Bioinformatics Applications of Parallel Processing

Janine Garnham
Zerene Sangma
Outline

- Introduction to Bioinformatics
- Applications of Bioinformatics
- Cluster & Grid Computing
- Bioinformatics Applications of Parallel Processing to be Addressed:
  - Phylogenetic Tree Calculation
  - Sequence Alignment
  - Protein Folding
  - RNA Folding
- Challenges & Future Work
Bioinformatics

The field of bioinformatics is concerned with the application of computers to biological data. Computers are used in the collection, storage, classification and analysis of this data. Much of this data is in the form of DNA (nucleotide bases) or protein sequences (amino acids).

Motivation for Parallel Processing:
- Large and exponentially increasing amounts of data (INSDEC alone contains 100 gigabases)
- Complex algorithms/computations on that data
Applications of Bioinformatics

Medical Application
- Understand life processes in healthy and disease states.
- Genetic Disease (SNPs)

Pharmaceutical and Biotech Industry
- To find and (develop) new and better drugs
- Gene-based or structure based drug design

Agricultural Applications
- Disease and Drought resistance plants
- Higher Yield crops
Cluster & Grid Computing

- Two common approaches to parallel processing in the bioinformatics field are cluster computing and grid computing due to their accessibility to ordinary researchers.

- Grid computing is a type of distributed and parallel computing that utilizes many geographically dispersed computers and/or clusters connected over a network. Machines in the grid may differ widely in hardware, operating systems, resources, etc.
Example Grids

- **TeraGrid** - Funded by NSF and is comprised of 8 US partners including universities, supercomputing centers and national labs. Over 102 teraflops of computing capability.

- **SHARCNET** - Shared Hierarchical Academic Research Computing Network consisting of 14 academic institutions in South Central Ontario. Total CPU performance of 5423 gflops.

- **European DataGrid** - Funded by European Union. At its largest was comprised of 25 sites in Europe, Russia and Taiwan with over 1000 processors. Concluded March 2004.

- **World Wide Grid (WWG)** ?
Bioinformatics Applications of Parallel Processing

Applications to be Addressed:

- Phylogenetic Tree Calculation
- Sequence Alignment
- Protein Folding
- RNA Folding
A phylogenetic tree displays the evolutionary relationships between various species believed to have a common ancestor.

Trees can be used for modeling of evolution at the molecular level, prediction of new gene functionality, identification of important genome regions, and species phylogeny determination.

Trees can be constructed from protein or DNA sequences.

Various algorithms exist for computing phylogenetic trees such as fastDNAml and branch and bound algorithm with maximum parsimony.

Phylogenetic tree calculation has high computing demands for large datasets.
Experiments in Phylogenetic Tree Calculation


- European DataGrid used to perform fastDNAml phylogenetic algorithm.
- Algorithm used to create tree for 22 DNA sequences.
- “Bootstrapping” for reliability, where a large number of independent tree calculations are performed on different grid nodes and the results are merged into the final bootstrapped tree.
- Up to 1000 bootstraps tested, with a granularity of 50 for a bootstrap of 1000.
- Speedup of 14 achieved relative to execution on a single machine for 1000 bootstraps.
Experiments in Phylogenetic Tree Calculation
Experiments in Phylogenetic Tree Calculation

Shaaban, M., Howard, C., and Buckley, L. “Parallelization of a Maximum Parsimony Branch and Bound Algorithm for Phylogenetic Inference,” (submitted).

- Parallelization of branch and bound algorithm with maximum parsimony on a 30-node cluster of AMD Opteron Linux workstations.
- Algorithm used to create tree for 11 DNA sequences
- Root dynamically distributes workload (trees) over nodes with a grain size of taxa-level 7
- Speedup of 20 obtained for 30 nodes
Experiments in Phylogenetic Tree Calculation
Sequence Alignment

- Sequence alignment is the base by base matching of a query sequence (protein or DNA) against another sequence or database of sequences.
- Longer query sequences and larger databases demand significant processing power.
- Database to database alignments also possible and more computationally intensive.
- Sequence alignment algorithms include Smith-Waterman and FASTA.
Experiments in Sequence Alignment


- The SSEARCH program, an implementation of the Smith-Waterman algorithm for pair-wise sequence alignment, was parallelized.
- Poor performance of SSEARCH on uniprocessor machine due to dynamically updated MxN scoring matrix.
- Parallelization performed on PARAM 1000, a cluster of 32 SMP compute nodes, each with 4 Sun UltraSparc-II processors.
- Data parallel implementation utilized where the master splits the database over the slaves.
- 4 DNA sequences of varying lengths were compared against a database containing 115056 sequences.
- Speedups obtained for the query sequences of lengths 400, 800, 1650 and 4023 nucleotides on 64 processors were 24.15, 33.64, 37.62, and 45.01 respectively.
Experiments in Sequence Alignment
Experiments in Sequence Alignment


- Fast Alignment Search Tool (FASTA) is a sequence analysis tool utilizing the FASTA algorithm.
- Parallelization performed on PARAM 1000, a cluster of 32 SMP compute nodes, each with 4 Sun UltraSparc-II processors.
- Data parallel implementation utilized where the master splits the database over the slaves.
- Query DNA sequence length of 148,849 nucleotides compared against database of 129,967 sequences on 2 to 64 processors.
- Speedup of 44 was seen with 64 processors.
Experiments in Sequence Alignment
Protein Folding

- Protein Folding problem is to study how a protein folds dynamically to its 3-d structure.
- The research focuses on:
  1. Protein structure prediction
  2. Study of protein folding process
- Determining a protein’s structure is important because its function is dependent on it.
- Several computational approaches:
  - Lattice models
  - Monte Carlo method
  - Molecular dynamics
  - Probabilistic Roadmap Method
  - Statistical mechanical models
- It is essentially a search for the best fitting molecular state and requires a lot of accurate and long simulations.
The approach is based on Probabilistic Roadmap Method.
Sequential folding algorithm pseudocode is given.
The sequential FOR LOOP in line 2 can be replaced with a parallel version, various nodes can be divided among processors.
Here, PRM is applied to proteins by replacing the collision-free environment with a potential energy.
The research was performed on a dedicated Linux cluster.

Input: The protein’s native state q_{native}.
Output: A roadmap R of the protein’s potential energy landscape.
1: Generate n nodes biased toward q_{native}.
2: for each q \in R do
3: Let N_k(q) be the k closest neighbors of q.
4: for each q \in N_k(q) do
5: if the local planner can find a path between q and q then
6: Add the edge (q, q) to R.
7: end if
8: end for
9: end for
10: return R.
Speedup on the Linux Cluster
Experiment 2 – Blue Gene

- **Computational Approach**: atomistic simulation in explicit solvents
- Blue Gene system uses 32x32x32 cube of building blocks with each building block consisting of a chip that integrates multiple processors, memory and communication logic – also known as “cellular” architecture
- Intra-processor communication -- shared memory
- Inter-processor communication – Message Passing
- Petaflops is achieved through the large number of replicated building blocks and each processor is capable of executing 1 G flop/s (assuming 2 flops in each cycle with 500 MHz clock rate)
- Challenges in the project:
  1. to understand how to extract the maximal amount of concurrency from the application
  2. Partition of the application in the parallel atomistic simulation
- **Performance Analysis**
  If successfully built this project will offer a super computing power predicted at a speed 1 petaflop/s
RNA Folding

- Problem: Predict the structure of a given RNA molecule.
- RNA are biopolymers formed from a sequence of nucleic acid; it forms a 3-d shape when the sequence folds back upon itself.
- The final 3-d structure largely determines the function the RNA molecule perform.
- Key objective is to develop methods for Computationally predicting the structure of a given molecule.
- Algorithms which have been used for RNA structure prediction are dynamic programming, comparative methods, kinetic folding and evolutionary algorithms.
P - RnaPredict

K. Wiese, Andrew Hendriks, Alain Deschenes and Belgacem Ben Youssef; “A Parallel Evolutionary Algorithm”, IEEE transaction on nanobioscience, vol 4, no.3 Sept. 2005

- RnaPredict is based on Genetic Algorithm
- The parallel version is named P – RnaPredict and is modified to support SIMD parallel programming paradigm and communications was based on MPI standard.
- P – RnaPredict was implemented on 128 – node Beowulf Cluster
- PRNG – Pseudorandom Random Number Generators for making coarse grained decisions like initialization of random population, random selection of parents, determining the random order of nodes of migration
- Secondary Structure Comparison
- The prediction accuracy of P – RnaPredict is about 89.2%

The Pseudocode for the distributed GA model:

1. Initialize random population
2. Evaluate fitness of individuals
3. For all NUM_GENERATIONS
4. For all DEME_COUNT
5. Select Parents in deme
6. Reproduce by crossover in deme
7. Mutate
8. Apply Elitism in deme
9. If MIGRATION_INTERVAL, migrate
Parallel GA results using three different PRNGs

<table>
<thead>
<tr>
<th>Demes Size</th>
<th>$P_{m}$</th>
<th>Demes Count</th>
<th>PRNG</th>
<th>Avg. $\Delta G$</th>
<th>Avg. Base Pair %</th>
<th>Best Base Pair %</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>0.25</td>
<td>10</td>
<td>DC</td>
<td>-200.61</td>
<td>25.09</td>
<td>41.30</td>
</tr>
<tr>
<td><strong>70</strong></td>
<td><strong>0.25</strong></td>
<td><strong>10</strong></td>
<td>RAND</td>
<td><strong>-199.65</strong></td>
<td><strong>24.32</strong></td>
<td><strong>48.55</strong></td>
</tr>
<tr>
<td>70</td>
<td>0.25</td>
<td>10</td>
<td>MCG</td>
<td>-198.76</td>
<td>24.44</td>
<td>40.57</td>
</tr>
<tr>
<td>70</td>
<td>0.8</td>
<td>10</td>
<td>DC</td>
<td>-204.17</td>
<td>26.28</td>
<td>47.82</td>
</tr>
<tr>
<td>70</td>
<td>0.8</td>
<td>10</td>
<td>RAND</td>
<td>-203.66</td>
<td>27.58</td>
<td>46.37</td>
</tr>
<tr>
<td><strong>70</strong></td>
<td><strong>0.8</strong></td>
<td><strong>10</strong></td>
<td>MCG</td>
<td><strong>-203.14</strong></td>
<td><strong>29.66</strong></td>
<td><strong>44.92</strong></td>
</tr>
<tr>
<td>50</td>
<td>0.25</td>
<td>14</td>
<td>RAND</td>
<td>-200.64</td>
<td>27.19</td>
<td>44.20</td>
</tr>
<tr>
<td>50</td>
<td>0.25</td>
<td>14</td>
<td>DC</td>
<td>-199.05</td>
<td>24.42</td>
<td>41.30</td>
</tr>
<tr>
<td>50</td>
<td>0.25</td>
<td>14</td>
<td>MCG</td>
<td>-198.83</td>
<td>26.81</td>
<td>41.30</td>
</tr>
<tr>
<td><strong>50</strong></td>
<td><strong>0.8</strong></td>
<td><strong>14</strong></td>
<td>MCG</td>
<td><strong>-207.08</strong></td>
<td><strong>27.75</strong></td>
<td><strong>48.55</strong></td>
</tr>
<tr>
<td>50</td>
<td>0.8</td>
<td>14</td>
<td>RAND</td>
<td>-202.93</td>
<td>26.64</td>
<td>45.65</td>
</tr>
<tr>
<td>50</td>
<td>0.8</td>
<td>14</td>
<td>DC</td>
<td>-199.23</td>
<td>22.89</td>
<td>38.40</td>
</tr>
</tbody>
</table>
Challenges & Future Work

- Need for more biological data and more accurate data.
- Better (parallel) algorithms.
- Improved computational performance.
- World Wide Grid?
Questions?
References


Shaaban, M., Howard, C., and Buckley, L. “Parallelization of a Maximum Parsimony Branch and Bound Algorithm for Phylogenetic Inference,” (submitted).


SHARCNET, [www.sharcnet.ca/](http://www.sharcnet.ca/)


European DataGrid, [www.cern.ch/eu-datagrid](http://www.cern.ch/eu-datagrid)