CAA3 - second part

Chapter 8

1. Consider a program that uses a database with 4 tables where each table has 4 columns. For each table the first two columns are alphanumeric and the remaining two are numeric. Create a database in PostgreSQL or MySQL (only one of two) to work with the application, and enter at least one dataset with no less than 10 entries per table/column (since this will be at least 4x4x10 entries we suggest using a method to load data from external files instead of manually entering the values). Working with the command line and with the database client, do the following:

   a. Generate a separate listing of each table, sorted by the values of column 3 in increasing order.
   b. Generate an integrated listing with the data of the 4 tables, sorted by the values of column 4 in decreasing order. Only the data from columns 1 and 4 should be included.
   c. In each table, replace the numerical value of column 4 by the numerical value of column 3 plus the current value of column 4 divided by 2. Visualize the new data of this new column.

Chapter 10

2. System performance.
   a. Analyse the performance of the system with the tools you consider appropriate and diagnose the usage of resources (CPU, disk, memory, swap and I/O). Determine the possible bottlenecks that could exist in the system. Simulate different loads on the system with programs that use both the CPU, output to the screen and disk read and/or write.
   Note: you can use the code *sumdis.c* given in the clusters chapter with some modifications.
   For example, you can use " sumdis 1 2000000" .
   b. Install the Munin package to monitor the system load and Monit to view the processes / services currently running. Monitor at least three services: apache, sshd, etc. In this exercise you must submit a description of the steps followed to install and configure the monitoring, including the configuration files generated and the results obtained.

Chapter 10

3. Install some package that implements MPI (Mpich or LAM/MPI) Compile and run the programs *cpi.c* and *monte.c* to calculate the value of Pi by launching at least 6 processes for each program (if possible in two or more processors, though not necessarily in different machines). Do a detailed analysis of the performance and draw conclusions on the advantages and disadvantages of distributed computing.

   The source code of these examples can be obtained from the installation itself or from the homepage of MPI http://www.mcs.anl.gov/research/projects/mpi/usingmpi/examples/main.htm

   The programs can be found in the Download section. The XMPi package can be used to visualize the execution and facilitate the collection of statistics.