

Short version of the keynote: Exascale in Biology: a long way to go!!

Exascale Challenges in Computational Biology
Barcelona, 13-15 Dec. 2010

Reinhard Schneider

Data Integration and Knowledge Management
EMBL-Heidelberg
Germany

Member of the working group 3.4 on Life Science and Health of
the European FP7 Support Action EESI "European Exascale
Software Initiative", (<http://www.eesi-project.eu>)



PRACE

The Partnership for Advanced Computing in Europe, PRACE, is a unique persistent pan-European Research Infrastructure for High Performance Computing (HPC). PRACE is a project funded in part by the EU's 7th Framework Programme.

PRACE forms the top level of the European HPC ecosystem.

PRACE provides Europe with world-class systems for world-class science and strengthens Europe's scientific and industrial competitiveness. PRACE will maintain a pan-European HPC service consisting of up to six top of the line leadership systems (Tier-0) well integrated into the European HPC ecosystem. Each system will provide computing power of several Petaflop/s (one quadrillion operations per second) in midterm.

On the longer term (2019) Exaflop/s (one quintillion) computing power will be targeted by PRACE. This infrastructure is managed as a single European entity.

Exa: 10^{18}



My hands-on experience
(1990-1997, PVM and MPI, 28 - 1024 CPU's)



Thinking machines CM5
Intel Touchstone
Parsytec GC 1024
IBM SP2 512
Kendal Square KSR-1
SGI PowerChallenge Array
Meiko CS2
Alliant FX2800

A supercomputer is like a F1 car



720 hp

18.000 rpm

300 km/h

75 liter / 100 km

Bioinformatics is a lot of plumbing.....



....and lot's of data!!



Sometime we care little about performance:

blastp

DB: nr.00 - nr.04 combined downloaded from ftp://ftp.ncbi.nlm.nih.gov/blast/db/

11,921,515 sequences; 4,071,357,309 total letters

Query Sequence: [ACL81455](#), 301 letters

# of Threads ¹⁾	NCBI Blast+	NCBI Blast+ for x86_64 ²⁾	NCBI Blast	NCBI Blast 2.2.20 for x86_64
1	241.0	126.1	202.9	124.8
2	156.4	69.5	145.7	78.9
4	114.6	38.3	116.7	56.6
8	93.3	20.6	102.6	45.9
12	90.0	17.1	100.6	43.7
16	87.7	15.1	100.7	42.1

3. Conclusions

3.1.blastn

64 bit NCBI Blast+ is the fastest for all numbers of threads.

Using more than 8 threads will not accelerate search speed any more.

If enough memory is available, launching more instances with 8 threads is the best strategy.

3.2.blastp

64 bit NCBI Blast+ is the fastest.

NCBI Blast+ is the slowest for 1 and 2 threads.

Using NCBI Blast+ on Core 2 Duo Macs is not a good idea. Use 64 bit NCBI Blast+ instead.

¹⁾ Specified -num_threads n in the command arguments.

²⁾ NCBI Blast+ was built with --with-64 configuration option. Please refer to [How to Build NCBI Blast+ for Mac OS X](#) for details to build. This 64 bit binary is available for download [here](#). It should work on Mac OS X 10.6 and 10.5 with 64 bit Intel processors, such as Core 2 Duo and Xeon.

<http://www.blaststation.com/>

How to measure performance?

Linpack Benchmark

....systems are ranked only by their ability to solve a set of linear equations, $Ax = b$, using a dense random matrix A .

...its scalability in the sense that it covers a performance range of 10 orders of magnitude.

... delivers performance figures that occupy the upper end of any other application performance.

...no other realistic application delivers a better efficiency (R_{max}/R_{peak}) of a system.

...running Linpack to measure the performance is kind of a first reliability test for new HPC systems.

The TOP500 Project: Looking Back over 15 Years of Supercomputing Experience
Hans Werner Meuer, University of Mannheim, Germany January 20, 2008
http://www.top500.org/files/TOP500_Looking_back_HWM.pdf

Problems with LinPack

- Top500 does not induce dataset size restrictions
- People will use largest matrix possible to have optimal computation to communication ratio
- Linpack uses $O(n^2)$ data and $O(n^3)$ computations ==> increase dataset size
- Roadrunner system $n=2.3 \times 10^6$; run took 2 hours
- Jaguar system at Oak Ridge Labs more memory (300 TeraByte), thus: $n = 4.7 \times 10^6$; run took 18 hours
- With increases in PetaFlops and Memory run-times for Linpack will get very long

FLOPS is not everything

Five years ago, Florida State University (FSU) faced a dilemma not uncommon to large universities supporting diverse research programs. The problem was that researchers were complaining that they lacked adequate computing and storage resources to support their research programs and, by extension, to fulfill their obligations to external funding agencies.

The dilemma was that these complaints were made while, at the same time, FSU was supporting a shared supercomputer that had recently run a highperformance Linpack (HPL) benchmark, placing it at number 34 among the world's Top 500 fastest computers.

<http://www.scientificcomputing.com/articles-HPC-Survival-in-the-Academic-Jungle-111010.aspx>

- Speedup

- Efficiency

Speedup

Speedup is the ratio between the run time of the original code and the run time of the modified code

$$\text{Speedup} = \frac{\text{Run time original code}}{\text{Run time modified code}}$$

or sometimes: the run time of the BEST serial implementation

Parallel Speedup

Parallel speedup is the ratio between the run time of the sequential code and the run time of the modified code

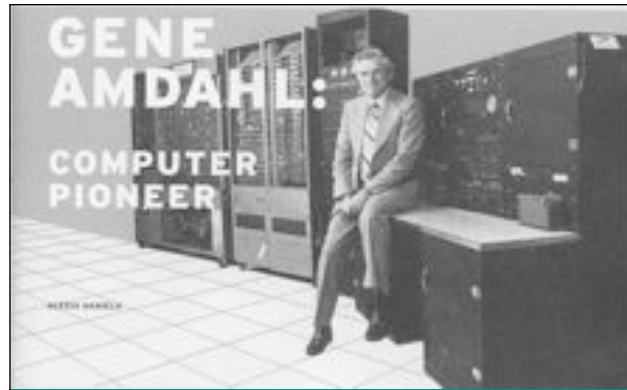
$$\text{Speedup} = \frac{\text{Run time sequential}}{\text{Run time parallel}}$$

Run time is measured as elapsed time (wallclock)

Efficiency

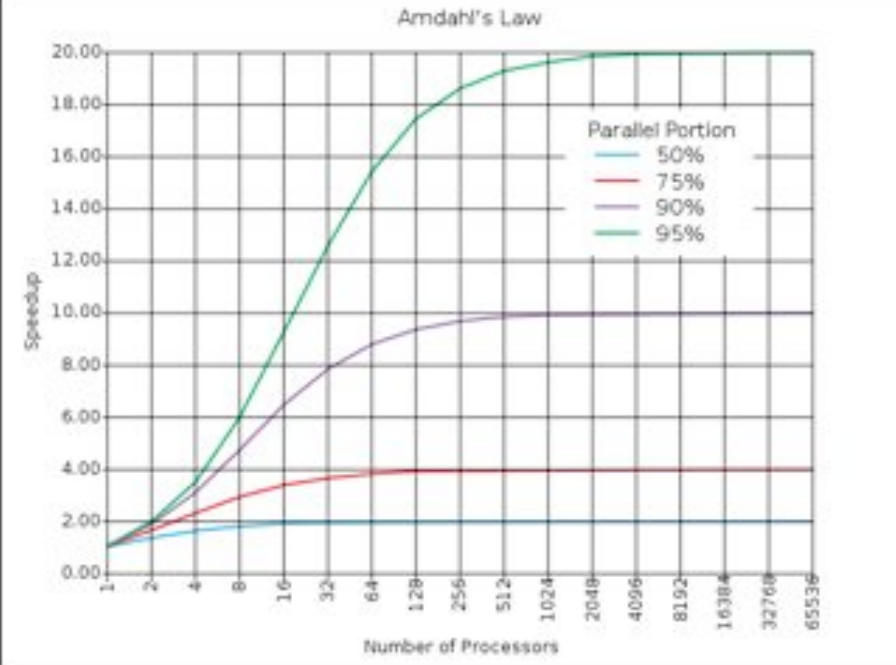
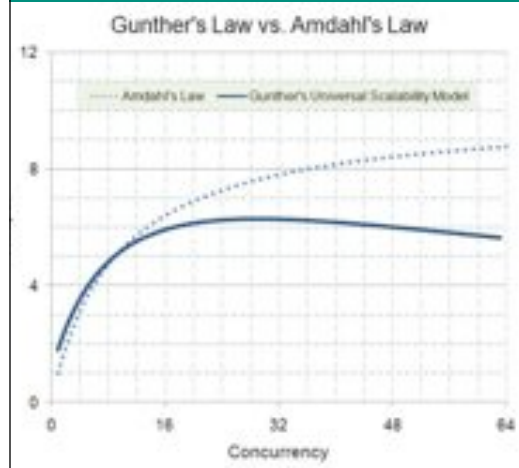
Parallel efficiency is defined as how well a program (your code) utilizes multiple processors (cores)

$$\text{Efficiency} = \frac{\text{Run time sequential}}{\text{Run time parallel} * \text{Nproc.}}$$



Overall Speedup

$$\frac{1}{(1 - P) + P/S}$$



Latency

$$C(p) = p / (1 + s(p-1) + kp(p-1))$$

Scalability is limited by sequential part

- Every program has a sequential portion, even if it is just the time needed to start all the threads or send initial data etc.
- $\text{Speedup} \leq 1 / (f + ((1-f)/p))$, where f is the fraction of the sequential part of the program
- For $p \rightarrow \text{infinity}$ the maximum speedup $S_{\text{max}} \leq 1/f$
- If f is 0.01 we get $S_{\text{max}} \leq 100$ (assuming linear speedups for the parallel part)
- Solutions
 - Make f small

HTC versus HPC

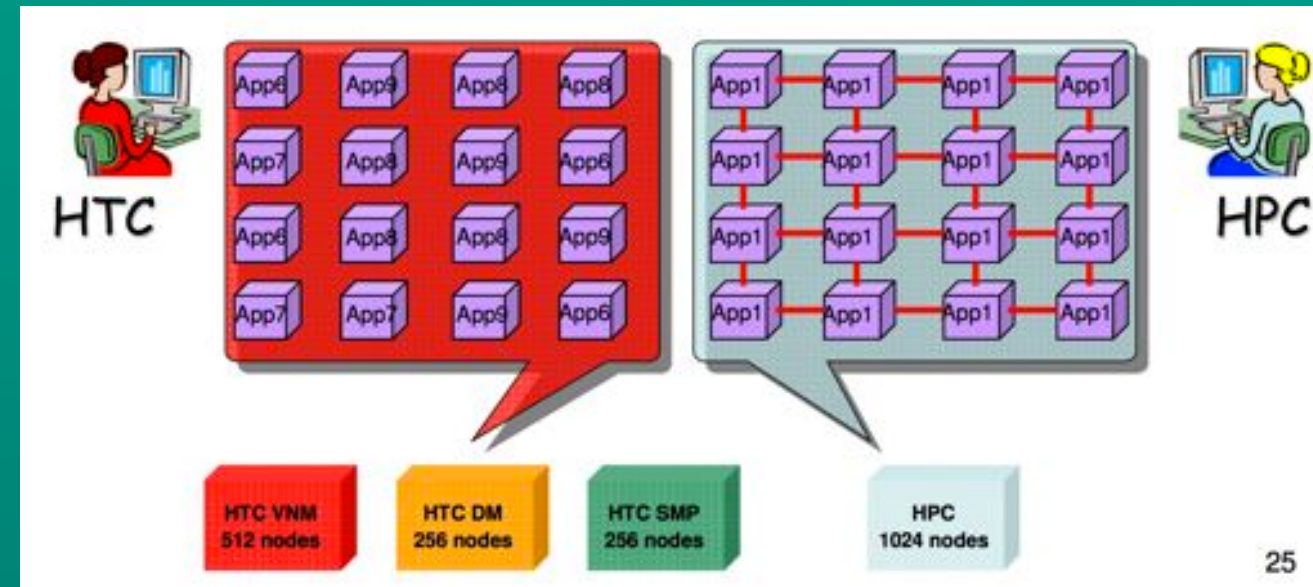
High Performance Computing (HPC) Mode – best for Capability Computing

- Parallel, tightly coupled applications
- Single Instruction, Multiple Data(SIMD) architecture
- Programming model: typically MPI
- Apps need tremendous amount of computational power over short time period

High Throughput Computing (HTC) Mode – best for Capacity Computing

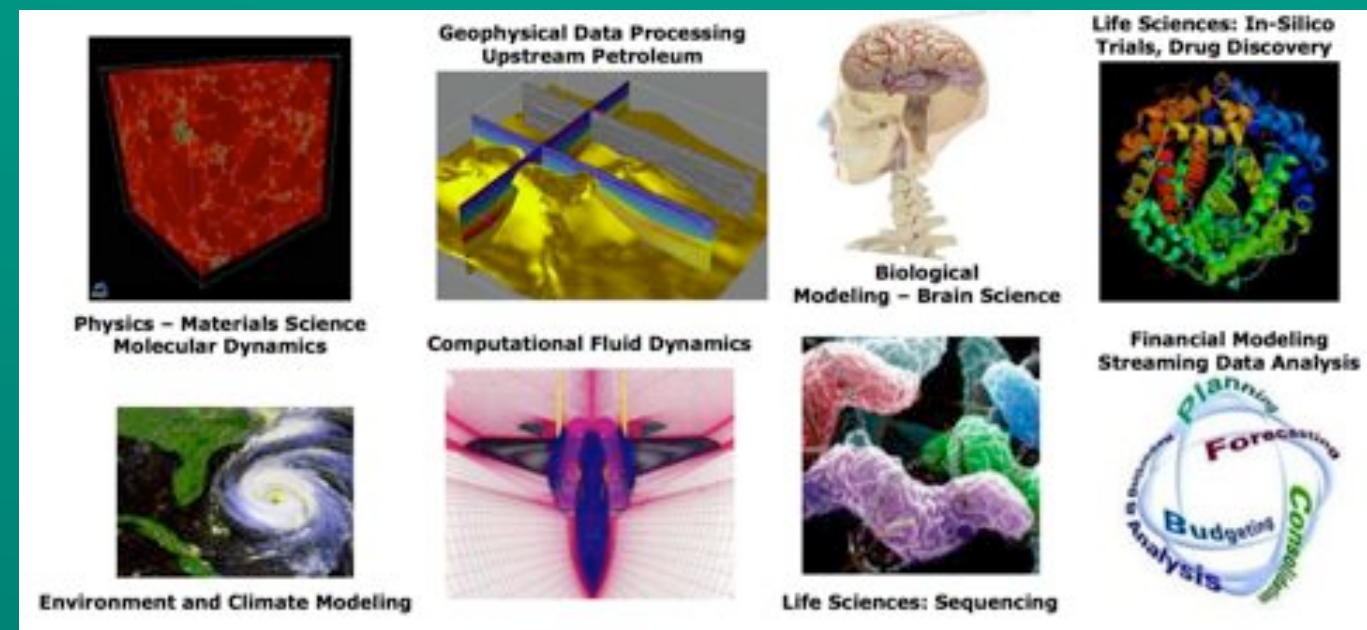
- Large number of independent tasks
- Multiple Instruction, Multiple Data(MIMD) architecture
- Programming model: non-MPI
- Apps need large amount of computational power over long time period
- Traditionally run on large clusters

IBM Massively Parallel Blue Gene: Application Development, Carlos P Sosa IBM and Biomedical Informatics & Computational Biology, University of Minnesota Rochester Rochester, Minnesota, http://www.msi.umn.edu/~cpsosa/MoscowStateUniv-JUL-2010_lecture.pdf



IBM Massively Parallel Blue Gene: Application Development, Carlos P Sosa IBM and Biomedical Informatics & Computational Biology, University of Minnesota Rochester Rochester, Minnesota, http://www.msi.umn.edu/~cpsosa/MoscowStateUniv-JUL-2010_lecture.pdf

The usual suspects.....



IBM Massively Parallel Blue Gene: Application Development, Carlos P Sosa IBM and Biomedical Informatics & Computational Biology, University of Minnesota Rochester Rochester, Minnesota, http://www.msi.umn.edu/~cpsosa/MoscowStateUniv-JUL-2010_lecture.pdf

How big are the systems today?

Number of Processors	Count	Share %	Rmax Sum (GF)	Rpeak Sum (GF)	Processor Sum
1025-2048	2	0.40 %	156020	199932	3328
2049-4096	60	12.00 %	2284042	2716312	221438
4k-8k	291	58.20 %	11685027	18923817	1769924
8k-16k	96	19.20 %	7607941	10538478	1037103
16k-32k	20	4.00 %	3375425	4351903	506760
32k-64k	17	3.40 %	5227787	8820874	783950
64k-128k	5	1.00 %	4655000	6694211	483678
128k-	9	1.80 %	8681851	12409784	1666146
Totals	500	100%	43673092.54	64655310.70	6472327

source: Top 500 list

TOP machines in Life science

- MD Anderson, 48.1 teraflop, 8,064-core HP Cluster Platform
- BC Genome Science Center machine, 47.3-teraflop, 5,040-core IBM iDataPlex
- University of Tokyo system, 34.7-teraflop, 3,552-core Fujitsu Primergy
- Pacific Northwest National Laboratory's Environmental Molecular Sciences Laboratory, 97.1-teraflop, 18,176-core HP cluster
- University of Tokyo's Human Genome Center, 54-teraflop, 5,760-core Sun Microsystems blade system
- Georgia Institute of Technology's Center for the Study of Systems Biology, 53.1-teraflop, 8,640-core system
- Janelia Farm Campus, 35.8-teraflop, 4,000-core Dell system
- Arizona State University and the Translational Genomics Research Institute, 30.1-teraflop Dell system

Computer System	Number of processors	CPU or Hybrid	PFLOPS	MFLOPS/Watt
Jaguar	224162	CPU	1.76	251
Roadrunner	122400	Hybrid	1.04	446
Jaguar@Tennessee (Cray XT5)	98928	CPU	0.831	269
Jugene	294912	CPU	0.825	365
TH-1	71680	Hybrid	0.563	380
Lawaides (SGI Altix@NASA)	56320	CPU	0.544	230
BlueGene@Lawrence Livermore	212992	CPU	0.478	206
Intrepid IBM BlueGene@Argonne	163840	CPU	0.458	363
Ranger SUNOpteron Blasde@TACC	62976	CPU	0.433	217
Sandia Labs SUN BLade	41616	CPU	0.424	177

<http://www6.cityu.edu.hk/cityu25/events/engineering/pdf/profdongarra.pdf>

Failure Rate

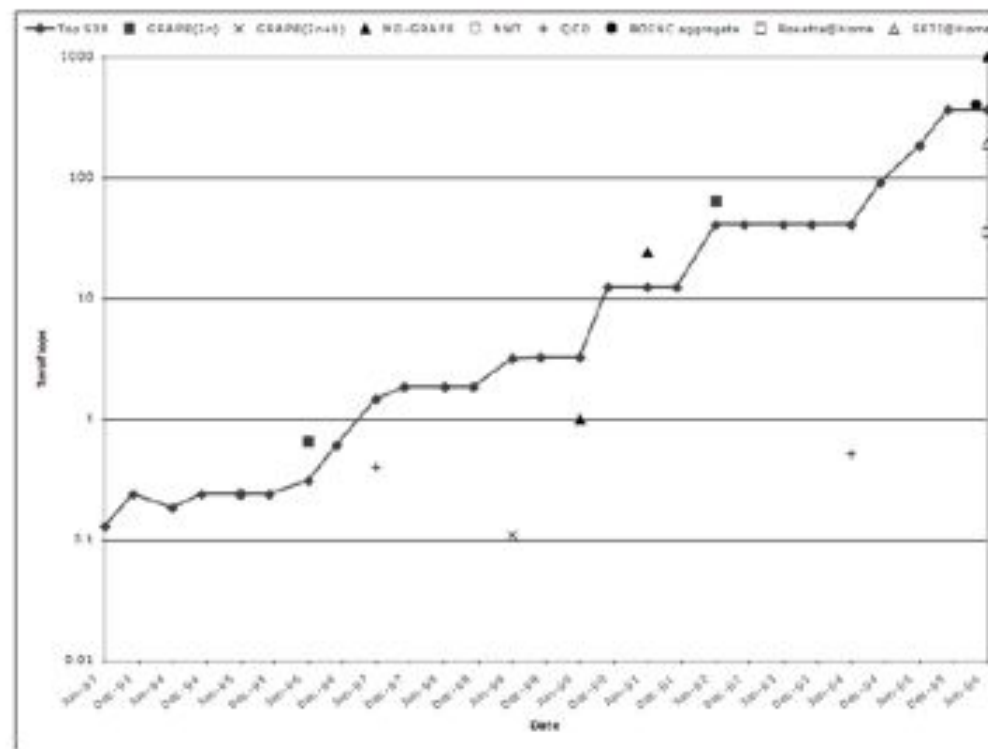
Jaguar, the current #1 system on the Top500 (as of Nov. 2009).

According to Jeff Veters at Oak Ridge National Labs, Jaguar has the following failure statistics:

Mean Time to Interrupt	32 hours
Mean Time to Failure	56 hours
Longest Up time for entire system	10 days

Failure here is also disk-replacement etc; meaning the machine is not usable

Dr. Jeff Layton, <http://www.delltechcenter.com/page/PetaFLOPS+for+the+Common+Man-+Pt+5+Challenges+with+PetaFLOPS+scale+systems>



GRAPE, NWT, BOINC

Fig. 1. Peak theoretical capacity of high performance computing systems over time. Shown are the peak theoretical capacity of the #1 ranked system on the Top500 List since its inception, along with the peak theoretical capability of selected special-purpose computing systems. Special-purpose systems represented include the Numerical Wind Tunnel, GRAPE family, MD-GRAPES, specialized QCD systems, and distributed BOINC applications [4], [5], [8], [10-19].

Progress Towards Petascale Applications in Biology:, Status in 2006, Craig A. Stewart, Matthias Mueller Malinda Lingwall

Table 1. Data about systems in Figure 1.

System	Classification	Peak theoretical capacity	Year	Reference
MDGRAPE-3	MD-GRAPe	1 PF	2006	4
BOINC combined statistics	BOINC aggregate	400.85 TF	2006	10
SETI@Home	SETI@Home	191.233 TF	2006	11
GRAPe-6	GRAPe(2n)	63.4 TF	2002	13
Rosetta@Home	Rosetta@Home	35.654 TF	2006	12
MDGRAPE-2	MD-GRAPe	24.6 TF	2001	14
MDGRAPE-2	MD-GRAPe	1 TF	2000	15
GRAPe-4	GRAPe(2n)	0.66 TF	1996	8
QCDOC	QCD	0.512 TF	2004	16
QCDSp	QCD	0.4 TF	1997	17
Numerical Wind Tunnel	NWT	0.2 TF	1995	18
GRAPe-5	GRAPe(2n+1)	0.11 TF	1999	19

Application	Discipline	Peak achieved rate	Year	Reference
Qbox	Physics	207.3 TF	2006	20
Solidification simulations	Physics	103 TF	2005	21
Peptide simulation	Biology/Molecular dynamics	55 TF	2006	22
Qbox	Physics	22.02 TF	2005	23
Corona simulation	Geology/Weather	15.6 TF	2006	24
Earth Simulator	Geology/Weather	15.2 TF	2004	25
LSMS	Physics	8 TF	2006	26
Weather forecast (NWS)	Geology/Weather	7.3 TF	2003	27
Earth Simulator	Geology/Weather	5 TF	2003	28
Lattice Boltzmann model	Fluid dynamics	4.7 TF	2005	29
Weather forecast (NOAA)	Geology/Weather	4 TF	2005	30
Blue Matter	Biology/Molecular dynamics	2.2 TF	2006	31
NAMD	Biology/Molecular dynamics	2.08 TF	2006	32
VASP	Physics	2 TF	2006	33
CPMD	Biology/Molecular dynamics	1.7 TF	2006	33
Wave propagation solver	Geology/Weather	1.21 TF	2003	34
Turbulence simulation	Fluid dynamics	1.18 TF	1999	35
DOWSER	Fluid dynamics	1 TF	2002	36

Scalability

I will show some speedup,
efficiency plot during this talk,
please try to project them to
let's say 1 million cores/
processors/nodes

Jülich Blue Gene/P Extreme Scaling Workshop, 2010,
Bernd Mohr and Wolfgang Frings, Jülich Supercomputing
Centre

Blue Gene/P system JUGENE which consists of 72 racks
with a total of 294,912 cores

Ten high-quality applications were selected (these are
the good codes)

MPI does not so easily scale!!

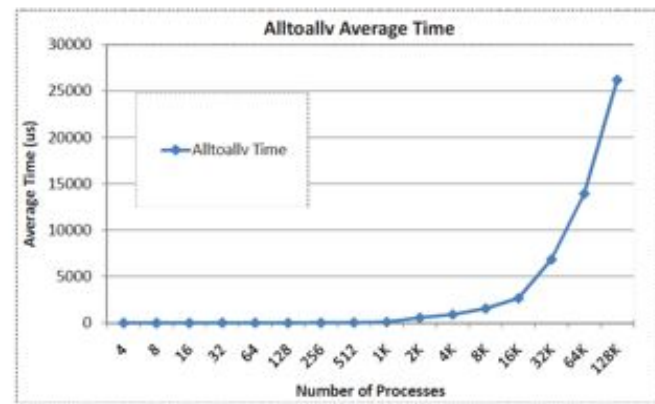


Fig. 1. Zero-byte Alltoallv time on IBM Blue Gene/P (no actual communication).

MPI is ready for scaling to a million processors barring a few issues that can be (and are being) fixed. Nonscalable parts of the MPI standard include irregular collectives and virtual graph topology. MPI implementations must pay careful attention to the memory requirements of functions and systematically root out data structures whose size grows linearly with the number of processes. To obtain scalable performance for collective communication, MPI implementations may need to become more topology aware or rely on global collective acceleration support.

MPI on a Million Processors, Pavan Balaji, Darius Buntinas, David Goodell, William Gropp, Sameer Kumar, Ewing Lusk, Rajeev Thakur, and Jesper Larsson Traeff, Proceedings of the 16th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface

Code1

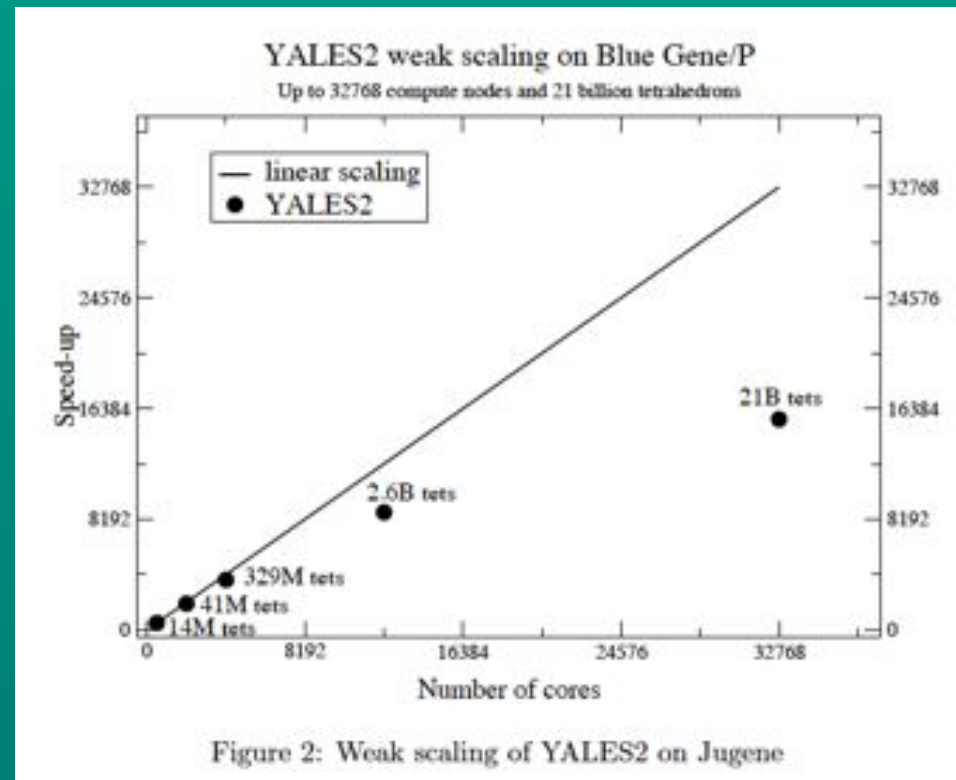
“To compile the program, the GNU C and C++ compiler in version 4.1.2 is used. Tests with the standard IBM XLC compiler showed a lower performance compared to the GNU compiler when compiling with flag -O2. Higher optimization was not possible due to internal compiler errors.”

“However, because of long runtimes when writing some output information on all nodes, we disabled all output besides the timing information for the benchmarking runs. Still, some of our full machine runs did not complete due to errors caused by exceeded MPI buffers.”

Test case	Lattice domain size per core	MLUPS	% of peak GFlop/s performance	% of peak memory bandwidth
A	40^3	3.0	4.12	14.61
	80^3	3.24	4.59	22.32
B	40^3	1.9	3.0	11.18
	80^3	1.92	3.15	15.63

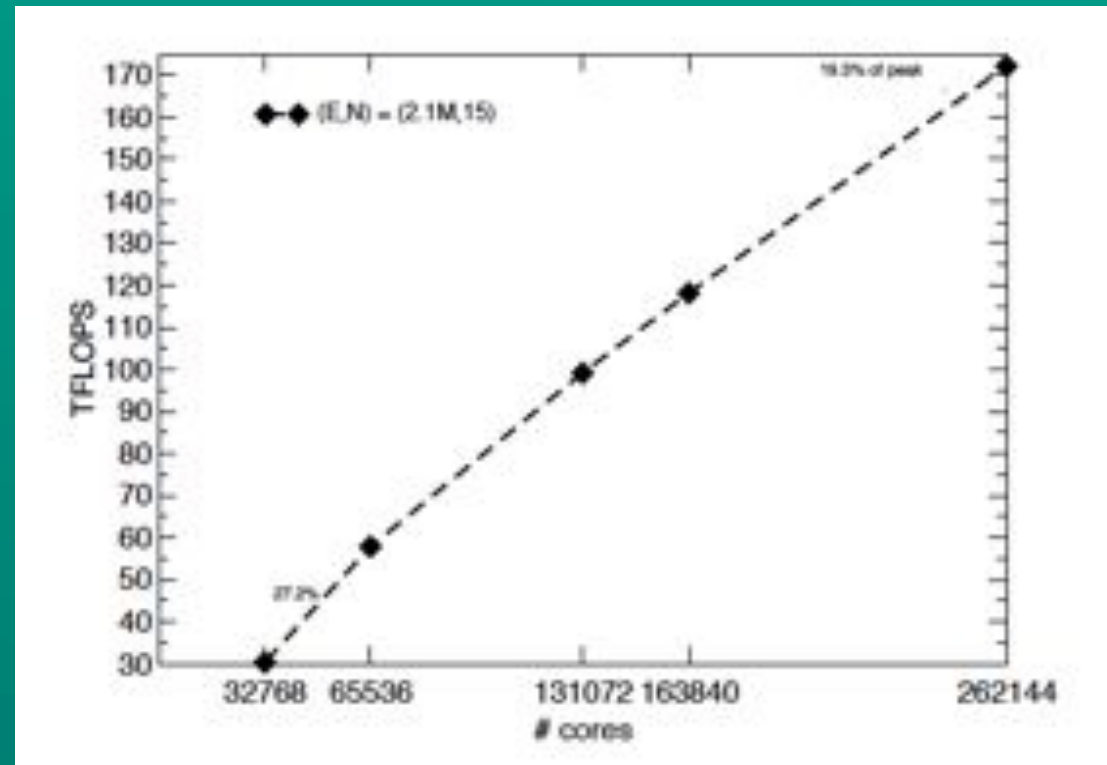
Table 1: Node performance of coupled fluid structure interaction simulations.

Code3

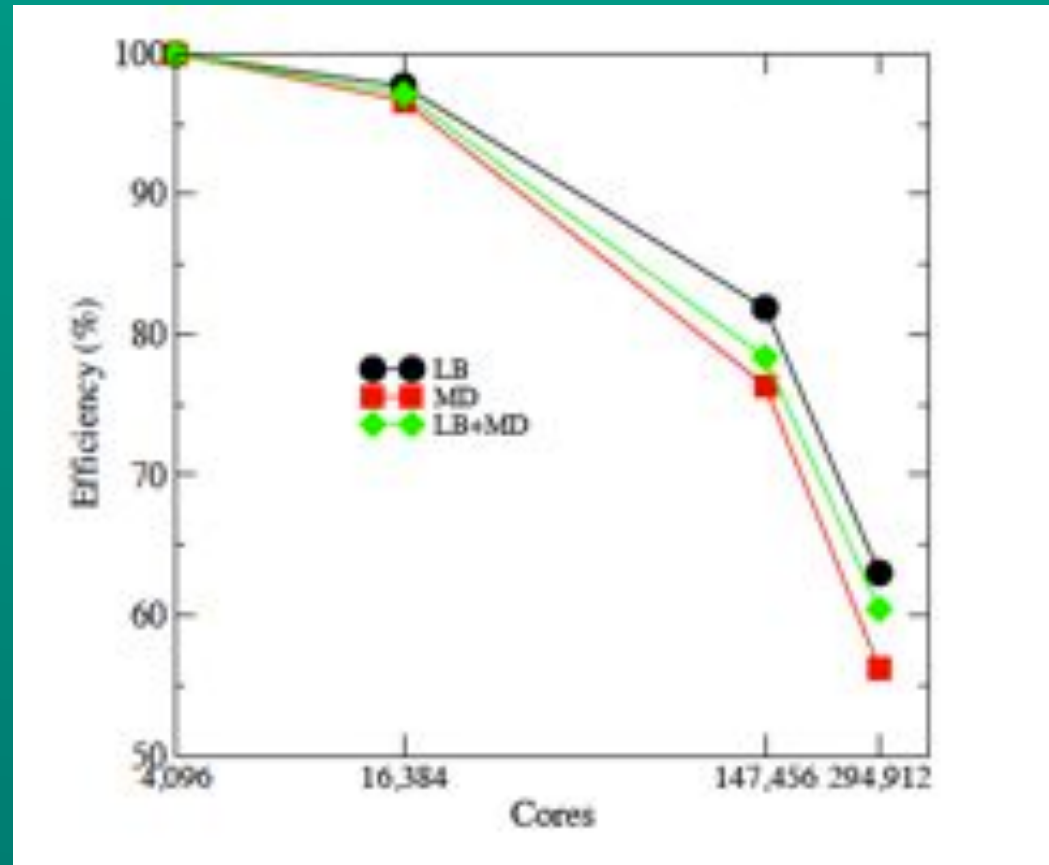


Code4

19,3% Peak



Code 5



Code 6

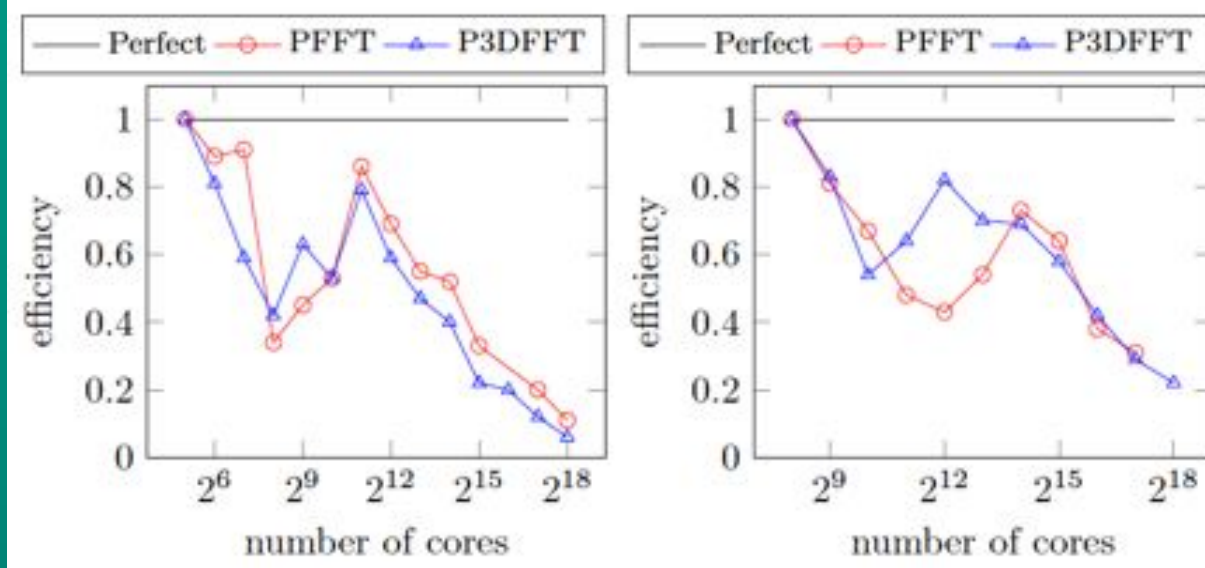
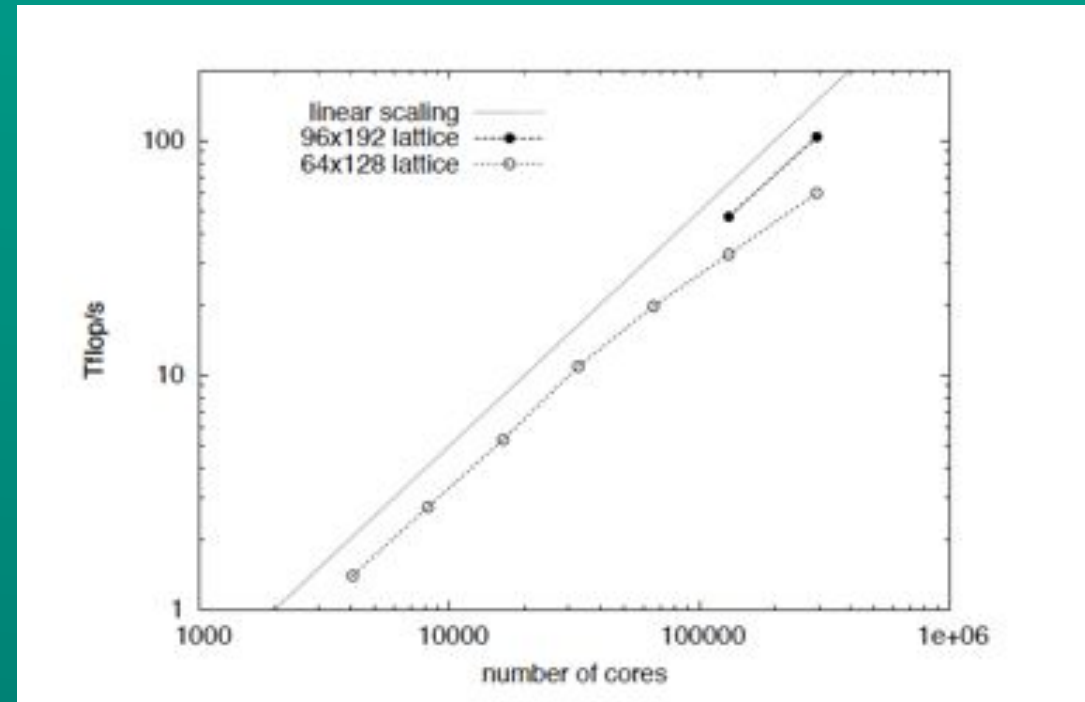


Figure 3: Efficiency for FFT of size 512^3 (left) and 1024^3 (right).

Code 7



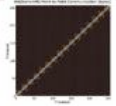
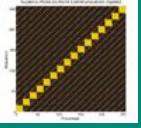
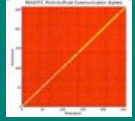
“The larger lattices scales well throughout. In this case we measured 10.4% of the peak performance on the full machine.”

“We measured I/O rates between 3 and 5 GByte/s”

Dwarfs

The dwarfs present a method for capturing the common requirements of classes of applications while being reasonably divorced from individual implementations.

Dwarfs from: The Landscape of Parallel Computing Research: A View From Berkeley

Dwarf	Description	Communication Pattern (Figure axes show processors 1 to 256, with black meaning no communication)	NAS Benchmark / Example HW
1. Dense Linear Algebra (e.g., BLAS [Blackford et al 2002], ScaLAP ACK [Blackford et al 1996], or MA TLAB [MathWorks 2006])	Data are dense matrices or vectors. (BLAS Level 1 = vector-vector; Level 2 = matrix-vector; and Level 3 = matrix-matrix.) Generally, such applications use unit-stride memory accesses to read data from rows, and strided accesses to read data from columns.	The communication pattern of MadBench, which makes heavy use of ScaLAP ACK for parallel dense linear algebra, is typical of a much broader class of numerical algorithms 	Block Triadiagonal Matrix, Lower Upper Symmetric Gauss-Seidel / Vector computers, Array computers
2. Sparse Linear Algebra (e.g., SpMV, OSKI [OSKI 2006], or SuperLU [Demmel et al 1999])	Data sets include many zero values. Data is usually stored in compressed matrices to reduce the storage and bandwidth requirements to access all of the nonzero values. One example is block compressed sparse row (BCSR). Because of the compressed formats, data is generally accessed with indexed loads and stores.	SuperLU (communication pattern pictured above) uses the BCSR method for implementing sparse LU factorization. 	Conjugate Gradient / Vector computers with gather/scatter
3. Spectral Methods (e.g., FFT [Cooley and Tukey 1965])	Data are in the frequency domain, as opposed to time or spatial domains. Typically, spectral methods use multiple butterfly stages, which combine multiply-add operations and a specific pattern of data permutation, with all-to-all communication for some stages and strictly local for others.	PARATEC: The 3D FFT requires an all-to-all communication to implement a 3D transpose, which requires communication between every link. The diagonal stripe describes BLAS-3 dominated linear-algebra step required for orthogonalization. 	Fourier Transform / DSPs, Zalink PDSP [Zalink 2006]

Dwarf	Performance Limit: Memory Bandwidth, Memory Latency, or Computation?
1. Dense Matrix	Computationally limited
2. Sparse Matrix	Currently 50% computation, 50% memory BW
3. Spectral (FFT)	Memory latency limited
4. N-Body	Computationally limited
5. Structured Grid	Currently more memory bandwidth limited
6. Unstructured Grid	Memory latency limited
7. MapReduce	Problem dependent
8. Combinational Logic	CRC problems BW; crypto problems computationally limited
9. Graph traversal	Memory latency limited
10. Dynamic Programming	Memory latency limited
11. Backtrack and Branch+Bound	?
12. Construct Graphical Models	?
13. Finite State Machine	Nothing helps!

Figure 9. Limits to performance of dwarfs, inspired by an suggestion by IBM that a packaging technology could offer virtually infinite memory bandwidth. While the memory wall limited performance for almost half the dwarfs, memory latency is a bigger problem than memory bandwidth

Blast is out

“A second thrust for the future of databases was in genetics, exemplified by the widely popular **BLAST** (Basic Local Alignment Search Tool) code. [Altschul et al 1990]

BLAST is a heuristic method used to find areas of DNA/protein sequences that are similar from a database. There are three main steps:

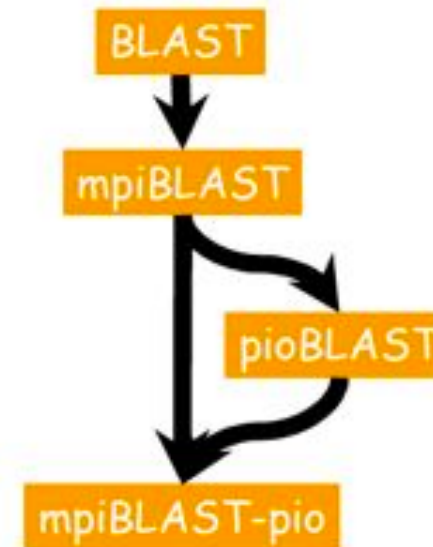
- Compile a list of high-scoring words from the sequence
- Scan database for hits from this list
- Extend the hits to optimize the match

Although clearly important, BLAST did not extend our list of dwarfs.”

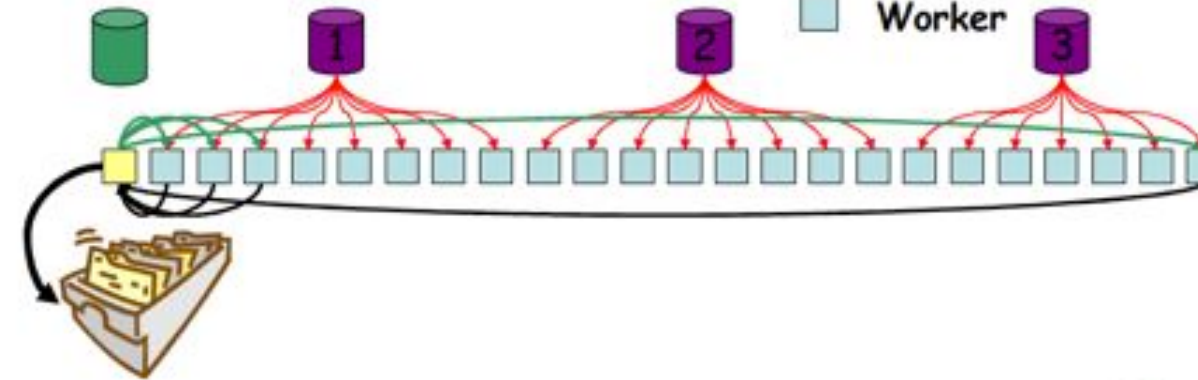
The Landscape of Parallel Computing Research: A View From Berkeley

BLAST examples

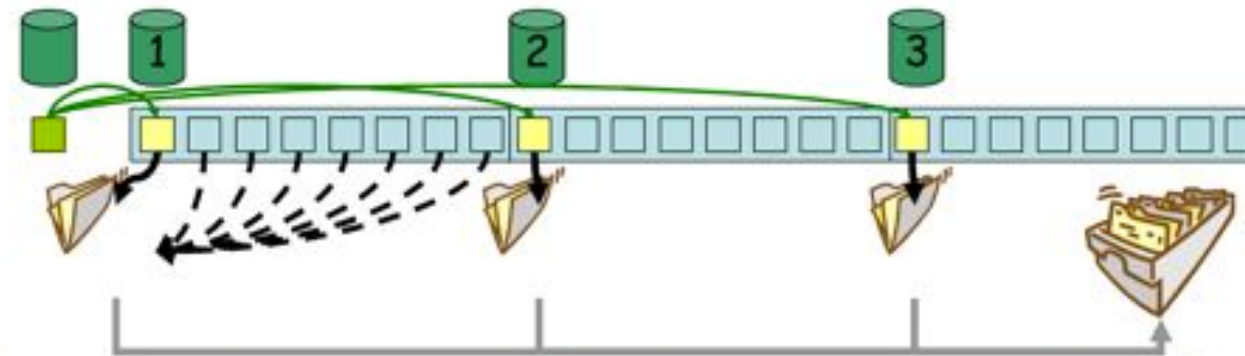
- o **mpiBLAST**
 - DB is partitioned and BLAST is executed in parallel
 - o **pioBLAST**
 - Uses parallel I/O to improve mpiBLAST
 - Dynamic (virtual) DB partitioning
 - Improved result merging
 - o **mpiBLAST-pio**
 - Incorporates the parallel-I/O performance enhancements of pioBLAST into mpiBLAST
-
- o A. Darling, L. Carey, and W. Feng. The design, implementation, and evaluation of mpiBLAST. In *Proceedings of the Cluster-World Conference and Expo, in conjunction with the 4th International Conference on Linux Clusters: The HPC Revolution*, 2003.
 - o H. Rangwala, E. Lantz, R. Musselman, K. Pinnow, B. Smith, , and B. Wallenfelt. Massively Parallel BLAST for the Blue Gene/L. In *High Availability and Performance Workshop*, 2005.
 - o C. Oehmen and J. Nieplocha. Scalablast: A scalable implementation of blast for high-performance data-intensive bioinformatics analysis. *IEEE Trans. Parallel Distrib. Syst.*, 17(8), 2006



- o Disk I/O overload
- o Master overworked

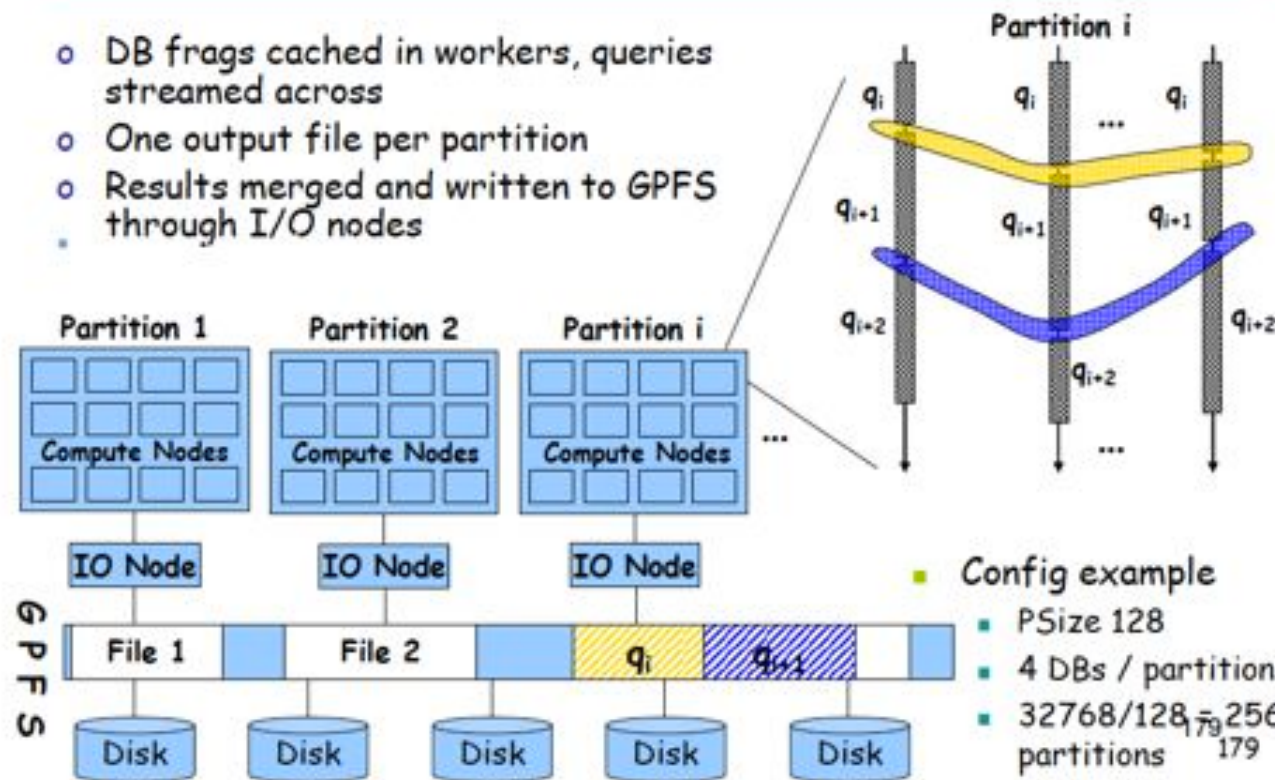


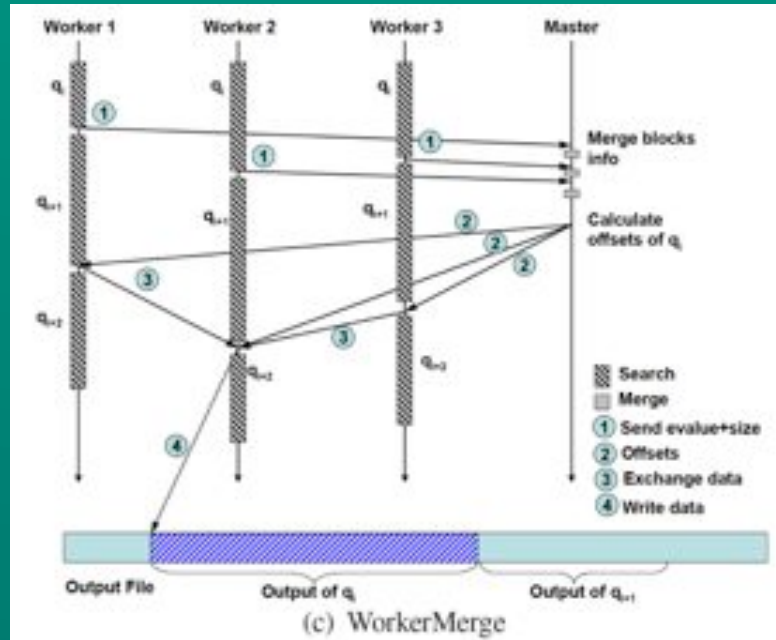
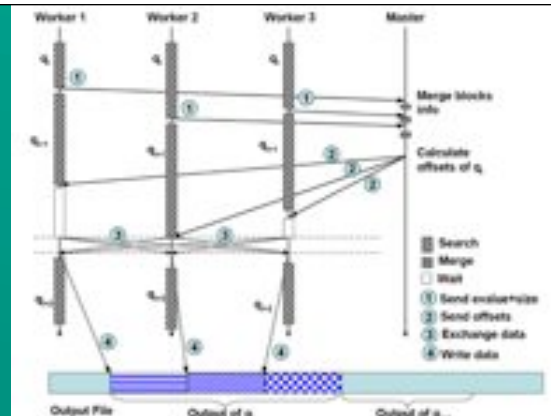
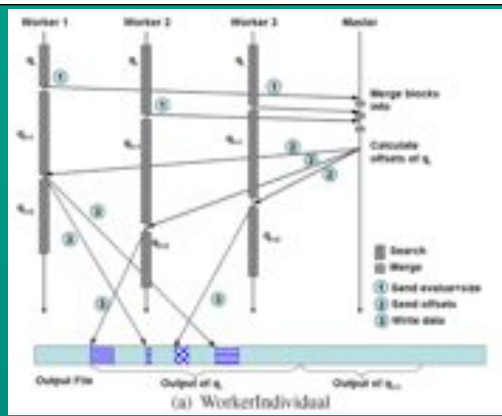
- o Query fragmentation
 - Load-balancing
- o Multiple output



• Oystein Thorsen, Karl Jiang, Amanda Peters, Brian Smith, Heshan Lin, Wu-chun Feng, Carlos P. Sosa, "Parallel Genomic Sequence-Search on a Massively Parallel System", Conference On Computing Frontiers Proceedings of the 4th international conference on Computing frontiers, Ischia, Italy, 59 - 68 (2007)

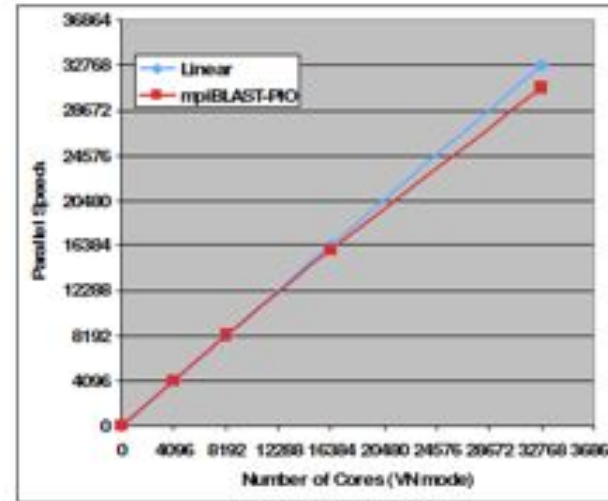
- DB frags cached in workers, queries streamed across
- One output file per partition
- Results merged and written to GPFS through I/O nodes







- o Self comparison of Microbial Genome database (5.2 GB raw size, 16 million sequences)



- o Scalability tests
 - Search a quarter million of randomly sampled sequences against the database itself
 - Achieve 93% parallel efficiency on 32768 cores (8-rack BG/P)
- o Complete genome-to-genome comparison
 - Finish searching 16 million vs. 16 million sequences within 12 hours

* H. Lin, P. Balaji, R. Poole, C. P. Sosa, X. Ma and W. Feng, "Massively Parallel Genomic Sequence Search on the Blue Gene/P Architecture," IEEE/ACM International Conference for High-Performance Computing, Networking, Storage and Analysis (SC), 2008

Different approach of Blast

- Sequence search all the 567 microbial genomes against each other in order to discover missing genes via mpiBLAST sequence-similarity computations, and
- Generate a complete genome sequence-similarity tree, based on the above sequence searching, in order to structure the sequence databases.
- Hardware connected over “Internet”

- 1) 2200-processor System X supercomputer at Virginia Tech.
- 2) 2048-processor BG/L supercomputer at Argonne National Laboratory.
- 3) 5832-processor Sicortex supercomputer at Argonne National Laboratory.
- 4) 700-processor Intel Jazz supercomputer at the Argonne National Laboratory.
- 5) A few hundred processors on the Teragrid system located at the San Diego Supercomputing Center and University of Chicago.
- 6) A few hundred processors located the Center for Computation and Technology located at Louisiana State University.
- 7) A few hundred processors on the Open Science Grid located at the Renaissance Computing Institute.
- 8) A few hundred processors on the Breadboard system at the Argonne National Laboratory.

I/O Resources

I/O Resources at the Tokyo Institute of Technology with support from Sun Microsystems. The details of this storage system are:

- 1) Clients: 10 quad-core SunFire X4200 and 2 16-core SunFire X4500 systems
- 2) Object Storage Servers (OSS): 20 SunFire X4500
- 3) Object Storage Targets (OST): 140 SunFire X4500 (each OSS has 7 OST)
- 4) RAID configuration for OST: RAID5 with 6 drives
- 5) Network: 1 Gigabit Ethernet
- 6) Kernel: 2.6
- 7) Lustre Version: 1.6.2

ParaMEDIC: Parallel Metadata Environment for Distributed I/O and Computing, P. Balaji§ W. Feng¶ J. Archuleta¶ H. Lin (Storage Challenge, Supercomputing 2007)

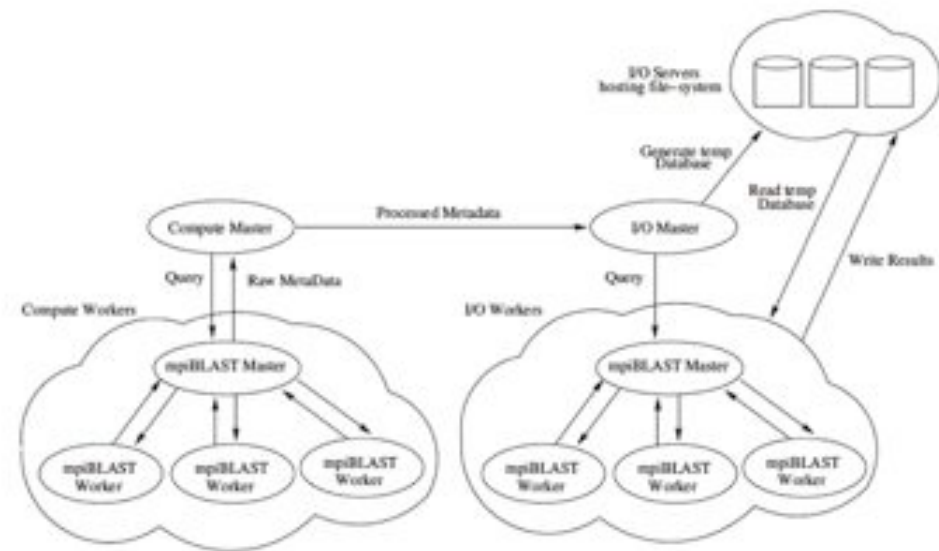


Fig. 3. The ParaMEDIC Framework

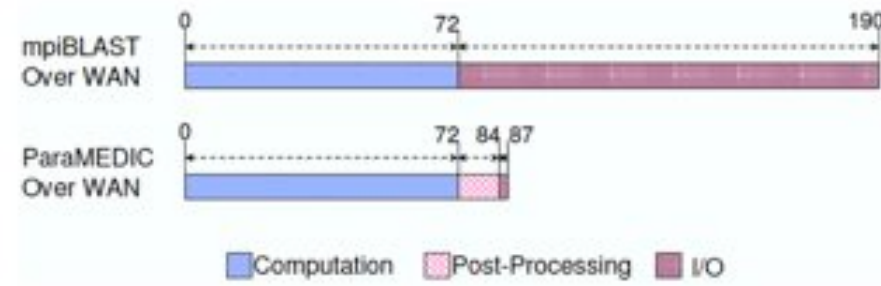


Fig. 4. Execution breakup for running mpiBLAST and ParaMEDIC over WAN with 100ms network latency. The performance numbers are measured in seconds.

Other alternatives?

- Cloud computing
- Mapreduce
- **example:** CloudBurst: highly sensitive read mapping with MapReduce
- **Sorting (<http://sortbenchmark.org/>)**

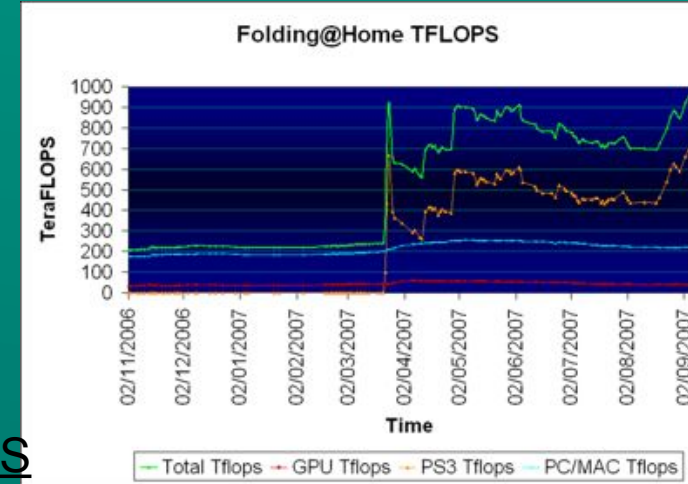
Hadoop

100 TB in 173 minutes
3452 nodes x (2 Quadcore Xeons, 8 GB memory, 4 SATA)
[Owen O'Malley](#) and [Arun Murthy](#), Yahoo Inc.

- **BOINC**
- **Specialist hardware like GRAPE, ANTON**

BOINC projects

- Folding@Home is, as of April 2010, sustaining over 6.2 PFLOPS. This level of performance is primarily enabled by the cumulative effort of a vast array of PlayStation 3, CPU, and powerful GPU units.
- The entire BOINC network averages about 5.1 PFLOPS as of April 21, 2010.
- As of April 2010, MilkyWay@Home computes at over 1.6 PFLOPS, with a large amount of this work coming from GPUs.
- As of April 2010, SETI@Home computes data averages more than 730 TFLOPS.
- As of April 2010, Einstein@Home is crunching more than 210 TFLOPS.
- As of April 2010, GIMPS is sustaining 44 TFLOPS.



<http://en.wikipedia.org/wiki/FLOPS>

Formula 1 car restrictions

2,4-Liter-V8-engines (currently, lot's of changes over time)

Fuel: super lead free

size of tank

no ABS

number of tires per race

engines must be good for 2 race

lots of aerodynamic restrictions

==> still a lot of innovations to overcome restrictions

What would happen if one would introduce restrictions on supercomputers (no public funding)?

- Flops per Watt
- total energy consumption
- 10-15 Production/Applications must reach 60+ of peak performance
-

How will an Exascale = Exaflop machine look like

- Estimated arrival 2018-2020
- Power consumption: 20 MW
(fixed due to political financing reasons, Jaguar has 7MW)
- Costs: 200 million per machine
(fixed due to political financing reasons)
- 50% of costs will go into memory (32-64 PB)
- Concurrency: 10-100-1000 million nodes/cores/threads
- Systemwide latencies on the order of tens of thousands of cycles
- Clock speed reduced to approx. 1 GHz
- Hardware reliability/stability: lower than today
- Software: disruptive technology ==> unclear

source: Jack Dongarra

- "...the hardware path, unlike the software path, is clear." (Jack Dongarra)*
- "There are actual applications running on Jaguar (1.7 PFlops) in the fields of material science and nanotechnology that exceed a petaflop. Unfortunately, only a handful applications today can get that close to the petaflop performance. " (Jack Dongarra)*
- "The way we write programs and develop software is typically slow to change but it will have to, and soon, as this change is upon us right now. It's causing a little bit of concern in the community as we understand the complexity and magnitude of this paradigm shift" (Pete Beckman, co-leader of the International Exascale software project, Director of Argonne's Exascale Computing and Technology Institute)**
- "Parallel machines are already hard to program and if you scale everything up and keep it as business as usual it's going to be intractable" (Pete Beckman, co-leader of the International Exascale software project, Director of Argonne's Exascale Computing and Technology Institute)**
- "The target is to have exascale systems developed and operational before 2020, but with the software challenge in mind, that has to be regarded as a 'soft' target" (Thom Dunning, Director National Center for Supercomputing Applications (NCSA)**
- "I have been convinced by my colleagues in the vendor community that reaching exascale within the next 10 years is not feasible, but desirable. (Stanley C. Ahalt, Director RENC, NC)**
- "...but unfortunately I do think this first milestone (ExaFlop) will be somewhat meaningless. It will be a benchmark - and not much more.....more likely to have a lag between the first exascale benchmarks, and the first application to actually take advantage of such a system - by several years" (Mike Bernhardt, Reporter, The Exascale Report)**

*<http://nextbigfuture.com/2010/06/jack-dongarra-interview-by-sander-olson.html>

** Scientific Computing World, Feb.-Mar. 2011, Issue 116

Genomics applications

-are data driven and have a high I/O requirement
-do not scale beyond a few hundred cores
-are suitable for the “Cloud” (increasing number of data sources can be find in the cloud)
-have a high (shared) memory demand
-are best suited for cluster systems
-are “new”; constantly under development
-are first implemented in scripting languages
- Programmers have little experience/access with HPC systems

My personal conclusions:

If a Exaflop machine will come up in the next 5-8 years it will be probably useless for any practical application, at least for most of the biological problems.

Biology needs balanced systems and not Linpack
ExaFlops!

END