PC for searching and for sorting the provided text files. Benchmarking should include 10 timed runs of both the search and the sort algorithms on a single PC with each of the various files in directory *inputFiles.* You should end up with a set of performance benchmarks for the different text files provided with fastest run, slowest run and average timing over all 10 runs for both the search and the sort algorithm. These benchmarks will be based on the time it takes from a client invoking the search/sort method to be returned the number of times that term occurs/receiving a sorted file from the service respectively.

3. Extend and parallelise the application to make use of the training lab Condor pool. The Condor version should allow a client to select the number of nodes the application should run across.

4. Perform benchmarking on the speed of the application across the Condor pool. As in step 2 you should record ten timed runs. What are the performance overheads in using Condor? How is the performance affected by the number of nodes used and the distribution of the different data sets across these nodes? What file size is needed to gain benefit from distributing the application across the nodes versus running the application on a single PC?

5. Wrap the *parallelised* application from step 3 as a GT4 Grid service on your allocated machine and develop a client to test it with each of the *inputFiles*.

6. Perform benchmarking on the speed of the GT4 service with the different *inputFiles* as in steps 2 and 4. What are the performance overheads in using GT4?

7. Extend the GT4 service to make use of the National Grid Service (NGS) Oxford node, i.e. the application should now submit jobs to NGS node at Oxford and not to the Condor pool. Information on how to submit jobs to the NGS and the URL for the headnodes is available at http://www.grid-support.ac.uk/content/view/118/69/. You should test job submission to the NGS with Globus toolkit version 2 (GT2). Information on GT2 for job submission and data retrieval is available in the /home/gc5/assignment/GT2-howto.pdf. You may also test out job submission to the NGS node at RAL. We note that the certificates you have been issued will only be recognized at the Oxford and RAL NGS nodes. The GT4 service should wrap these GT2 command line invocations, e.g. through Java runtime objects.

8. Perform benchmarking on the speed of the application when using the NGS resource as in steps 2, 4 and 6. How does the overall performance change when using the NGS Oxford node compared to the Condor pool with the different *inputFiles*?

9. Extend the GT4 service to use both the Condor pool and the NGS node at Oxford. Perform benchmarking on this extended service. Explain the issues in performing benchmarking in a heterogeneous environment. Outline how you have attempted to exploit the benchmarking results from steps 2, 4, 6 and steps 8 above in the final service developed in step 9. For example, based on the size of input file and the number of jobs you wish to submit you may prefer to select a given resource to submit the jobs to or decide to run the job on your allocated lab PC.

Marking

Your assignment must be submitted by 5:00PM on Friday, 9th March 2007 in the locked box outside the teaching office. It must be submitted in an unsealed A4 envelope with your name, name of the course, and assessment number clearly written on the front. You must include your pink declaration of authorship form in the envelope. Please note that failure to provide an envelope may result in other students seeing your mark.

The marks will be assigned:

- 10% for steps 1 and 2;
- 15% for steps 3 and 4;
- 15% for steps 5 and 6;
- 15% for steps 7 and 8;
- 15% for step 9.

A further 30% is available for the way in which performance aspects are addressed and discussed in the write up (including the issues in performance benchmarking on a shared set of resources with different file sizes). You should submit the source code for the Grid Services, any design notes you have made, and a written report discussing the performance of the system across the range of scenarios. You should provide graphs/plots for the various performance-benchmarking you have made in steps 2,4,6,8 and 9 and explain these results. You will be expected to demonstrate your system at the final Grid Computing lab session on 16th March.