WHITE PAPER

SGI® High Throughput Computing (HTC) Wrapper Program for Bioinformatics on SGI ICE™ and SGI UV™ Systems

Haruna Cofer*, PhD

January, 2012

Abstract

The SGI High Throughput Computing (HTC) Wrapper Program for Bioinformatics allows customers to maximize throughput and utilization of their SGI systems for running standard bioinformatics applications like BLAST, FASTA, ClustalW and HMMER. Developed by SGI Application Engineers, HTC works with the original application binaries and allows users to obtain the same results far more efficiently and easily than with a standard batch scheduler. HTC is freely available from SGI and designed to run on SGI Rackable™ Standard-Depth, SGI ICE and SGI UV systems.

*SGI Applications Engineering

TABLE OF CONTENTS

1.0 Introduction

The DNA sequencing centers and laboratories of today are capable of generating the nucleic bases of thousands to millions of DNA fragments in a single day. These fragments are cleaned and assembled into long contiguous sequences using sequence assembly algorithms like Velvet[1] and further analyzed using sequence similarity algorithms like BLAST[2]. The sequence information obtained by these methods is then merged into existing public domain or proprietary databases, to be searched and analyzed by researchers worldwide.

In order for these facilities to keep up with the dramatic increase in data flow, large amounts of compute power are necessary. Algorithms like BLAST and HMMER[3] are parallelized for execution on multiple processors, but as processing power increases and code implementation improves, the scalability generally decreases and may be no more than four to eight processing cores. Therefore, a new approach is needed to fully utilize the compute resources available for analyzing the thousands of sequences generated each day.

Typical large-scale sequence analyses using codes like BLAST are implemented within an automated pipeline using Perl scripts and batch scheduler job arrays. However, there are small but finite startup overheads associated with launching jobs using these methods. The overhead may only be two to five seconds, but compared to a 20 second BLAST job, this 10% to 25% overhead can become very significant when multiplied by thousands of jobs.

As a result of customer demand for a more efficient high throughput analysis tool, SGI application engineers developed a wrapper program called the HTC (High Throughput Computing) Wrapper Program for Bioinformatics. There are other programs available to support the high throughput processing of bioinformatics programs, but they can be difficult to install and configure, and may not be free. In contrast, HTC from SGI is a single binary that transparently reads all the inputs, load balances the jobs, and submits the jobs to maximize system utilization and efficiency. It is distributed free of charge for running on SGI systems and can be downloaded at www.sgi.com/solutions/research/htc.html

2.0 HTC (High Throughput Computing) Wrapper Program for Bioinformatics

The HTC (High Throughput Computing) Wrapper Program for Bioinformatics is a driver program for bioinformatics applications like BLAST, FASTA, ClustalW and HMMER that is optimized for use in a high throughput production environment running on a single shared or distributed cluster of SGI servers. HTC supports high volume processing where multiple query sequences are being searched against possibly multiple databases.

Figure 1 is a simple schematic representation of the processing performed by HTC. From left to right, first HTC optionally pre-sorts the input sequences based on sequence length in order to achieve better load balancing of the jobs. For example, input sequences with longer lengths generally correspond to longer running jobs, so the user may achieve better throughput performance if HTC is allowed to pre-sort and run those input files first. HTC then divides the optionally pre-sorted input sequences into blocks, each containing k sequences. The blocked sequence files are then input to multiple instantiations of the application program (e.g., BLAST, HMMER) and launched with dynamic scheduling to all available processor cores for maximum system utilization and efficiency.

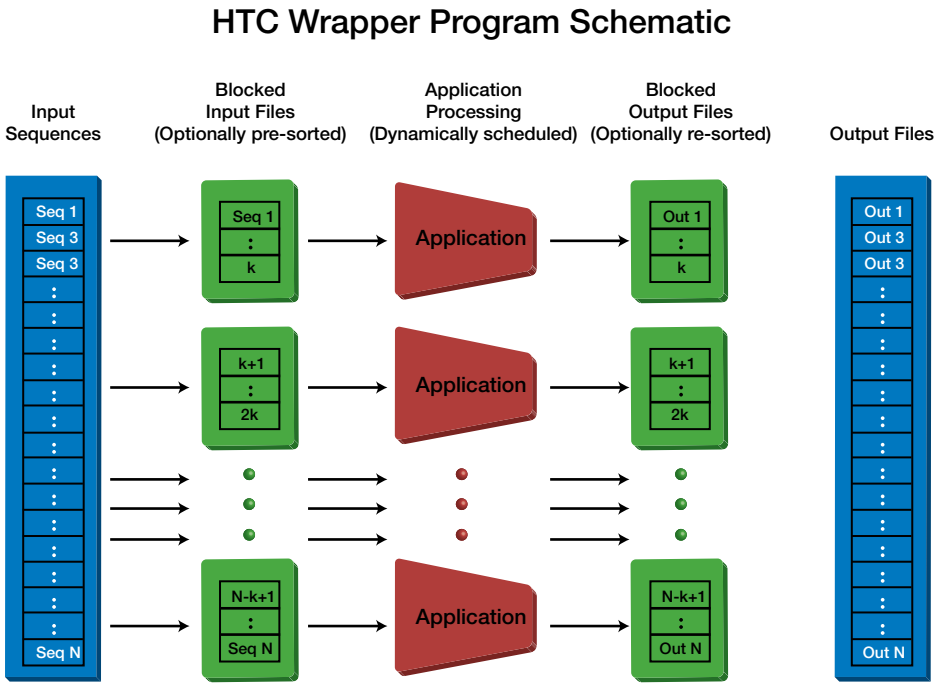


Figure 1. Schematic representation of HTC Wrapper Program.

The final program outputs are then collected and optionally re-sorted back into the original order for the user.

HTC works with the original unmodified binaries for the latest BLAST+, FASTA, ClustalW, HMMER3 and any user-supplied script that calls an alternate program of choice (e.g., older BLAST, HMMER2). However, for maximum throughput and efficiency, it is recommended that the program be built serially without pthreads or other parallel programming constructs. Since parallelism now resides at the top HTC driver level, removing these extra parallel programming constructs will further reduce compute overhead.

2.1 BLAST

BLAST (Basic Local Alignment Search Tool) [1,2] is one of the most widely used similarity search tools for biological sequences. BLAST is actually a collection of programs that use the BLAST algorithm to compare protein or DNA sequence queries to protein or DNA sequence databases, in order to rapidly identify statistically significant matches between newly sequenced fragments and existing databases of known sequences. These searches enable the scientist to gain insight and make inferences about the possible function and structure of the new unknown sequence, or to initially screen a large number of new sequences for further analysis using more sensitive and compute-intensive algorithms. As the acronym implies, the BLAST algorithm implements a heuristic method that is designed for speed, seeking local as opposed to global alignments that are better able to detect relationships among sequences with only isolated regions of similarity.

The latest version of BLAST from the NCBI (National Center for Biotechnology Information) is called BLAST+. BLAST+ represents a complete re-write of the legacy BLAST program originally based on the NCBI C Toolkit, to a program based on the NCBI C++ Toolkit. The new BLAST+ algorithms include many performance and feature improvements over the legacy BLAST algorithms, generally resulting in faster and more efficient execution. This performance improvement is most noticeable with the BLASTN algorithm, where SGI has observed up to a 4x performance improvement over the legacy BLASTN implementation.

2.1.1 HMMER

HMMER [3] is a popular sequence analysis tool for protein sequences. As the name implies, HMMER uses probabilistic models called profile hidden Markov models (profile HMMs) to represent and identify homologous protein sequences, enabling HMMER to be very accurate and sensitive in detecting remote homologs.

The latest version of HMMER is called HMMER3, and it represents a complete re-write of the previous HMMER2 code. HMMER3 implements several performance enhancing algorithms such as a heuristic filter and a log-likelihood model, as well as the use of vector instruction sets (e.g., SSE2, Altivec/VMX), resulting in execution times that are comparable to BLAST.

3.0 Benchmarks

The benchmarks below use BLAST+ and HMMER3 as examples of the excellent performance and scaling that HTC can achieve to process thousands of inputs. The benchmarks were run on a shared memory SGI UV 1000 and a distributed memory SGI ICE 8400 system with the following specifications:

SGI UV 1000

- 32 dual-socket nodes, each socket with 6-core Intel® Xeon® Processor X7542 (18MB Cache, 2.66 GHz, 5.86 GT/s Intel® QPI), total 384 cores
- Total Memory: 2TB
- Speed: 1067 MHz (0.9 ns)
- SLES11SP0 OS, SGI ProPack 7SP1
- HyperThreading not available
- TurboBoost available but not used

SGI ICE 8400

- 128 dual-socket nodes, each socket with 6-core Intel® Xeon® Processor X5680 (12MB Cache, 3.33 GHz, 6.40 GT/s Intel® QPI), total 1536 cores
- Total Memory: 3TB
- Speed: 1333 MHz (0.8 ns)
- SGI Tempo Admin Node 2.1
- SLES11SP0 OS, SGI ProPack 7SP1
- HyperThreading available but not used
- TurboBoost available but not used

3.0.1 About the SGI Systems

SGI UV 1000

SGI UV scales to extraordinary levels — up to 256 sockets (2,560 cores, 4096 threads) with architectural support to 32,768 sockets (262,144 cores). Supporting up to 16TB of global shared memory in a single system image, the UV enables the efficient execution and scaling of applications ranging from in-memory sequence database searches and *de novo* sequence assemblies, to a diverse set of general data and compute-intensive HPC applications.

SGI ICE 8400

The SGI ICE integrated blade cluster was designed for today's data intensive problems. This innovative platform from SGI raises the efficiency bar, easily scaling to meet virtually any processing requirement without compromising ease of use, manageability or price/performance. SGI ICE delivers unsurpassed customer value with breakthrough density, efficiency, reliability and manageability to meet the computational demands of supporting a high throughput sequence analysis pipeline, as well as serving a broad and varied research community.

3.1 HTC-BLAST vs. BLAST Performance

The benefits of using HTC-BLAST become very clear when searching large volumes of queries. HTC-BLAST is designed to maximize throughput and utilization of the system while running transparently for the user. The benchmark below compares the throughput performance of running BLAST in two ways using:

1) HTC 4.0 wrapper program
2) PBSPro 10.4 job array scheduler

The comparison is performed on both a shared memory SGI UV 1000 and a distributed memory SGI ICE 8400. The test case consists of running BLASTP searches using 10,000 sequences from the RefSeq protein database (3,221,557 residues in 10,000 sequences) against a Non-Redundant Protein Database (1,336,756,300 residues in 3,878,509 sequences) using BLASTP from NCBI, BLAST version 2.2.23+.

The results below show that running BLAST with HTC (HTC-BLAST) is faster than running BLAST with a PBSPro job array scheduler, even on just 12 cores (two sockets). HTC-BLAST on 12 cores of a 2.67GHz SGI UV 1000 is 1.6 times faster than running the same test using PBSPro job arrays over the same number of cores. Similarly, using HTC-BLAST on 12 cores of a 3.33GHz SGI ICE 8400 demonstrates a 1.5-fold improvement in throughput over using PBSPro job arrays.

System	BLAST using HTC 4.0	BLAST using PBSPro 10.4 Job Array	Speedup of HTC over PBSPro Job Array
2.67GHz SGI UV 1000	33 jobs/min	20 jobs/min	1.6x
3.33GHz SGI ICE 8400	40 jobs/min	26 jobs/min	1.5x

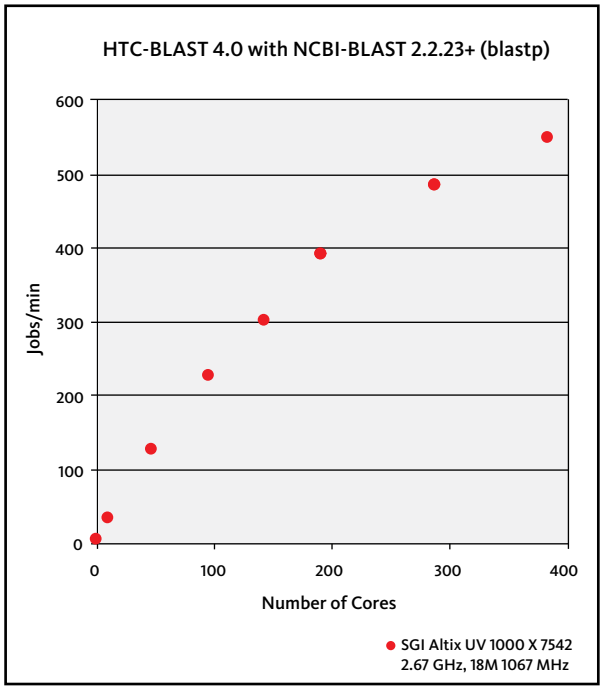
3.2 HTC Scaling

The graphs below demonstrate the excellent scaling of the HTC program with two bioinformatics programs on two different systems:

1) BLAST+ on a shared memory SGI UV 1000 system
2) HMMER3 on a distributed memory SGI ICE 8400 system

3.2.1 HTC-BLAST Scaling

In the first benchmark (Figure 2), 10,000 sequences (3,221,557 residues) from the RefSeq protein database are queried against a Non-Redundant Protein Database (1,336,756,300 residues in 3,878,509 sequences) using BLASTP from NCBI, BLAST version 2.2.23+. HTC-BLAST from SGI reduces the execution time from over 60 hours on a single core to 18 minutes on 384 cores of an SGI UV 1000 system, yielding results for nearly 800,000 BLASTP queries per day. This corresponds to an increase in throughput rate from 3 jobs per minute on a single core to almost 550 jobs per minute on 384 cores.



SGI UV 1000 X7542
2.67 GHz, 18M 1067 MHz

Executable: HTC-Bio 4.0 with N CBI BLAST 2.2.23+ (blastp), compiled using Intel C Compiler version 11.1.064.

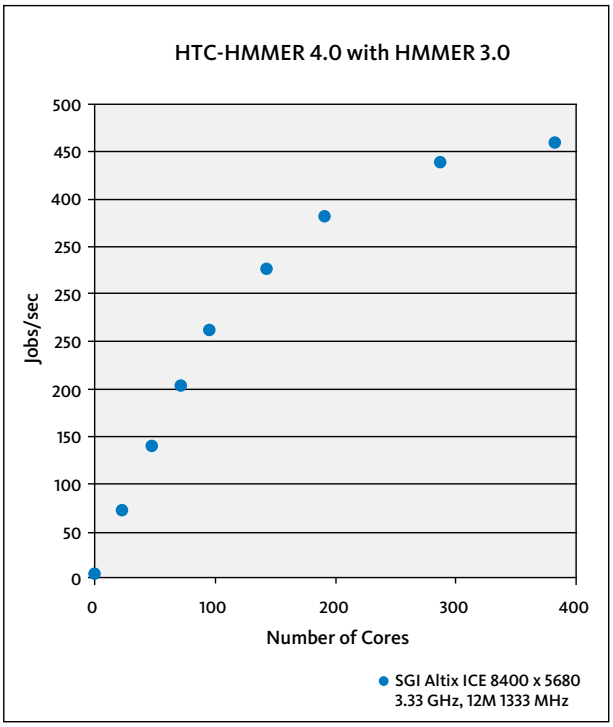
Query: 10,000 sequences from RefSeq protein database (3, 221,557 residues in 10,000 sequences).

Database: Non-Redundant Protein Database (1, 336,756,300 residues in 3,878,509 sequences).

Figure 2. Performance of HTC-BLAST on SGI UV 1000 system.

3.2.2 HTC-HMMER Scaling

In the next benchmark (Figure 3), 100,000 sequences (38,330,252 amino acids) from SwissProt 57.6 were queried against the Pfam-A 24.0 database (11,912 families with 7,079,739 sequences and 1,627,712,293 residues) using HMMSCAN from HMMER version 3.0. HTC-HMMER from SGI reduces the execution time from almost six hours on a single core to only a few minutes on 384 cores of an SGI ICE 8400 cluster, representing a throughput rate of almost 40 million queries per day. This corresponds to an increase in throughput rate from five jobs per second on a single core to over 450 jobs per second on 384 cores.



SGI ICE 8400 X5680
3.33 GHz, 12M 1333 MHz

Executable: HTC-Bio 4.0 with HMMER 3.0 (March 2010), compiled using Intel C Compiler version 11.1.064.

Query: 100, 000 sequences (38, 330,252 amino acids) from SwissProt 57.6.

Database: Pfam-A 24.0 database (11, 912 families with 7, 079,739 sequences and 1,627,712,293 residues).

Figure 3. Performance of HTC-HMMER on SGI ICE 8400 system.

4.0 Conclusion

HTC has been shown to demonstrate significantly higher performance and scalability than a general batch scheduler for processing thousands of query sequences using the popular bioinformatics packages BLAST+ and HMMER3. The performance improvement is mainly due to the automatic load balancing provided by HTC and the reduction in startup overhead when launching thousands of jobs. The near-linear scalability of HTC not only ensures maximum utilization of compute resources, but also simplifies future system expansions as the complexity and volume of query sequences increase over time.

5.0 References

[1] Zerbino, D.R. and E. Birney. “Velvet: algorithms for *de novo* short read assembly using de Bruijn graphs.” *Genome Research* 18:821-829 (1998).

[2] Altschul, S.F., and W. Gish. “Local alignment statistics.” Ed. R. Doolittle. *Methods in Enzymology* 266:460-80 (1996).

[2] Karlin, S., and S.F. Altschul. “Applications and statistics for multiple high scoring segments in molecular sequences.” *Proc. Natl. Acad. Sci.* 90:5873-7 (1993).

[3] Durbin, R., S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acid*. Cambridge University Press, 1998.

Global Sales and Support: sgi.com/global