COMPUTER & COMPUTATIONAL SCIENCES

Los Alamos National Laboratory RADIANT

www.lanl.gov/radiant

Green Destiny + mpiBLAST = Bioinfomatics

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

NNSA

Los Alamos NATIONAL LABORATORY

COMPUTER & COMPUTATIONAL SCIENCES

Los Alamos National Laboratory RADIANT www.lanl.gov/radiant

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

10th International Conference on Parallel Computing (ParCo'03), Dresden, Germany, 3 September 2003

Los Alamos Unlimited Release: LA-UR-03-6651

NNSA

Los Alamos NATIONAL LABORATORY

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

COMPUTER & COMPUTATIONAL SCIENCES

Where is Supercomputing?

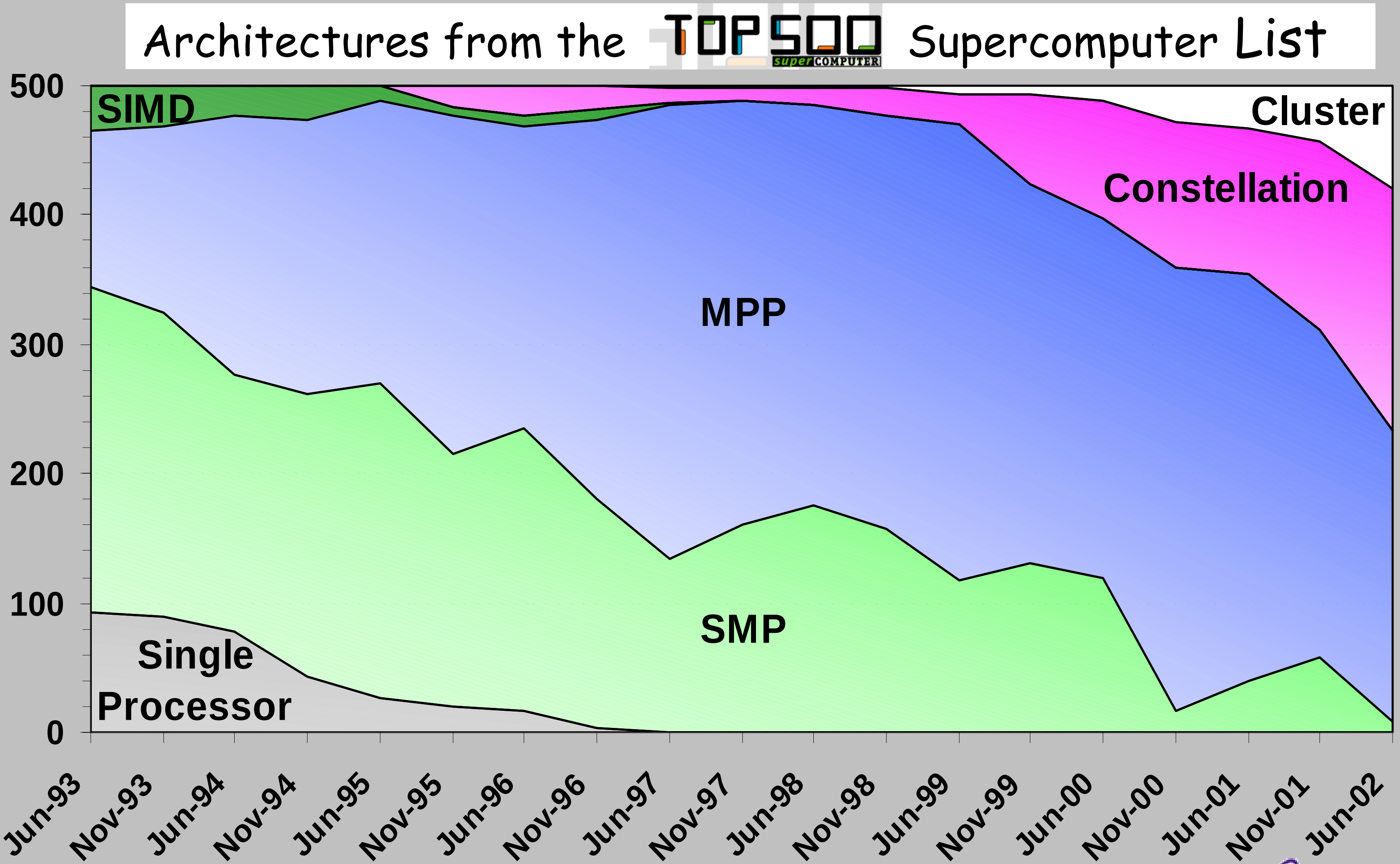
(Picture Sources: Thomas Sterling, Caltech & NASA JPL and Wu Feng, LANL)

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: *Performance* (i.e., Speed)
    - Floating-Operations Per Second (flops)
    - Example: Japanese Earth Simulator @ 35 Tflops.
- Emergence of Beowulf Clusters
  - New Evaluation Metric: *Price/Performance*
    - Acquisition Cost / Floating-Operations Per Second (flops)

Where is Supercomputing?



The Need for *New* Evaluation Metrics

- Analogy: Buying a car. Which metric(s) to use?
  - *Raw Performance:* Ferrari 550.
  - *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?

Service	Cost of One Hour of Downtime
Brokerage Operations	$6,450,000
Credit Card Authorization	$2,600,000
Amazon.com	$275,000
eBay	$225,000
Package Shipping Services	$150,000
Home Shopping Channels	$139,000

Sources:

1. Contingency Planning Research, Inc.
2. "Business Recovery over Wide-Area Networks: Are You Ready?," *AT&T White Paper*, 1999.
3. "The Cost of Downtime," *InternetWeek*, July 30, 1999.

"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.
  - 240-node *Green Destiny*, Apr. 2002.
- 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 α Higher Temperature α 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?

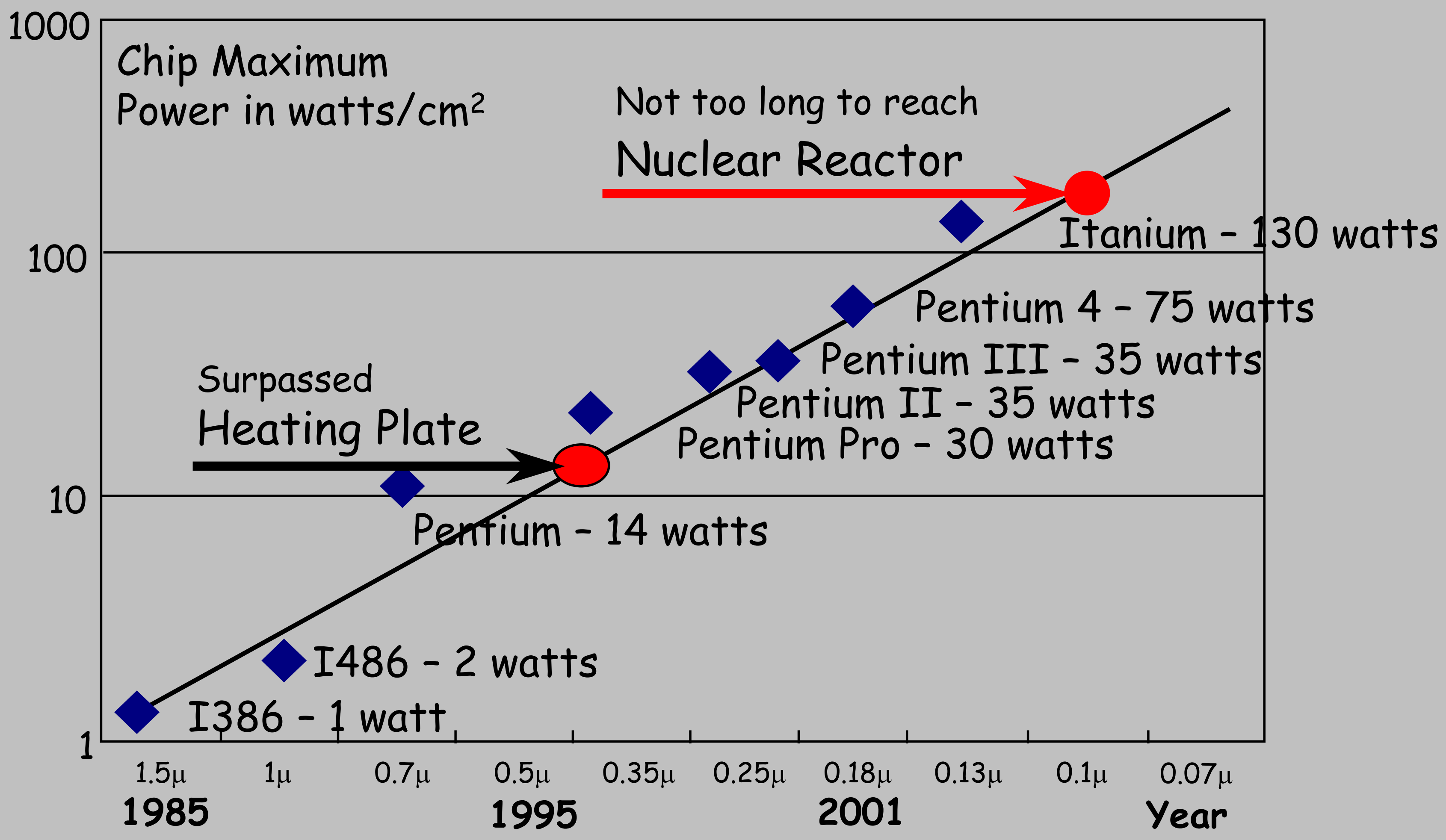
- Why is TCO hard to quantify?
  - Components
    - Acquisition + Administration + Power + Downtime + Space

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



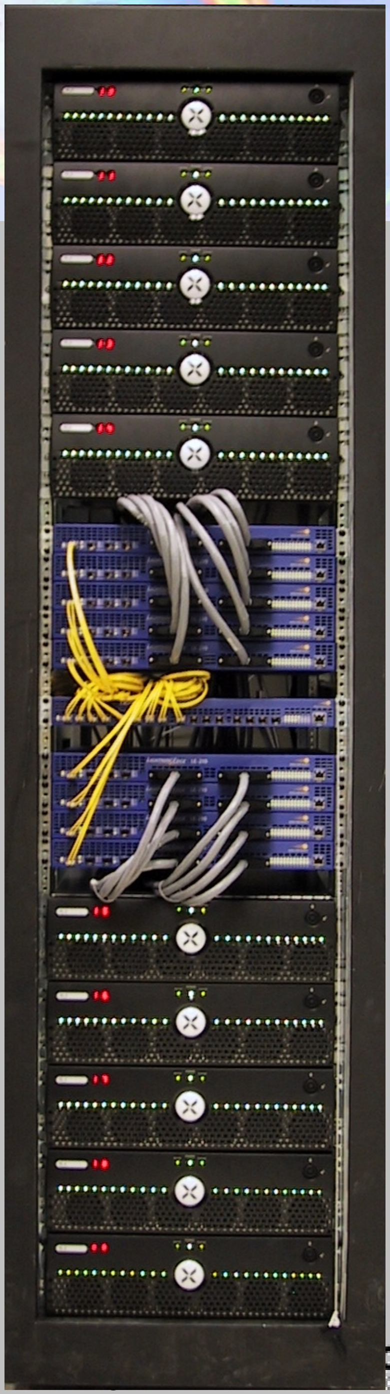
Source: Fred Pollack, Intel. New Microprocessor Challenges in the Coming Generations of CMOS Technologies, MICRO32 and Transmeta.

Power, Temperature, Reliability

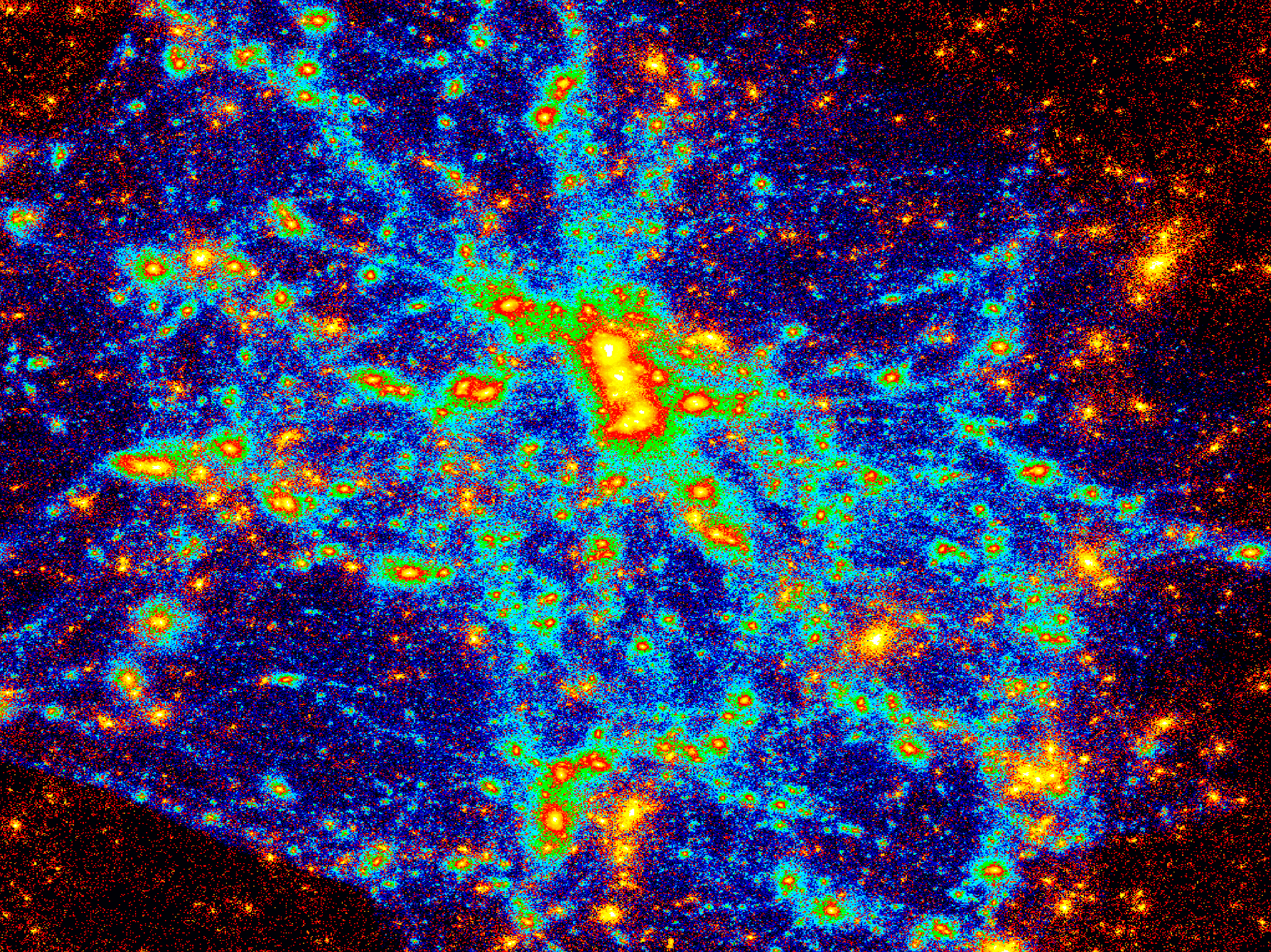
- What's wrong with high power?
  - Costs $$$ to power *and* cool such a system.
  - Causes reliability problems. Why?
    - Higher power implies higher temperatures.
- Arrhenius' Equation
  (circa 1890s in chemistry → circa 1980s in computer & defense industries)
  - As temperature increases by 10° 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)



Wu-chun Feng
feng@lanl.gov



Treecode Benchmark for n-Body

Site	Machine	CPUs	Gflops	Mflops/CPU
LANL	Green Destiny+	**212**	***58.00***	***273.6***
NERSC	IBM SP-3	256	57.70	225.0
LANL	SGI O2K	64	13.10	205.0
LANL	Green Destiny	**212**	**38.90**	**183.5**
SC '01	MetaBlade2	**24**	**3.30**	**138.0**
LANL	Avalon	128	16.16	126.0
LANL	Loki	16	1.28	80.0
NASA	IBM SP-2	128	9.52	74.4
SC '96	Loki+Hyglac	32	2.19	68.4
SNL	ASCI Red	6800	464.90	68.4
CalTech	Naegling	96	5.67	59.1
NRL	TMC CM-5E	256	11.57	45.2

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

Machine	Avalon Beowulf	ASCI Red	ASCI White	Green Destiny+
Year	1996	1996	2000	2002
Performance (Gflops)	18	600	2500	58
Area (ft^2)	120	1600	9920	6
Power (kW)	18	1200	2000	5
DRAM (GB)	36	585	6200	150
Disk (TB)	0.4	2.0	160.0	4.8
DRAM density (MB/ft^2)	300	366	625	25000
Disk density (GB/ft^2)	3.3	1.3	16.1	800.0
Perf/Space ($Mflops/ft^2$)	150	375	252	9667
Perf/Power (Mflops/watt)	1.0	0.5	1.3	11.6

Parallel Computing Platforms Running the N-body Code

Machine	Avalon Beowulf	ASCI Red	ASCI White	Green Destiny+
Year	1996	1996	2000	2002
Performance (Gflops)	18	600	2500	58
Area (ft^2)	120	1600	9920	6
Power (kW)	18	1200	2000	5
DRAM (GB)	36	585	6200	150
Disk (TB)	0.4	2.0	160.0	4.8
DRAM density (MB/ft^2)	300	366	625	25000
Disk density (GB/ft^2)	3.3	1.3	16.1	800.0
Perf/Space ($Mflops/ft^2$)	150	375	252	9667
Perf/Power (Mflops/watt)	1.0	0.5	1.3	11.6

Parallel Computing Platforms Running the N-body Code

Machine	Avalon Beowulf	ASCI Red	ASCI White	Green Destiny+
Year	1996	1996	2000	2002
Performance (Gflops)	18	600	2500	58
Area (ft^2)	120	1600	9920	6
Power (kW)	18	1200	2000	5
DRAM (GB)	36	585	6200	150
Disk (TB)	0.4	2.0	160.0	4.8
DRAM density (MB/ft^2)	300	366	625	25000
Disk density (GB/ft^2)	3.3	1.3	16.1	800.0
Perf/Space ($Mflops/ft^2$)	150	375	252	9667
Perf/Power (Mflops/watt)	1.0	0.5	1.3	11.6

Green Destiny vs. Earth Simulator: LINPACK Benchmark

Machine	Green Destiny+	Earth Simulator
Year	2002	2002
LINPACK Performance (Gflops)	101	35,860
Area (ft^2)	6	70,290
Power (kW)	5	7,000
Cost Efficiency ($/Mflop)	3.35	11.15
Space Efficiency (Mflops/ft^2)	16,833	510
Power Efficiency (Mflops/watt)	20.00	5.12

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 with Pharmaceuticals + Feedback from J. Craig Venter

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.

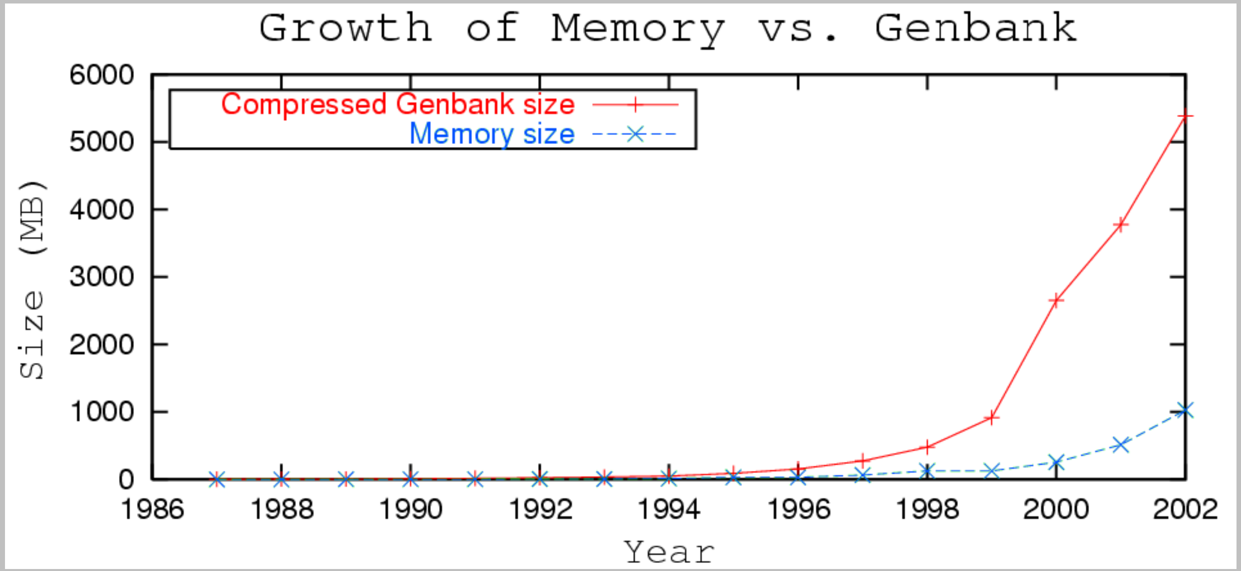
How Can BLAST Be Used?

- 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.

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

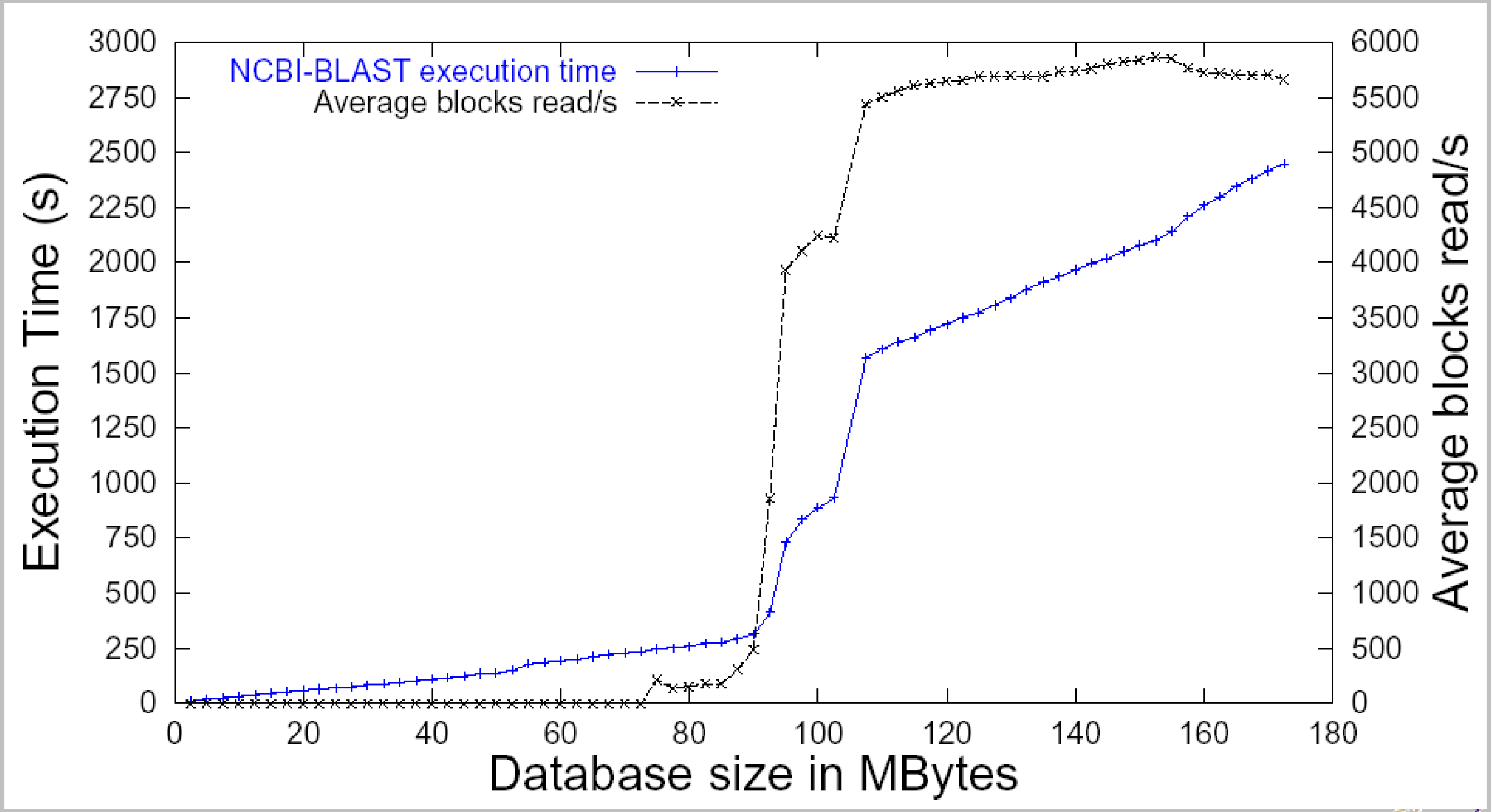
Motivation: BLAST Trends

- 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.

Parallelism in BLAST

Query:

>Perilla frutescens mRNA

CATCTACTCAAATTAAGAAATAGATAGAAATGGTTACGAGTGCAATGGGTCCAAGCCCGCGGGTGGAGG
AACTGGCCCGAAGCGGACTCGACACGATCCCAAAAGTATACGTGCGGCCCGAAGAGCACCTGAAAA

Database:

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

GCCTCAAACAGTTTAATTTTCTTCAAAACTAGTTTTTTTTTGGTTTTAGTTGGTATCCACGGAAGAGAGA
GAAAATGTTGGGAATTTTCAGCGGACGTATAGTATCATTGCCGGAAGAGCTGGTGGCTGCCGGGAACC

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

TTTTACTAGAGGAGTATCCCCGCTCCCGTGTACCTCTGGGCCCGTGTGGGAGGGTGGCTGGTGGGTATTG
GCGGCCCGAAGGGCCCGCCGCGCATTTAAGGAGTCGCCGCCCCGACTCTGTGTCTTCGGGTGACTTGGT
GCGCCGCCGTCAGCTAGTCTCCGATCTGCCCCGACCGACGGCTCCTGCCACCCGAACATG

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

TTTTTTTCTTGATGCTGAAATCTATCCAAACATCACCAGTGACATTTCCTTGAAAGTAGTGCTTTTGTCTT
TCAGACTTGCCCTCACGAGTCCTTGACCAAATTCTTGCTTTCTGGCACAATCTGAAGCCCAAAGGCTCTA

Match:

```
GTGGGTATTGGCGGCCCGAAGGGCCCGCCGCG
      ++---+++++++++++-++-+
AAGACTACGTGCGGCCCGAAGAGCACCTGAAA
```

Wu-chun Feng
feng@lanl.gov

Query Segmentation

Queries

>Perilla Frutescens CDS 0001

TTGGTATCCACGGAAGAGAGAGAAAATGTTGGGAATTTTCAGCGGAC
GTATAGTATCATTGCCGGAAGAGCTGGTGGCTGCCGGGAACC

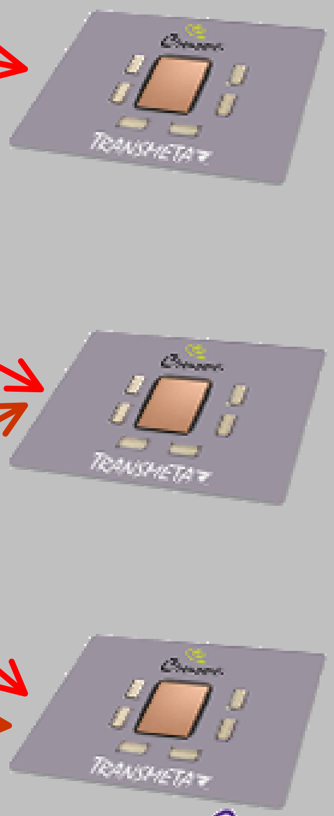
>Perilla Frutescens CDS 0002

GGAGGGTGGCTGGTGGGTATTGGCGGCCCGACCGATCTGCCCCGACC
GACGGCTCCTGCCACCCGAACATGTGATAGAAAGGAQQQQQQQ

>Perilla Frutescens CDS 0003

TTTTTTTCTTGATGCTGAAATCTATCCAAACATCACCAGTCCTCACGA
GTCCTTGACCAAATTCTTGCTTTCTGGCACAATCTGAAGCCCAAAGGC

Worker nodes



Database

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

TTGGTATCCACGGAAGAGAGAGAAAATGTTGGGAATTTTCAGCGGAC
GTATAGTATCATTGCCGGAAGAGCTGGTGGCTGCCGGGAACC

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

GGAGGGTGGCTGGTGGGTATTGGCGGCCCGACCGATCTGCCCCGACC
GACGGCTCCTGCCACCCGAACATG

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

TTTTTTTCTTGATGCTGAAATCTATCCAAACATCACCAGTCCTCACGA
GTCCTTGACCAAATTCTTGCTTTCTGGCACAATCTGAAGCCCAAAGGC

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
GTATAGTATCATTGCCGGAAGAGCTGGTGGCTGCCGGGAACC

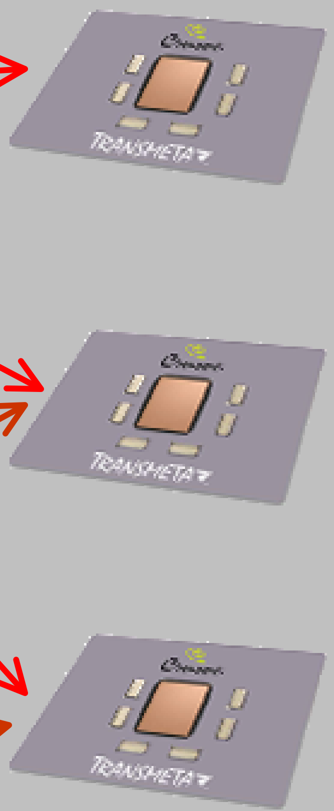
>Perilla Frutescens CDS 0002

GGAGGGTGGCTGGTGGGTATTGGCGGCCCGACCGATCTGCCCCGACC
GACGGCTCCTGCCACCCGAACATGTGATAGAAAGGAQQQQQQQ

>Perilla Frutescens CDS 0003

TTTTTTTCTTGATGCTGAAATCTATCCAAACATCACCAGTCCTCACGA
GTCCTTGACCAAATTCTTGCTTTCTGGCACAATCTGAAGCCCAAAGGC

Worker nodes



Database

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

TTGGTATCCACGGAAGAGAGAGAAAATGTTGGGAATTTTCAGCGGAC
GTATAGTATCATTGCCGGAAGAGCTGGTGGCTGCCGGGAACC

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

GGAGGGTGGCTGGTGGGTATTGGCGGCCCGACCGATCTGCCCCGACC
GACGGCTCCTGCCACCCGAACATG

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

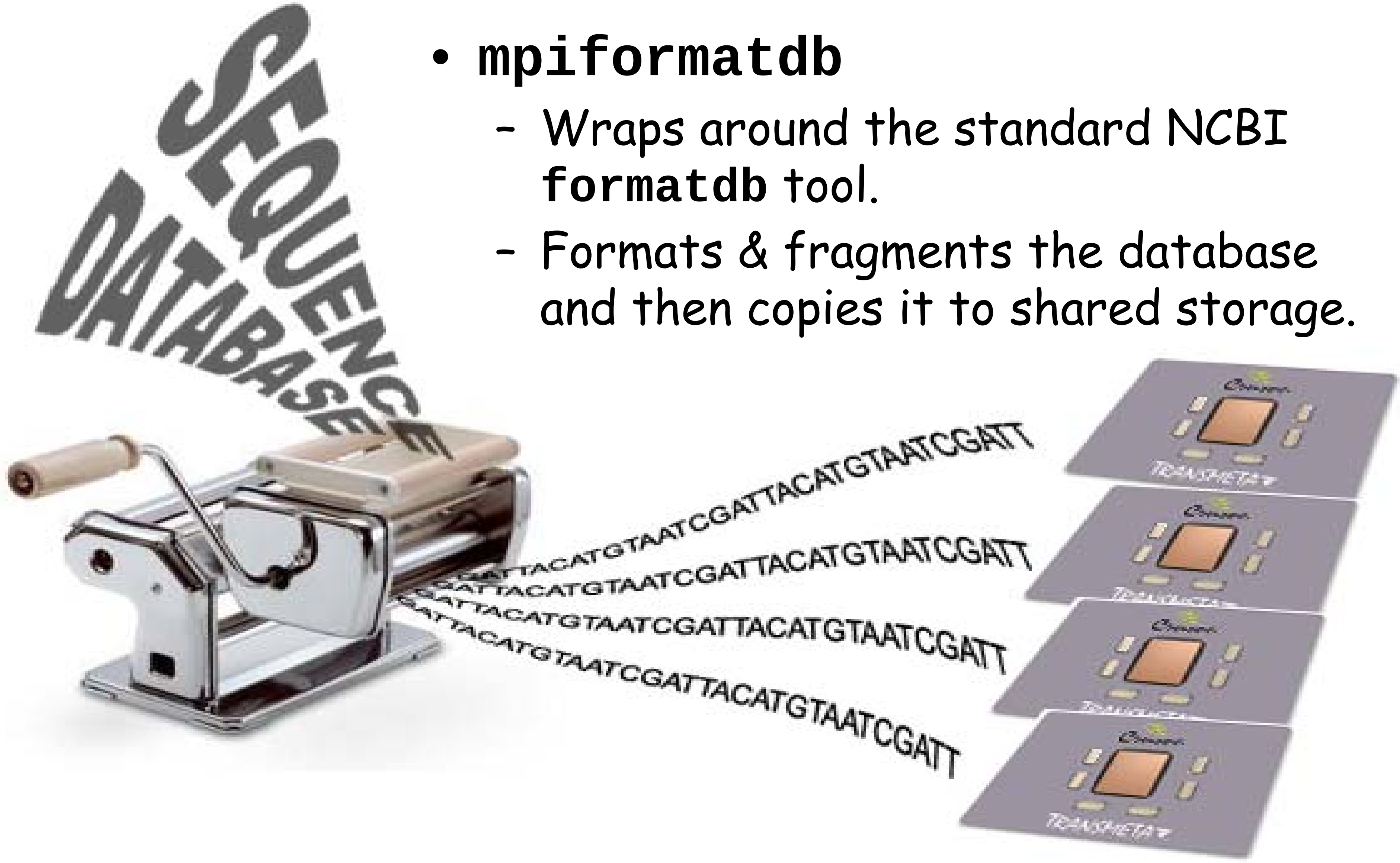
TTTTTTTCTTGATGCTGAAATCTATCCAAACATCACCAGTCCTCACGA
GTCCTTGACCAAATTCTTGCTTTCTGGCACAATCTGAAGCCCAAAGGC

"Bioinfomagic" Outline

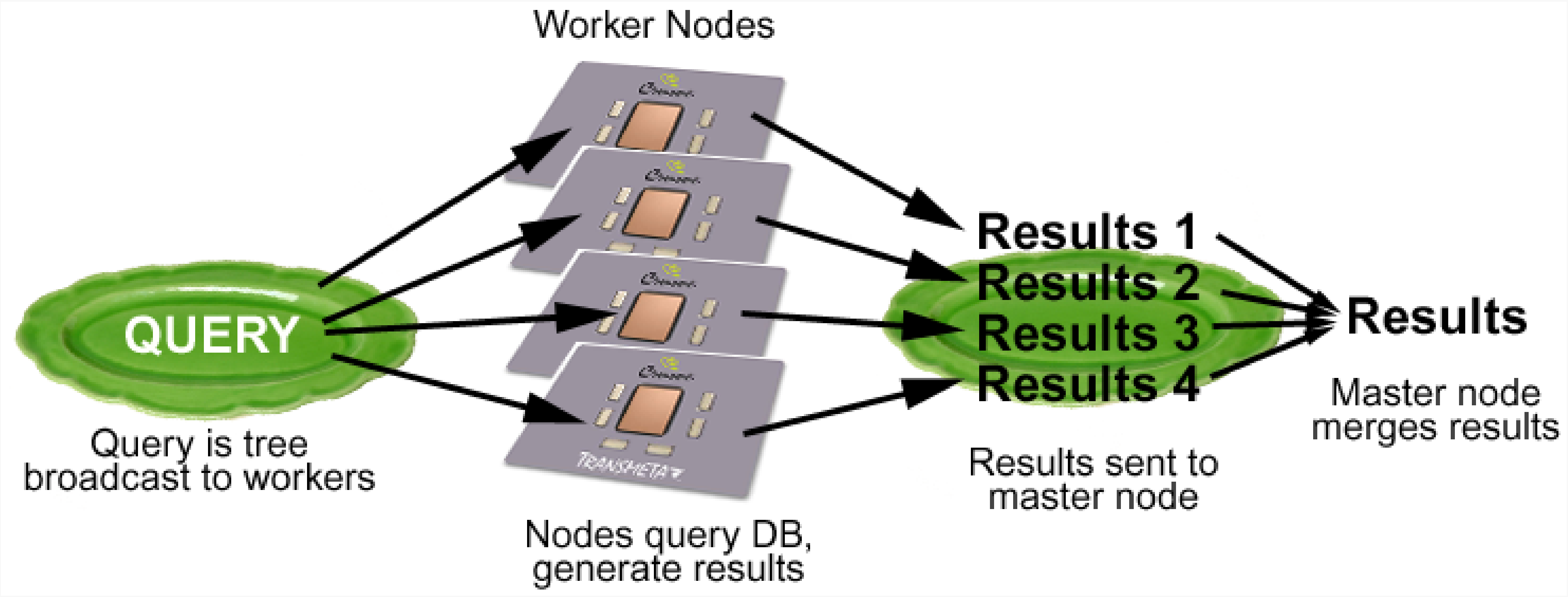
- Green Destiny
  - Problem Statement
  - Where is Supercomputing?
  - "Supercomputing in Small Spaces" Project
  - Experimental Results
  - Inspiration for mpiBLAST
- mpiBLAST
  - All About BLAST
    - What? How To Use?
  - Motivation
  - Uncovering the Parallelism
  - Algorithm & Implementation
  - 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



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

- Operating System: Linux, Windows, Solaris.
- Programming Language: C++ with MPI.
- Job Scheduling: PBS and Sun Grid Environment.

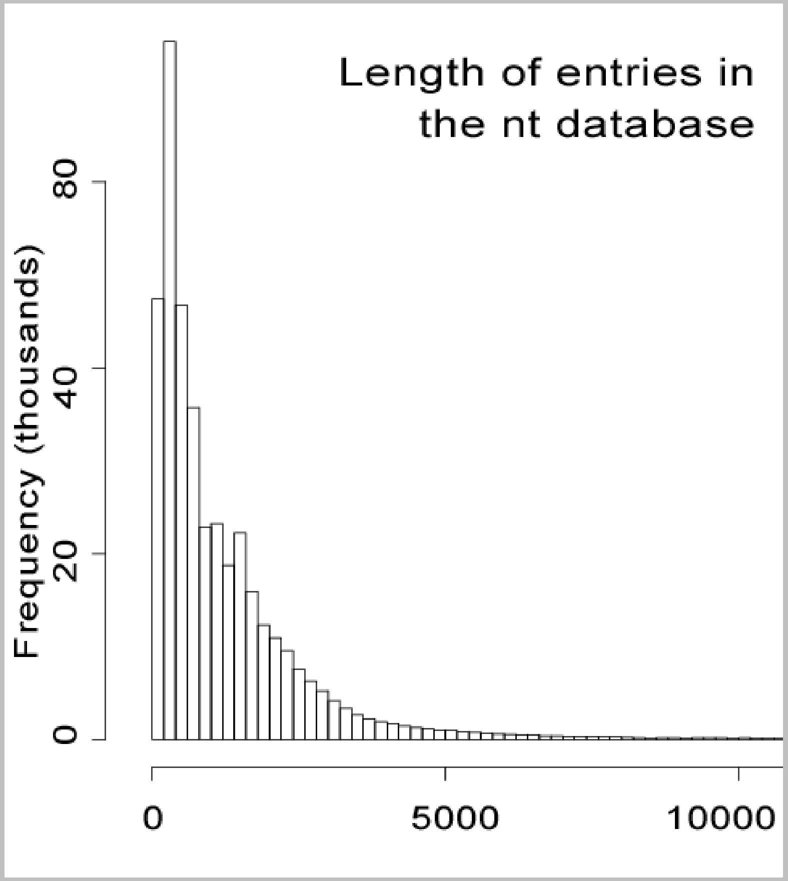
Wu-chun Feng
feng@lanl.gov

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.

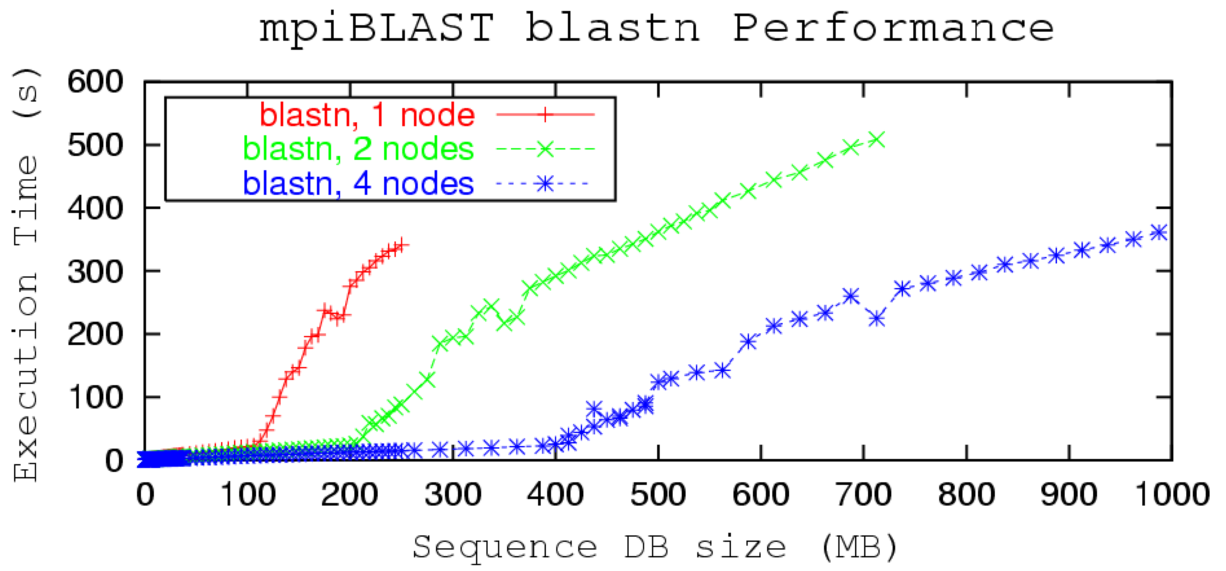
BLAST Usage Model

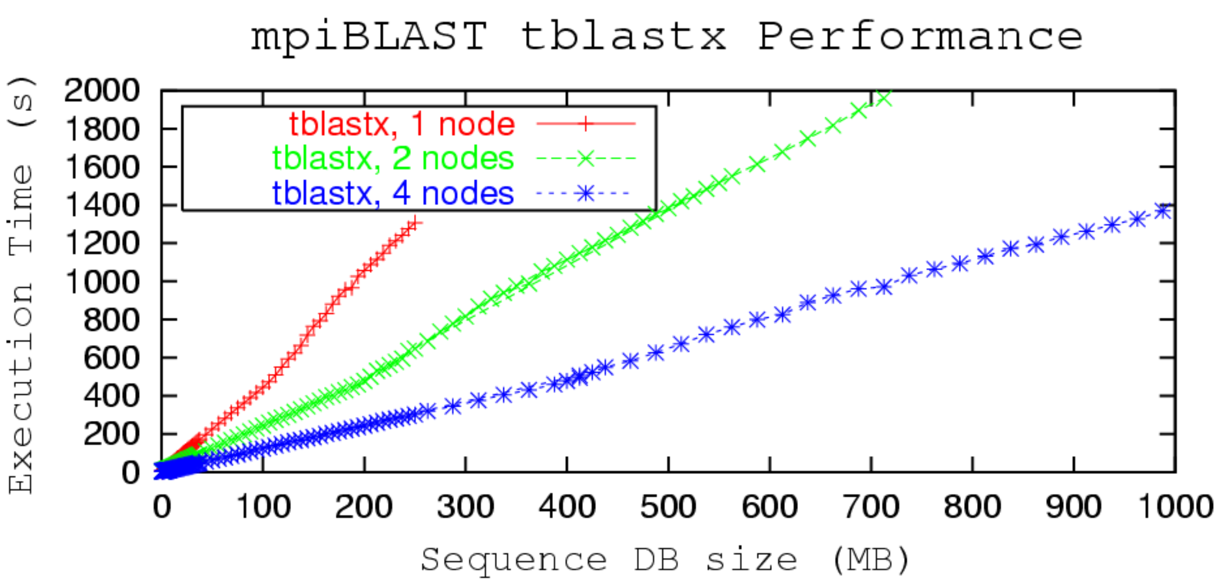
- We model BLAST as it would be used in a high-throughput sequence annotation pipeline.
- Queries are predicted genes from a newly sequenced bacterial genome, lengths are exponentially distributed with a mean of 747.2.
- Database is the GenBank *nt* database, sequence lengths are approximately exponentially 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

Nodes	Runtime (s)	Speedup over 1 node
1	80774.93	1.00
4	8751.97	9.23
8	4547.83	17.76
16	2436.60	33.15
32	1349.92	59.84
64	850.75	94.95
128	473.79	170.49

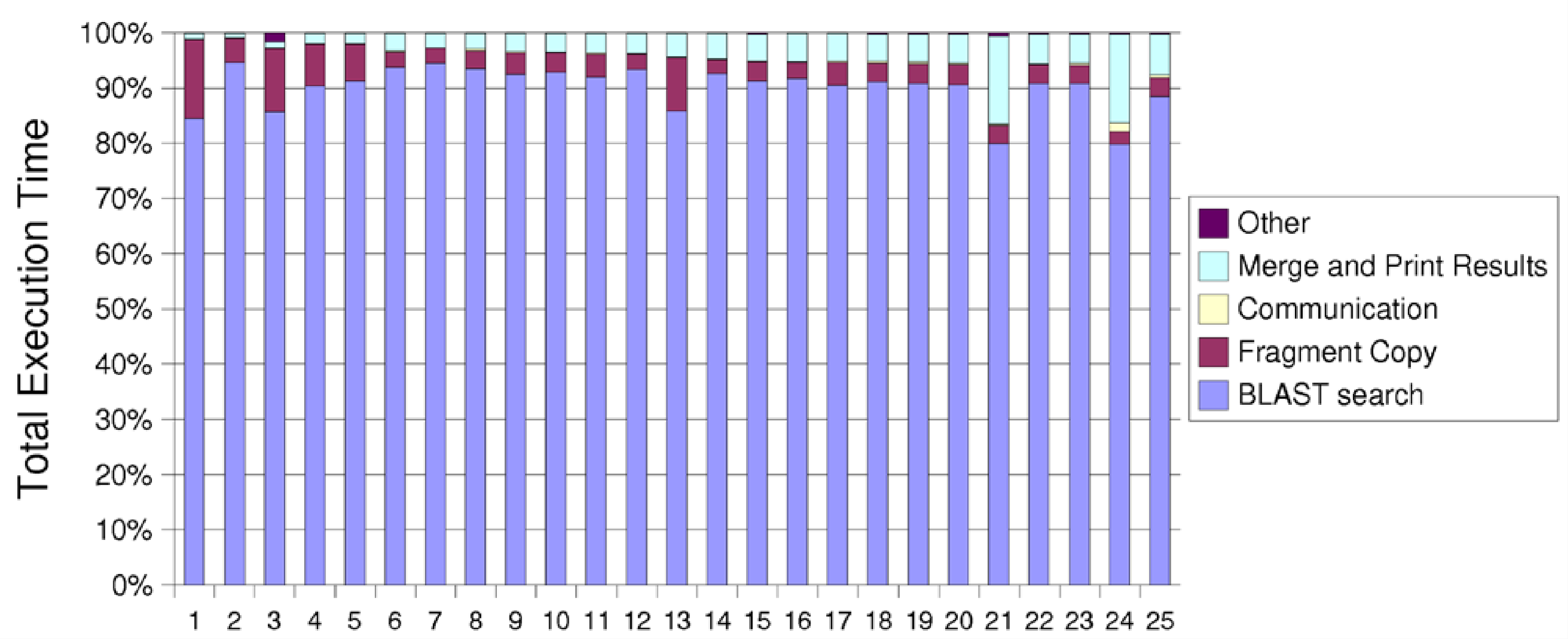
The Bottom Line

mpiBLAST reduces search time from 1346 minutes (or 22.4 hours) to under 8 minutes!

Where Does The Time Go?

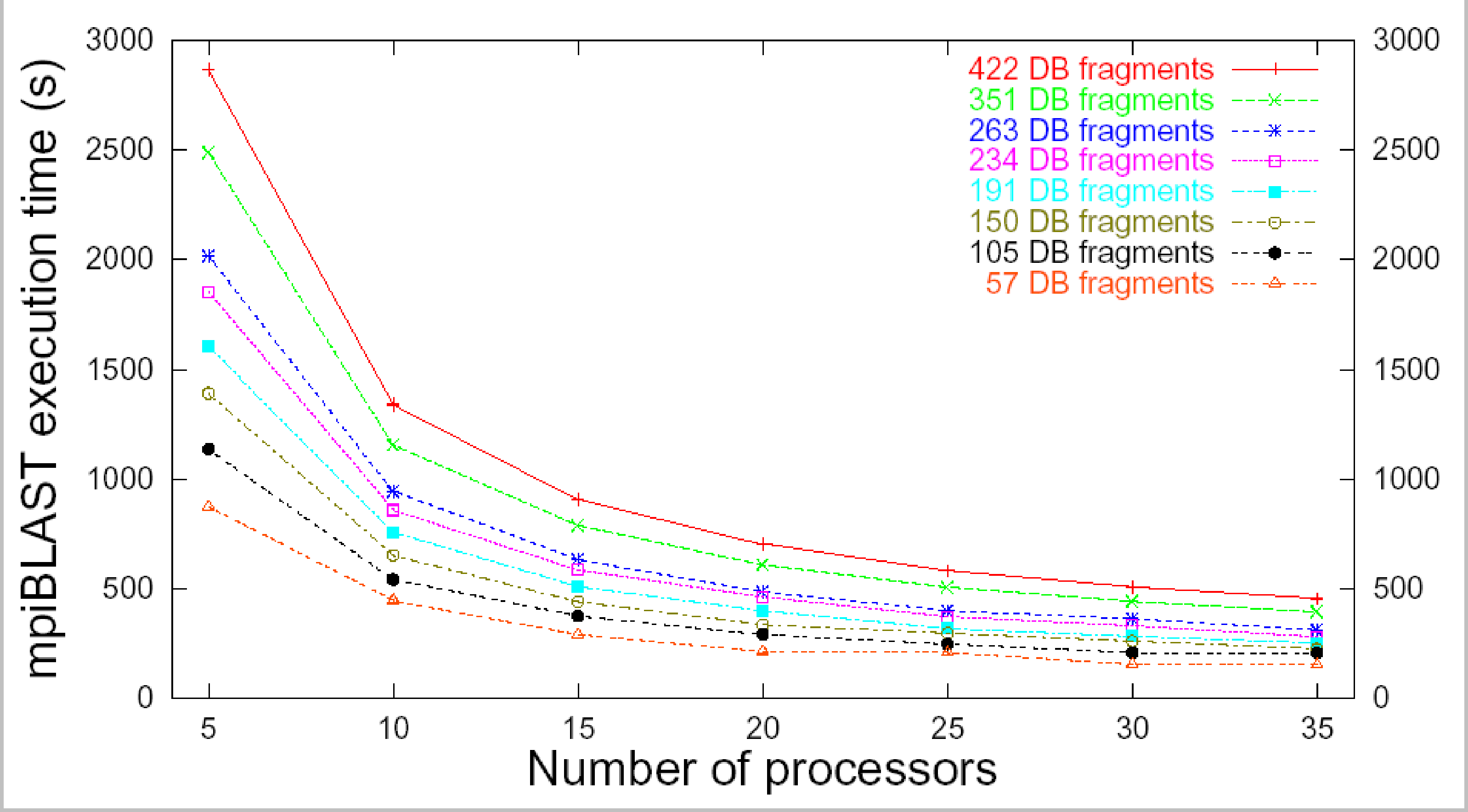
- Components of mpiBLAST Execution
  1. MPI and mpiBLAST initialization
  2. Copying of database fragments
  3. BLAST search
  4. Communication
  5. Result merging and formatting
- Notes
  1. "Result merging and formatting" is serial.
  2. mpiBLAST was instrumented with MPE to collect timing statistics for each of the above components.

Where The Time Goes …



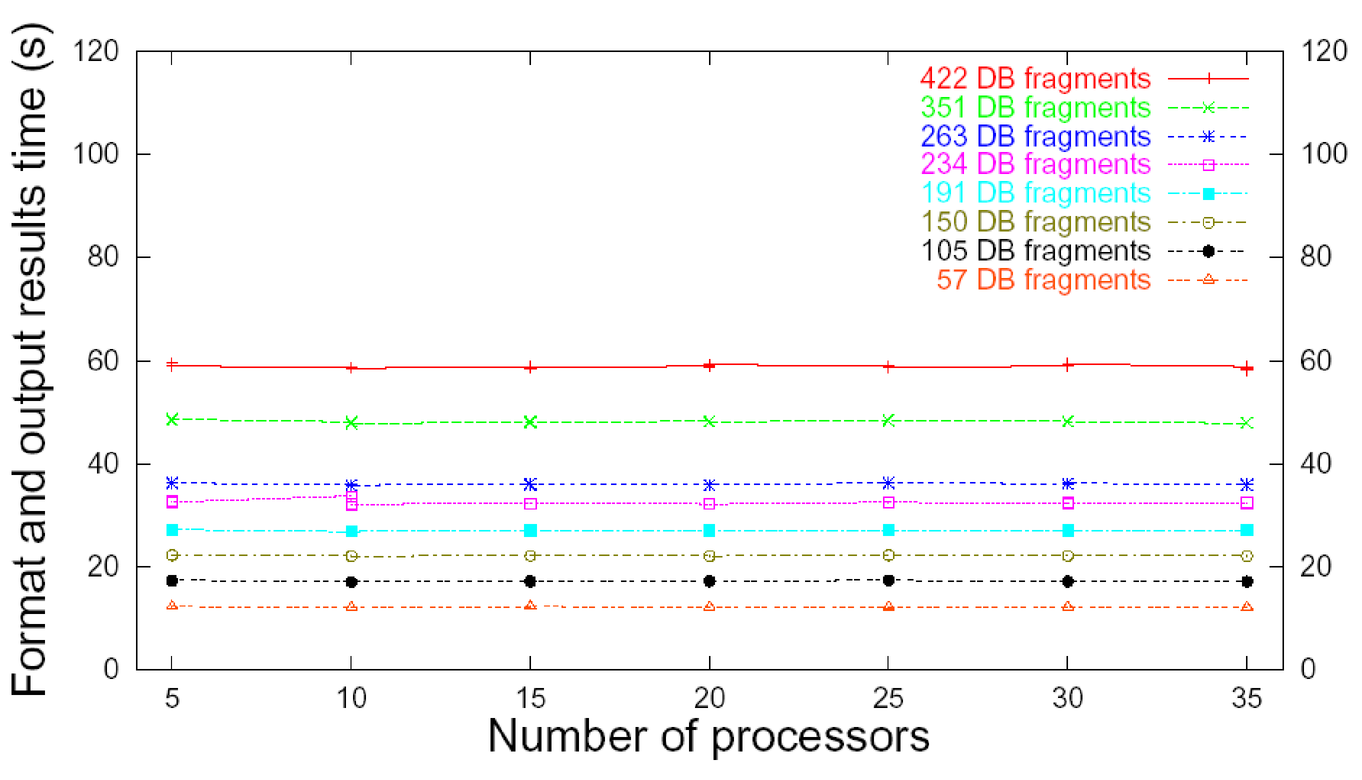
Overhead for Fragmentation

- 10-KB query searched against a 2-GB database.



Overhead for Result Formatting

- 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.

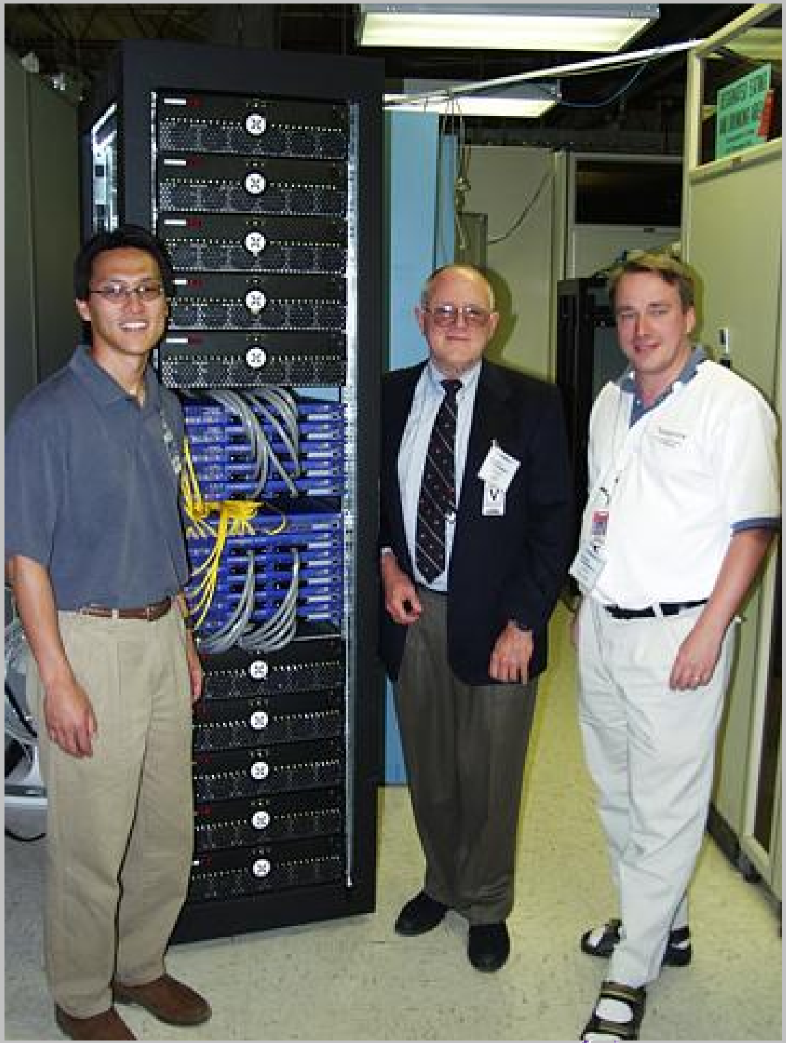
COMPUTER & COMPUTATIONAL SCIENCES

Relevant Publications

- Green Destiny
  - "Honey, I Shrunk the Beowulf!," *International Conference on Parallel Processing*, Aug. 2002.
  - "The Bladed Beowulf: A Cost-Effective Alternative …," *IEEE Cluster*, Sept. 2002.
  - "High-Density Computing: A 240-Processor Beowulf in One Cubic Meter," *SC 2002,* Nov. 2002.
- mpiBLAST
  - "mpiBLAST: Delivering Super-Linear Speedup with an Open-Source Parallelization of BLAST" (poster), *Pacific Symposium on Biocomputing*, Jan. 2003.
  - "The Design, Implementation, and Evaluation of mpiBLAST" <u>Best Paper: Applications Track</u>, *ClusterWorld Conference & Expo*, Jun. 2003.

Media Coverage

- "Los Alamos Lends Open-Source Hand to Life Sciences," *The Register*, 6/29/03. http://www.theregister.com/content/61/31471.html.
- "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.
- "At Los Alamos, Two Visions of Supercomputing," *The New York Times*, 6/25/02.
- "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

< logo to be determined >

☺

http://sss.lanl.gov

http://mpiblast.lanl.gov

Wu-chun (Wu) Feng

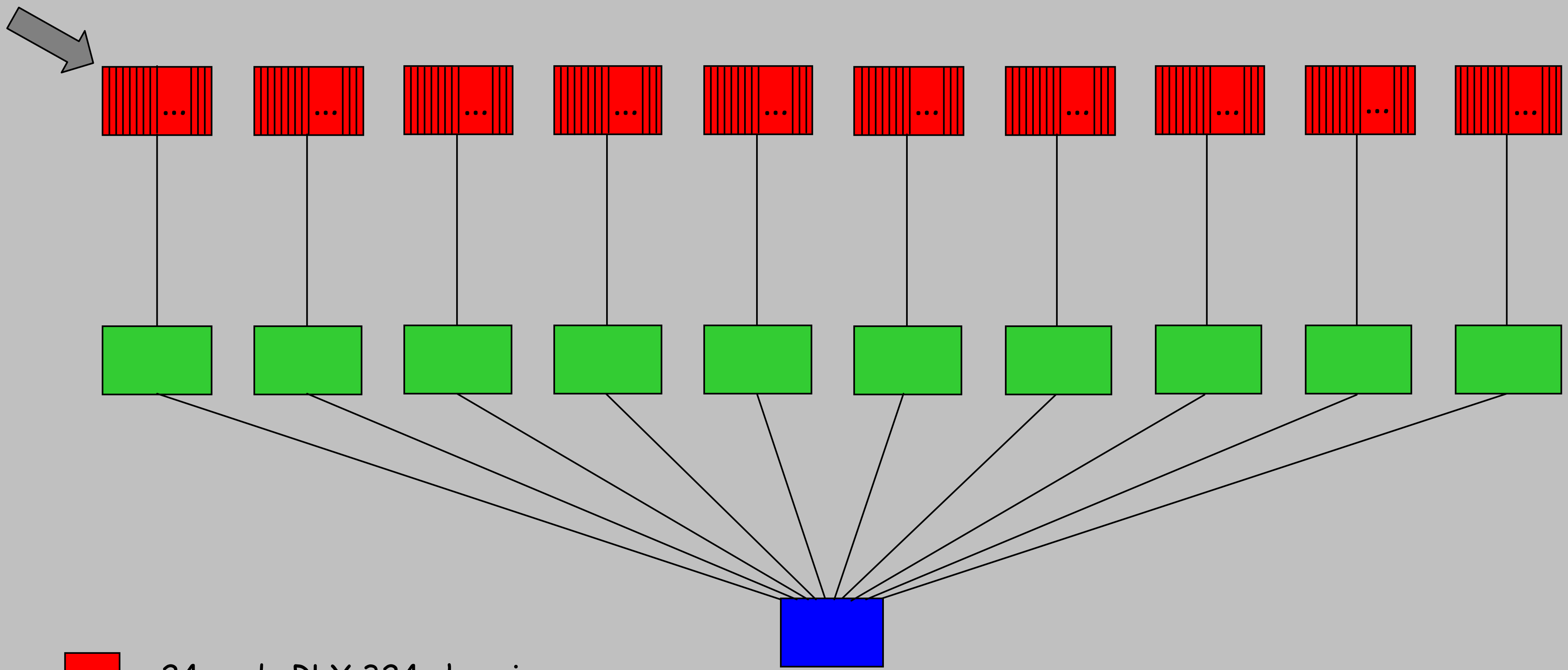
Research and Development in Advanced Network Technology

RADIANT

http://www.lanl.gov/radiant

EXTRA SLIDES

Architecture For Our Bladed Beowulf Cluster: Green Destiny



24-node RLX 324 chassis

24-port Fast Ethernet switch

16-port Gigabit Ethernet switch

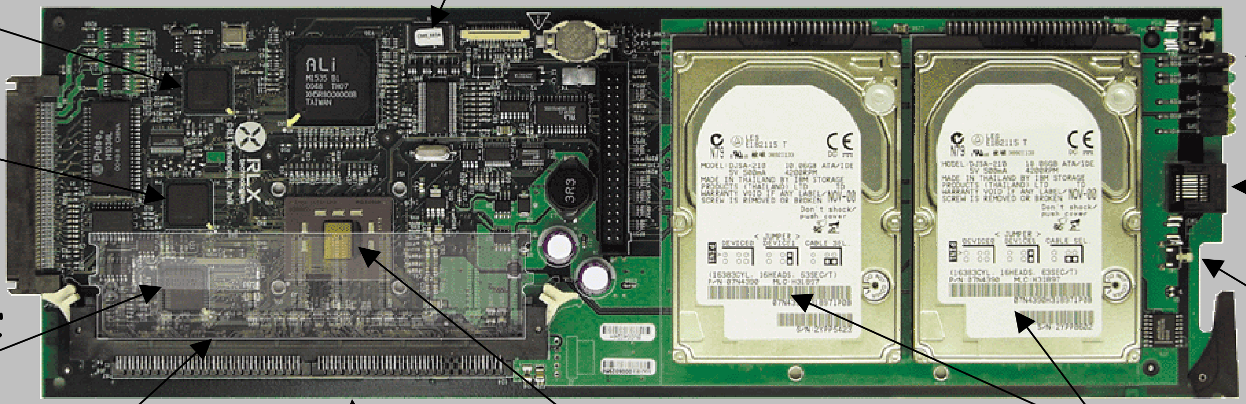
100-Mb/s Fast Ethernet link

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)



Code Morphing Software (CMS), 1 MB

Public NIC
33 MHz PCI

Private NIC
33 MHz PCI

Management NIC
33 MHz PCI

Status LEDs

Serial RJ-45
debug port

Reset Switch

128MB, 256MB, 512MB
DIMM SDRAM
PC-133

512KB
Flash ROM

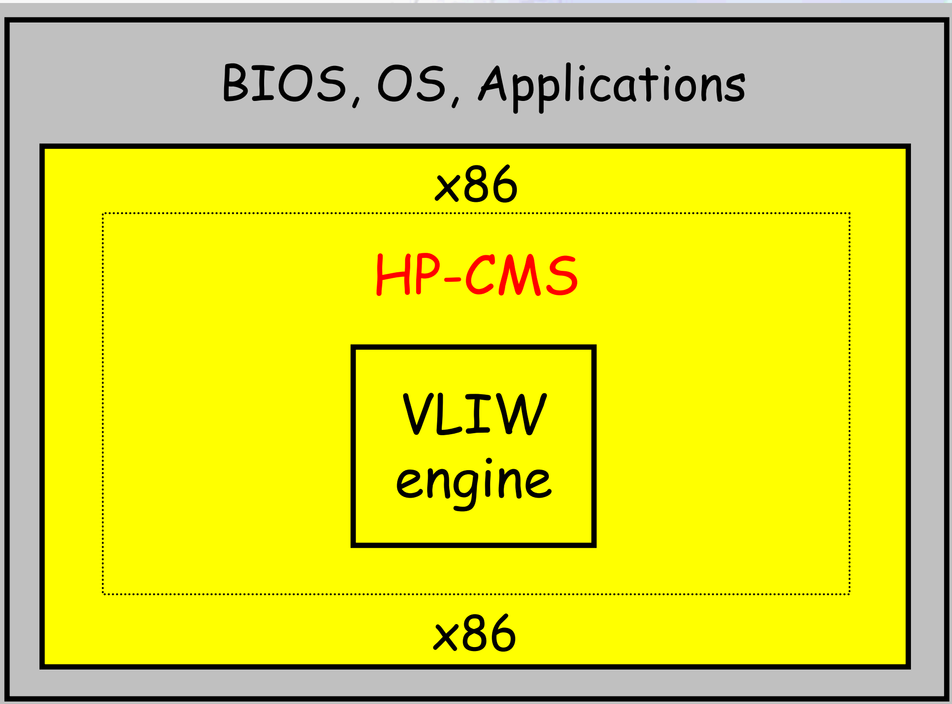
Transmeta™
TM5600 633 MHz

ATA 66
0 or 1 or 2 - 2.5" HDD
10 or 30 GB each

RLX ServerBlade™ 1000t
$999

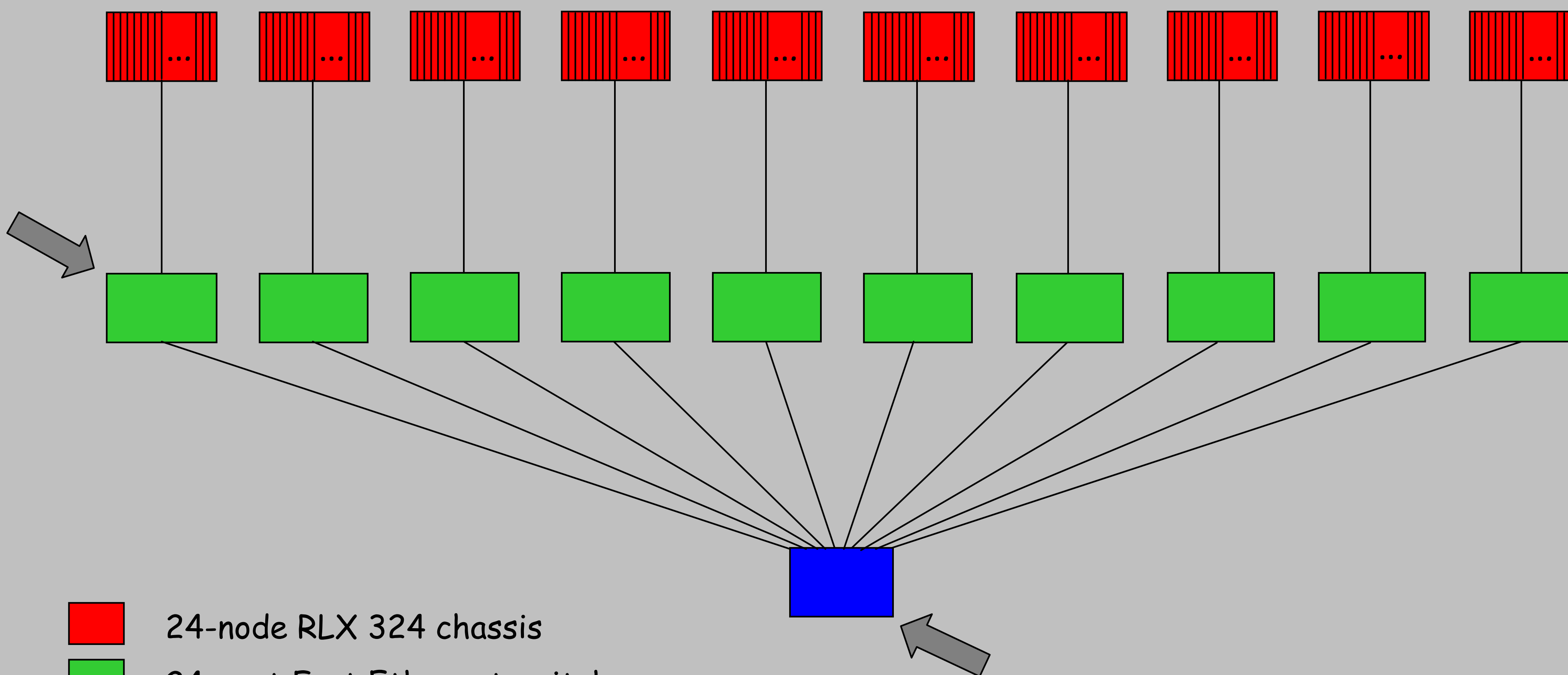
128KB L1 cache, 512KB L2 cache
LongRun, Northbridge, x86 compatible

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")
  - ~¼ of Intel PIII → ~ 6x-7x less power dissipation
  - Less power → lower "on-die" temp. → 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

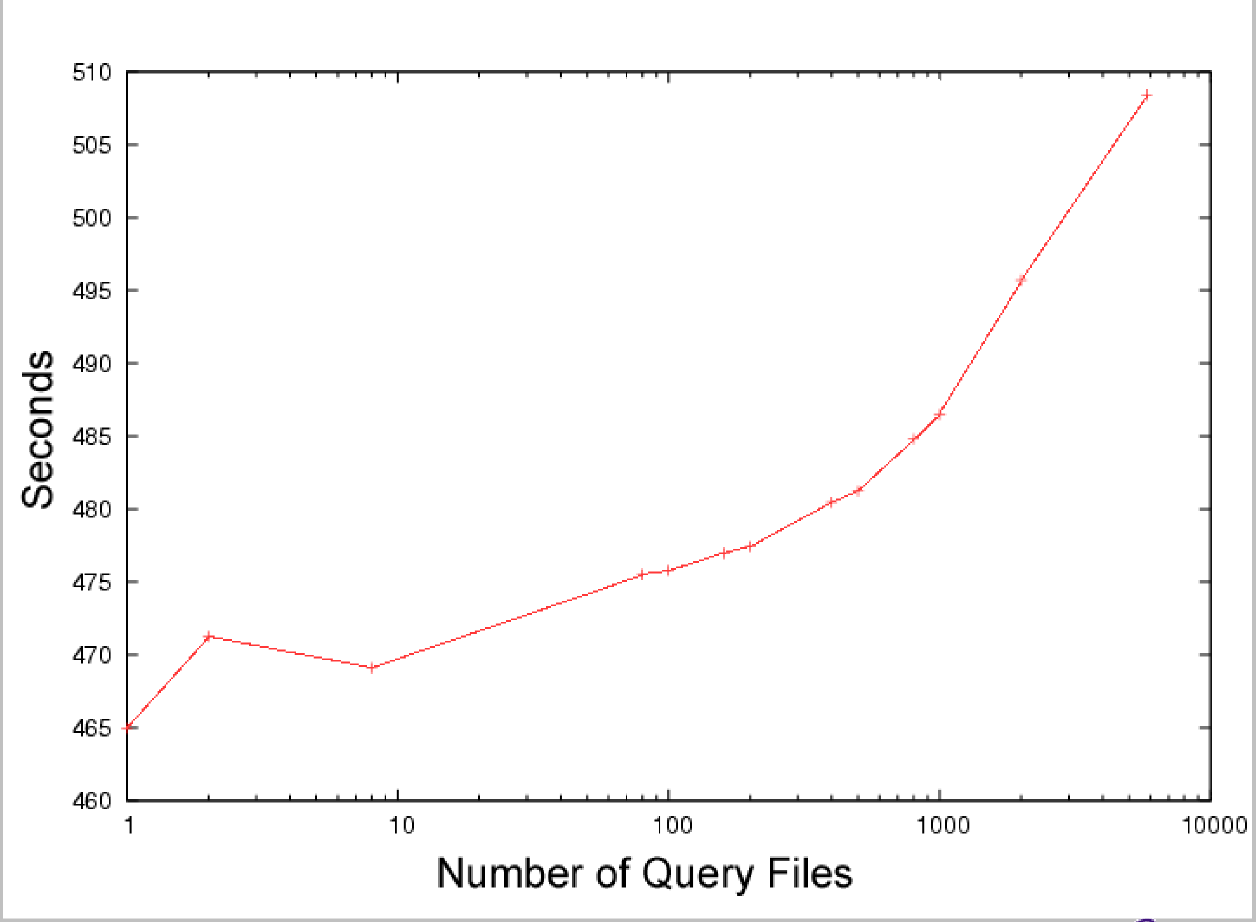
100-Mb/s Fast Ethernet link

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:



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.