Practical 2

This is the second and final assessed coursework for the Modelling and Simulation course. Both courseworks are equally weighted and account for 25% of the final mark. This is an individual coursework exercise designed to assess your skill and ability. This means that your submission must be entirely your own work unless you declare it to be otherwise. Please see the School of Informatics guidelines on plagiarism for further clarification. These are available on-line at `http://www.inf.ed.ac.uk/admin/ITO/DivisionalGuidelinesPlagiarism.html`.

Instructions for postgraduate and undergraduate students

The conditions of assessment for postgraduate students submitting this coursework exercise for the Level 11 version of the course are different from the instructions for undergraduate students submitting this coursework exercise for the Level 10 version of the course.

Postgraduate students on the Level 11 version of the course should complete **all four parts** of this exercise (Parts A, B, C and D below).

Undergraduate students on the Level 10 version of the course should complete **any three of the four** parts of this exercise (any three of Parts A, B, C and D below).

Description

This practical will have you working with the Eclipse PEPA Plug-in Project[1], and the SSJ library. Using these tools you will create models of a small Condor pool. **Unless told so explicitly**, you can assume the exponential rate for all delays.

Part A

In this part of the practical you are asked to model a small Condor pool. Research groups are allowed to submit one **job** at a time, typically submitting to the Condor pool once every ten hours *after* the previous job has completed. A job can be of non-specific (general) type or specialised towards particle physics. A job also varies in length. For the purposes of this part of the practical a job can be **1-3 blocks** in length. This leads to a total of 6 possible combinations being submitted to the pool.

The Condor pool will consist of 4 machines each of which can be either a general purpose unit, or specialised towards particle physics jobs. The general purpose machines will, on average, execute one block of computation in 10 hours regardless of the type of job. The specialised machines differ, being able to process individual blocks of particle physics jobs five times faster than the

[1] `http://homepages.inf.ed.ac.uk/mtribast/plugin/index.html`

general machines, but at the expense of other jobs. Blocks of general (non-specific) jobs take 14 hours. At this stage no scheduling is happening, compute jobs are randomly assigned to a machine.

a) Construct a PEPA representation of the Condor pool as described above. There should be descriptions of two types of machines, capable of handling two different types of compute jobs at the rates defined. The model should also allow for different sized jobs. Note the last part of this question asks for analysis of completion rate.

b) Set the pool to contain two general purpose and two particle physics machines, and an even ratio between the two types of jobs being submitted. How many research groups can the Condor pool handle before reaching 95% saturation of processing time? Explain your reasoning.

c) Using the Eclipse PEPA Plug-in, generate graphs showing job completion rates for varying ratios of normal jobs to particle physics jobs. Vary the ratio from 10% to 90%. Supply one graph for each combination of machines in the Condor pool (4 general machines, 1 general machine and 3 particle machines, 2 of each etc). Think experimentation.

10 marks

Part B

Using the model description from Part A, the following changes are made. Instead of the coarse definition of job length, we will use a Poisson distribution with $\lambda = 5$, where sampling from the distribution will tell us how many million computations are required (so on average 5 million). Simply throw away any samples of value 0. The general purpose machines can crunch one million computations of either job type in, on average, 5 hours. The particle physics machines are still five times faster for particle jobs and run general compute jobs at 9 hours per million computations. All times should be treated as the mean for an exponential distribution. Also, a new job is submitted to the Condor pool at a fixed rate of μ regardless of completion of previous jobs (so no research groups). There will be no buffer for queuing jobs, so if all the processors are in use a new job is simply removed from the queue.

a) Construct a discrete-event simulation using the SSJ package. Each job submitted will have an equal chance of being either a general job or a particle physics job. The Condor pool should consist of two of each type of machine. The duration of the simulation will be one month (31 days or 744 hours).

b) Experiment with the simulation to discover the rate μ that achieves 95% processor usage.

c) Have your simulation calculate the average processing time for a job, along with the standard deviation.

10 marks

Part C

In this part you will model a different aspect of the small Condor pool. Here the jobs will be of fixed size and the pool will consist of two sites. Site one will consist of two normal machines and site two will contain two particle physics machines. Your task will consist of modelling two different schedulers of your choosing and evaluating their performance. You will be supplied with a basic definition for both the processors and a job which you not obliged to use. The partial PEPA file can be found on the Modelling and Simulation homepage. Refer to the file for rate definitions.

a) Model two different scheduling algorithms. For those inexperienced with schedulers examples to choose from are round-robin (switch between each site in turn) and prioritise (send particle jobs to the particle machines when possible). You can even invent your own algorithms if you wish. If you do this ensure to comment the definitions as to their intended behaviour, otherwise oddities will be considered mistake.

b) Using the modes of analysis available in the PEPA Plugin Project, come to a conclusion as to which algorithm is better. When justifying your decision, make sure to reference the data so your answer can be verified by the marker. If you perform experiments which you use to defend your point, include the graphs and as much detail as required to reproduce your analysis. For example, how do the two algorithms compare if you vary the probability of the job type.

10 marks

Part D

For part D you will create a SSJ simulation based on Part C (construct two scheduling algorithms). As simulation allows for larger models than is feasible with CTMC analysis, the model will be larger than that described above. The jobs will vary in length as in Part B, and each site will contain 10 homogeneous machines, site A with 10 general-purpose machines and site B with 10 particle physics machines. Use the rates defined in Part B. Feel free to expand the capability of the scheduler if you wish as well.

a) Construct a discrete-event simulation using the SSJ package. Simulation duration will be one month as in Part 2. Job submission will occur at rate μ and as before there is no queuing of submitted jobs (if a machine isn't available the job is dropped). The actual rate μ used is at your discretion.

b) Using a rate for μ that ensures a steady use of the Condor pool, compare your two scheduling algorithms by measuring average processing time per job and by the rate of dropped jobs and report your results. If you find that under different conditions the better scheduler changes then report when this can happen.

10 marks

Deadline information

This practical exercise has been issued on Friday 31st October 2008 and the deadline for submission is **Friday 28th November 2008 at 11:00am**.

Submission instructions

Your submission for this practical exercise is to be electronic only, with no additional paper documents. Please prepare a directory somewhere in your DiCE filespace entitled `Prac2` and put inside this sub-directories named `PartA`, `PartB`, and so forth for the parts of the practical exercise which you have completed. All files should be formatted to 80 characters for ease of printing (if required). In Eclipse this can be achieved with the Format command in the Source drop-down menu.

The central deliverable of your submission should be a report in the Prac2 directory, detailing all of the answers obtained from your experiments, with any necessary supporting information. Please format this using LaTeX or other document preparation software.

In these directories you may wish to include README files to explain the contents, and any necessary details about processing and using the files contained in the directory.

Submit your work for assessment using one of the two following commands (as appropriate).

```
submit cs4 ms-4 cw2 Prac2
submit msc ms-5 cw2 Prac2
```

You may issue this command multiple times. Later submissions will overwrite earlier ones. The date of submission for assessment purposes will be taken as the date of the last submission.

Frequently-asked questions

Q How will the coursework be marked?

A Your PEPA models and Java SSJ programs will be read and, in case of any questions concerning your results we will re-run your PEPA model in the PEPA Eclipse plugin and execute your Java programs.

Q What is the assessment criteria?

A In the assessment of this practical exercise the assessors are looking for PEPA models which are clear and concise. The model should be well-structured in order that it is evident what is being modelled and what assumptions are being made. The components of the PEPA model should have clear and well-defined roles relevant to the modelling problem under study.

The same criteria also apply to the Java code submitted for assessment. This too should be clear and concise, with transparent use of the SSJ library in order that the reader can be convinced that the simulation is really measuring the performance metrics which it is intended to measure.

Good use of comments in both the PEPA and Java files will help the marker understand what you were trying to achieve.

In any accompanying notes and descriptions the assessors are seeking thoughtful justifications of the conclusions draw from the results whether they are computed with the PEPA Eclipse Plug-in or with the SSJ simulation library.

Q I'm an postgraduate student, can I take Modelling and Simulation as a Level 10 course and only do three parts of the practical?

A No, that is not possible.

Q I'm an undergraduate student, can I do all four parts of the practical?

A Yes, you can.

Q Is there any advantage in me doing all four parts?

A Yes. All four parts of your submission will be marked and the best three of these marks will be used to compute your coursework mark. This is likely to increase your final mark overall. Also, you many gain a deeper understanding of stochastic simulation by doing all four parts.

Q Is there any disadvantage in me doing all four parts?

A It will take longer. However, you may understand more, and may get a better mark.

Q When I compile my SSJ programs I get a lot of confusing errors of the form "`cannot find symbol; symbol :  variable Sim`".

A This is a CLASSPATH error. You need to tell the Java compiler where to find the SSJ files using commands of the following form:

`export SSJHOME=~/ssj` (or wherever you put SSJ)

`export LD_LIBRARY_PATH=$SSJHOME/lib:$LD_LIBRARY_PATH`

`export CLASSPATH=.:$SSJHOME/lib/ssj.jar:$SSJHOME/lib/colt.jar:`
`$SSJHOME/lib/optimization.jar:$SSJHOME/lib/Blas.jar:$CLASSPATH`