UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
Departamento de Engenharia Informática

Trabalho Prático N.º 1
Computação de Alto Desempenho

2012/13 – 2º Semestre

MEI

Prazo: 19-4-2013

Nota: Este trabalho é igual ao do ano anterior. Serão comparadas as soluções deste ano com as do ano anterior e serão tomadas medidas contra qualquer tipo de fraude.
Nota 2: Os alunos deverão apresentar regularmente o seu trabalho durante as aulas, de forma a receberem a orientação necessária.

Shared Memory Programming

Project Goals

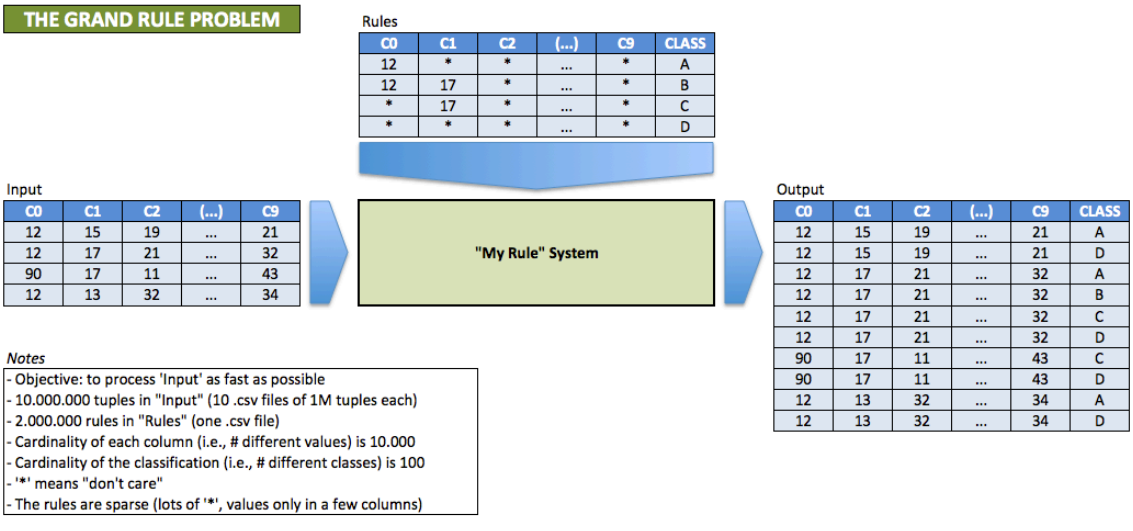
The main goal of this project is to give students a deeper insight on shared memory parallel programming, including performance evaluation concerns.

Project Description

Note: this problem was proposed and written by Professor Paulo Marques.

An industrial client (a bank) has recently approached a company in our town to classify the expenses of its clients according to a number of rules. In practice, for each bank transaction of an input file, they want to see if it matches a number of predefined rules. A bank transaction is a tuple with 10 attributes – $(a_0, a_1, a_2, a_3, \ldots, a_9)$, where each attribute is an integer (32 bit). Each rule is also a tuple with 10 attributes plus a classification "c". I.e., $(r_0, r_1, r_2, r_3, \ldots, r_9, c)$. Each rule attribute can either be a number (32 bit), or a "*", which means "don't care". For example, the transaction (1,2,3,4,5,6,7,8,9,0) matches the rule (1,2,3,4,5,6,7,8,9,0,111) and the rule (1,*,*,*,*,*,*,*,*,*,0,222), but it does not match either the rules (2,2,3,4,5,6,7,8,9,0,333) or (2,*,*,*,*,*,*,*,*,*,0,444).

Your challenge is to implement a system that allows efficiently solving this problem for large amounts of data. The next image illustrates the problem.



In particular, the objective is to process 10 input files containing the transactions of the last 10 days. Each input file has 1M tuples. The rule table is fixed and has 2M rules. Most of the rules are actually empty (i.e., with "*"), having numbers in only a few columns of each rule. The number of unique values in each column is 10.000 and the number of different classifications is 100. Your solution should process the data as efficiently as possible, generating an output file for each input one. By "efficiently" we mean:

- Having a reasonably good algorithm for performing the computation.
- Utilizing the resources of a machine to their fullest.
- If you use a distributed approach, even so, use each machine to its full potential.

The Work

In their implementation, students should use one or more of the following technologies (they should discuss any other alternatives with the professor):

- Java
- Pthreads
- OpenMP
- Cilk
- SSE

NOTE: Although Java may be a very good language to solve this problem, implementations in Java will not serve for the second assignment, because students will have to solve the same problem in the Message Passing Interface.

Students will be given access to a test data set and the complete dataset for the project. One piece of advice: try to implement the better algorithm that you can before making it concurrent. You must use only one shared memory machine. As an indication, you should be able to process, at least, 6.000 transactions/sec on a Core2-Duo 2.5GHz with 4Gb of RAM. Doing twice that is readily attainable.

The data set is huge in size. You can download it from http://eden.dei.uc.pt/~filipius/CAD/dataset.tgz. In there you will find three directories:

- debug_extra_small: a very small rule table and input transaction file
- debug_small: the 2M rule table, and a very small input transaction file
- THE_PROBLEM: the 2M rule set and the 10x 1M transactions input files

The data set includes the expected output. To compare with the output of their own program, students should sort their own output and should remove duplicate entries. Students can readily execute both operations using standard Unix commands, such as sort and uniq. The –u option of sort can discard the need for uniq.

Also, in the rule input files:

- For the columns 0..9, "0" means "don't care". Values go from 1 up to 10.000.
- The the column 10 (i.e., the classification) values go from 0 up to 99.

Students should solve a problem and should iterate over their solution to improve performance as much as they can.

Evaluation Criteria

The evaluation of the work will be based on the following criteria:

- Correction of the code. Students should ensure that their algorithm is working properly.
- Performance. Students should minimize execution time of their algorithm. This will be one of the most valued aspects of the work. Students should keep in mind that they should balance performance of the code against clarity and readability of their solutions.
- Careful evaluation of the solutions proposed. In addition to simple performance, students must carefully evaluate and describe the improvements they did to their algorithm (i.e., they should tell a sort of story of their algorithm).
- Analysis of the scalability of the solution: how does the algorithm behave when the number of threads grows.

Report of the Work

Besides the software that you will develop, you should also write a report describing the work. It should contain the following sections:

(A) Introduction
(B) Description of the best solution
(C) Analysis of performance (which optimizations where made, which measurements were taken). Your analysis should include graphical plots.
(D) Analysis of performance for different numbers of threads (include the architecture where the you ran the tests). This analysis should include plots to summarize the results.
(E) Conclusions.

The report should be made as small as possible, yet with enough length to be clear and readable.

Assignment upload

Students should deliver everything, including the report, in a .zip file. This file should include a README file with all the INFORMATION NECESSARY to execute and test the assignment without the presence of the students. Assignments that do not contain this README with all the necessary instructions will not be evaluated. Assignments that do not execute correctly will also not be evaluated.

You should upload the .zip file in the Inforestudante.

However, it is mandatory to deliver the report in the locker of the professor of the theoretical classes up to 1 working day after the deadline stated above.

Good Work!