mpiBLAST benchmarks

SGI/IZO
www.ehu.es/sgi

18 de agosto de 2009

Índice

1. Introduction

In this report we show some benchmarks performed with mpiBLAST in the machines of the SGI-IZO. This report can be used to understand the behavior of mpiBLAST in order to prepare your runs in our machines. mpiBLAST is a parallel version of the well known BLAST NCBI but with the ability of submit one job to several computers at the same time. This allows to reduce computation time linearly when increasing the number of computers, as we will show. On the other hand, BLAST uses data bases of several GB of size which do not feet in normal computers. mpiBLAST breaks this data base in smaller fragments that are submitted to the computers, this reduce the need of read data in the disk and improves the speed.

2. Machines

In the service mpiBLAST has not been installed in the Itanium nodes because the performance of this program in the Itanium architecture is very bad. We will study the performance of mpiBLAST y the Opteron nodes of the Arina cluster and in the Grid Péndulo.

The Opteron nodes has 8 cores at 2.4 GHz and 16 GB of RAM memory (one node has 32 GB of RAM). Péndulo uses regular PC's as computation nodes and we have focused in the nodes with Core2duo processors at 2.1 GHz, 1 GB of RAM and gigabit ethernet connexion.

3. The benchmark

We have study the performance of calculations and its dependence as a function of the size of the input file as well as a function of the number of cores. There are two magnitudes important for us, the time spent by the calculation and the memory used. The last one is very important in Péndulo because we need calculations that need less memory than 500 MB per core, which is the available RAM memory in Péndulo.

We will use three input files with gene sequences. The important parameter will be the size of these files because the bigger the file the longer the calculation. The size of the files is directly related with the number of sequences they have. We chose files of 26, 260 and 2600 KB.

We have used the *nr* data base. It is 4 GB big when uncompressed and must be formated to be useful, after this formating it uses 7.8 GB. This is two big for the RAM memory of normal PC so the computer must store the data base in the disk during the calculation and read it, which slows the calculation. There is not this problem in computing clusters like Arina with large amount of RAM.

When formating the data base with mpiBLAST we chose to split it in several pieces. This smaller pieces are better for computation because they fit in the RAM of the computers. The larger the number of pieces the smaller the size the occupy in the RAM memory.

We have run the same calculation by using 6, 12, 18 and 24 workers. A worker is a process that use one core (single cpu) and one of the fragments of the data base. One calculation uses as many cores as workers (data base fragments) plus

two more, one for read/write process and other for scheluding, i.e., we will use 8, 14, 20 and 26 cores in the calculations.

4. Formating the data base

First we must format the data base. We must configure the `$home/.ncbirc` file with the proper definitions and run

```
/software/bin/mpiformatdb -N 6 -i nr -o T
```

to format the data base for 6 workers. Similarly for 12, 18 and 24 workers.

5. Running the job

Once the data base is formated we have run the calculations with the fo-llowing command

```
/software/bin/mpiblast -use-virtual-frags -use-parallel-write -p
blastx -d nr -m 7 -I T -i input_file.fna -o out_file.xml
```

and we have measured the times. As mentioned previously the 6 fragment data base was submitted to the cluster to 8 cores, the 12 fragment data base was submitted to 14 cores, the 18 fragment data base to 20 cores, etc. If this relation is not fulfilled each worker will read more data base fragments increasing the use of RAM memory and time. The size of the input_file.fna file was 26 , 261 and 2616 KB.

6. Results

6.1. Opteron nodes

First we run the calculations in the opteron nodes. We show in table 1 the amount of memory each worker used. We see that as the number of workers increases the amount of memory decreases linearly. The amount of memory used by each worker is equal to the fragment of the data base it reads plus about 50 MB of the mpiBLAST executable and the input file. For example, our data base directory has 7.8 GB. If we use 6 workers each one will read 7,8/6 = 1,3 GB, plus another $\approx$ 50 MB for the binary and input file we have 1350 MB.

	6 workers	12 workers	18 workers	24 workers
mem/worker (MB)	1350	690	480	370

Table 1: Memory per worker used measured in MB.

Taking this information for this data base we see that in Péndulo we must use more than 18 workers because each PC has 1 GB and two cores, i.e., 500 MB per core. We recommend in fact to use 24 workers or more.

In table 2 we show the time spent by each calculation as a function of the number of cores. In each column we see that as the file size increases ($\times$ 10) the computing time increase in the same ratio ($\times$ 10). In each row we see the similar linear behavior as the numbers of workers increases very close to and ideal behavior up to 18 workers. For 24 works the speedup is 3.3 compared to 6 workers, while ideally should be 4.

File size	6 workers	12 workers	18 workers	24 workers
26	628	354	227	200
261	6270	3390	2480	1935
2616	64105	33160	23610	19592

Table 2: Time spent (seconds) for 3 different calculation with different file sizes (MB) as a function of the number of workers.

Therefore, taking into account the data for the larger input file we obtain that with 18 workers we need 2.6 hours of computing time per MB input file and if we run 24 workers we need 2.1 hours per MB of input file.

6.2. Péndulo nodes

Due to memory limits shown in table 1 we must run mpiBLAST with more than 18 workers, so we did in 24 cores. Nevertheless by using only one of the cores of each PC it can use the whole GB so we have ran the benchmark in 12 cores under these condition. In table 3 we show the time spent by each calculation as a funcion of the number of cores. The resutls are very similar to opteron ones and the performance is similar.

File size	12 workers	24 workers
26	884	1350
261	4036	3474
2616	35757	19613

Table 3: Time spent (seconds) for 3 diferent calculation with different file sizes (MB) as a function of the number of workers.

For small input files we observe larger times that in opteron nodes. This is because the inizialization of the calculation, load the data base in the nodes, takes longer in Pendulo due to the lower performance of its nodes. This inizialization time, increases with the number of workers and also is apreciable in the table 2 for the smaller input file. Nevertheles, real input files are of the order of some MB and this effect is negligible.

Taking into account the data for the larger input file we obatin and 24 workers we need 2.1 hours of computing time per MB input file. Taking into account that at nights Pendulo has 11 hours of computing time the largest file we can compute es 5.2 MB big. At weekend with 55 hours of computing time the maximun file size is 26 MB. Nevertheless, this is not a limit for the posibilities of Pendulo because this files are easily cut in smaller peaces and we can take advantage of the high number of computers in Pendulo.

7. Conclusions

mpiBLAST scales very well in both in Opteron nodes in Arina and Péndulo, this becomes this computers a well suited ones for this kind of analysis. We provide some guidelines to approximately configure your calculations in Péndulo.