Thursday 24rd April 2008

Operating Systems Practical Assignment,
Spring Semester 2008,
Teacher: Nies Huijsmans

OS Website: http://www.liacs.nl/~shenstra/os

Lab Assignment #7
Original author: Fabrice Colas, student assistant
LIACS, room 133, fcolas@liacs.nl, 071 527 7033

***Outline of the assignment***

In this assignment, I use a computing problem from genetics to discuss OS-related issues when using parallel computing. So, first, I quickly introduce the problematic in genetics, I state the research goal and I present the technique we plan to use to answer the problematic, i.e. itemset mining. In that regard, I ask you to answer two questions. The first one concerns the space necessary to store the result of our computation, whereas the other one concerns the compute time. In order to do our computation, we decide to 'parallelize' our computing problem. So, I give some additional background to help you understand the sketch and then I ask you in the last question to set-up a 'user-space' scientific computing environment which can communicate with a central database server.

**Reports to be sent by email for the 14th of May at the latest,**
(1) Use the template file to report your answers (http://www.liacs.nl/~shenstra/os/documents/assignment7_report.doc)
(2) And into a single ZIP/TAR.GZ file, send the 'user-space' install described of question (3). As the file may be too big to email, you can either pass by my office with a usb key, or upload your archive on a FTP/HTTP server and you just tell me the location.

**Grading of the lab assignment**
(1) Refer to webpages, books or colleagues that you got help from
(2) Go as far as you can with question (3), the minimum requirement is to install in the user space the R environment. But try to go further by installing mysql, rdbi and rmysql.

***Introduction***

Last decade showed a tremendous improvement of genotyping techniques leading to the sequencing of the whole human genome [1]. Availability of the human genome permitted to further study diseases having an heredity component. With more precise knowledge of the DNA structure and technical advance, genome-wide analysis are now permitted. However, this is at the expense of some conceptualization of the DNA structure. For instance, some locus of the genome were discovered to have a DNA that differs by a single nucleotid; these locus are referred to as Single Nucleotid Polymorphisms and their location on the genome is recorded in public databases [2]. A SNP record defines the most commonly shared allele variation and the less frequent polymorphism; the population occurence probabilities are also given for having the most common polymorphism on the (x2) alleles, on (x1) or none (x0).

We consider an association study with patients affected and unaffected by a disease. There are 1000 participants involved, 500 are cases and 500 controls. The genotyping proceeds on 100.000 SNP locations. The “class” indicates whether participants are cases (1) or controls (0). Consequently, the data set is made of 1000 rows and 100.000 columns plus one.

*We want to Identify combinations of SNP markers that highly associate with the disease occurence.*

To answer this question, we select the item set mining family of algorithm. Thus, we search for combinations of markers -or rules- that are highly predictive of the class:

(1 item) Marker(i) => Class,
(2 items) Marker(i) AND Maker(j) => Class,
(3 items) Marker(i) AND Maker(j) AND Maker(k) => Class.

Therefore, when searching exhaustively for combinations, the number of tests increases exponentially with the combinations:
(100.000 x (100.000-1) x (100.000-2) x ...).
In fact, with this SNP study, there are about 10^10 tests for two combinations and 10^15 for three. In practice, we compute the interestingness of a rule from contingency tables -see the following-.

| **-- SNP Combination --** | | **-- Class: Case / Control --** | |
|---|---|---|---|
| **Marker i** | **Maker j** | **0** | **1** |
| 0 | 0 | | |
| 0 | 1 | | |
| 0 | 2 | | |
| 1 | 0 | | |
| 1 | 1 | | |
| 1 | 2 | | |
| 2 | 0 | | |
| 2 | 1 | | |
| 2 | 2 | | |

| | | | | |
|---|---|---|---|---|
| | **Total** | **500** | **500** | **1000** |

From the computer science point of view, there are two major issues: (1) the computing speed and the (2) data storage. Today OS assignment covers both aspects.

***Question (1):*** If we proceed to 10^10 statistical tests, we want to store the result of the tests. Therefore, given the different sketches (a), (b) and (c) described here after, estimate the space to store 10^10 float numbers.

(a) We store the float numbers together with their integer index as characters in a flat CSV file,
(b) We store the numbers together with their index in a MySQL database. Make your choice between FLOAT, DOUBLE types for the numbers and SIGNED/UNSIGNED TINYINT indeces.
(c) We store the numbers in a binary format of your own together with the indeces.

Try to minimize the storage costs while preserving the full number accuracy. In addition, remark that each hypothesis may depend on the operating system!

***Question (2):*** *Given the following sketches, make some rough estimation of the time necessary to compute 10^10 association tests:*

(a) each test take 1s
(b) each test take 0.1s
(c) each test take 0.01s
(d) each test take 0.001s

Give your results in days...

***Preliminary to question (3):*** In questions (1) and (2), we illustrated the very large storage space and computing time, so that we can not easily compute the 10^10 tests on a single computer. We decide to carry out our analysis using parallel computing facilities and in fact, we further characterize our computation problem as *massively parallel* because each computation may run independently of the others: a single task involves only two markers and the class information. As the whole data is relatively small, we decide to copy it on all computing node. Thus, we avoid a bottleneck due to the network when the nodes request their copy of the data from a central server. Finally, concerning the set of statistical tests to compute and the storage of the results, we use a database.

To do our computation, we rely on the free scientific computation environment R, which is an equivalent to Matlab or Scilab. We make a 'user-space' installation of R which every single node can run using a NFS-mounted disk. We also install all the additional packages that we require for our computation like RDBI and RMySQL for the communication with the database.

***Question (3) 'howto':***

***(a)*** Create a subdirectory under your home directory like "`/home/mylogin/os_tmp/`" to easily delete the directory while you are trying to get the OS assignment done.

***(b)*** As the Unix nodes may have outdated MySQL software, we do a user-space install of MySQL. So, download mysql [3], set the configure and environment variables with static libraries for a user-space installation and compile. Notice that in the following, when compiling software requiring MySQL development files, you have to tell the specific location of your user-space MySQL install. For instance, check your PATH by doing "`which mysql`".

***(c)*** Download R [4], set the configure and environment variables similarly and compile it.

***(d)*** Do "`which R`" to check the first R executable in your path; eventually update your PATH or execute R directly "`./os_tmp/bin/R`".

***(e)*** Download DBI [5] and RMySQL [6] and try install them both: "`R CMD INSTALL DBI_0.2-4.tar.gz`", "`R CMD INSTALL RMySQL_0.6-0.tar.gz`"

***(f)*** While installing RMySQL, you may run into trouble because you need to inform on the specific location of the just-built MySQL libraries.

***References***


[1] Everyone's genome. Nature 409, p. 813 , 2001. [PDF]
[2] The International HapMap Consortium. A Haplotype Map of the Human Genome. Nature 437, 1299-1320. 2005. [PDF]
[3] http://dev.mysql.com/get/Downloads/MySQL-5.1/mysql-5.1.23-rc.tar.gz/from/ftp://mysql.proserve.nl/pub/mysql/Downloads
[4] http://cran.r-project.org/src/base/R-2/R-2.6.2.tar.gz
[5] http://cran.r-project.org/src/contrib/DBI_0.2-4.tar.gz
[6] http://cran.r-project.org/src/contrib/RMySQL_0.6-0.tar.gz