EULAG PARALLELIZATION AND DATA STRUCTURE

Andrzej Wyszogrodzki
NCAR

Parallelization - methods

Shared Memory (SMP) :

- Automatic Parallelization
- Compiler Directives (OpenMP)
- Explicit Thread Programming (Pthreads, SHMEM)

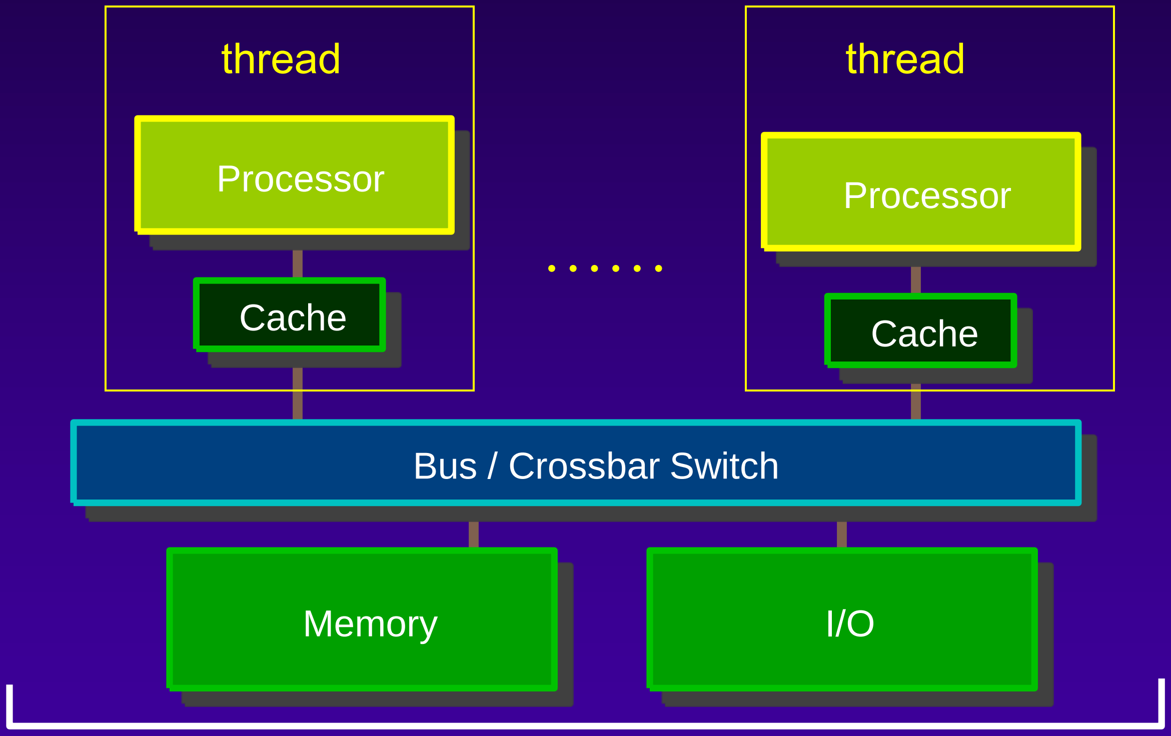
Distributed Memory (DMP) / Massively Parallel Processing (MPP) :

- PVM – currently not supported
- SHMEM – Cray T3D, Cray T3E, SGI Origin 2000
- MPI – highly portable

Hybrid Models: MPI+OpenMP

SMP Architecture

- Common (shared) memory for tasks communication (threads).
- Memory location fixed during task access
- Synchronous communication between threads.



All computational threads in a group belong to a single process

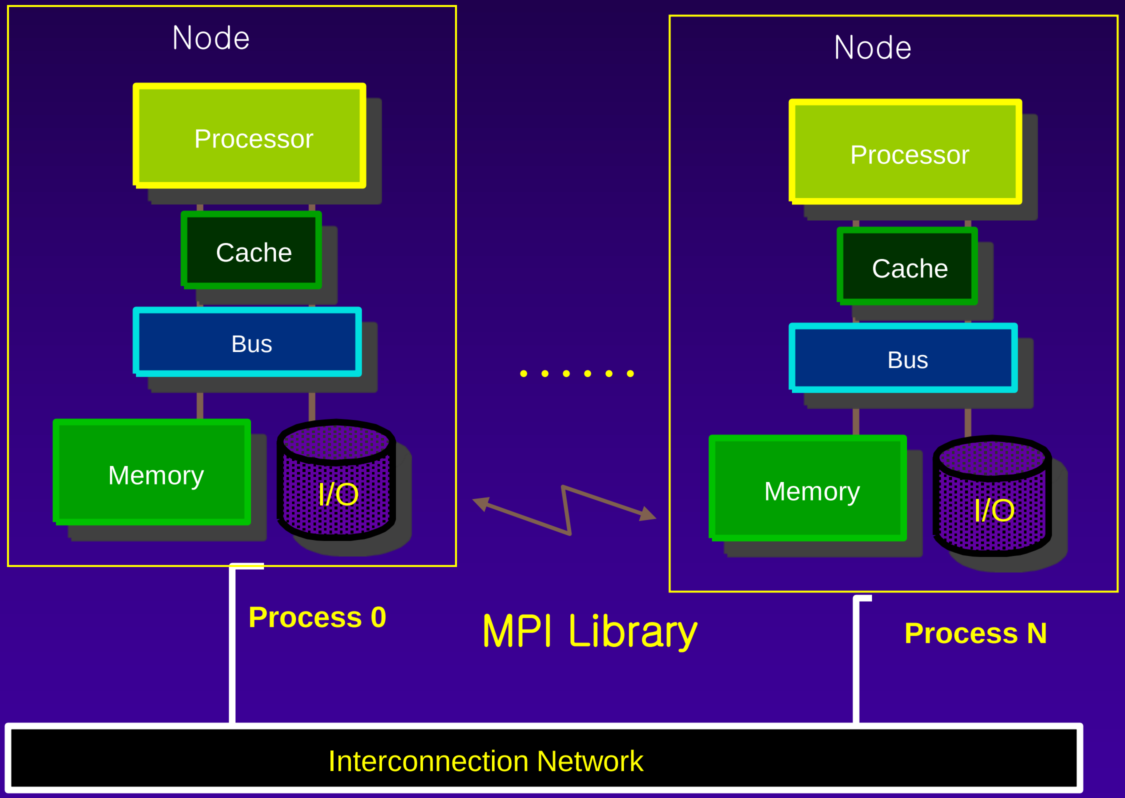
Process

Performance and scalability issues:

- Synchronization overhead
- Memory bandwidth

MPP Architecture

- Each node has its own memory subsystem and I/O.
- Communication between nodes via Interconnection network
- Exchange message packets via calls to the MPI library



- Each task is a Process.
- Each Process Executes the same program and has its own address space
- Data are exchanged in form of message packets via the interconnect (switch, or shared memory)

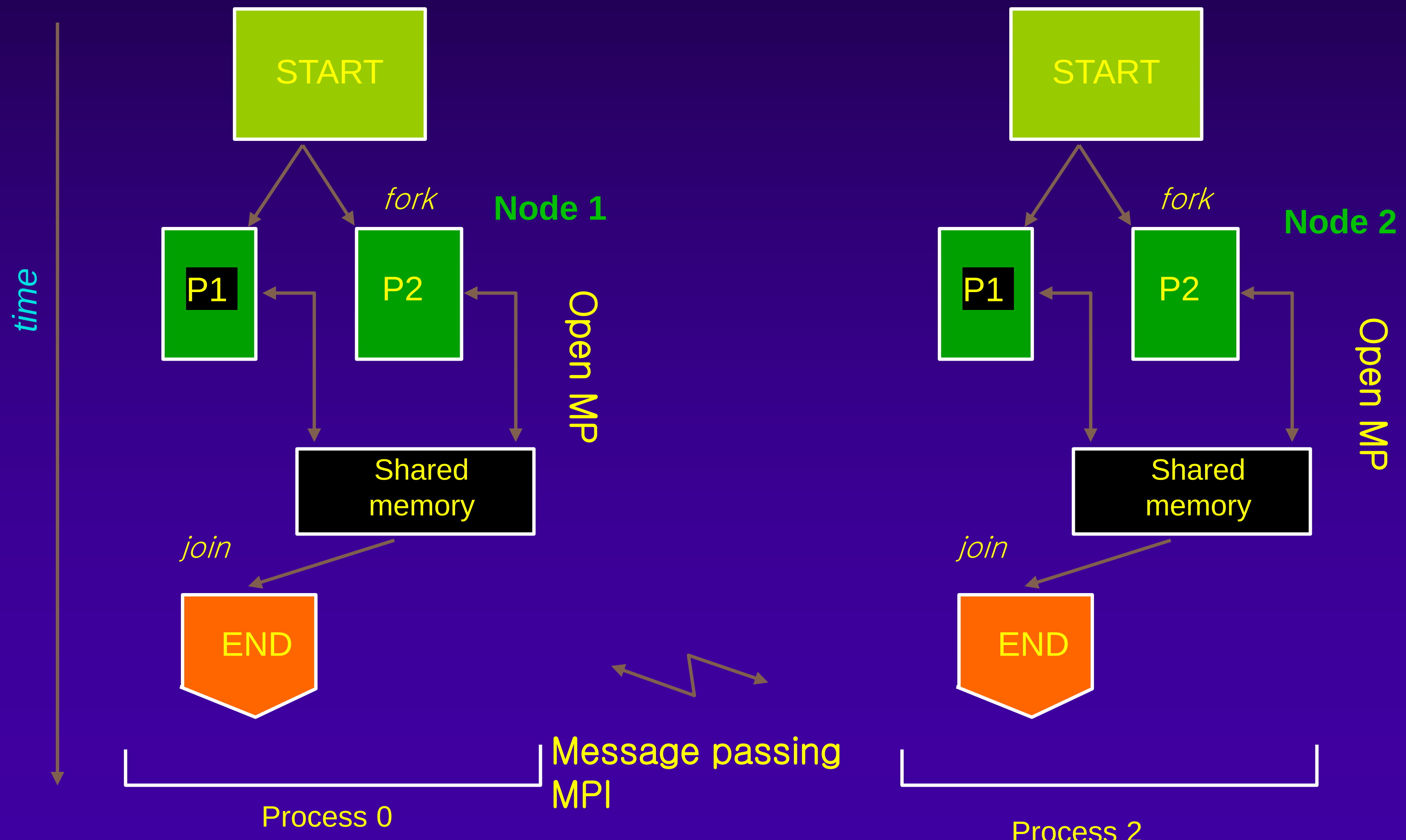
- Performance and scalability issues:
  - Overhead ~ to the size and number of packets
  - Good scalability on large processor systems.

Multithread tasks per node

Optimize performance on "mixed-mode" hardware (e.g. IBM SP, Linux Superclusters)

Optimize resource utilization (I/O)

- MPI is used for "Inter-node" communication,
- Threads (OpenMP / Pthreads) are used for "Intra-node" communication



OpenMP

Components to specify shared memory parallelism:

- Directives
- Runtime Library
- Environment Variables

EXAMPLE:

```
!$OMP PARALLEL DO PRIVATE (I)
      do i=1,n
         a(i) = a(i)+1
       end do
 !$OMP END PARALLEL DO
```

PROS:

- Portable / multi-platform working on major hardware architectures
- Systems including UNIX and Windows NT
- C/C++ and FORTRAN implementations
- Application Program Interface (API)

CONS:

- Scoping - variables in a parallel loop: private or shared?
- Parallel loops may calls subroutines, include many nested do loops
- Non parallelizable loops - automatic compiler parallelization?
- Not easy to get optimal performance:
  Effective use of directives, code modification, new computational algorithms
- Need to reach more than 90% of parallelization to hope for good speedup

Message Passing Interface - MPI

MPI – library, not a language
Library of around 100 subroutines (… most codes uses less than 10)

Message-passing: collection of processes communicating via messages:

- Collective or global - group of processes exchanging messages
- Point-to-point - pair of processes communicating with each other

MPI 2.0 standard released in April 1997, extention to MPI 1.2

- Dynamic Process Management (spawn)
- One-sided Communication (put/get)
- Extended Collective Operations
- External Interfaces
- Parallel I/O (MPI-I/O)
- Language Bindings (C++ and FORTRAN-90)

Parallelization strategies:

- Choose data decomposition / domain partition
- Map model sub-domains to processor structure
- Check data load balancing
- Use parallel algorithms if possible (e.g. parallel FFT)
- Set up Communication

MPP vs SMP

	Advantages	Disadvantages
Compiler	- Very easy to use - No rewriting of code	- Marginal performance - Loop level parallelization
Open MP	- Easy to use - Limited rewriting of code - OpenMP - standard	- Average performance
MPI	- High performance - Portable - Scales outside a Node	- Extensive code rewriting - May have to change algorithm - Communication overhead - Dynamical load balancing

EULAG PARALLELIZATION

ISSUES:

- Data partitioning
- Load balancing
- Code portability
- Parallel I/O
- Debugging
- Performance profiling

HISTORY:

Compiler parallelization – 1996-1998, Vector Crays J90 at NCAR
MPP/SMP – PVM/SHMEM version at Cray T3D (W. Anderson 1996)
MPP – use MPI & porting SHMEM to 512 PE Cray T3E at NERSC (Wyszogrodzki 1997)
MPP – porting EULAG on number of systems HP, SGI, NEC, Fujitsu, 1998-2005
SMP – attempt to use OpenMP by M. Andrejczuk ~ 2004 ???
MPP – porting EULAG on BG/L at NCAR and BG/W IBM Watson in Yorktow Heights

CURRENT STATUS:

PVM – not supported anymore, no systems available with PVM
SHMEM – partially supported (global, point to point), no systems currently available
MPI – fully supported and developed
Open MP – not supported, planned for future development

EULAG PORTABILITY

PREVIUS IMPLEMENTATIONS:

Serial processor workstations: Linux, Unix

Vector computers with automatic compiler parallelizations: Crays J90, ….

MMP systems: Cray t3D, Cray T3E (NERSC 512 PE), HP Exemplay, SGI Origin 2000, NEC (ECMWF), Fujuttsu

SMP systems: Cray t3D, Cray T3E , SGI Origin 2000, IBM SP

Recent systems at NCAR (last 3 years):

IBM power4 BG/L 2048 CPUs (frost)

IBM power6 4048 CPUs (bluefire) 76.4 TFp/s,TOP#25?

IBM p575+ 1600 CPUs (blueice)

IBM p575 576 CPUs (bluevista)

IBM p690 1600 CPUs (bluesky)

Other recent supercomputers:

IBM power4 BG/W 40000 CPUs (Yorktown Heights)

l'Université de Sherbrooke - Réseau Québécois de Calcul de Haute Performance (RQCHP):

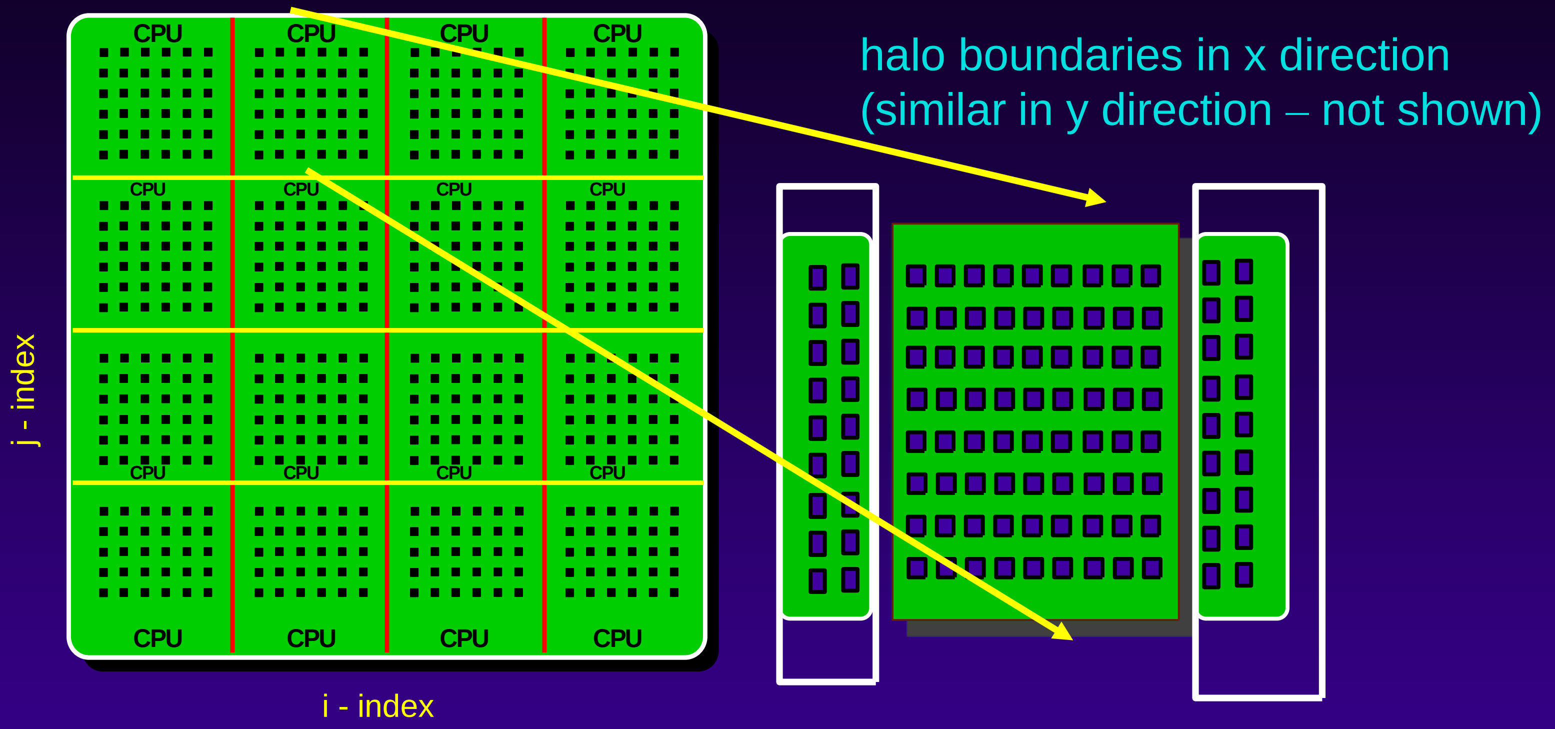
Dell 1425SC Cluster

Dell PowerEdge 750 Cluster

PROBLEMS:

Linux clusters, different compilers, no EULAG version working currently in double precision

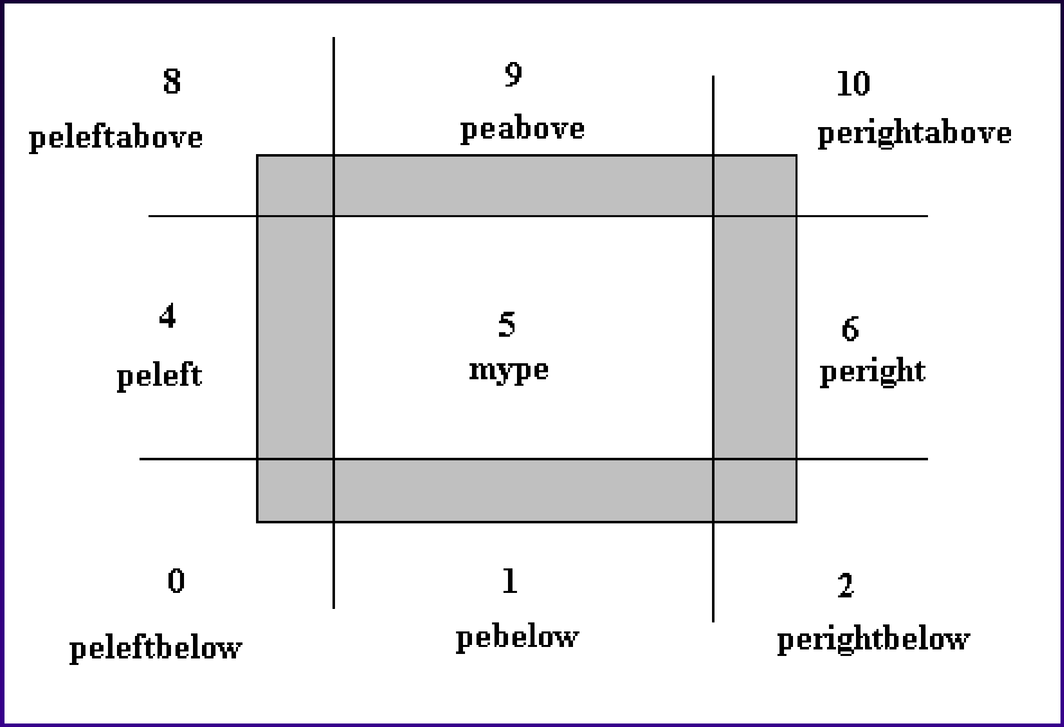
Data decomposition in EULAG



halo boundaries in x direction
(similar in y direction – not shown)

- 2D horizontal domain grid decomposition
- No decomposition in vertical Z-direction
- Hallo/ghost cells for collecting information from neighbors
- Predefined halo size for array memory allocation
- Selective halo size for update to decrease overhead

Typical processors configuration



8 peleftabove

9 peabove

10 perightabove

4 peleft

5 mype

6 peright

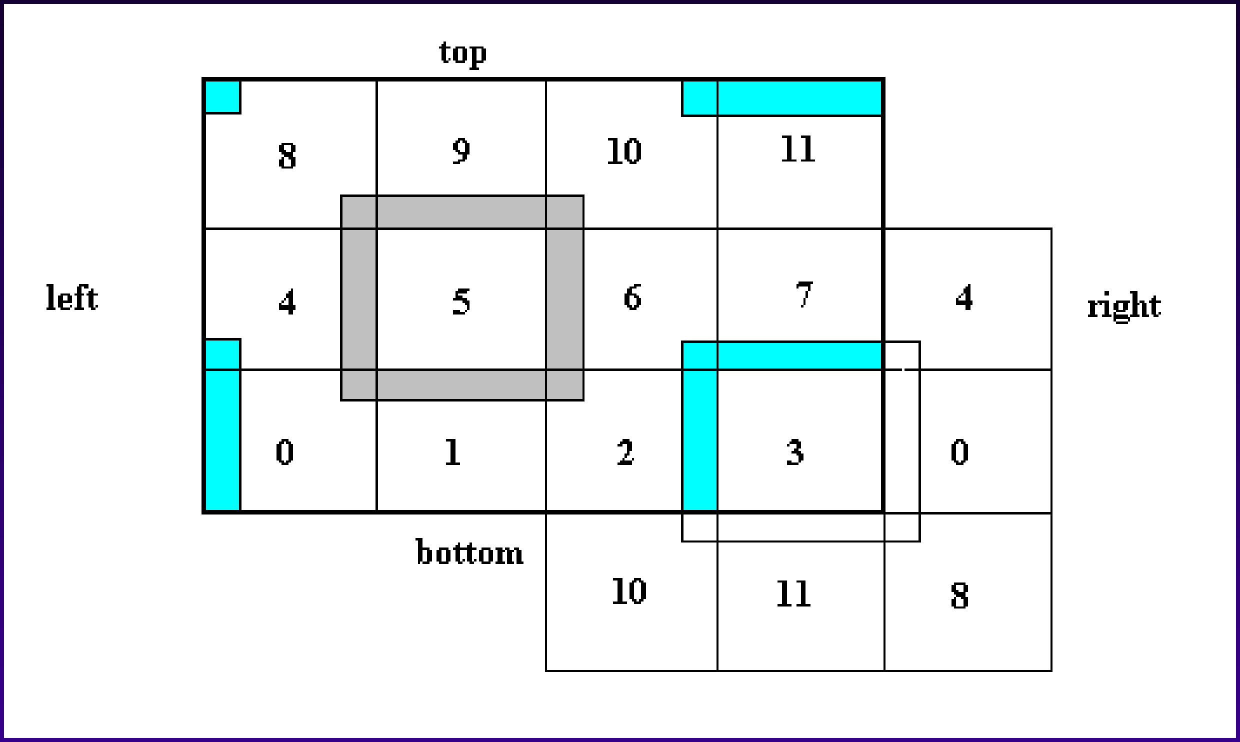
0 peleftbelow

1 pebelow

2 perightbelow

- Computational 2D grid is mapped onto an 1D grid of processors
- Neighboring processors exchange messages via MPI
- Each processor know its position in physical space (column, row, boundaries) and location of neighbor processors

EULAG – Cartesian grid configuration



← In the setup on the left

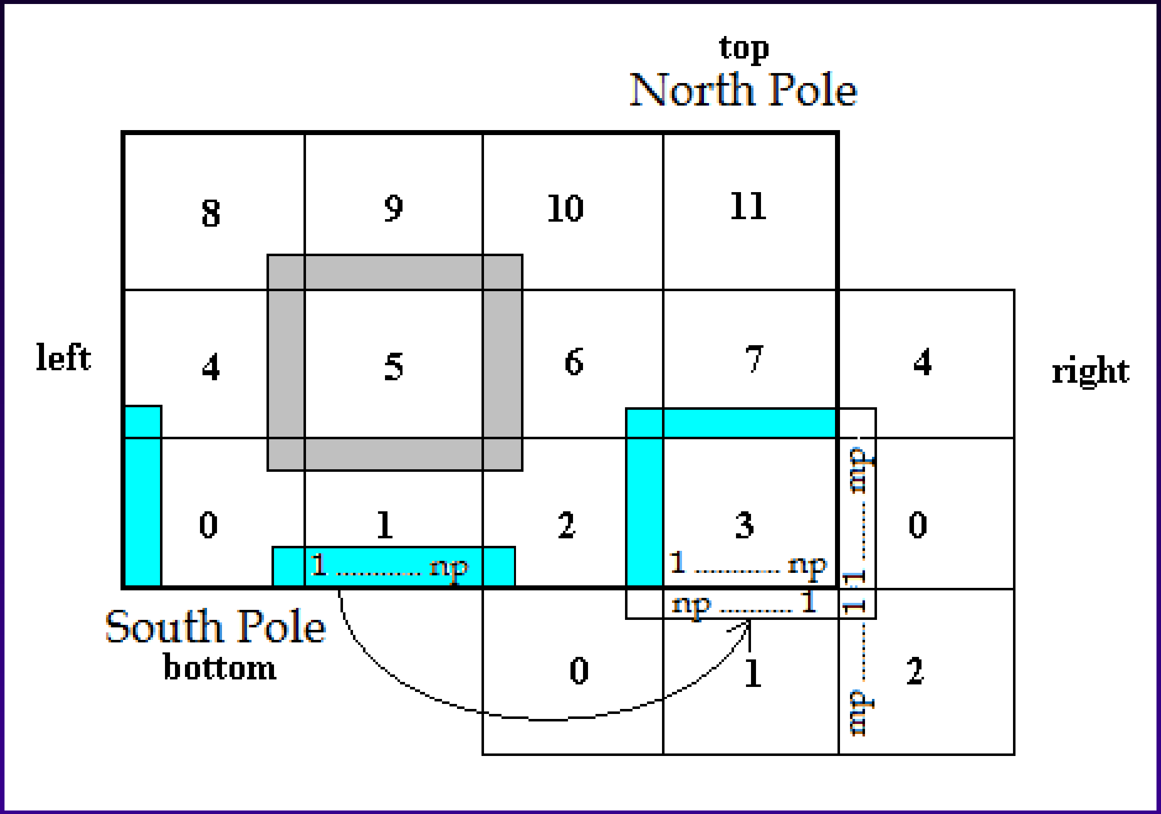
- nprocs=12
- nprocx = 4, nprocy = 3
- if np=11, mp=11

  then full domain size is

  N x M = 44 x 33 grid points

- Parallel subdomians ALWAYS assume that grid has cyclic BC in both X and Y !!!
- In Cartesian mode, the grid indexes are in range: 1…N, only N-1 are independent !!!
- F(N)=F(1) –> periodicity enforcement
- N may be even or odd number but it must be divided by number of processors in X
- The same apply in Y direction.

EULAG Spherical grid configuration with data exchange across the poles



← In the setup on the left

- nprocs=12
- nprocx = 4, nprocy = 3
- if np=16, mp=10

then full domain size is

N x M = 64 x 30 grid points

- Parallel subdomians in longitudinal direction ALWAYS assume that grid has cyclic BC !!!
- At the poles processors must exchange data with appropriate across the pole processor.
- In Spherical mode, there is N independent grid cells F(N)≠ F(1) … required by load balancing and simplified exchange over the poles -> no periodicity enforcement
- At the South (and North) pole grid cells are placed at Δy/2 distance from the pole.

MPI point to point communication functions

	BLOCKING	NONCKING
standard	send	isend
buffered	bsend	ibsend
synchronous	ssend	issend
ready	rsend	irsend

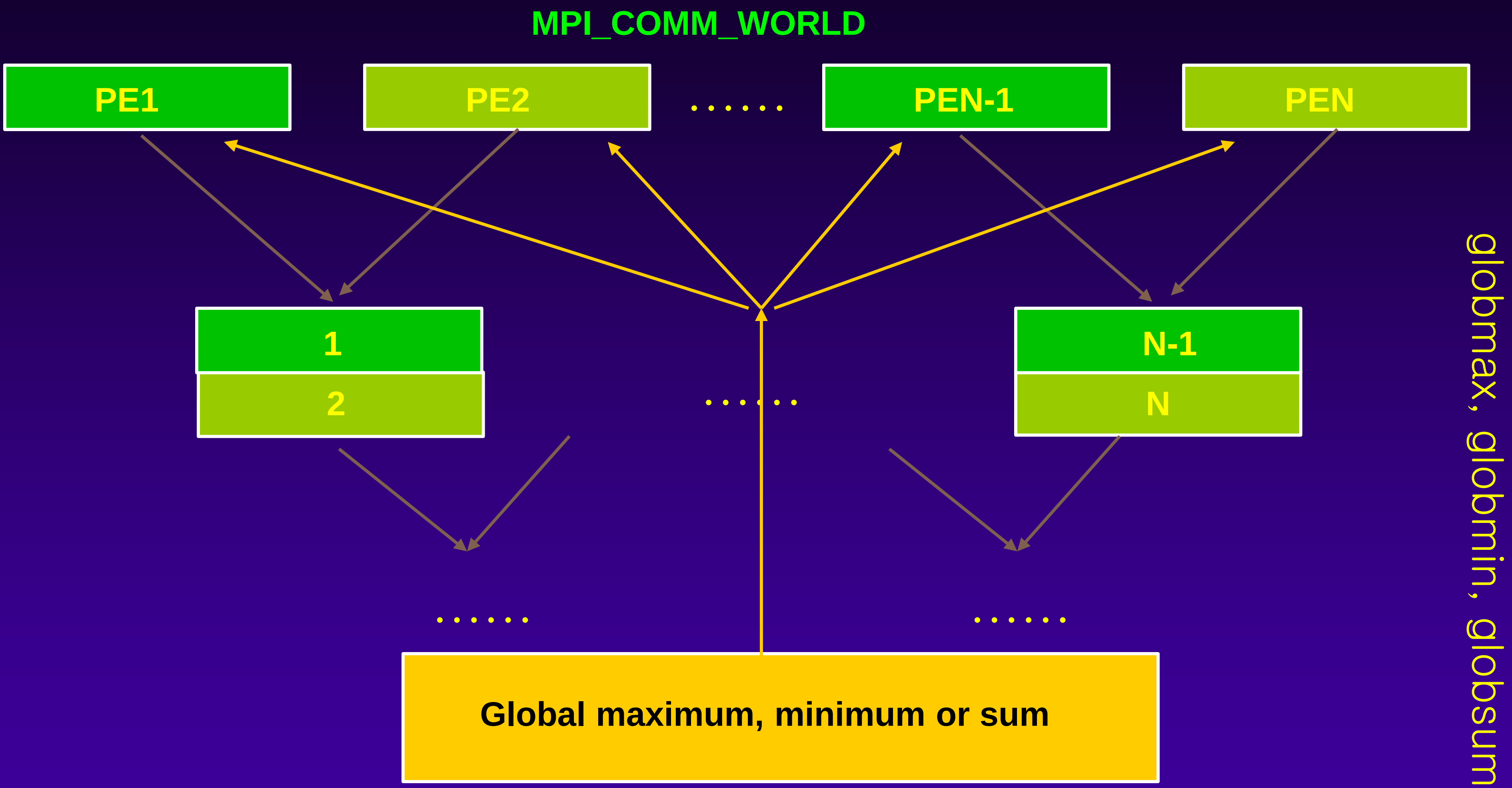
send_recv +
8 different
types
send/recv

- Blocking: Processor sends and waits until everything is received.
- Nonblocking: Processor sends and does not wait for data to be received.

MPI collective communication functions

- broadcast
- gather
- scatter
- reduction operations
- all to all
- barrier synchronization point between all MPI processes

EULAG reduction subroutines



EULAG I/O

Requirements of I/O Infrastructure:

- Efficiency
- Flexibility
- Portability

I/O in EULAG

- full dump of model variables in raw fortran binary format
- short dump of basic variables for postprocessing
- Netcdf output
- Parallel Netcdf
- Vis5D output in parallel mode
- MEDOC (SCIPUFF/MM5)

PARALLEL MODE

- PE0 collects all sub-domains and save to hard drive
- Memory optimization in parallel mode (sub-domains are sequentially saved without creating single serial domain, require reconstruction of the full domain in post processing mode)

CONS: full output need to be self-defined, lack of time stamps

Performance and scalability

Weak Scaling

- Problem size/proc fixed
- Easier to see Good Performance

- Beloved of Benchmarkers, Vendors, Software Developers –Linpack, Stream, SPPM

Strong Scaling

- Total problem size fixed.
- Problem size/proc drops with P

- Beloved of Scientists who use computers to solve problems. Protein Folding, Weather Modeling, QCD, Seismic processing, CFD

EULAG SCALABILITY

NCAR's IBM POWER 5 SMP

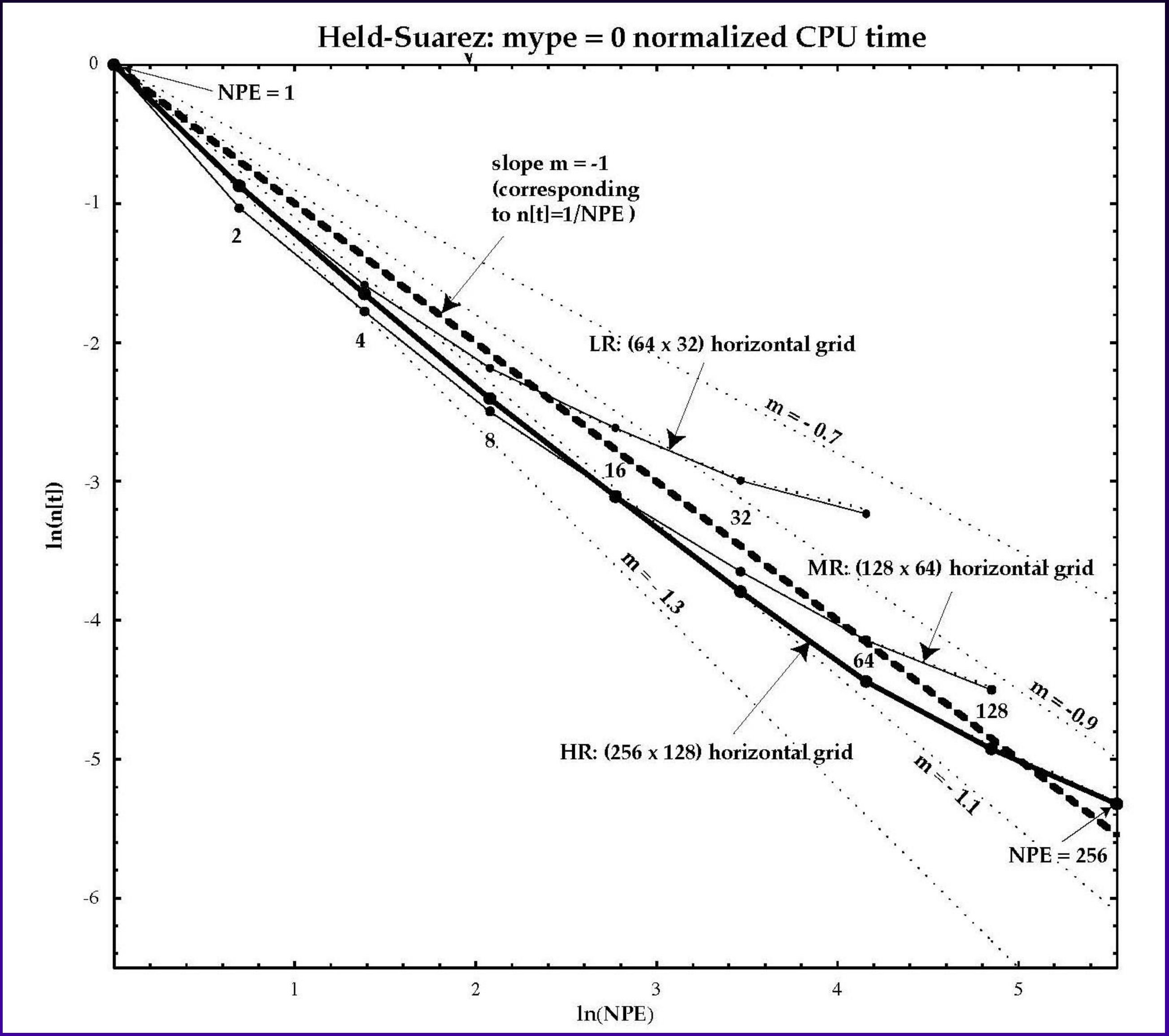
Grid sizes:

LR (64x32)
MR (128x64)
HR (256x128)

Each test case use the same number of vertical levels (L=41).

- **Bold dashed line - ideal scalability, wall clock time scales like 1 / *NPE*.**
- **Excellent scalability up to number of processors *NPE=sqrt*(*N*M*): 16 PE's (LR) 64 (MR), 256 (HR)**
- **Max speedups - 20x; 90x; 205x**
- **Performance sensitive to the particular 2D grid decomposition**

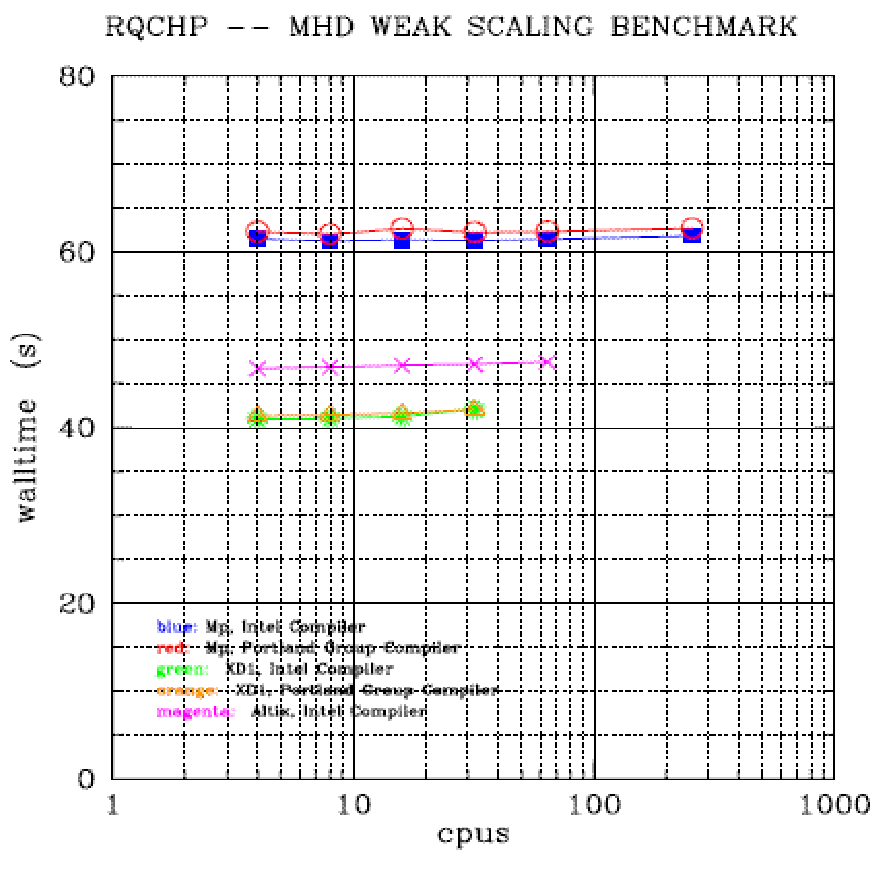
Held-Suarez test on the sphere and Magneto-Hydrodynamic (MHD) simulations of the solar convection

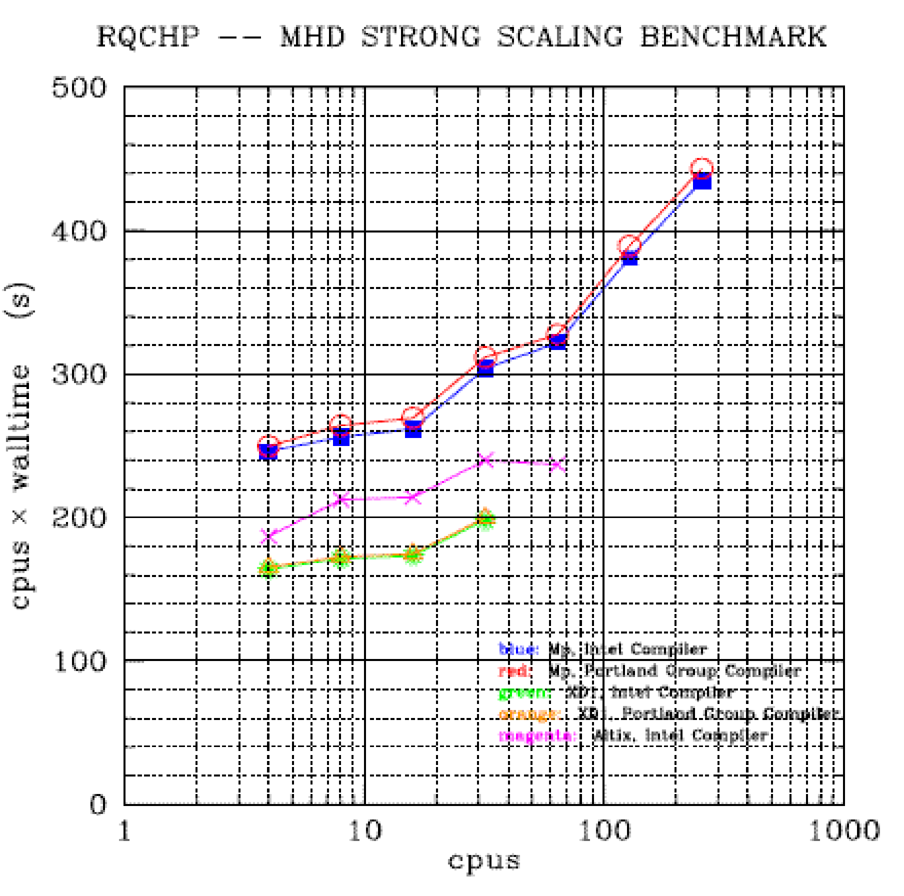


weakening of the scalability is due to increased ratio of the amount of information required to be exchanged between processors to the amount of local computations

EULAG SCALABILITY

Benchmark results from the Eulag-MHD code at
l'Université de Sherbrooke - Réseau Québécois de Calcul de Haute Performance (**RQCHP**),
Dell 1425SC and Dell PowerEdge 750 Clusters



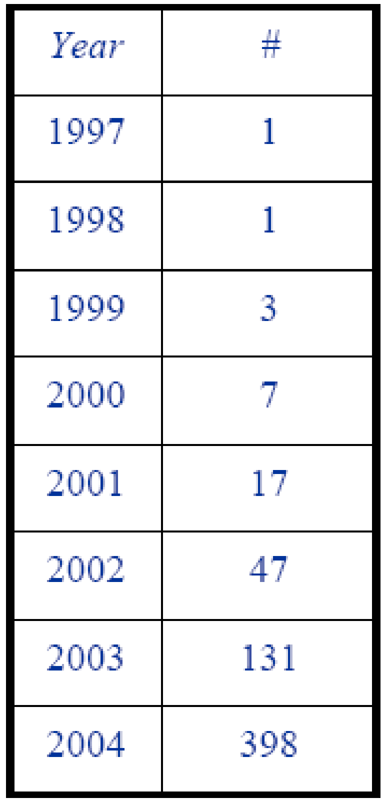


Curves corresponding to different machines and two compilers running on the same machine.

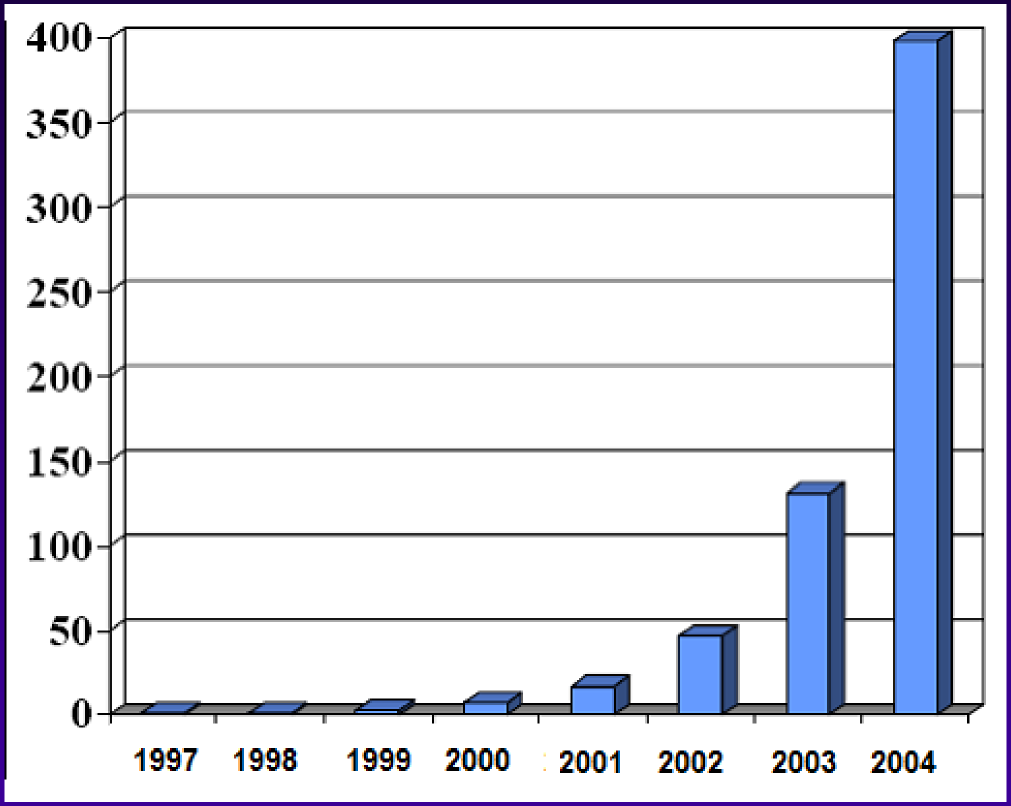
Weak scaling: code performance follow the best possible result where the curve stays flat.
Strong scaling: communication/calculation ratio goes up with number of used processors.
Performance reach best solution (a linear growth), for the largest runs on the biggest machine.

Top500 machines exceed 1 Tflop/s (2004)

1 TF = 1000,000,000,000 Flops



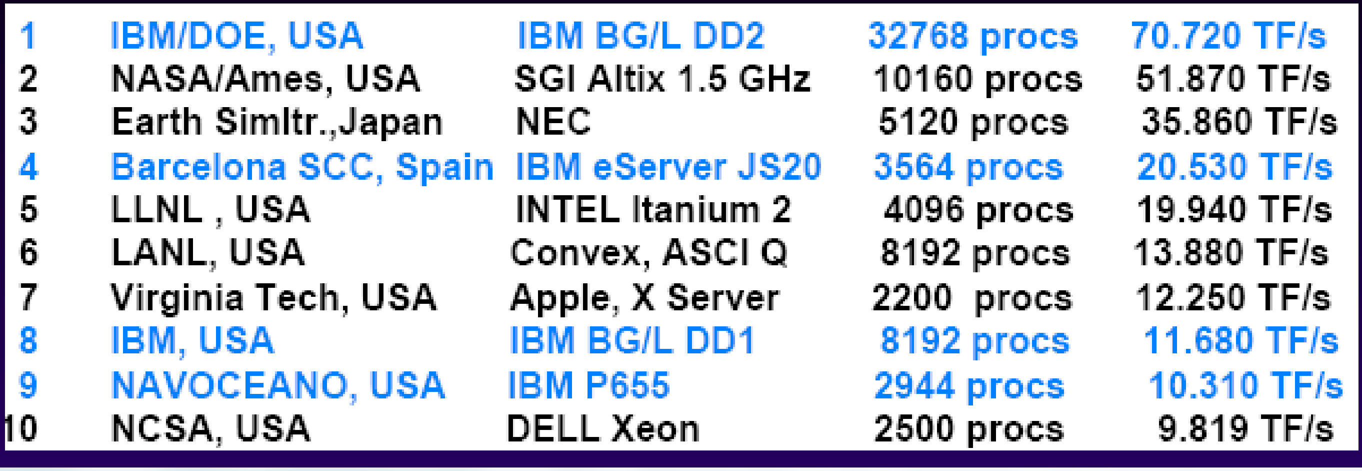
Year	#
1997	1
1998	1
1999	3
2000	7
2001	17
2002	47
2003	131
2004	398



TERA SCALE systems became commonly available !

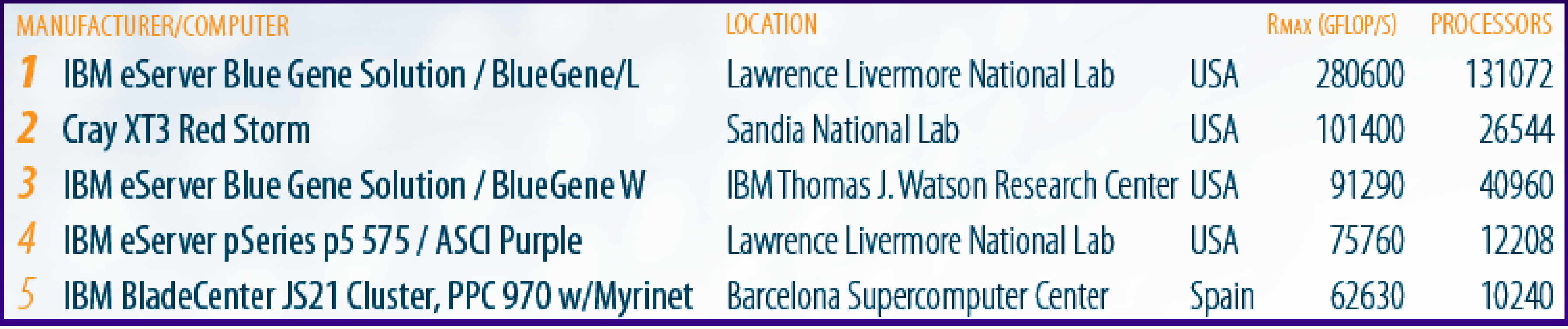
TOWARD PETA SCALE COMPUTING

2004



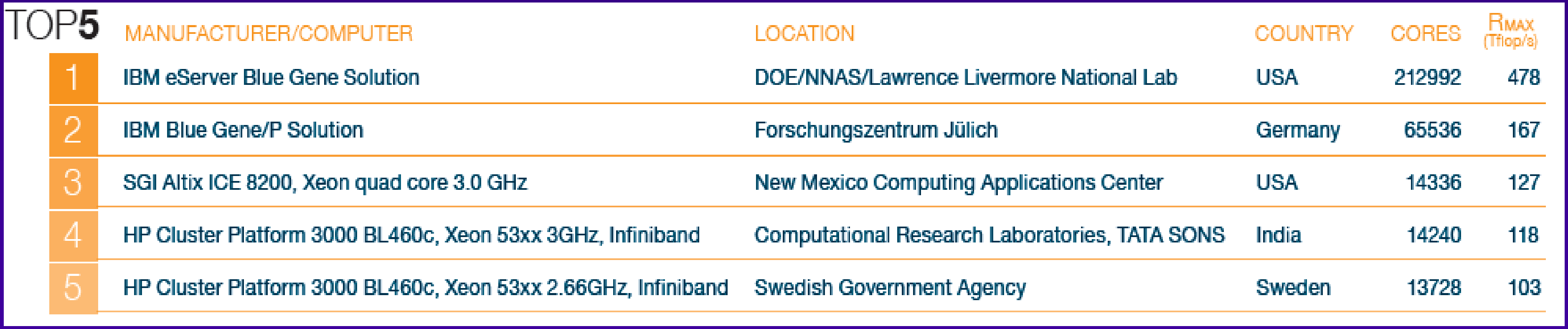
1	IBM/DOE, USA	IBM BG/L DD2	32768 procs	70.720 TF/s
2	NASA/Ames, USA	SGI Altix 1.5 GHz	10160 procs	51.870 TF/s
3	Earth Simltr.,Japan	NEC	5120 procs	35.860 TF/s
4	Barcelona SCC, Spain	IBM eServer JS20	3564 procs	20.530 TF/s
5	LLNL , USA	INTEL Itanium 2	4096 procs	19.940 TF/s
6	LANL, USA	Convex, ASCI Q	8192 procs	13.880 TF/s
7	Virginia Tech, USA	Apple, X Server	2200 procs	12.250 TF/s
8	IBM, USA	IBM BG/L DD1	8192 procs	11.680 TF/s
9	NAVOCEANO, USA	IBM P655	2944 procs	10.310 TF/s
10	NCSA, USA	DELL Xeon	2500 procs	9.819 TF/s

2006



	MANUFACTURER/COMPUTER	LOCATION		RMAX (GFLOP/S)	PROCESSORS
1	IBM eServer Blue Gene Solution / BlueGene/L	Lawrence Livermore National Lab	USA	280600	131072
2	Cray XT3 Red Storm	Sandia National Lab	USA	101400	26544
3	IBM eServer Blue Gene Solution / BlueGene W	IBM Thomas J. Watson Research Center	USA	91290	40960
4	IBM eServer pSeries p5 575 / ASCI Purple	Lawrence Livermore National Lab	USA	75760	12208
5	IBM BladeCenter JS21 Cluster, PPC 970 w/Myrinet	Barcelona Supercomputer Center	Spain	62630	10240

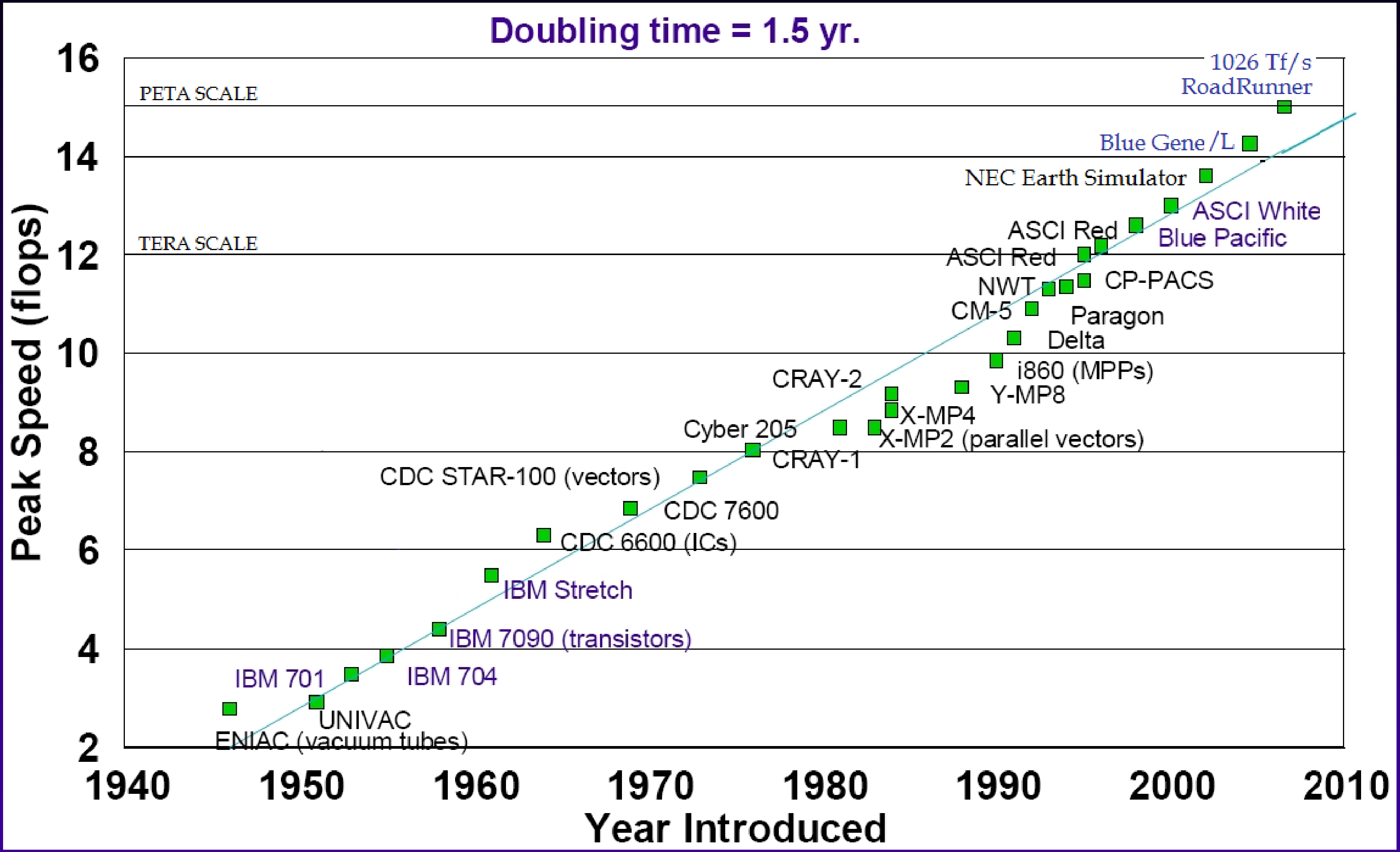
2007



TOP5	MANUFACTURER/COMPUTER	LOCATION	COUNTRY	CORES	RMAX (Tflop/s)
1	IBM eServer Blue Gene Solution	DOE/NNAS/Lawrence Livermore National Lab	USA	212992	478
2	IBM Blue Gene/P Solution	Forschungszentrum Jülich	Germany	65536	167
3	SGI Altix ICE 8200, Xeon quad core 3.0 GHz	New Mexico Computing Applications Center	USA	14336	127
4	HP Cluster Platform 3000 BL460c, Xeon 53xx 3GHz, Infiniband	Computational Research Laboratories, TATA SONS	India	14240	118
5	HP Cluster Platform 3000 BL460c, Xeon 53xx 2.66GHz, Infiniband	Swedish Government Agency	Sweden	13728	103

IBM Blue Gene system was leader in HPC since 2004

2008 first peta system at LANL



LANL (USA):
IBM Blade Center QS22/LS21 Cluster (RoadRunner)
Processors: PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 Ghz
Advanced versions of the processor in the Sony PlayStation 3
122400 cores, peak performance 1375.78Tflops (sustained 1026Tflops)

BLUE GENE SYSTEM DESCRIPTION

Earth Simulator – used to be # 1 on 500 list ~ 35 TF/s on Linpack

IBM BG/L 16384 nodes (Rochester, 2004)

Linpack: 70.72 TF/s sustained, 91.7504 TF/s peak

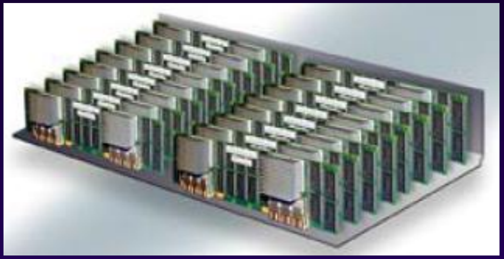
Cost/performance optimized

Low power factor

	ASCI White	ASCI Q	Earth Simulator	ASCI Purple	BlueGene/L[†]
Machine Peak Speed (Tflop/s)	12.3	30	40	100	180 / 360[*]
Total Memory (Tbytes)	8	33	10	50	32
Footprint (ft.^{2})	10,000	20,000	34,000	12,000	2,500
Total Power (MW)	1.0	3.8	10	4.5	1.2
Cost (M$)	~100	~200	~350	~250	<< 100
Installation Date	9/2000	~9/2002	2/2002	12/2004	~12/2004
No. of Nodes	512	4,096	640	197	65,536
CPUs per Node	16	4	8	64	2
Clock Frequency (MHz)	375	1,000	500	~2,000	700
Power Dissipation/Node (W)	2,000	920	16,000	23,000	15
Peak Speed/Node (Gflop/s)	24.0	7.3	64.0	512	5.6
Memory/Node (GiB)	16	8	16	250	0.5

Blue Gene BG/L - hardware

Massive collection of low-power CPUs instead of a moderate-sized collection of high-power CPUs



Chip	Compute card	Node card	Rack	System
2 CPU cores	2 chips	16 comp cards	32 node cards	64 raks
	1x2x1	32 chips 4x4x2	8x8x16	64x32x32
Peak 5,6 GF/s	11.2 GF/s	180 GF/s	5.6 TF/s	360 TF/s
Memory 4 MB	1 GB	16 GB	512 GB	32 TB

Power and cooling

700MHz IBM PowerPC 440 processors

Typical 360 Tflops machine ~ 10-20 megawatts

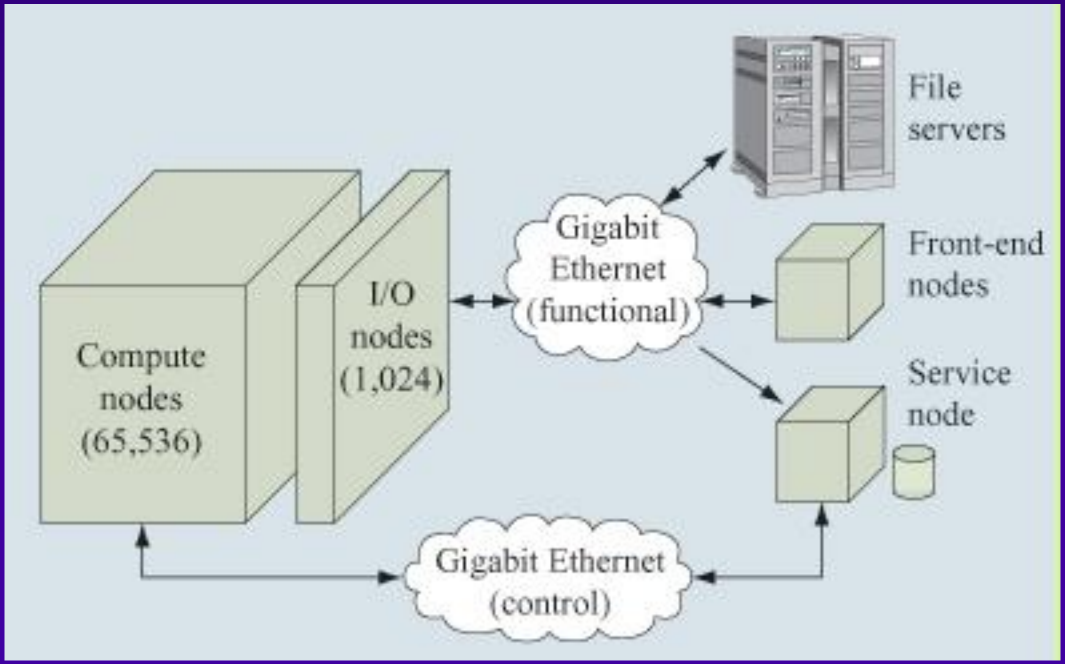
BlueGene/L uses only 1.76 megawatts

High ratios:

- **power / Watt**
- **power / square meter of floor space**
- **power / $$$**

Reliability and maintenance

20 fails per 1,000,000,000 hours =
1 node failure every 4.5 weeks



Blue Gene BG/L – main characteristics

Mode 1 (Co-processor mode - CPM):

one process per compute node
CPU0 does all the computations
CPU1 does the communications
Communication overlap with computation
Peak comp perf is 5.6/2 = 2.8 Gflops

Mode 2 (Virtual node mode - VNM):

one process per processor
CPU0, CPU1 independent “virtual tasks”
Each does own computation and communication
The two CPU’s talk via memory buffers
Computation and communication cannot overlap
Peak compute performance is 5.6 Gflops

NETWORK:

Torus Network (High-speed, high-bandwidth network, for point-to-point communication)

Collective Network (Low latency, 2.5 µs, does MPI collective ops in hardware)

Global Barrier Network (Extremely low latency, 1.5 µs)

I/O Network (Gigabit Ethernet)

Service Network (Fast Ethernet and JTAG)

SOFTWARE:

MPI (MPICH2)

IBM XL Compilers for PowerPC

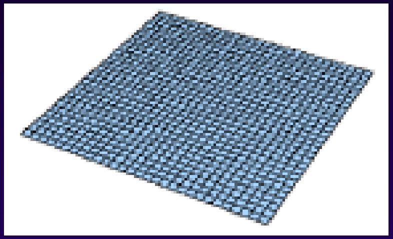
Math Library:

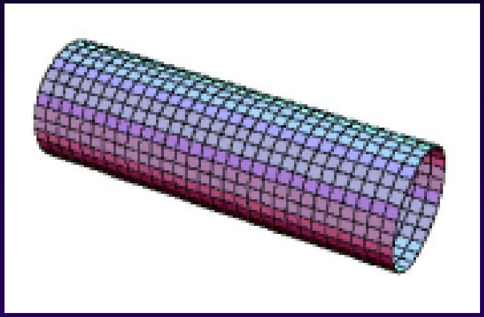
ESSL: dense matrix kernels

MASSV: reciprocal, square root, exp, log

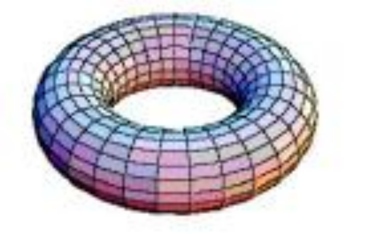
FFT: Parallel Implementation developed by Blue Matter Team

Blue Gene BG/L – torus geometry





3-d Torus



Torus topology instead of crossbar: 64 x 32 x 32 3D torus of compute nodes.

Each compute node is connected to its six neighbors: x+, x-, y+, y-, z+, z-

Compute card is 1x2x1

Node card is 4x4x2 (16 compute cards in 4x2x2 arrangement)

Midplane is 8x8x8 (16 node cards in 2x2x4 arrangement)

Supports cut-through routing, with deterministic and adaptive routing.

Each uni-directional link is 1.4Gb/s, or 175MB/s.

Each node can send and receive at 1.05GB/s.

Variable-sized packets of 32,64,96…256 bytes

Guarantees reliable delivery

Blue Gene BG/L – physical node partition

Node partitions are created when jobs are scheduled for execution

Processes are spread out in a pre-defined mapping (XYZT)

Alternate and sophisticated mappings are possible

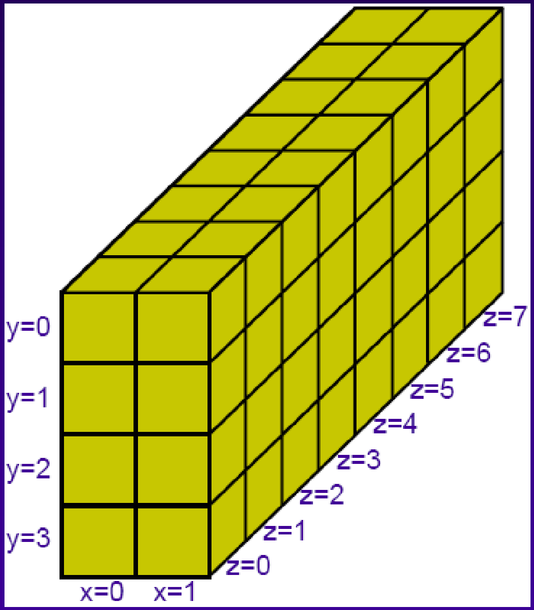
User may specify desired processor

configuration when submitting job:

e.g. **submit lufact 2x4x8**

partition of 64 compute nodes, with shape

2 (on x-axis) by 4 (on y-axis) by 8 (on z-axis)



A contiguous, rectangular subs
ection of the compute nodes

Blue Gene BG/L – mapping processes to nodes

In MPI, logical process grids are created with **MPI_CART_CREATE**

The mapping is performed by the system, matching physical topology

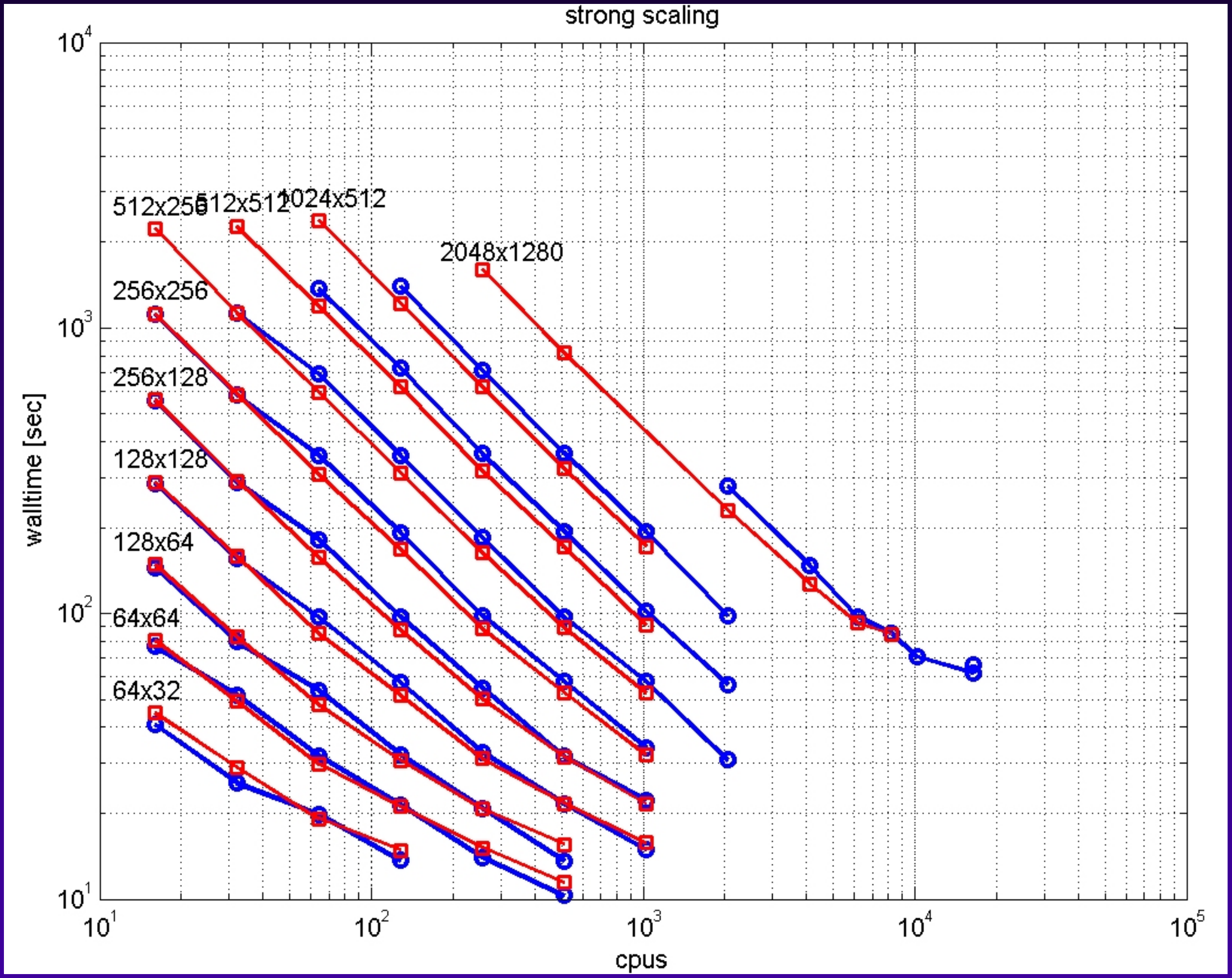
- Each xy-plane is mapped to one column
- Within Y column, consecutive nodes are neighbors
- Logical row operations in X correspond to operations on a string of physical nodes along the z-axis
- Logical column operations in Y correspond to operations on an xyplane
- Row and column communicators are created with **MPI_CART_SUB**

0,0,0	0,0,1	0,0,2	0,0,3	0,0,4	0,0,5	0,0,6	0,0,7
0,1,0	0,1,1	0,1,2	0,1,3	0,1,4	0,1,5	0,1,6	0,1,7
0,2,0	0,2,1	0,2,2	0,2,3	0,2,4	0,2,5	0,2,6	0,2,7
0,3,0	0,3,1	0,3,2	0,3,3	0,3,4	0,3,5	0,3,6	0,3,7
1,0,0	1,0,1	1,0,2	1,0,3	1,0,4	1,0,5	1,0,6	1,0,7
1,1,0	1,1,1	1,1,2	1,1,3	1,1,4	1,1,5	1,1,6	1,1,7
1,2,0	1,2,1	1,2,2	1,2,3	1,2,4	1,2,5	1,2,6	1,2,7
1,3,0	1,3,1	1,3,2	1,3,3	1,3,4	1,3,5	1,3,6	1,3,7

EULAG 2D grid decomposition is distributed over contiguous, rectangular 64 compute nodes with shape 2x4x8

EULAG SCALABILITY on BGL/BGW

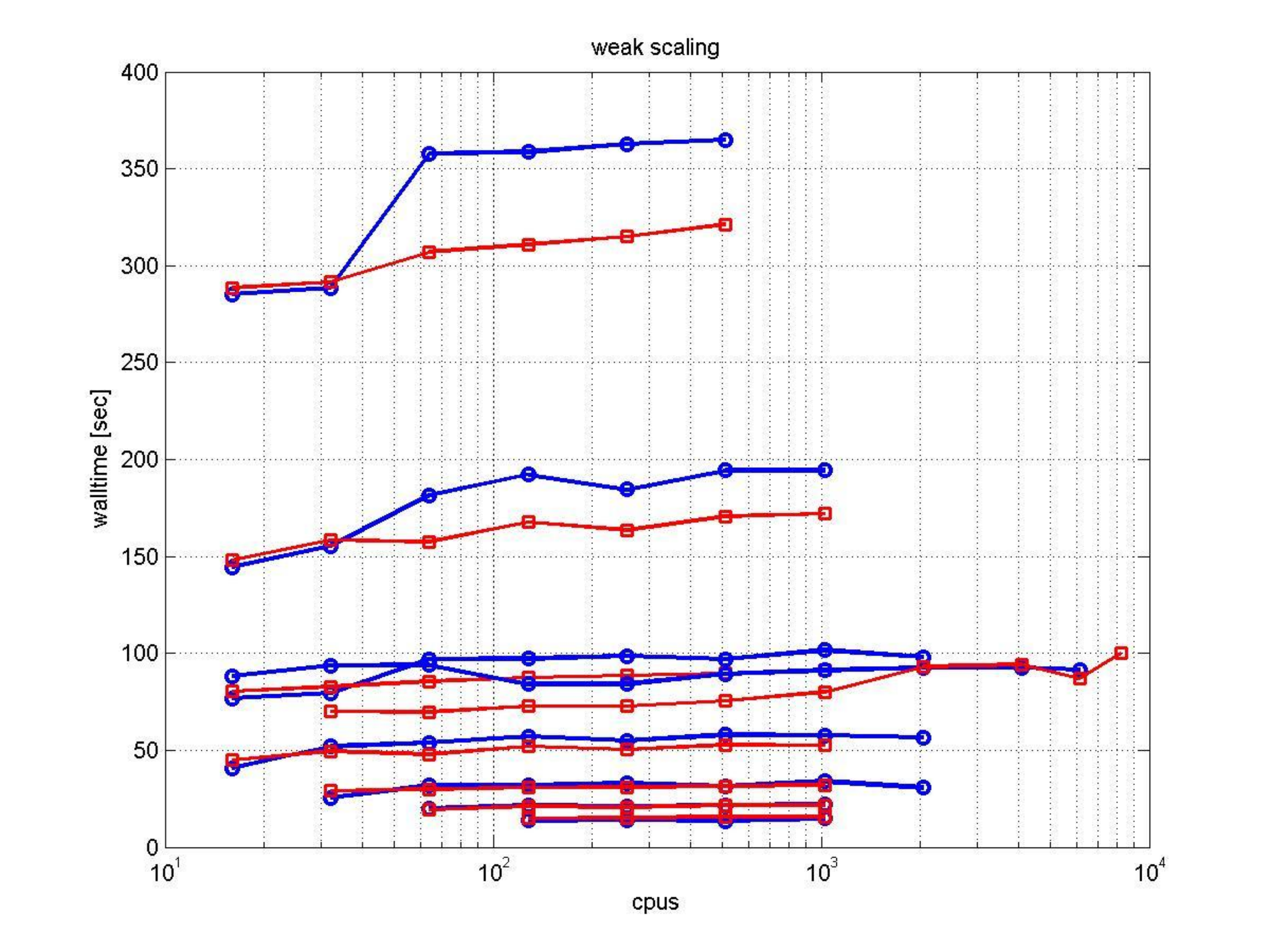
Benchmark results from the Eulag-HS experiments
NCAR/CU BG/L system 2048 processors (frost),
IBM/Watson Yorktown heights BG/W … up to 40 000 PE, only 16000 available during experiment



All curves except 2048x1280 are performed on BG/L system.
Numbers denote horizontal domain grid size, vertical grid is fixed l=41
The Elliptic solver is limited to 3 iterations (iord=3) for all experiments
Red lines – coprocessor mode, blue lines virtual mode

EULAG SCALABILITY on BGL/BGW

Benchmark results from the Eulag-HS experiments
NCAR/CU BG/L system 2048 processors (frost),
IBM/Watson Yorktown heights BG/W ... up to 40 000 PE, only 16000 available during experiment



Red lines – coprocessor mode, blue lines virtual mode

CONCLUSIONS:

EULAG is scalable and perform well on available supercomputers

SMP implementation based on Open MP is needed

Additional work is needed to run model efficiently at PETA scale

- profiling to define bottlenecks

- 3D domain decomposition

- optimized mapping for increase locality

- preconditioning for local elliptic solvers

- parallel I/O