

C C A T T 0 1 0 0 0
G A G G A 0 1 1 0 1
G A A T T 0 0 1 1 0
A C A A G 0 0 1 0 0
T A C C A 0 0 1 1 0
T T A C A 0 1 0 0 0
A C C T C 0 0 0 1 0
A A G G A 0 0 0 0 0
G A T G A 0 1 1 0 0
T A G A T 0 0 1 0 0
G A T G A 1 0 1 0 0
T G T A G 1 0 0 0 0
T A G T A 0 0 0 0 0
G A T A T 1 0 0 0 0
G A G T G 1 1 0 0 0
A G A T T 1 1 0 0 0
G A G T A 1 1 0 0 0
T G A T G 1 1 0 0 0
A T T A G 1 1 0 0 0
T A G A T 1 1 0 0 0
G A G A 1 1 0 0 0
G T A 1 1 0 0 0
G A T 1 1 0 0 0
T A G 1 1 0 0 0
A G A 1 1 0 0 0
G A 1 1 0 0 0
A 1 1 0 0 0
T 1 1 0 0 0

White Paper

White paper on *MPI Support for CLC Bioinformatics Cell 2.1*

March 6, 2008



Contents

1 Introduction	3
2 Introduction to the benchmarks	3
3 Benchmark of Smith-Waterman BLAST	4
3.1 Smith-Waterman BLASTn	4
3.2 Smith-Waterman BLASTp	5
4 Benchmark of ClustalW	5
5 HMMER benchmark	6
5.1 hmmpfam	6
5.2 hmmsearch	6
6 Conclusion	7

1 Introduction

This white-paper describes how the performance of the algorithms on the *CLC Bioinformatics Cell* scale when running on a computer cluster. With version 2.1, the *CLC Bioinformatics Cell* has MPI support, which means that its high performance can now be utilized on a larger scale by installing it on a computer cluster.

The *CLC Bioinformatics Cell* is a software package which takes advantage of the computing powers of new CPUs not utilized by standard software. It uses the Single Instruction Multiple Data (SIMD) technique to parallelize, and thereby accelerate Smith-Waterman database searches, ClustalW alignments and HMMER (hmmsearch and hmmpfam for searching for domains in a protein family). Read more about the *CLC Bioinformatics Cell* at <http://www.clccell.com> which includes more information about the algorithms, interfaces and system requirements.

The central issue when running the algorithms on a cluster is how the performance scales. Ideally, using two computers should double the speed compared to one computer. Such a linear scale cannot be realized in real life, since the calculations performed on each of the computer have to be coordinated and there is a little overhead in dispatching the calculation jobs to the computers.

Even though the scalability is not linear, the MPI implementation in the *CLC Bioinformatics Cell* has reduced the overhead significantly, and the performance of the algorithms scale like this on a eight-node cluster (one master and seven slave nodes, each with two cores):

- Smith-Waterman BLASTn: 5.53
- Smith-Waterman BLASTp: 4.63
- ClustalW: 4.08
- hmmpfam: 5.62
- hmmsearch: 4.24

The benchmarks of each algorithm will be described in details in the following sections. For general information about the performance of the *CLC Bioinformatics Cell*, we refer to the white-paper at <http://www.clcbio.com/white-paper>.

2 Introduction to the benchmarks

The scalability of *CLC Bioinformatics Cell* is shown by measuring the performance when adding a node to the cluster. The tables below describe performance measures for each algorithm going from one to eight nodes.

The algorithms are run twice, and the statistics for the run with the best performance are shown here.

The cluster used in these benchmarks is set up like this:

- Eight nodes (seven slaves and one master):
 - Core 2 Duo E6550 (2.33 Ghz)

- 1 GB DDR2 PC8500 RAM
- Hard disk: Seagate Barracuda 7200.10 80GB, 8MB cache, SATA-II
- Connected in a gigabit ethernet network

Each node in the cluster has two cores, so when all the seven slaves are in use, it means that 14 cores are calculating. The software on each computer takes care of parallelizing the calculations on the two cores (similar to the situation where you run the Cell on a single computer with multiple cores).

Even though this is a small cluster, it gives an unambiguous indication of the scalability.

The data sets are similar to the ones used in the *CLC Bioinformatics Cell* white-paper found at <http://www.clcbio.com/white-paper>. Here you can see detailed statistics about e.g. number of hits and the general performance on different data sets with different E-values.

3 Benchmark of Smith-Waterman BLAST

Smith-Waterman BLASTn was tested on a data set consisting of all horse EST sequences in GenBank (36,914 sequences with a total number of residues of 19,762,562). The full data set was used as database, and a subset of 100 sequences were used as query sequences (52,489 residues).

Smith-Waterman BLASTp was tested on a database consisting of 50,000 randomly chosen protein sequences from Swiss-Prot (the total number of residues is 18,396,764). 100 of these sequences were used as query sequences (33,701 residues).

3.1 Smith-Waterman BLASTn

BLASTn with up to three slaves shows almost linear scalability (2.68). After that it drops a little, and with seven slaves it has a speed up of 5.53 (see table 1).

Number of slaves	Time (seconds)	Speed-up
1	147.80	1.00
2	77.07	1.91
3	55.09	2.68
4	44.49	3.32
5	35.44	4.17
6	30.79	4.80
7	26.68	5.53

Table 1: Benchmark of Smith-Waterman BLASTn.

The command used for the benchmark is:

```
blastall_cell_mpi -c cell_sw -e 10 -b 1000 -p blastn  
-i horse_est_100.fasta -d horse_est.fasta
```

3.2 Smith-Waterman BLASTp

Scalability of BLASTp is not quite as good as for BLASTn. Up til three slaves, it has the same speed-up of 2.68, but after that it decreases compared to BLASTn. With seven slaves, it is 4.63 (see table 2).

Number of slaves	Time (seconds)	Speed-up
1	88.45	1.00
2	47.34	1.86
3	32.99	2.68
4	27.77	3.18
5	23.35	3.78
6	20.37	4.34
7	19.07	4.63

Table 2: Benchmark of Smith-Waterman BLASTp.

The command used for the benchmark is:

```
blastall_cell_mpi -c cell_sw -e 10 -b 1000 -p blastp  
-i uniprot_sprot_100.fasta -d uniprot_sprot_50000.fasta
```

4 Benchmark of ClustalW

For benchmarking ClustalW, a set of 62 nucleotide sequences from the HIV-1 subtype reference alignment (with an average 9014.9 nucleotides per sequence) was used. Retrieved from the HIV database at Los Alamos National Laboratory http://www.hiv.lanl.gov/content/hiv-db/SUBTYPE_REF/align.html.

Compared to the database searches (Smith-Waterman BLAST and hmmpfam), the ClustalW alignment needs more coordination and only part of the algorithm is parallelized. This means that it does not scale as well as the others. With seven slaves, the speed-up is 4.08 (see table 3).

Number of slaves	Time (seconds)	Speed-up
1	1096.71	1.00
2	608.31	1.80
3	441.98	2.48
4	383.76	2.85
5	331.75	3.30
6	285.47	3.84
7	268.79	4.08

Table 3: Benchmark of ClustalW.

The command used for the benchmark is:

```
clustalw_cell_mpi hiv.fasta
```

5 HMMER benchmark

The hmmpfam benchmark used the full PFAM database (ls) and 100 randomly selected protein sequences from Swiss-Prot as query (40,993 residues).

For hmmsearch, one model from the PFAM database was used, and in order to get measurable calculations, we use a four-fold copy of the full Swiss-Prot 53.3. The total number of sequences used is about 1,097,180 and the number of residues is 402,785,756.

5.1 hmmpfam

The scalability for hmmpfam resembles that of BLASTn. For the first three slaves, it is almost linear (2.82). With seven slaves, the speed-up is as high as 5.62 (table 4).

Number of slaves	Time (seconds)	Speed-up
1	70.02	1.00
2	36.06	1.94
3	24.79	2.82
4	19.38	3.61
5	15.91	4.40
6	13.84	5.05
7	12.44	5.62

Table 4: Benchmark of hmmpfam.

The command used for the benchmark is:

```
hmmpfam_cell_mpi Pfam_ls.bin uniprot_sprot_100.fasta
```

5.2 hmmsearch

For hmmsearch, speed-up with seven slaves is a little less with 4.24 (see table 5). The command

Number of slaves	Time (seconds)	Speed-up
1	98.06	1.00
2	54.00	1.81
3	40.60	2.41
4	32.21	3.04
5	27.81	3.52
6	25.06	3.91
7	23.10	4.24

Table 5: Benchmark of hmmsearch.

used for the benchmark is:

```
hmmsearch_cell_mpi Pfam_ls_1_f.bin uniprot_sprot_4x.fasta
```



6 Conclusion

The benchmarks of the MPI implementation of the *CLC Bioinformatics Cell* show that the database searches in particular scale really well. Figure 1 shows an overview of the five algorithms. BLASTn and hmmpfam display the best performance when they are scaled.

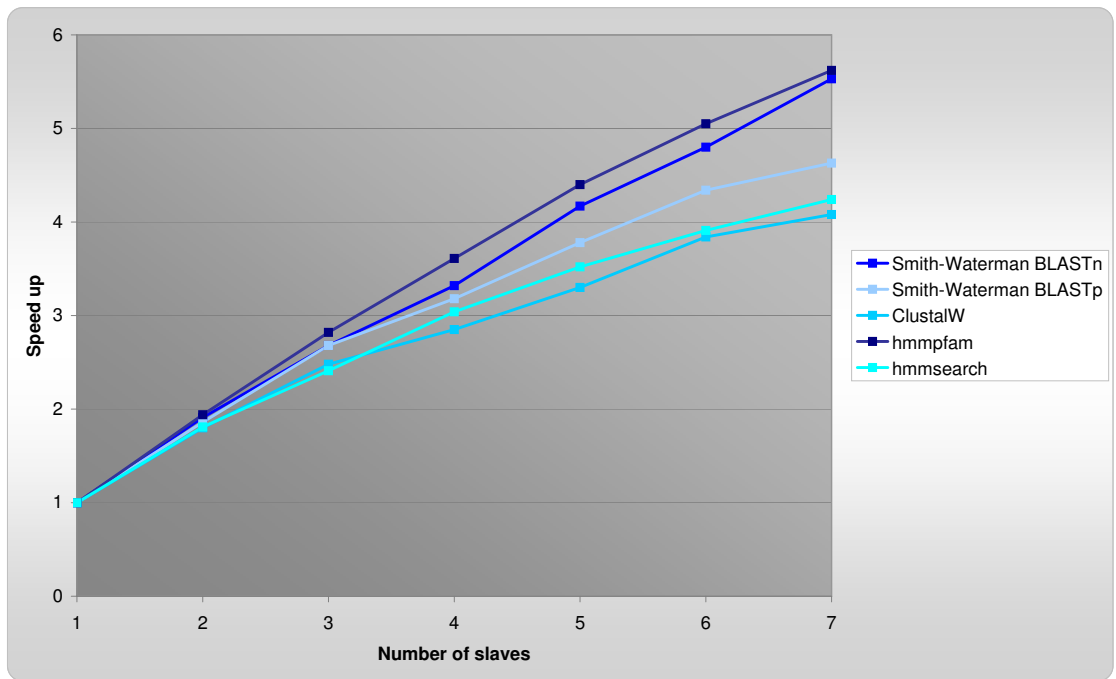


Figure 1: The results of the benchmarks summarized.

In general, searches in large databases scale better than smaller databases.