Green Destiny + mpiBLAST = Bioinfomastics

Wu-chun Feng  
feng@lanl.gov

For more on Green Destiny, go to http://sss.lanl.gov  
For more on mpiBLAST, go to http://mpiblast.lanl.gov  

10th International Conference on Parallel Computing (ParCo’03), Dresden, Germany, 3 September 2003  
Los Alamos Unlimited Release: LA-UR-03-6651
Green Destiny + mpiBLAST = Bioinfomagic

Wu-chun Feng
feng@lanl.gov

For more on Green Destiny, go to http://sss.lanl.gov
For more on mpiBLAST, go to http://mpiblast.lanl.gov
The Components of “Bioinfomagic”

- **Green Destiny**
  - A 240-node supercomputer in a “telephone booth”
    - Footprint: 6 square feet (or 0.55 square meters).
    - Power Consumption: 3.2 kW.

- **mpiBLAST**
  - An open-source parallelization of BLAST that achieves super-linear speed-up.
“Bioinfomagic” Outline

- **Green Destiny**
  - Problem Statement
  - Where is Supercomputing?
  - “Supercomputing in Small Spaces” Project
  - Experimental Results
  - Inspiration for mpiBLAST

- **mpiBLAST**
  - All About BLAST
    - What? How To Use?
  - Motivation
  - Uncovering the Parallelism
  - Algorithm & Implementation
  - Experimental Results
  - Conclusion
Problem Statement

• Operating Environment
  - 85-90°F (30-32°C) warehouse at 7,400 feet (2195 meters) above sea level.
  - No air conditioning, no air filtration, no raised floor, and no humidifier/dehumidifier.

• Computing Requirement
  - Parallel computer to enable high-performance network research in simulation and implementation.

• Old Solution: Traditional Cluster
  - 100-processor cluster computer that failed weekly in the above operating environment.

• New Solution: Low-Power, Always-Available Cluster
  - A 240-processor cluster in six square feet → Green Destiny
Where is Supercomputing?
(Picture Sources: Thomas Sterling, Caltech & NASA JPL and Wu Feng, LANL)

We have spent decades focusing on performance, performance, performance.
Where is Supercomputing?

- Top 500 Supercomputer List
  - Formula One Racecars of Computing

- Benchmark
  - LINPACK / LAPACK: Linear algebra.
  - Evaluation Metric: \textit{Performance} (i.e., Speed)
    - Floating-Operations Per Second (flops)
    - Example: Japanese Earth Simulator @ 35 Tflops.

- Emergence of Beowulf Clusters
  - New Evaluation Metric: \textit{Price/Performance}
    - Acquisition Cost / Floating-Operations Per Second (flops)
Where is Supercomputing?

Architectures from the TOP500 Supercomputer List

SIMD
Cluster
Constellation
MPP
SMP
Single Processor
The Need for New Evaluation Metrics

• Analogy: Buying a car. Which metric(s) to use?
  - Price/Performance: Ford Mustang GTO.
  - Fuel Efficiency: Honda Insight.
  - Reliability: Toyota Camry.
  - Storage: Honda Odyssey.
  - Off-Road Efficiency: Jeep Cherokee.
  - All-Around: Volvo XC90.

• So many metrics to evaluate a car ... why not to evaluate a supercomputer?

• But which metrics?
Where is Supercomputing?
(Picture Sources: Thomas Sterling, Caltech & NASA JPL and Wu Feng, LANL)

We need new metrics to evaluate efficiency, reliability, and availability (ERA) as they will be the key issues of this decade.
### Why Metrics for ERA?

<table>
<thead>
<tr>
<th>Service</th>
<th>Cost of One Hour of Downtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brokerage Operations</td>
<td>$6,450,000</td>
</tr>
<tr>
<td>Credit Card Authorization</td>
<td>$2,600,000</td>
</tr>
<tr>
<td>Amazon.com</td>
<td>$275,000</td>
</tr>
<tr>
<td>eBay</td>
<td>$225,000</td>
</tr>
<tr>
<td>Package Shipping Services</td>
<td>$150,000</td>
</tr>
<tr>
<td>Home Shopping Channels</td>
<td>$139,000</td>
</tr>
</tbody>
</table>

Sources:
1. Contingency Planning Research, Inc.
“Bioinfomagic” Outline

• Green Destiny
  - Problem Statement
  - Where is Supercomputing?
  - “Supercomputing in Small Spaces” Project
  - Experimental Results
  - Inspiration for mpiBLAST

• mpiBLAST
  - All About BLAST
    • What? How To Use?
  - Motivation
  - Uncovering the Parallelism
  - Algorithm & Implementation
  - Experimental Results
  - Conclusion
"Supercomputing in Small Spaces"

- **Project Initiation:** Oct. 2001.
  - 24-node MetaBlade (5.25” x 19” x 25”), Nov. 2001.

- **Project Goals**
  - Sustainable production computing in a “hostile” environment.
  - Determine future directions in *efficient* high-performance computing.

- **Not a “replacement”** for traditional clusters or supercomputers *(which generally require special infrastructure to house)*, but
  - A complementary solution, particularly for those who are space-, power-, or budget-constrained, e.g., bioinformatics and pharmaceutical companies.
  - Potentially a first step in the “right direction” for future supercomputing.
New Evaluation Metrics for ERA

• **Total Price/Performance Ratio (ToPPeR)**
  - Price is more than the *cost of acquisition* (a la traditional “price/performance” ratio).
  - “Total Price” is the *total cost of ownership (TCO)* of a cluster.

• **Performance/Power Ratio → “Power Efficiency”**
  - How efficiently does a computing system use energy?
  - Performance has increased by 2000 since the Cray C90; performance/watt has only increased by 300.
  - How does this affect reliability and availability?
    - Higher Power Dissipation $\propto$ Higher Temperature $\propto$ Higher Failure Rate

• **Performance/Space Ratio → “Space Efficiency” or “Compute Density”**
  - How efficiently does a computing system use space?
  - Performance has increased by 2000 since the Cray C90; performance/sq. ft. has only increased by 65.
Quantifying TCO?

- Why is TCO hard to quantify?
  - Components
    - Acquisition + Administration + **Power** + Downtime + **Space**
  - Institution-Specific
    - Too Many Hidden Costs
  - Traditional Focus: Acquisition (i.e., equipment cost)
    - Cost Efficiency: Price/Performance Ratio
  - New *Quantifiable* Efficiency Metrics
    - “Power” Efficiency: Performance/Power Ratio
    - “Space” Efficiency: Performance/Space Ratio
Moore's Law for Power

**Chip Maximum Power in watts/cm²**

- **I386** - 1 watt
- **I486** - 2 watts
- **Pentium** - 14 watts
- **Pentium Pro** - 30 watts
- **Pentium II** - 35 watts
- **Pentium III** - 35 watts
- **Pentium 4** - 75 watts
- **Itanium** - 130 watts

**Year**
- 1985
- 1995
- 2001

**Not too long to reach**
- **Heating Plate**
- **Nuclear Reactor**

Power, Temperature, Reliability

• What’s wrong with high power?
  - Costs $$$ to power and cool such a system.
  - Causes reliability problems. Why?
    • Higher power implies higher temperatures.

• Arrhenius’ Equation
  (circa 1890s in chemistry → circa 1980s in computer & defense industries)
  - As temperature increases by 10° C ...
    • The failure rate of a system doubles.
    • The reliability of a system is cut in half.
  - Twenty years of unpublished empirical data.
“Green Destiny” Bladed Beowulf

- A 240-Node Beowulf in One Cubic Meter
- Each Node
  - 667-MHz Transmeta TM5600 CPU with HP-CMS
    - Upgraded to 1-GHz Transmeta TM5800 CPUs
  - 640-MB RAM
  - 20-GB hard disk
  - 100-Mb/s Ethernet (up to 3 interfaces)
- Total
  - 160 Gflops peak (240 Gflops with upgrade)
  - 240 nodes
  - 150 GB of RAM (expandable to 276 GB)
  - 4.8 TB of storage (expandable to 38.4 TB)
## Treecode Benchmark for n-Body

<table>
<thead>
<tr>
<th>Site</th>
<th>Machine</th>
<th>CPUs</th>
<th>Gflops</th>
<th>Mflops/CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANL</td>
<td>Green Destiny+</td>
<td>212</td>
<td>58.00</td>
<td>273.6</td>
</tr>
<tr>
<td>NERSC</td>
<td>IBM SP-3</td>
<td>256</td>
<td>57.70</td>
<td>225.0</td>
</tr>
<tr>
<td>LANL</td>
<td>SGI O2K</td>
<td>64</td>
<td>13.10</td>
<td>205.0</td>
</tr>
<tr>
<td>LANL</td>
<td>Green Destiny</td>
<td>212</td>
<td>38.90</td>
<td>183.5</td>
</tr>
<tr>
<td>SC ’01</td>
<td>MetaBlade2</td>
<td>24</td>
<td>3.30</td>
<td>138.0</td>
</tr>
<tr>
<td>LANL</td>
<td>Avalon</td>
<td>128</td>
<td>16.16</td>
<td>126.0</td>
</tr>
<tr>
<td>LANL</td>
<td>Loki</td>
<td>16</td>
<td>1.28</td>
<td>80.0</td>
</tr>
<tr>
<td>NASA</td>
<td>IBM SP-2</td>
<td>128</td>
<td>9.52</td>
<td>74.4</td>
</tr>
<tr>
<td>SC ’96</td>
<td>Loki+Hyglac</td>
<td>32</td>
<td>2.19</td>
<td>68.4</td>
</tr>
<tr>
<td>SNL</td>
<td>ASCI Red</td>
<td>6800</td>
<td>464.90</td>
<td>68.4</td>
</tr>
<tr>
<td>CalTech</td>
<td>Naegling</td>
<td>96</td>
<td>5.67</td>
<td>59.1</td>
</tr>
<tr>
<td>NRL</td>
<td>TMC CM-5E</td>
<td>256</td>
<td>11.57</td>
<td>45.2</td>
</tr>
</tbody>
</table>
Parallel Computing Platforms  
(for “Apples-to-Oranges” Comparison)

- **Avalon (1996)**  
  - 140-CPU Traditional Beowulf Cluster
- **ASCI Red (1996)**  
  - 9632-CPU MPP
- **ASCI White (2000)**  
  - 512-Node (8192-CPU) Cluster of SMPs
- **Green Destiny (2002)**  
  - 240-CPU Bladed Beowulf Cluster
## Parallel Computing Platforms Running the N-body Code

<table>
<thead>
<tr>
<th>Machine</th>
<th>Avalon</th>
<th>Beowulf</th>
<th>ASCI Red</th>
<th>ASCI White</th>
<th>Green Destiny+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>1996</td>
<td>1996</td>
<td>2000</td>
<td>2002</td>
<td></td>
</tr>
<tr>
<td>Performance (Gflops)</td>
<td>18</td>
<td>600</td>
<td>2500</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>Area (ft²)</td>
<td>120</td>
<td>1600</td>
<td>9920</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Power (kW)</td>
<td>18</td>
<td>1200</td>
<td>2000</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>DRAM (GB)</td>
<td>36</td>
<td>585</td>
<td>6200</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>Disk (TB)</td>
<td>0.4</td>
<td>2.0</td>
<td>160.0</td>
<td>4.8</td>
<td></td>
</tr>
<tr>
<td>DRAM density (MB/ft²)</td>
<td>300</td>
<td>366</td>
<td>625</td>
<td>25000</td>
<td></td>
</tr>
<tr>
<td>Disk density (GB/ft²)</td>
<td>3.3</td>
<td>1.3</td>
<td>16.1</td>
<td>800.0</td>
<td></td>
</tr>
<tr>
<td>Perf/Space (Mflops/ft²)</td>
<td>150</td>
<td>375</td>
<td>252</td>
<td>9667</td>
<td></td>
</tr>
<tr>
<td>Perf/Power (Mflops/watt)</td>
<td>1.0</td>
<td>0.5</td>
<td>1.3</td>
<td>11.6</td>
<td></td>
</tr>
</tbody>
</table>
## Parallel Computing Platforms
### Running the N-body Code

<table>
<thead>
<tr>
<th>Machine</th>
<th>Avalon Beowulf</th>
<th>ASCI Red</th>
<th>ASCI White</th>
<th>Green Destiny+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>1996</td>
<td>1996</td>
<td>2000</td>
<td>2002</td>
</tr>
<tr>
<td>Performance (Gflops)</td>
<td>18</td>
<td>600</td>
<td>2500</td>
<td>58</td>
</tr>
<tr>
<td>Area (ft²)</td>
<td>120</td>
<td>1600</td>
<td>9920</td>
<td>6</td>
</tr>
<tr>
<td>Power (kW)</td>
<td>18</td>
<td>1200</td>
<td>2000</td>
<td>5</td>
</tr>
<tr>
<td>DRAM (GB)</td>
<td>36</td>
<td>585</td>
<td>6200</td>
<td>150</td>
</tr>
<tr>
<td>Disk (TB)</td>
<td>0.4</td>
<td>2.0</td>
<td>160.0</td>
<td>4.8</td>
</tr>
<tr>
<td>DRAM density (MB/ft²)</td>
<td>300</td>
<td>366</td>
<td>625</td>
<td>25000</td>
</tr>
<tr>
<td>Disk density (GB/ft²)</td>
<td>3.3</td>
<td>1.3</td>
<td>16.1</td>
<td>800.0</td>
</tr>
<tr>
<td>Perf/Space (Mflops/ft²)</td>
<td>150</td>
<td>375</td>
<td>252</td>
<td>9667</td>
</tr>
<tr>
<td>Perf/Power (Mflops/watt)</td>
<td>1.0</td>
<td>0.5</td>
<td>1.3</td>
<td>11.6</td>
</tr>
</tbody>
</table>
## Parallel Computing Platforms
### Running the N-body Code

<table>
<thead>
<tr>
<th>Machine</th>
<th>Avalon Beowulf</th>
<th>ASCI Red</th>
<th>ASCI White</th>
<th>Green Destiny+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>1996</td>
<td>1996</td>
<td>2000</td>
<td>2002</td>
</tr>
<tr>
<td>Performance (Gflops)</td>
<td>18</td>
<td>600</td>
<td>2500</td>
<td>58</td>
</tr>
<tr>
<td>Area (ft²)</td>
<td>120</td>
<td>1600</td>
<td>9920</td>
<td>6</td>
</tr>
<tr>
<td>Power (kW)</td>
<td>18</td>
<td>1200</td>
<td>2000</td>
<td>5</td>
</tr>
<tr>
<td>DRAM (GB)</td>
<td>36</td>
<td>585</td>
<td>6200</td>
<td>150</td>
</tr>
<tr>
<td>Disk (TB)</td>
<td>0.4</td>
<td>2.0</td>
<td>160.0</td>
<td>4.8</td>
</tr>
<tr>
<td>DRAM density (MB/ft²)</td>
<td>300</td>
<td>366</td>
<td>625</td>
<td>25000</td>
</tr>
<tr>
<td>Disk density (GB/ft²)</td>
<td>3.3</td>
<td>1.3</td>
<td>16.1</td>
<td>800.0</td>
</tr>
<tr>
<td>Perf/Space (Mflops/ft²)</td>
<td>150</td>
<td>375</td>
<td>252</td>
<td>9667</td>
</tr>
<tr>
<td>Perf/Power (Mflops/watt)</td>
<td>1.0</td>
<td>0.5</td>
<td>1.3</td>
<td>11.6</td>
</tr>
</tbody>
</table>
Green Destiny vs. Earth Simulator: LINPACK Benchmark

<table>
<thead>
<tr>
<th>Machine</th>
<th>Green Destiny+</th>
<th>Earth Simulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>2002</td>
<td>2002</td>
</tr>
<tr>
<td>LINPACK Performance (Gflops)</td>
<td>101</td>
<td>35,860</td>
</tr>
<tr>
<td>Area (ft²)</td>
<td>6</td>
<td>70,290</td>
</tr>
<tr>
<td>Power (kW)</td>
<td>5</td>
<td>7,000</td>
</tr>
<tr>
<td>Cost Efficiency ($/Mflop)</td>
<td>3.35</td>
<td>11.15</td>
</tr>
<tr>
<td>Space Efficiency (Mflops/ft²)</td>
<td>16,833</td>
<td>510</td>
</tr>
<tr>
<td>Power Efficiency (Mflops/watt)</td>
<td>20.00</td>
<td>5.12</td>
</tr>
</tbody>
</table>

Disclaimer: This is not exactly a fair comparison. Why?

(1) LINPACK performance is extrapolated for Green Destiny+.
(2) Use of area and power does not scale linearly.
(3) Goals of the two machines are different.
“Bioinfomagic” Outline

- **Green Destiny**
  - Problem Statement
  - Where is Supercomputing?
  - “Supercomputing in Small Spaces” Project
  - Experimental Results
  - Inspiration for mpiBLAST

- **mpiBLAST**
  - All About BLAST
    - What? How To Use?
  - Motivation
  - Uncovering the Parallelism
  - Algorithm & Implementation
  - Experimental Results
  - Conclusion
Q&A Exchange with Pharmaceutical Companies

- Pharmaceutical: “Can you get the same type of results for bioinformatics applications?”
- Wu: “What is your primary application?”
- Pharmaceutical: “BLAST …”

J. Craig Venter in GenomeWeb on Oct. 16, 2002.

“… to build something that is replicable so any major medical center around the world can have a chance to do the same level of computing … interested in IT that doesn’t require massive air conditioning. The room at Celera cost $6M before you put the computer in. [Thus, I am] looking at these new green machines being considered at the DOE that have lower energy requirements” & therefore produce less heat.
What is BLAST?

• **Basic Local Alignment Sequence Tool**
  - Ubiquitous sequence database search tool used in molecular biology.

• **Overall Approach**
  - *Given a query DNA or amino-acid (AA) sequence,*
    - BLAST finds similar sequences in the database.
    - BLAST reports the statistical significance of similarities between the query and database.
How Can BLAST Be Used?

- Newly sequenced genomes are typically BLAST-searched against a database of known genes.
  - Similar sequences may have similar functions in a new organism.
- BLAST can be used to help identify coding regions in eukaryotes.

<table>
<thead>
<tr>
<th>Search Name</th>
<th>Query Type</th>
<th>Database Type</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>blastn</td>
<td>Nucleotide</td>
<td>Nucleotide</td>
<td>None</td>
</tr>
<tr>
<td>tblastn</td>
<td>Peptide</td>
<td>Nucleotide</td>
<td>Database</td>
</tr>
<tr>
<td>blastx</td>
<td>Nucleotide</td>
<td>Peptide</td>
<td>Query</td>
</tr>
<tr>
<td>blastp</td>
<td>Peptide</td>
<td>Peptide</td>
<td>None</td>
</tr>
<tr>
<td>tblastx</td>
<td>Nucleotide</td>
<td>Nucleotide</td>
<td>Query and Database</td>
</tr>
</tbody>
</table>
Motivation: BLAST Trends

• Widely used, i.e., more than 100,000 queries per day on the public NCBI servers alone.
• Up to 95% of compute cycles spent on BLAST.
• Computationally demanding.
• Sequence databases growing exponentially.
Motivation: BLAST with Low Memory

- Standard BLAST running on a system with 128 MB of memory.
Existing Parallelizations of BLAST

- **Multithreaded**
  - NCBI-BLAST.

- **Query Segmentation**
  - Numerous free, open-source implementations exist.

- **Database Segmentation**
  - TurboBLAST from TurboWorx, Inc.
    - Commercial, closed source, expensive $$
    - Only linear speed-up.
  - parallelblast at Caltech
    - Free but not directly integrated with NCBI toolbox.
Query:
> Perilla frutescens mRNA

CATCTACTCAAAATTAAGAAATAGATAGAAATGGTTACGAGTGCAATGGGTCCAAGCCCGGGTGGAGG
AACTGGCCCGAAGCGGACTCGACACGATCCCAAAAGTATACGTGCGGCCCGAAGAGCACCTGAAAA

Database:
> gi|3123744|dbj|AB013447.1|AB013447
GCCTCAAACAGTTTAATTTTCTTCAAACCTAGTTTTTTTTTGTTTTTAGTTGTATCCACGGAAGAGAGA
GAAAAATGTGGGAATTTTTCAGCGGACGTATAGTATCACGCTGCCGAAGAGACGTGGCTGCCCGGGAACC

> gi|221778|dbj|D00026.1|HS2HSV2P4 Herpes simplex virus type 2 gene
TTTTTACTAGGAGTATCCCCGCTCCCCGTTTAATCCCTGCGGGCGGGCGTGTTGGGAAGGATGGCTGGTGATTTG
GCGGCCCCGAGGCAGCAGCAGCATTTAAGGAGTCGCGCCCGCCCGAAGCTCGGTCTCTGGGTGACCTTGG
GCGCGCCGCTAGCTAGCTCCGATCTGCCCGACCACCGCCTGCTGCCACCGAACGTG

> gi|7328961|dbj|AB032155.1|AB032154S2 Homo sapiens PGFS gene
TTTTTTTCTTGGATCTAAATCTAATCCAAACATCAACGATGACATTTCCTTTGAAGATGCTTTTGTCTTT
TCAGACTTGGCTCCGAGTCTCCTGACCACATTTTCTTTCTGCTGCAACAAATCGCAGATCGAAAGCCACCTCTA

Match:
GTGGGTATTGGCGGCCCGAAGGGCCCGGCCCGC
++ - - +++++++ ++ +
AAGACTACGTGCGGCGCCGAGAGGACACCTGAAAA
Query Segmentation

Queries

>Perilla Frutescens CDS 0001
TTGGTATCCACGGAAGAGAGAGAAAATGTTGGGAATTTTCAGCGGAC
GTATAGTATCATTTGCCGGAAGAGCTGGTGGCTGCCGGGAACC

>Perilla Frutescens CDS 0002
GGAGGGTGGCTGGTGATTTGGCGGCCCGACCGATCTGCCCCGACC
GACGGCTCTCGCCAACCCGAACATGTGATAGAAAGGAQQQQQQQQ

>Perilla Frutescens CDS 0003
TTTTTTCTTGTGCTGAAATCTATCCAAACATACCCAGTCTCCACGA
GTCCTTGACAAATCTTGTGCTGGCAAACTCGAAGCCCAAAGGC

Database

>gi|3123744|dbj|AB013447.1|AB013447
TTGGTATCCACGGAAGAGAGAGAAAATGTTGGGAATTTTCAGCGGAC
GTATAGTATCATTTGCCGGAAGAGCTGGTGGCTGCCGGGAACC

>gi|221778|dbj|D00026.1|HS2HSV2P4
GGAGGGTGGCTGGTGATTTGGCGGCCCGACCGATCTGCCCCGACC
GACGGCTCTCGCCAACCCGAACATGTGATAGAAAGGAQQQQQQQQ

>gi|7328961|dbj|AB032155.1|AB032154S2
TTTTTTCTTGTGCTGAAATCTATCCAAACATACCCAGTCTCCACGA
GTCCTTGACAAATCTTGTGCTGGCAAACTCGAAGCCCAAAGGC
Avoiding Extra Disk I/O

Approaches

1. Use database scan sharing (i.e., query merging)
2. Buy expensive hardware with a large shared-memory model.
3. Buy a cluster and utilize the aggregate memory of the cluster.

In general, clusters provide a better performance/price ratio because they utilize commodity technology.
Database Segmentation

Queries

>Perilla Frutescens CDS 0001
TTGGTATCCACGGAAGAGAGAGAAAATGTTGGGAATTTTCAGCGGAC
GTATAGTATCATATGCGGAAAGAGCTGGTGGCTGCCGGGAACC

>Perilla Frutescens CDS 0002
GGAGGGTGGCTGGTGGGTATTGGCGGCCCGACCGATCTGCCCCGACC
GACGGCTCCTGCAACCCGAACATGTGATAGAAAGG

>Perilla Frutescens CDS 0003
TTTTTTTCTTGATGCTGAAATCTATCCAAACATCACCAGTCCTCACGA
GTCCTTGACAAAATTTTTGCGCAACATCTGAAGCCTAAAAGGC

Database

>gi|3123744|dbj|AB013447.1|AB013447
TTGGTATCCACGGAAGAGAGAGAAAATGTTGGGAATTTTCAGCGGAC
GTATAGTATCATATGCGGAAAGAGCTGGTGGCTGCCGGGAACC

>gi|221778|dbj|D00026.1|HS2HSV2P4
GGAGGGTGGCTGGTGGGTATTGGCGGCCCGACCGATCTGCCCCGACC
GACGGCTCCTGCAACCCGAACATGTGATAGAAAGG

>gi|7328961|dbj|AB032155.1|AB032154S2
TTTTTTTCTTGATGCTGAAATCTATCCAAACATCACCAGTCCTCACGA
GTCCTTGACAAAATTTTTGCGCAACATCTGAAGCCTAAAAGGC

Worker nodes
“Bioinfomagic” Outline

• Green Destiny
  - Problem Statement
  - Where is Supercomputing?
  - “Supercomputing in Small Spaces” Project
  - Experimental Results
  - Inspiration for mpiBLAST

• mpiBLAST
  - All About BLAST
    • What? How To Use?
  - Motivation
  - Uncovering the Parallelism
  - Algorithm & Implementation
  - Experimental Results
  - Conclusion
mpiBLAST Algorithm: Fragmenting the Database

- **mpiformatdb**
  - Wraps around the standard NCBI formatdb tool.
  - Formats & fragments the database and then copies it to shared storage.
mpiBLAST Algorithm: Querying the Database

Query is tree broadcast to workers
Nodes query DB, generate results
Results sent to master node
Master node merges results
mpiBLAST Algorithm: Assigning Database Fragments

- Master assigns database fragments to workers.
- A worker copies the fragment from shared storage to the local hard drive (if the fragment is not already stored locally).
- The master tries to minimize the number of copy operations and duplicate fragments stored on worker nodes.
mpiBLAST Algorithm: Searching a Database Fragment

1. Worker receives a “fragment assignment” from master and copies the fragment, if necessary.
2. Worker searches the fragment.
3. Worker report results to master.
4. If there is still work to do
   Then master assigns new fragment to the worker.
   Else worker exits.

When all workers have finished, master writes out merged results.
mpiBLAST Implementation

mpiBLAST encapsulates the freely available NCBI C toolbox for
- Formatting the database.
- Executing the actual BLAST algorithm.
- Formatting and writing the results file.

mpiBLAST Specifics
- Programming Language: C++ with MPI.
- Job Scheduling: PBS and Sun Grid Environment.
Measuring the Performance of mpiBLAST

• BLAST Search Performance
  - Depends on many parameters, e.g.,
    • Number and length of queries.
    • Number and size of database entries.
    • Extent of similarity between queries and database entries.

• Goal of mpiBLAST Performance Evaluation
  - Select values that accurately reflect typical BLAST usage patterns.
• We model BLAST as it would be used in a high-throughput sequence annotation pipeline.
• Queries are predicted genes from a newly sequenced bacterial genome, lengths are exponentially distributed with a mean of 747.2.
• Database is the GenBank nt database, sequence lengths are approximately exponentially distributed with a mean of 1370.
mpiBLAST: Low-Memory Performance

- Environment
  - 1, 2, or 4 nodes.
  - Each node w/ dual 550-MHz CPUs and 128-MB memory.
  - Same query and database used.

- Conclusions
  - blastn is I/O bound. Superlinear speed-up possible.
  - tblastx is CPU bound.
mpiBLAST: Performance on Green Destiny

BLAST Run Time for 300-kB Query against nt

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Runtime (s)</th>
<th>Speedup over 1 node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80774.93</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>8751.97</td>
<td>9.23</td>
</tr>
<tr>
<td>8</td>
<td>4547.83</td>
<td>17.76</td>
</tr>
<tr>
<td>16</td>
<td>2436.60</td>
<td>33.15</td>
</tr>
<tr>
<td>32</td>
<td>1349.92</td>
<td>59.84</td>
</tr>
<tr>
<td>64</td>
<td>850.75</td>
<td>94.95</td>
</tr>
<tr>
<td>128</td>
<td>473.79</td>
<td>170.49</td>
</tr>
</tbody>
</table>

The Bottom Line
mpiBLAST reduces search time from 1346 minutes (or 22.4 hours) to under 8 minutes!
Where Does The Time Go?

• **Components of mpiBLAST Execution**
  1. MPI and mpiBLAST initialization
  2. Copying of database fragments
  3. BLAST search
  4. Communication
  5. Result merging and formatting

• **Notes**
  1. “Result merging and formatting” is serial.
  2. mpiBLAST was instrumented with MPE to collect timing statistics for each of the above components.
Where The Time Goes ...
Overhead for Fragmentation

- 10-KB query searched against a 2-GB database.
Efficiency of result formatting

- Independent of the number of processors.
- Variable with the number of database fragments.
Future Work

• R&D and Additional Features
  - Load Balancing.
  - Transparent Fault Tolerance.
  - Query Segmentation.
  - MPI I/O Implementation for I/O.
  - Grid-based mpiBLAST (to be demonstrated at SC 2003).
  - Auto-updating of Database Fragments
• Performance Evaluation
  - Platforms & Associated Optimizations (e.g., w/ and w/o SSE)
    • NCSA TeraGrid Clusters: Platinum (IA-32) and Titan (IA-64)
    • AMD Opteron cluster: Pending
    • LANL Apple G4 and Apple G5 (w/ and w/o Altivec) clusters: Pending
  - Compilers
    • Intel compiler vs. gcc compiler.
  - Profiling NCBI C Toolbox
    • Why the overhead for additional fragments?
    • Replace NCBI BLAST search.
Conclusion

• mpiBLAST Performance
  - Linear speed-up at a minimum.
  - Super-linear speed-up when the database size is larger than a single node’s memory.

• mpiBLAST Impact
  - With only one optimization, i.e., database segmentation, mpiBLAST reduces the search time of a 300-kB query on a 5.1-GB nt database from nearly one day to merely eight minutes.
  - Database segmentation, coupled with query segmentation and load balancing, could reduce search to mere seconds.
mpiBLAST Status

• Research & Development
  - Several mpiBLAST enhancements and optimizations have been implemented but not yet released, e.g., efficient fragment distribution, GUI.

• Impact
  - 4000+ downloads (April 2003 to July 2003)
    • Examples: UIUC/NCSA, Indiana U., UC-Berkeley, and a plethora of pharmaceutical companies.
  - Media interest from bioinformatics and information technology communities.
Relevant Publications

• Green Destiny

• mpiBLAST
Media Coverage

• “LANL Researchers Outfit the 'Toyota Camry' of Supercomputing for Bioinformatics Tasks,” BioInform / GenomeWeb, 2/3/03.
• “Developments to Watch: Innovations,” BusinessWeek, 12/2/02.
• “Craig Venter Goes Shopping for Bioinformatics to Fill His New Sequencing Center,” GenomeWeb, 10/16/02.
• “Bell, Torvalds Usher Next Wave of Supercomputing,” CNN, 5/21/02.
Acknowledgments

• Green Destiny
  - Technical Co-Leads: Mike Warren and Eric Weigle
  - Encouragement: C. Gordon Bell and Linus Torvalds

• mpiBLAST
  - Inspiration: J. Craig Venter
  - Lead Developer & Slide Contributor: Aaron Darling
SUPERCOMPUTING in SMALL SPACES


Wu-chun (Wu) Feng
Research and Development in Advanced Network Technology

http://www.lanl.gov/radiant
Architecture For Our Bladed Beowulf Cluster: Green Destiny

- 24-node RLX 324 chassis
- 24-port Fast Ethernet switch
- 16-port Gigabit Ethernet switch
- 100-Mb/s Fast Ethernet link
RLX System™ 324

- 3U vertical space
- 5.25” x 17.25” x 25.2”
- Two hot-pluggable 450W power supplies
  - Load balancing
  - Auto-sensing fault tolerance
- System midplane
  - Integration of system power, management, and network signals.
  - Elimination of internal system cables.
  - Enabling efficient hot-pluggable blades.
- Network cards
  - Hub-based management.
  - Two 24-port interfaces.

RLX System™ 300ex
- Interchangeable blades
  - Intel, Transmeta, or both.
- Switched-based management
RLX ServerBlade™ 633 (circa 2000)

- Public NIC 33 MHz PCI
- Private NIC 33 MHz PCI
- Management NIC 33 MHz PCI
- Transmeta™ TM5600 633 MHz
- ATA 66
  - 0 or 1 or 2 - 2.5" HDD
  - 10 or 30 GB each
- Code Morphing Software (CMS), 1 MB
- 512KB Flash ROM
- 128MB, 256MB, 512MB DIMM SDRAM PC-133
- 512KB L1 cache, 512KB L2 cache
- LongRun, Northbridge, x86 compatible
- RLX ServerBlade™ 1000t
  - $999

Wu-chun Feng
feng@lanl.gov

10th International Conference on Parallel Computing (ParCo’03)
Transmeta TM5600 CPU: VLIW + CMS

- VLIW Engine
  - Up to four-way issue
  - In-order execution only.
  - Two integer units
  - Floating-point unit
  - Memory unit
  - Branch unit

- VLIW Transistor Count ("Anti-Moore’s Law")
  - \(\frac{1}{4}\) of Intel PIII \(\rightarrow\) \(\sim\) 6x-7x less power dissipation
  - Less power \(\rightarrow\) lower "on-die" temp. \(\rightarrow\) better reliability & availability

- Transforming Transmeta’s CMS into a high-performance CMS (HP-CMS)
Architecture For Our Bladed Beowulf Cluster: Green Destiny

- 24-node RLX 324 chassis
- 24-port Fast Ethernet switch
- 16-port Gigabit Ethernet switch
- 100-Mb/s Fast Ethernet link

Wu-chun Feng
feng@lanl.gov
Low-Power Network Switches

- WWP LE-410: 16 ports of Gigabit Ethernet
- WWP LE-210: 24 ports of Fast Ethernet via RJ-21s
- (Avg.) Power Dissipation / Port: A few watts.
Query Fragmentation Overhead

Sum of NCBI-BLAST execution times when queries are broken into many files:

![Graph showing the sum of NCBI-BLAST execution times over the number of query files.](image)
mpiBLAST in a Nutshell

- BLAST is a widely-used search tool for biological sequence databases.
- Problem: BLAST searches can be slow because large databases are out of core memory.
- Solution: Use the aggregate memory of a cluster!
- mpiBLAST parallelizes BLAST searches using database segmentation.
- Database is fragmented and put on shared storage.
- Workers search fragments and relay results to the master.
Performance in a Nutshell

- Super-linear speedup when the database is out-of-core on a single node.
- Near linear speedup in other cases.
- Efficiency declines when scaled to hundreds of nodes because serial result-merging and output dominates.

Load balancing:
- Coarse grained load-balancing achieved through database segmentation.
- There is heavy overhead in database fragmentation.