TimeLogic®
biocomputing solutions

Technical Note

Accelerated BLAST Performance with Tera-BLAST™: a comparison of FPGA versus GPU and CPU BLAST implementations

TimeLogic Division, Active Motif Inc, 1914 Palomar Oaks Way, Suite 150, Carlsbad, CA 92008

ABSTRACT

A number of technologies have emerged for accelerating similarity search algorithms in bioinformatics, including the use of field programmable gate arrays (FPGA), graphics processing units (GPU), and clusters of standard multi-core CPUs. Here we present Tera-BLAST™, an FPGA-accelerated implementation of the BLAST algorithm, and compare the performance to GPU-accelerated BLAST and the industry standard NCBI BLAST+ on high performance computers. Our results show that Tera-BLAST, running on the TimeLogic J-series FPGA Similarity Search Engine, performs 100's of times faster than BLAST running on generic NVIDIA Tesla M2090 GPU cards or standard Intel Xeon multi-core CPU's.

INTRODUCTION

The use of algorithms that perform similarity searches between nucleotide or protein sequences is commonplace within the field of bioinformatics. The Smith-Waterman algorithm (Smith & Waterman, 1981) has long been the 'Gold Standard' for sequence alignment tools because it delivers the best local alignment for a given query/target pair. Unfortunately, dynamic programming algorithms such as Smith-Waterman are too cumbersome for high-throughput bioinformatics work due to the large size of the data sets involved. As a result, heuristic methodology has been utilized to improve algorithmic performance. One such approach, the Basic Local Alignment Search Tool (BLAST) (Altschul et al, 1997), has been enormously popular as it very successfully achieves a balance of performance and search sensitivity. However, continued improvements in DNA sequencing technology along with the corresponding growth of public database repositories have meant that even algorithms like BLAST have struggled to keep up with demand. Addressing this challenge has traditionally required larger and larger CPU clusters with significant acquisition and operational costs.

To tackle this persistent problem, heuristic algorithms have been combined with hardware acceleration technologies like general-purpose graphics processing units (GPU) and field programmable gate arrays (FPGA). Here we compare BLAST running on CPUs, GPUs, and FPGAs to show that the FPGA-accelerated Tera-BLAST algorithm from TimeLogic provides the industry's fastest and most cost effective similarity search tool for high-throughput bioinformatics.

MATERIALS AND METHODS

Hardware and Software

All tests were run on a Dell R720 server with dual Xeon E5-2690 2.90 GHz 8-core hyper-threaded CPUs (32 cores) and 64GB RAM, under the 64-bit CentOS v6 Linux operating system.

CPU tests were performed with NCBI BLAST+ v2.2.27 (Camacho et al., 2009) with default parameters. All other software tests were run using the equivalent of these defaults: -num_descriptions 500 -num_alignments 250 -word_size 3 -threshold 11 -evalue 10 -seg no.

GPU tests were performed with a single high-end NVIDIA Tesla M2090 GPU card with 512 processors at 1.3 GHz, and 6GB of global memory (retail cost ~$2500.00). Setup required the download and installation of the CUDA toolkit v4.2.9, 64-bit Red Hat driver v295.41, and the GPU Computing SDK v 4.2.9, followed by application specific software for CUDA-BLASTP v2.0 (Liu, et al., 2011) and GPU-BLASTP v1.1 (Vouzis and Sahinidis, 2011). Because CUDA-BLASTP has a limitation of accepting only a single query sequence, a perl script was written to iterate over multiple query sequences in series. For GPU-BLAST, default values were used for '-gpu_blocks' and '-gpu_threads' parameters.

Tera-BLAST tests were performed with one and two TimeLogic J-series FPGA-accelerated Similarity Search Engines running DeCypher v9.0 software.

Data Sets

Query sequence data was taken from a subset of the Uniprot Swissprot database comprising the first 1.2 mega-symbols of amino acid residues (3,266 sequences).

Database sequence data was taken from a subset of the NCBI non-redundant (nr) protein database comprising the first 1 giga-symbols of amino acid residues (3,219,757 sequences). Binary BLAST databases were built following established protocols for each respective BLAST implementation.

RESULTS AND DISCUSSION

FPGA-Accelerated Tera-BLAST

Tera-BLAST searches with one and two TimeLogic J-series accelerators installed in a 32 CPU-core server ran in 2.9 and 1.9 minutes, respectively. Comparatively, GPU-BLAST searches ran between 303.1 and 104.9 minutes, and NCBI BLAST+ between 817.7 and 77.7 minutes, depending on the number of CPU cores used in these searches (Table 1 and Figure 1). NCBI BLAST+ and GPU-BLAST search performance is dependent on the number of CPU cores available, and increases with increasing CPU allocation. However, as CPU cores increase, GPU performance eventually falls below CPU-only performance. When 16 CPU cores are used, the benefit of using a GPU accelerator is negligible (107.8 minutes vs 104.9 minutes), and when 32 CPU cores are used, GPU accelerated versions actually run 41% slower than CPU-only versions.

	Time (minutes)			
CPU Cores	NCBI BLAST+ (2x Xeon E5-2690)	GPU-BLAST (1x Tesla M2090)	Tera-BLAST (1x J3 accelerator)	Tera-BLAST (2x J3 accelerators)
1	817.7	303.1	n/a	n/a
2	445.4	233.0	n/a	n/a
4	257.2	154.3	n/a	n/a
8	167.8	110.8	n/a	n/a
16	107.8	104.9	n/a	n/a
32	77.7	109.7	2.9	1.9

Table 1: Search times for each BLAST version search using increasing numbers of CPU cores.

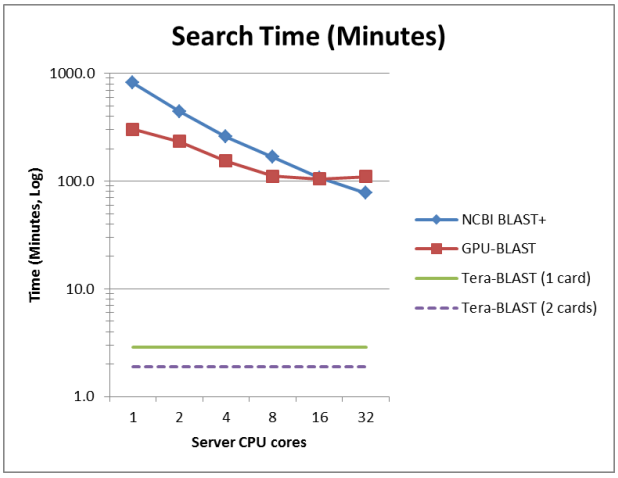

Figure 1: BLAST search times in minutes for Tera-BLAST (FPGA), GPU-BLAST (GPU), and NCBI-BLAST+ (CPU).

Using two J-series similarity search engines, Tera-BLAST runs up to 434.2x faster than NCBI BLAST+ running on a single core (Table 2 and Figure 2). Tera-BLAST runs up to 41.2x faster than NCBI BLAST+ running on 32 processor cores (Figure 2), or 1319.7x per core equivalent (Table 3 and Figure 3).

	Acceleration Factor vs. 1 core			
CPU Cores	NCBI BLAST+ (2x Xeon E5-2690)	GPU-BLAST (1x Tesla M2090)	Tera-BLAST (1x J3 accelerator)	Tera-BLAST (2x J3 accelerators)
1	1.0x	2.7x	n/a	n/a
2	1.8x	3.5x	n/a	n/a
4	3.2x	5.3x	n/a	n/a
8	4.9x	7.4x	n/a	n/a
16	7.6x	7.8x	n/a	n/a
32	10.5x	7.5x	283.6x	434.2x

Table 2: Tera-BLAST runs up to 434.2x faster than NCBI BLAST+ running on a single core. NCBI BLAST+ runs only 10.5x faster when 32 cores vs. 1 core are used. GPU-BLAST runs slower than NCBI BLAST+ once more than 16 CPU cores are available.

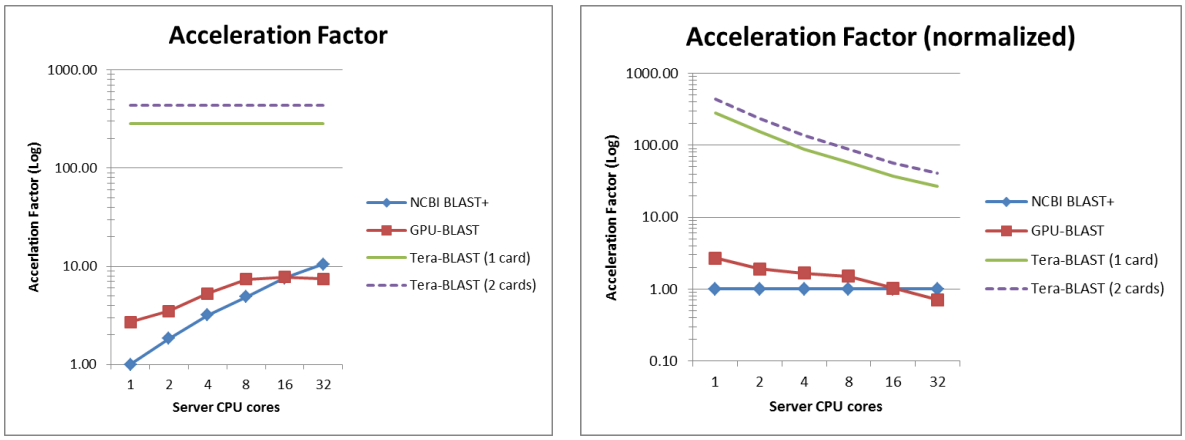

Figure 2: Tera-BLAST performs up to 434.2x faster than NCBI BLAST+ and up to 161.0x faster than GPU-BLAST when only 1 CPU is used. When 32 CPU cores are used, Tera-BLAST performs up to 41.2x faster than NCBI-BLAST+ and up to 58.2x faster than GPU-BLAST.

	Acceleration Factor as core equivalents			
CPU Cores	NCBI BLAST+ (2x Xeon E5-2690)	GPU-BLAST (1x Tesla M2090)	Tera-BLAST (1x J3 accelerator)	Tera-BLAST (2x J3 accelerators)
1	1.0x	2.7x	n/a	n/a
2	2.0x	3.8x	n/a	n/a
4	4.0x	6.7x	n/a	n/a
8	8.0x	11.1x	n/a	n/a
16	16.0x	16.4x	n/a	n/a
32	32.0x	22.7x	862.0x	1319.7x

Table 3: CPU Core Equivalents are calculated by multiplying the normalized acceleration factor by the number of cores used.

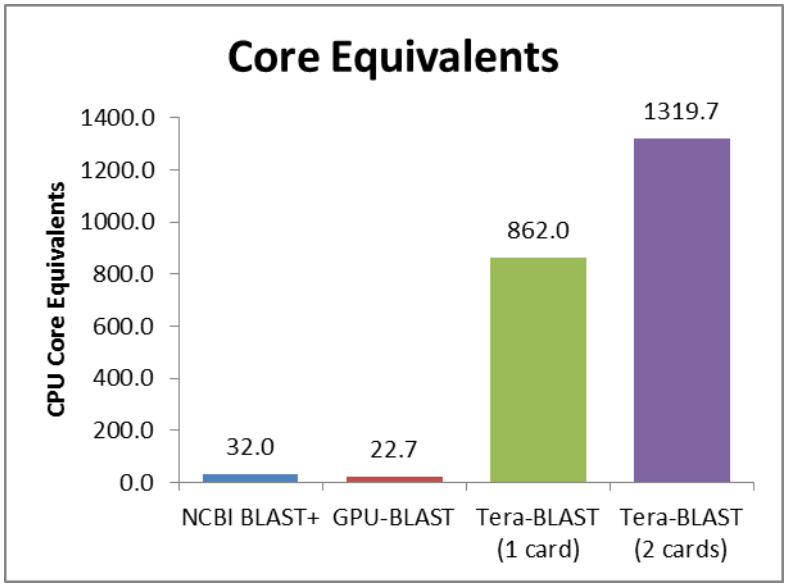

Figure 3: Tera-BLAST supplies up to 1319.7 core equivalents of performance in a single 2U node.

Types of GPU-Accelerated BLAST

The CUDA-BLASTP software is published as performing up to 10x faster than NCBI BLAST+ when compared to a single CPU core (Liu, et al., 2011). In practice, performance is generally lower due to several significant limitations. First, optimal performance is highly data specific and dependent on query sequence length. Second, CUDA-BLASTP was designed to only accept a single query sequence. This severely limits its use in high performance computing, as the only way to analyze larger data sets is to iterate serially over multiple query sequence files. Such iteration, according to the CUDA-BLASTP author, will add significant performance overhead. Using this iterative method, which was unnecessary for all other software in this experiment, **CUDA-BLASTP performed slower than standard CPUs.** In this case, performance was decreased by approximately 50 percent (data not shown). CUDA-BLASTP also reported errors on approximately 5% of the search iterations: "Unspecified launch failure" errors and "Segmentation fault (core dumped)".

Alternatively, the GPU-BLAST software is built upon the standard NCBI-BLAST+ code and, like the original, can handle both multiple queries and multiple CPU processors. It is reported to provide a speedup of 3-4 times when compared to NCBI BLAST+ running on a single CPU core, but with our data, GPU-BLAST performed only approximately 2 times faster (data shown above). When more commonly available multi-core CPUs are used, for example the 32 CPU core Dell R720 servers used for these tests, **GPU-BLAST ran 41% slower than NCBI BLAST+.**

CONCLUSION

In summary, Tera-BLAST offers the best performance for the BLAST algorithm based on resource investment. Tera-BLAST exceeds CPU performance by up to 434.2x, and GPU-accelerated performance by up to 161.0x. Furthermore, Tera-BLAST provides the equivalent of up to 1319.7 CPU cores of performance in a single 2U server. In addition to superior performance, TimeLogic's DeCypher Tera-BLAST offers far more flexibility than GPU-based implementations in that it can be used for all flavors of BLAST (BLASTN, BLASTP, BLASTX, TBLASTN, TBLASTX), as well as Tera-Probe™, which these GPU alternatives do not support.

GPU-accelerated BLAST algorithms perform only slightly better than standard CPUs, and only when small numbers of CPU cores are utilized. When more than 8 CPU cores are used, this performance gain disappears altogether. When all 32 available CPU cores were utilized, GPU performance was actually worse than NCBI BLAST+ running in software. As multi-core CPUs are now the minimal standard offering, GPU computing for BLAST does not appear to be a cost-effective investment.

When considering price for performance in high-throughput biocomputing, DeCypher Tera-BLAST is by far the best solution for accelerated high-throughput analysis.

REFERENCES

1. Altschul, S., Madden, T., Schäffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25:3389-3402. (PMID: 9254694)

2. Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., Madden, T.L. (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421. (PMID: 20003500)

3. Liu, W., Schmidt, B., and Muller-Wittig, W. (2011) CUDA-BLASTP: Accelerating BLASTP on CUDA-Enabled Graphics Hardware. IEEE/ACM Trans. Comput. Biol. Bioinform. 8(6):1678-84. (PMID: 21339531)

4. Smith, T., Waterman, M. (1981) Identification of common molecular subsequences. J Molecular Biology 147:195–197. (PMID: 7265238)

5. Vouzis, P. and Sahinidis, N. (2011) GPU-BLAST: using graphics processors to accelerate protein sequence alignment. Bioinformatics, 2011, 27(2):182-8. (PMID: 21088027)