

mpiBLAST:

Parallelization of BLAST for **Computational Clusters**

RADIANT LOGO HERE



Aaron Darling Dept. of Computer Science University of Wisconsin-Madison darling@cs.wisc.edu

Wu-chun Feng Advanced Computing Laboratory Los Alamos National Laboratory feng@lanl.gov



All About BLAST

What is **BLAST**?

- BLAST is a family of sequence database search algorithms [1,2]

- Foundation for much research in molecular genetics

- Searches for similarities between a short query sequence and a large set of database sequences.

- Query and database sequences are either DNA or amino acid (protein) sequence, BLAST will translate between DNA and amino acid sequenceson-the-fly.

Search Name	Query Type	Database Type	Translation
blastn	Nucleotide	Nucleotide	None
tblastn	Peptide	Nucleotide	Database
blastx	Nucleotide	Peptide	Query
blastp	Peptide	Peptide	None
tblastx	Nucleotide	Nucleotide	Query and Database

Table 1. Standard BLAST database search types. DNA and amino acid sequences are translated by BLAST, making searches with several combinations of database and query type possible.

mpiBLAST Performance

Modelling BLAST Usage

- The length, number and content of the query and database sequences significantly affect BLAST runtime [11]
- Must choose queries and the database to accurately reflect a typical BLAST usage pattern

- We model a common problem in genomics: using BLAST on all predicted ORFs in a nevvly sequenced genome [16] Sequence database: nt

Queries: Predicted ORFs from the E. Chrysanthemi genome



Figure 5. Histograms of the query sequence lengths and database sequence lengths that were used for benchmarking NCBI BLAST and mpi-BLASI

Querv

>Perilla frutescens mRNA for leucoanthocyanidin dioxygenase CATCTACTCAAATTAAGAAATAGATAGAAATGGTTACGAGTGCAATGGGTCCAAGCCCGCGGGTGGAGGAACTGGCCCGA AGCGGACTCGACACGATCCCAAAAGTATACGTGCGGCCCGAAGAGCACCTGAAAAGTATCATAGGCAACATTTTGGCGGA ΑΤΑΑΑΑΤΤΑΑΑΤΑΑΤΤΤΤΤΟΑΑΤΤGTACTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Database

gi|3123744|dbi|AB013447.1|AB013447 Brassica napus mRNA for aluminum-induce AATTTTCAGCGGACGTATAGTATCATTGCCGGAAGAGCTGGTGGCTGCCGGGAACCGAACT CGGATCGGTCCTAGTCAACAAGTTCGTAGAGAAAAATCCCTCTGCCGTGTCCGTACAGGTCGGCGACTACGTGCAGCTT |HS2HSV2P4 Herpes simplex virus type 2 gene for glycoprotei CGACCGACGGCTCCTGCCACCCGAACAT ATATGGAACTTGTTACATCTCCTTCTAGTTGTCAAAGGTCTTTGGAAGGCAGGATCTGTTTCATAGATGAGCTTCTG ATACTTTTAGAAGATATATAAAATTTATTTCTATGAAAAAGGTTATTACTTGACAATAA TGTTATTTAAAAATTTGCCTCTAGATTCTTGTGGGCCTTCTAGGTACA ATATCACATTGTCCCAAACCTGCTCAGCTCCTTATCAAAATCAAAAACATTTCCATCAACTTTGTGGTCCAGGTGCCAAT CCACCTCCTTCATATGGAATTGCTTGCTAGATCCTG

Hit Extension GTGGGTATTGGCGGCCCGAAGGGCCCGCCGC AAGACTACGTGCGGCCCGAAGAGCACCTGAAA

Figure 1. An example of the BLAST algorithm

Why Parallelize BLAST?

Sequence DB is larger than the main memory of a standard workstation.

- DB searches must utilize slow disk I/O

- Sequence DBs are growing faster than they can be processed [14]

- Solution 1: Buy a computer with more memory

- Solution 2: Use inexpensive commodity components in parallel

BLAST searches are very CPU intensive

- But sequence DB entries are essentially independent and can be searched in parallel.
- Communication costs during a parallel search are low
- Parallel BLAST can utilize efficient, low cost clusters like Green Destiny

The BLAST Algorithm

1. The query sequence is indexed with a string matching data structure 2. Database sequences are 'streamed' past the query index 3. Exact subsequence matches (hits) of some minimum size are identified 4. Hits are extended to the surrounding matching region by scoring each successive extension with a statistically motivated function. 5. The extended hits, called High Scoring Pairs (HSPs), are sorted based on score and are reported to the user in one of several formats



Figure 2. Growth of computer memory size vs. Genbank size. Genbank is a public repository for DNA sequences that is maintained by NCBI. The compressed size of Genbank is an approximation based on 2-bit per base encoding of the data at [3]



Figure 6. blastn searches as the database grows larger than the system buffer-cache. Disk I/O sharply increases when the database becomes too large, degrading blastn performance.



Figure 7. Performance of blastn searches using mpiBLAST blastn performance suffers severely when the database becomes too large for the system buffer-cache. In this case, mpiBLAST yields a super-linear speedup versus a single node.

Performance in Low Memory

- Goal:
- Determine BLAST behavior when the database is too large to fit in the system buffer-cache
- Method:
- NCBI BLAST and mpiBLAST were benchmarked using increasingly large database sizes
- Attempt to cache the DB by doing an untimed search before each timed search

Results:

- When the DB is too large for buffer-cache, disk I/O increases sharply as does BLAST search time
- mpiBLAST on a single node behaves similarly to NCBI BLAST - When run with 2 or 4 workers, the DB can be respectively 2 or 4 times larger before heavy disk I/O occurs



Figure 8. Performance of tblastx searches using mpiBLAST. tblastx is not affected by heavy disk I/O because it is CPU-bound. mpiBLAST achieves linear speedup for tblastx searches.



BLAST in Parallel

Parallel BLAST Approaches

Multithreading

- Each thread searches a distinct portion of the database
- Implemented in NCBI BLAST

Query Segmentation

- Each query is farmed out to a distinct node in a cluster
- Each node searches the entire database
- Implemented in HT-BLAST[4], cBLAST[15], U. lowa BLAST[5], TurboBLAST[6,7]

Database Segmentation

- Each node in a cluster searches a distinct portion of the database
- The query is sent to all nodes in the cluster
- Implemented in TurboBLAST[6,7], Blackstone PowerBLAST, and Platform LSF

mpiBLAST is a freely available open-source implementation of database segmentation for BLAST searches

mpiBLAST Algorithm

Format the Database

- Execute a wrapper for standard NCBI formatdb
- Fragments database and copies the fragments to shared storage

Figure 3. Our 'hokey' depiction of database segmentation. Each worker node receives a distinct portion of the entire sequence database. The process does not actually involve a spaghetti press.



mpiBLAST, etc.

mpiBLAST Speedup

Goal:

- Determine the speedup and scalability of mpiBLAST on a large cluster

Method:

- Benchmark mpiBLAST on Green Destiny[9,10]
- Each node has a 667 MHz Transmeta TM5600, 640MB RAM, and a 20GB hard disk.
- Nodes are interconnected with switched 100Base-TX ethernet
- Operating system is Linux 2.4

Results:

- mpiBLAST exhibits super-linear speedup when run on multiple cluster nodes.
- Efficiency decreases as the number of workers increases

mpiBLAST can be downloaded from http://mpiblast.lanl.gov





Figure 9 (above). The speedup of mpiBLAST on Green Destiny. Speedup is based on the search time for 300 kb of query sequences against 5.1 GB of the nt database.

Figure 10 (right). Green Destiny, a 240 node bladed beowulf based on the Transmeta Crusoe processor.

Future Directions

- Perform a detailed analysis of 'Where the time goes' when doing parallel BLAST searches
- Implement query segmentation for large clusters with high startup costs
- Use the NCBI toolkit to directly output the merged BLAST results
- in several formats including XML, ASN.1 and tab delimited text
- Implement seamless fault tolerance so that a search need not be restarted when a node goes down.

1. S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Research, 25, pp. 3389-3402, 1997

2. Stephen Altschul, Warren Gish, Webb Miller, Eugene Myers, and David Lipman, "Basic local alignment search tool", Journal of Molecular Biology, 215:403-410, 1990. 3. D. A. Benson, M. S. Boguski, D. J. Lipman, J. Ostell, "GenBank", Nucleic Acids Research, 25:1, pp. 1-6, 1997.

4. N. Camp, H. Cofer, and R. Gomperts, "High-Throughput BLAST", SGI White Paper, September 1998.

5. R.C. Braun, K.T. Pedretti, T.L. Casavant, T.E. Scheetz, C.L. Birkett, and C.A. Roberts, "Parallelization of Local BLAST Service on Workstation Clusters", Future Generation Computer Systems, 17:6 pp. 745-754, April 2001.

6. Ruzhu Chen, Clarisse Taaffe-Hedglin, Nathan Willard, and Andrew H. Sherman, "Benchmark and Performance Analysis of TurboBLAST on IBM xSeries Server Cluster", IBM Red Paper, 2002. 7. R.D. Bjornson, A.H. Sherman, S.B. Weston, N. Willard, and J. Wing, "TurboBLAST: A Parallel Implementation of BLAST Based on the TurboHub Process Integration Architecture.", IPDPS 2002 Workshops, April 2002.

- Database segmentation

- Load balancing - Basic fault tolerance
- Integration with scheduler systems (PBS)

mpiBLAST Approach

- Open source, MPI based [13], available at http://mpiblast.lanl.gov







- If a worker is assigned a fragment it does not have on local storage, the fragment is copied to local storage - As fragment searches complete, results are reported to the master node. - Master node merges results into a single results file

Figure 4. A schematic of the mpiBLAST query algorithm. The query is sent to each worker where a unique portion of the database is searched. Results are sent to the master node to be merged.

8. Al Shpuntof of AFS Informatics and Christopher Hoover of TimeLogic, Personal Communication, August 2002

9. M. Warren, E. Weigle, and W. Feng, "High-Density Computing: A 240-Node Beowulf in One Cubic Meter", To appear at SC2002: High-Performance Networking and Computing Conference, (Baltimore, MD, Nov 16 - Nov 22, 2002).

10. W. Feng and M. Warren and E. Weigle, "The Bladed Beowulf: A Cost-Effective Alternative to Traditional Beowulfs", To appear at IEEE Cluster 2002.

11. E. Chi, E. Shoop, J. Carlis, E. Retzel, and J. Riedl, "Efficiency of shared-memory multiprocessors for a genetic sequence similarity search algorithm", Technical Report TR97-005, University of Minnesota Computer Science Department, 1997.

12. Ilya Sharapov, "Computational Applications for Life Sciences on Sun Platforms: Performance Overview", Sun White Paper, November 2001. 13. W. Gropp, E. Lusk, A. Skjellum, "Using MPI: Portable Parallel Programming with the Message Passing Interface", 2nd Edition, 1999 14. W. Jim Kent, "BLAT - The BLAST-Like Alignment Tool", Genome Research, 12:656-664, April 2002.

15. cBLAST. http://www.cbbc.murdoch.edu.au/research/cluster/.

16. Jeremy D. Glasner, Guy Plunkett III, Paul Liss, Aaron Darling, Frederick R. Blattner, and Nicole T. Perna, "ASAP, a systematic annotation package for community analysis of genomes", Nucleic Acids Research, January 2003.