Towards PetaScale simulations of turbulence in precipitating clouds

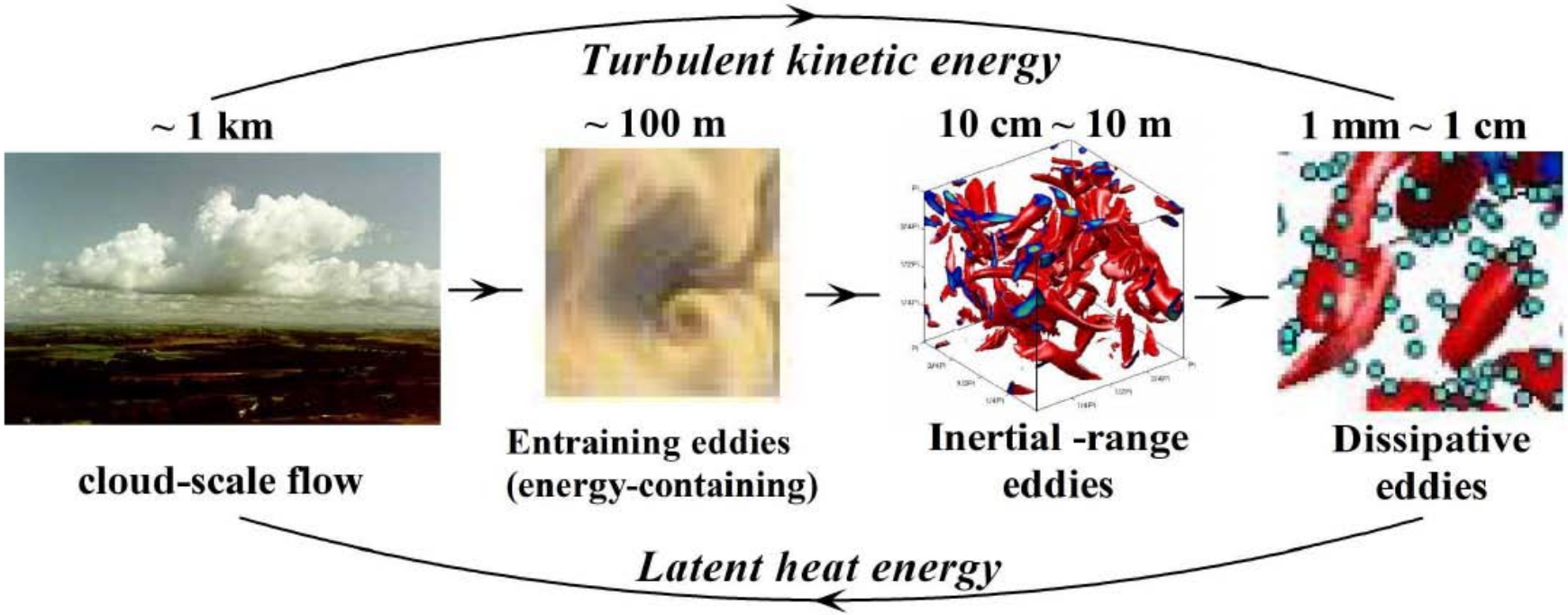
Andrzej Wyszogrodzki
Zbigniew Piotrowski
Wojciech Grabowski

National Center for Atmospheric Research
Boulder, Co, USA

September 13, 2010
Sopot, Poland

Multiscale interactions in atmospheric clouds

The turbulent kinetic energy flows from cloud-scale motion to dissipative eddies

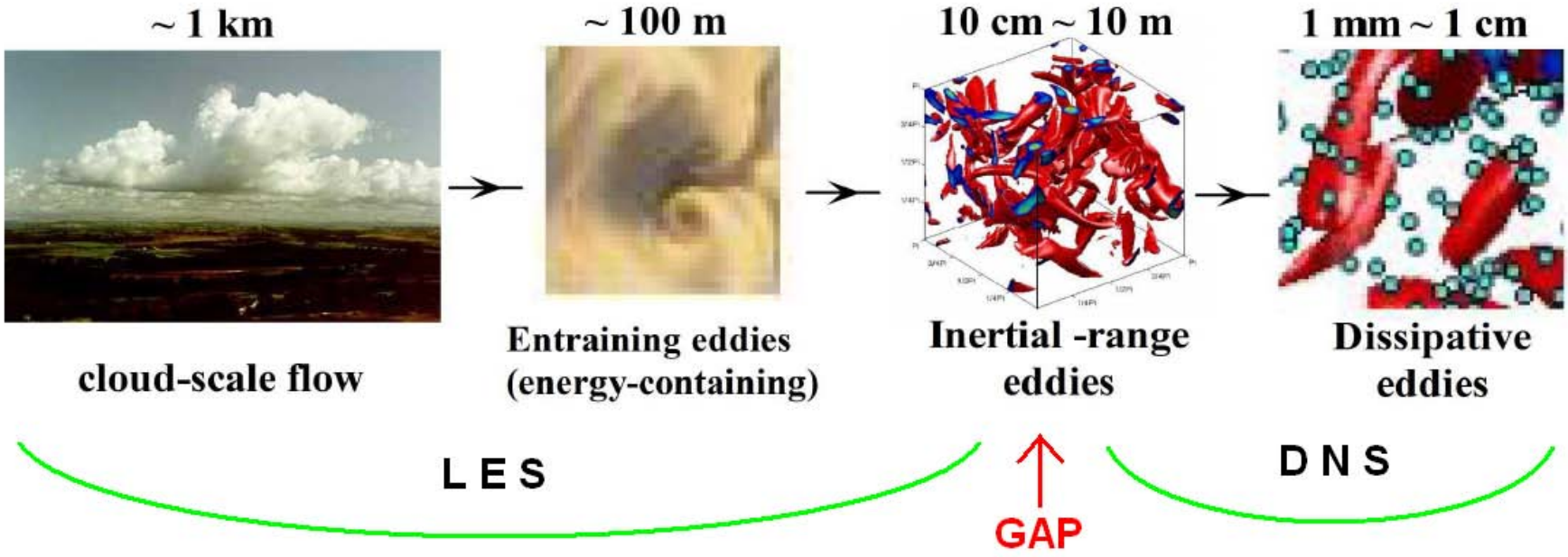


Latent heat energy flows from individual droplets to cloud-scale motion.

typical cloud of dimension 1 km
could consist of O(10^{17}) droplets

Cloud Turbulence

Full spectrum of scales divided into two ranges: LES and DNS

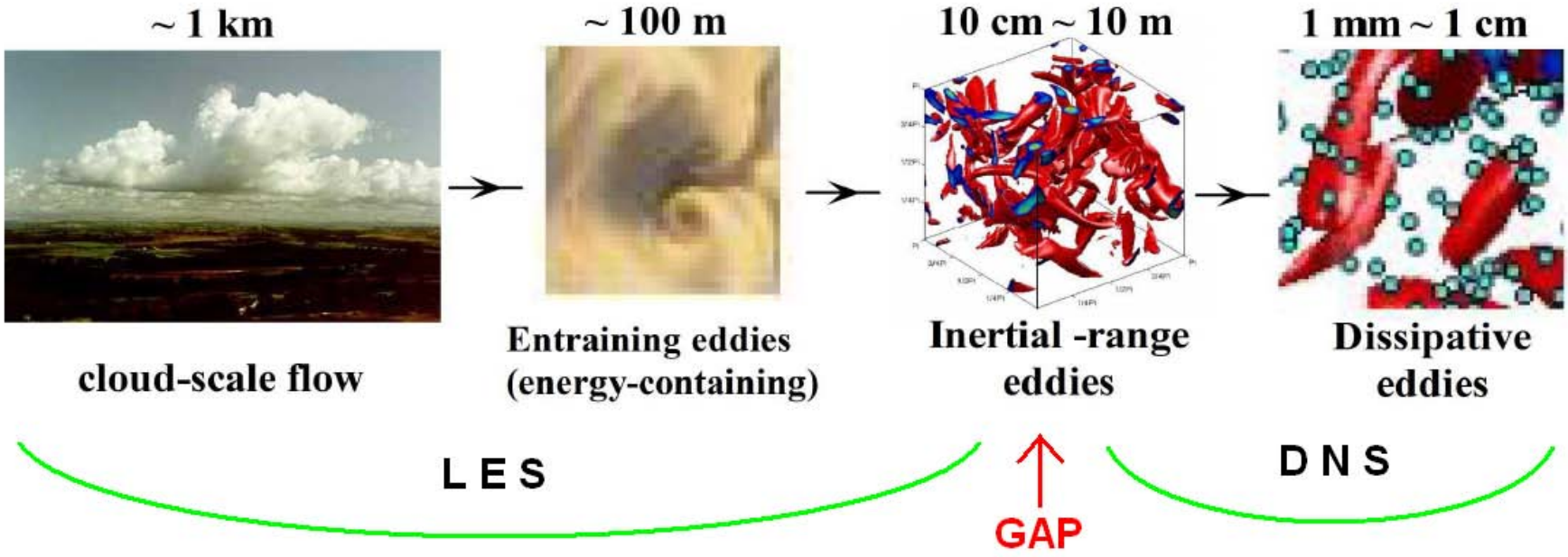


Bin-based microphysics

Limitation to low Reynolds numbers

Cloud Turbulence

Full spectrum of scales divided into two ranges: LES and DNS



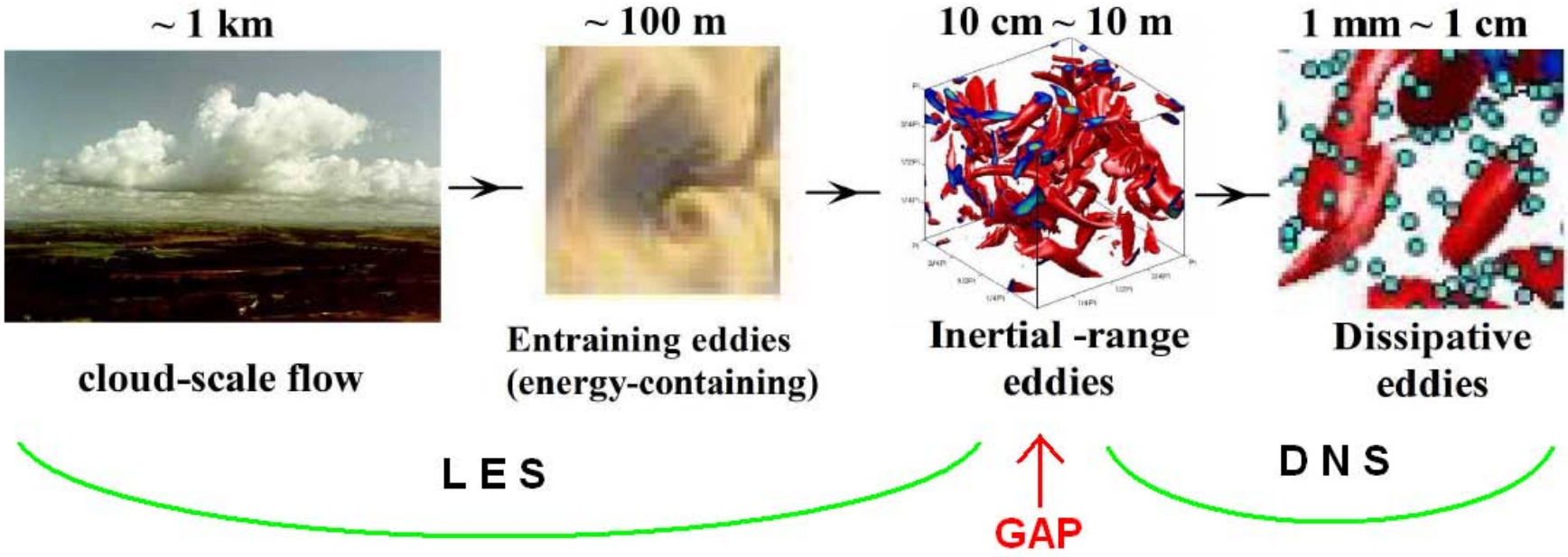
Cloud-resolving LES
L ~ 1 km
dx ~ 1 m

Filling the gap by LES and DNS models:
e.g. turbulent enhancement of the collision kernel for broad range of dissipation rates

Hybrid DNS
L ~ 1 m
dx ~1 mm

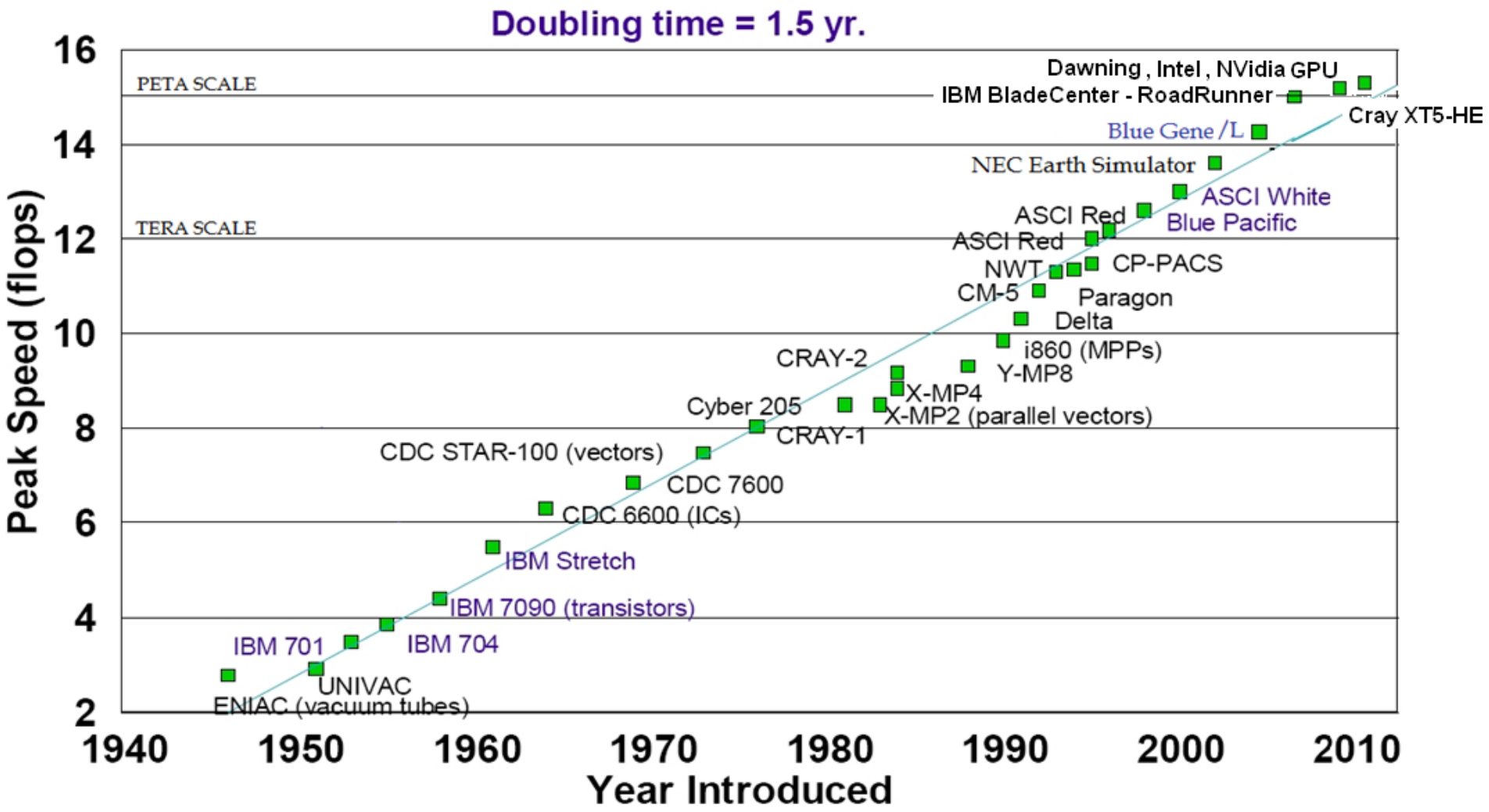
Cloud Turbulence

Full spectrum of scales divided into two ranges: LES and DNS



LES and DNS models needs to efficiently use Peta Scale computer srchitectures

Peta-scale systems



TOWARD PETA SCALE COMPUTING

NCAR

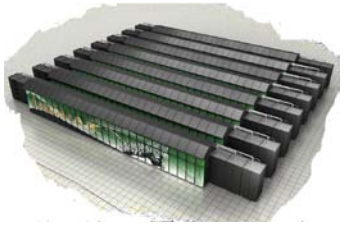
	NAME/MANUFACTURER/COMPUTER	LOCATION	COUNTRY	CORES	R_{max}	Peak
1	**Jaguar**, Cray XT5 6-core 2.6 GHz	DOE / OS / ORNL	USA	224162	1.76	2.33
2	**Nebulae**, Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU	(NSCS)	China	120640	1.27	2.98
3	**Roadrunner**, IBM BladeCenter QS22/LS21 Cluster, PowerXCell 3.2 Ghz / Opteron 1.8 GHz, Voltaire Iband	DOE / NNSA / LANL	USA	122400	1.04	1.37
4	**Kraken**, Cray XT5 6-core 2.6 GHz	NSF / U of Tennessee	USA	98928	.832	1.03
5	**Jugene**, IBM Blue Gene/P Solution	Forschungszentrum Juelich	Germany	294912	.826	1.00

IBM

Power 5 / 6 / 7

Blue Gene / L / C / P / Q
Power PC 440 / 450
Trading the speed for lower power consumption

CRAY
THE SUPERCOMPUTER COMPANY



曙光

Dawning Information Industry
insists on the pattern of in-house technological innovation

TOWARD PETA SCALE COMPUTING

Current systems available for us

Franklin - Cray XT4 (NERSC)
38,128 Opteron cores
peak performance - 352 Tflops
#17 @ Top500

IBM Blue Gene/L (NCAR)
700MHz PowerPC-440 CPUs
4096 compute nodes – 8192 cores
22.9 TFlops

Hopper – Cray XT5 (NERSC)
2 quad-core AMD 2.4 GHz processors
5312 total cores

IBM Bluefire (NCAR)
4,096 POWER6™ 4.7 GHz processors
77 Tflops, #90 @ Top500

Lynx - Cray XT5m (NCAR)
2 hex-core 2.2 GHz AMD Opteron
912 cores
8.03 TFLOPs

TOWARD PETA SCALE COMPUTING

PetaScale systems in near future

Late summer 2010
Hopper II – Cray XE6 (NERSC)
2 twelve-core AMD 'MagnyCours' 2.1 GHz
153,408 total cores
>1 PetaFlop peak performance

2011
IBM Blue Waters (NCSA)
~10 Petaflops peak
> 1 Petaflops sustained

IBM Cyclops-64 (C64) / BlueGene/C
80/160 processors (or cores) per chip
80 Gflops/chip
13.824 nodes (2.211.840 cores total)
1.1 PetaFlops

IBM Blue Gene/Q – Sequoia
LLNL, 2011-2012
98,304 nodes - 1.6 million cores
~20 Petaflop peak

EULAG PARALLELIZATION HISTORY

1996-1998: compiler parallelization on NCAR's vector Crays J90

1996-1997: first MPP (PVM)/SMP (SHMEM) version at NCAR's Cray T3D based on 2D domain decomposition **(Anderson)**

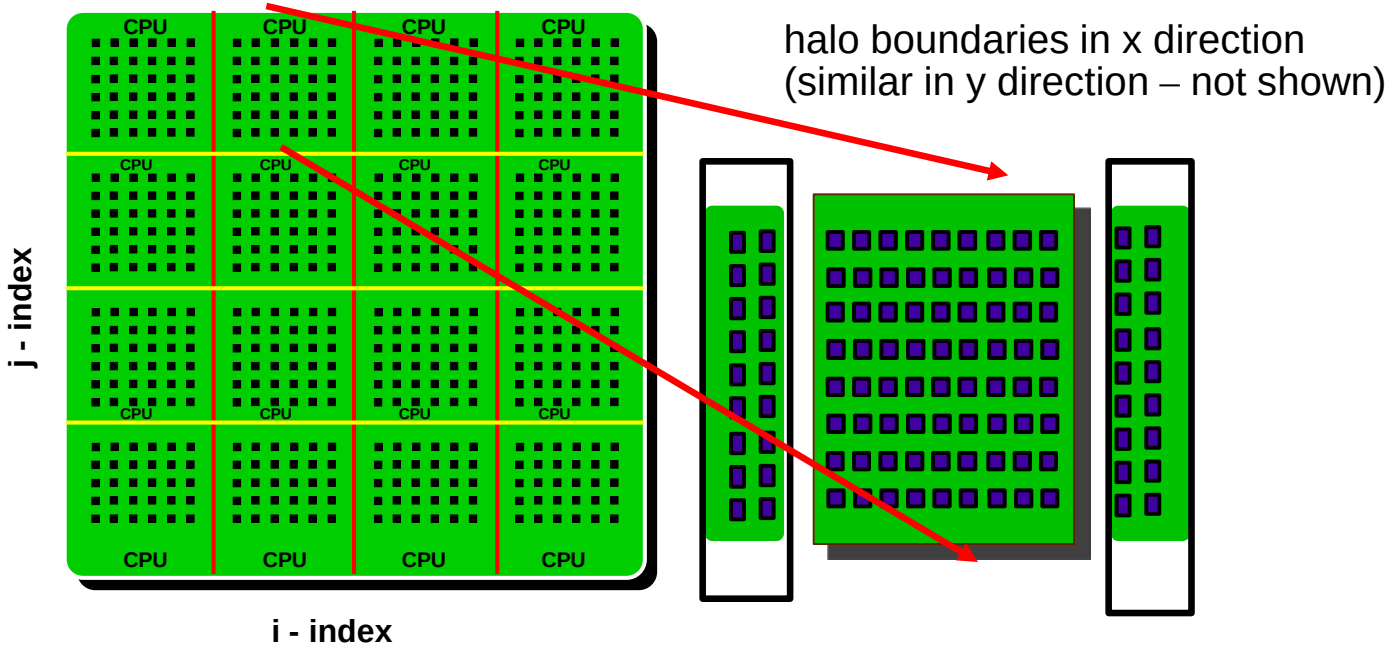
1997-1998: extension to MPI, removal of PVM **(Wyszogrodzki)**

2004: attempt to use OpenMP **(Andrejczuk)**

2009-2010: development of OpenMP and GPU/OpenCL version **(Rojek & Szustak)**

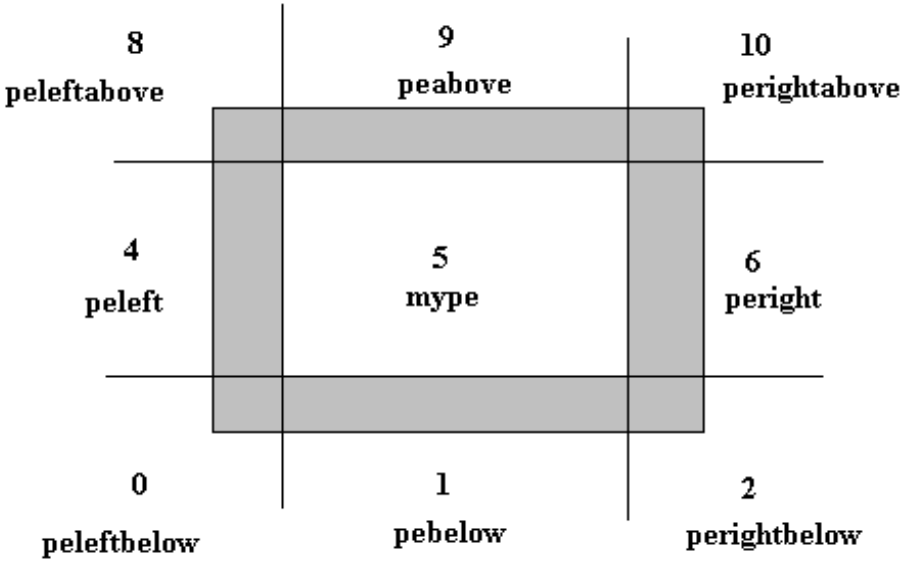
2010: extending 2D decomposition to 3D MPP **(Piotrowski & Wyszogrodzki)**

Data decomposition in EULAG



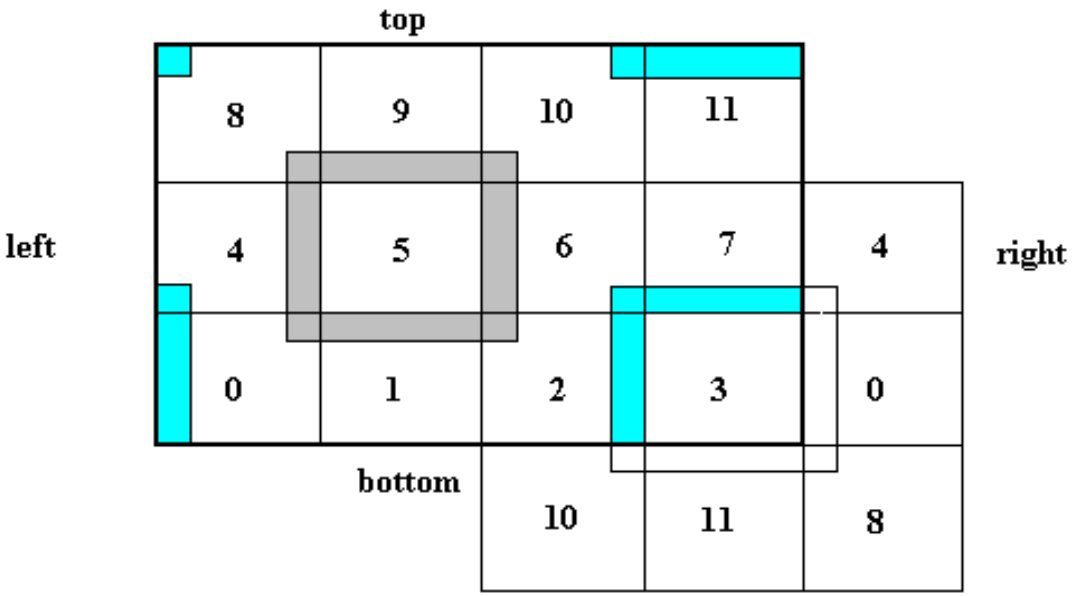
- 2D horizontal domain grid decomposition
- No decomposition in vertical Z-direction
- Hallo/ghost cells for collecting information from neighbors
- Predefined halo size for array memory allocation
- Selective halo size for update to decrease overhead

Typical processors configuration



- 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

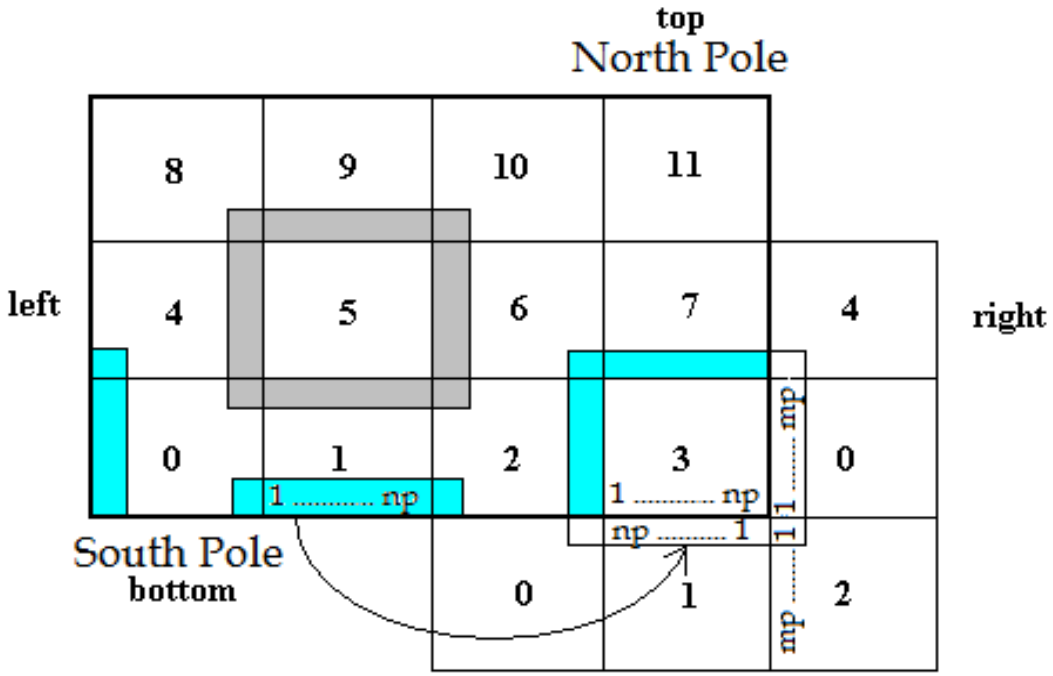
- nprocs=12
- nprocx = 4, nprocy = 3
- if np=11, mp=11

  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 grid in 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 $\Delta y/2$ distance from the pole.

EULAG SCALABILITY TESTS

Weak Scaling

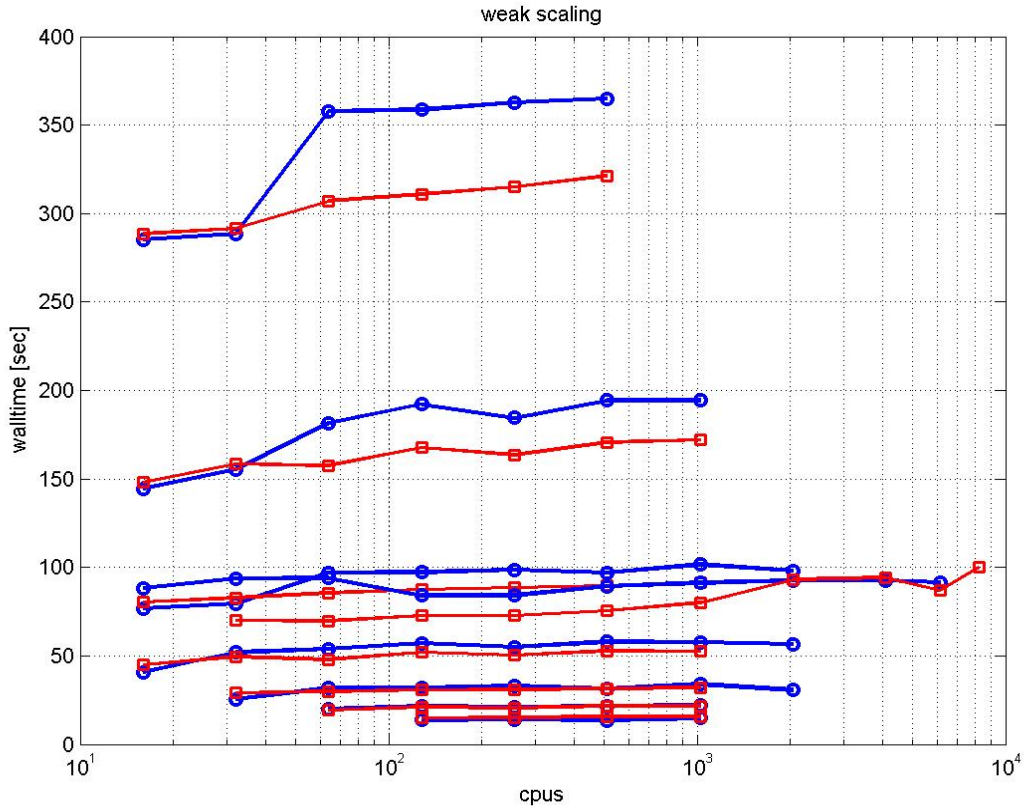
- Problem size/proc fixed
- Easier to see Good Performance
- Beloved of Benchmarkers, Vendors, Software Developers –Linpack, Stream, SPPM

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 TESTS

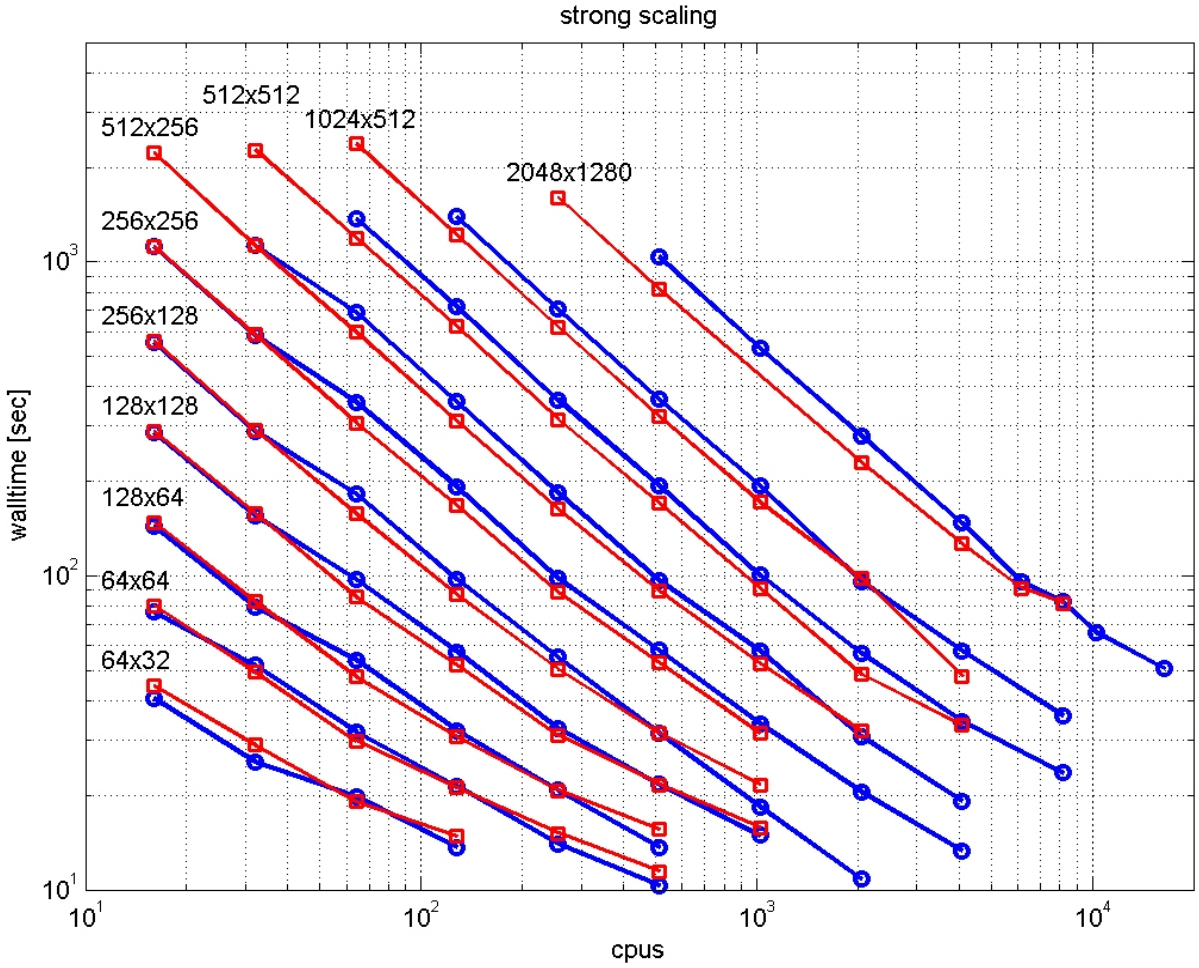
Benchmark results from the Eulag-HS experiments
NCAR/CU BG/L system 2048 processors (frost),
IBM/Watson Yorktown heights BG/W ... up to 40 000 PE, only 16000 available during experiment



Red lines – coprocessor mode, blue lines virtual mode

EULAG SCALABILITY

Benchmark results from the Eulag-HS experiments
NCAR/CU BG/L system 8384 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)

Red lines – coprocessor mode, blue lines virtual mode

Excellent scalability up to number of processors *NPE=sqrt(N*M)*

Problems in achieving high model efficiency

Performance and scalability bottlenecks:

Data locality & domain decomposition

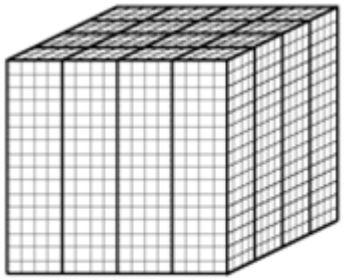
Peak performance

Tradeoffs: efficiency vs accuracy and portability

Load balancing & optimized processor mapping

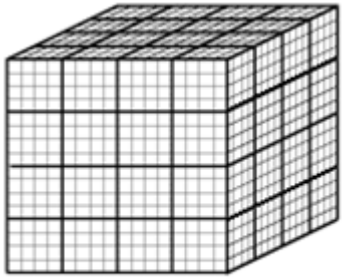
I/O

Development of EULAG 3D *domain decomposition (Piotrowski)*



2D

Purpose: increase data locality by minimize maximum number of neighbors (messages)



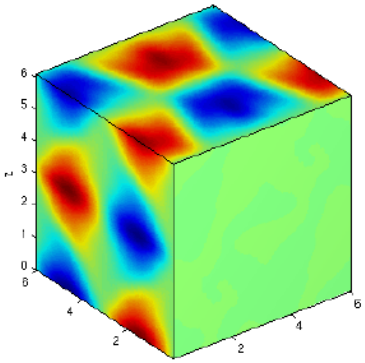
3D

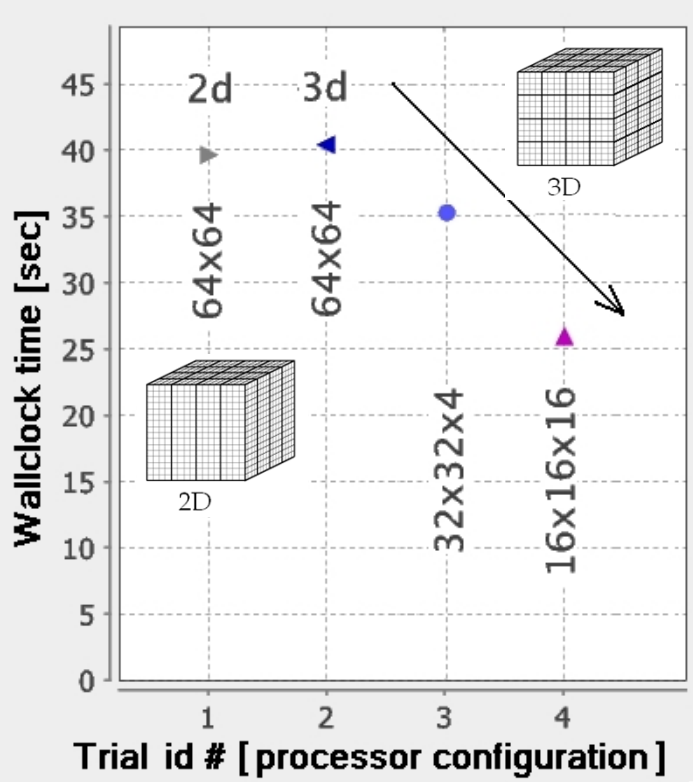
Changes to model setup and algorithm design

- ***New processor geometry setup***
- ***Halo updates in vertical direction***
- ***Optimized halo updates at the cube corners***
- ***Changes in vertical grid structure for all model variables***
- ***New loops structure due to differentiation and BC in vertical***

EULAG 3D *domain decomposition*

Taylor Green Vortex (TGV) system
Turbulence Decay
Triple periodic cubic grid box





Only pressure solver and model initializations, no preconditioneer

Fixed number of iterations

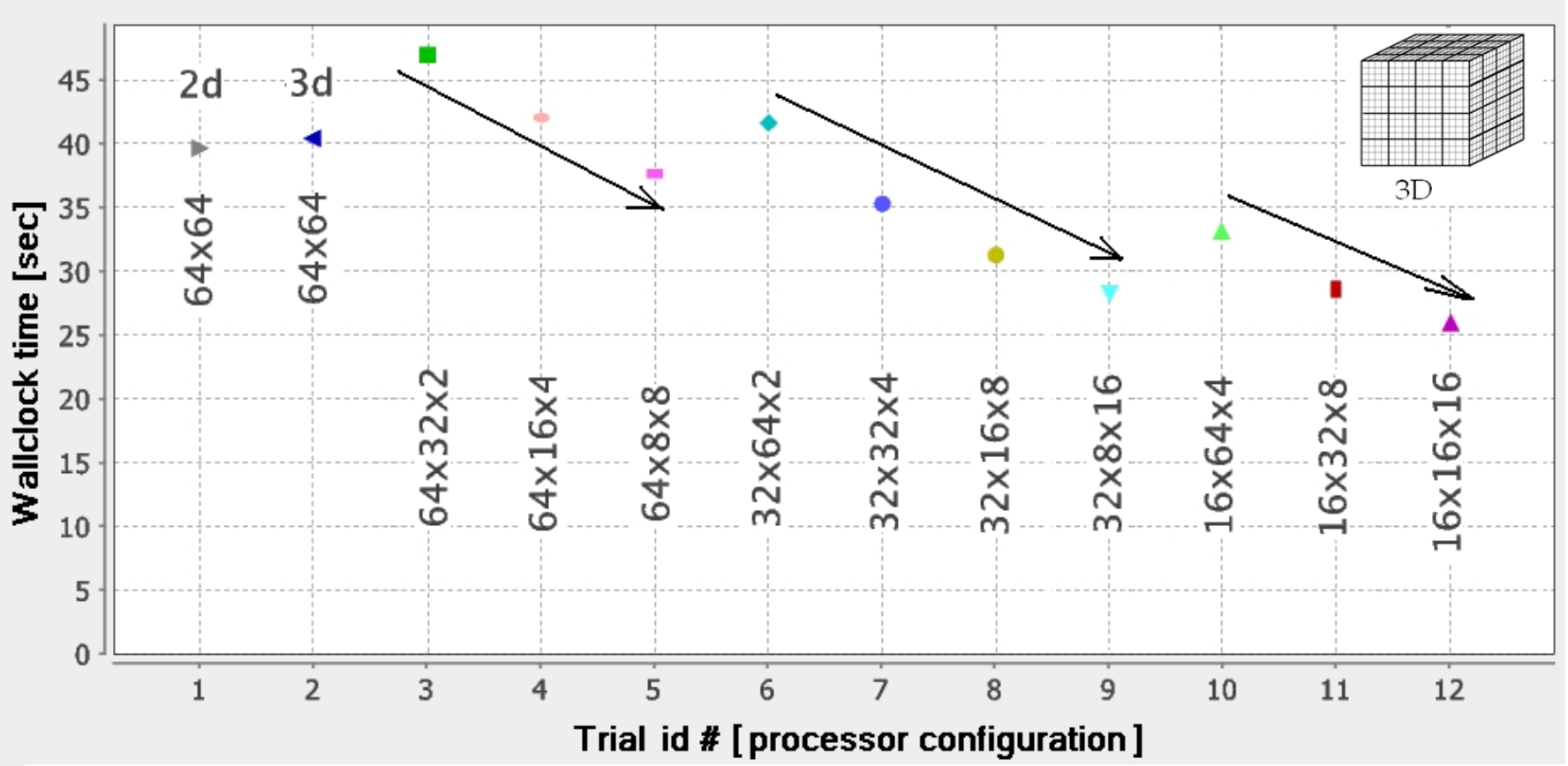
100 calls to solver

512^3 grid points

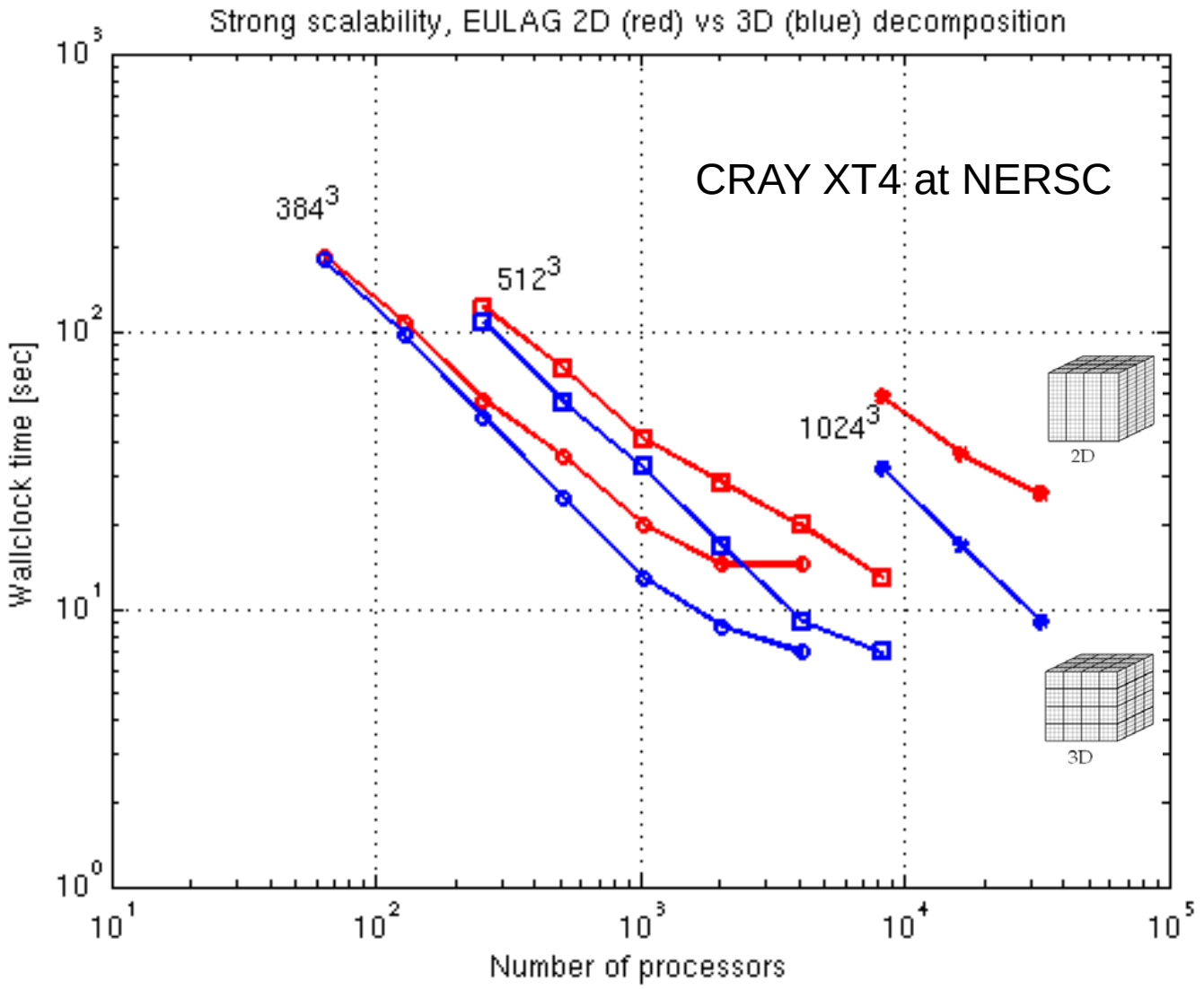
IBM BG/L system
with 4096 PEs

EULAG 3D *domain decomposition*

Decomposition patterns



EULAG 3D *domain decomposition*



BOTTLENECK – DATA LOCALITY

Nonhydrostatic, anelastic, Navier-Stokes eqns

$$\frac{d\mathbf{v}}{d\bar{t}} = -\widetilde{\mathbf{G}}\overline{\nabla}\pi' - \mathbf{g}\frac{\theta'}{\theta_h} - \beta\mathbf{v} + \mathcal{D}_m(e, \overline{\nabla}\mathbf{v}) - \alpha_m\mathbf{v}'$$

$$\frac{d\theta'}{d\bar{t}} = -\overline{\mathbf{v}}^s \bullet \overline{\nabla}\theta_e - \beta(\theta - \theta_B) + \mathcal{D}_h(e, \overline{\nabla}\theta) - \alpha_h\theta'$$

$$\overline{\nabla} \bullet (\rho^*\overline{\mathbf{v}}^s) = 0 \qquad \overline{\mathbf{v}}^s = \widetilde{\mathbf{G}}^T\mathbf{v} \qquad \widetilde{\mathbf{G}} \sim (\partial\overline{\mathbf{x}}/\partial\mathbf{x})$$

Renormalized Jacobi matrix of transf coeff.

Preconditioned Generalized Conjugate Residual (GCR) solver for nonsymmetrical elliptic pressure eqn

$$\left\{\frac{\delta t}{\rho^*}\overline{\nabla} \bullet \rho^*\widetilde{\mathbf{G}}^T\left[\hat{\hat{\mathbf{v}}} - (\mathbf{I} - 0.5\delta t\mathbf{R})^{-1}\widetilde{\mathbf{G}}(\overline{\nabla}\pi'')\right]\right\}_{\mathbf{i}} = 0$$

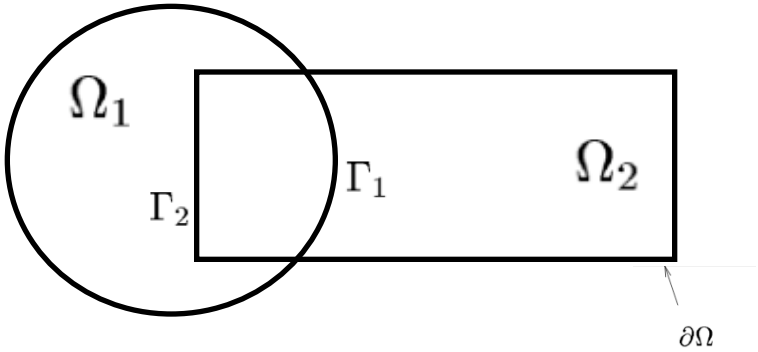
$$\overline{\mathbf{v}}^s = \hat{\hat{\mathbf{v}}} - \mathbf{Grad}\ \phi$$

solnoidal velocity

SOLUTION REQUIRES EFFICIENT PARALLEL SOLVER FOR GLOBAL REDUCTION OPERATIONS

New domain decomposition ideas for elliptic solvers (J-F Cossette)

How to solve the Dirichlet problem on non-trivial domains?



$$\Delta u = 0 \quad \text{in } \Omega, \quad u = g \quad \text{on } \partial\Omega$$

$$\begin{aligned} \Delta u_1^{n+1} &= 0 \quad \text{in } \Omega_1, \\ u_1^{n+1} &= u_2^n \quad \text{on } \Gamma_1, \end{aligned} \qquad \begin{aligned} \Delta u_2^{n+1} &= 0 \quad \text{in } \Omega_2, \\ u_2^{n+1} &= u_1^{n+1} \quad \text{on } \Gamma_2, \end{aligned} \qquad \begin{aligned} \mathcal{L} u_j^{n+1} &= f \quad \text{in } \Omega_j, \\ u_j^{n+1} &= u_k^{n+1_{jk}} \quad \text{on } \Gamma_{jk}, \end{aligned}$$

Schwarz Alternating decomposition method:

- form a sequence of local solutions found on simpler subdomains that converges to the global solution
- readily extends to arbitrary partitions
- used as a preconditioner in Newton-Krylov Schwarz methods (*NKS) – CFD problems* (e.g. *low Mach number compressible flows, tokamak edge plasma fluid)*

New domain decomposition ideas for elliptic solvers (J-F Cossette)

Discretized forms of the Schwarz method that solve the linear system use:

- *restriction operators (global to local)* to collect boundary condition
- *prolongation (local to global) operators* to redistribute partial solutions

$$Au = f$$

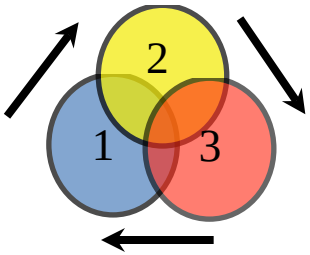
$$\left(\sum R_i^T A_i^{-1} R_i\right) A = \left(\sum R_i^T A_i^{-1} R_i\right) \mathbf{f}$$

$\mathbf{i} = 1, 2 \ldots,$ number of subdomains

Additive Schwarz

Restrictive Additive Schwarz method (RASM) eliminates the need for transmission conditions - faster convergence and CPU time Cai and Sarkis (1999)

Parallel computing: solution on each subdomain is found simultaneously (Lions, 1998)

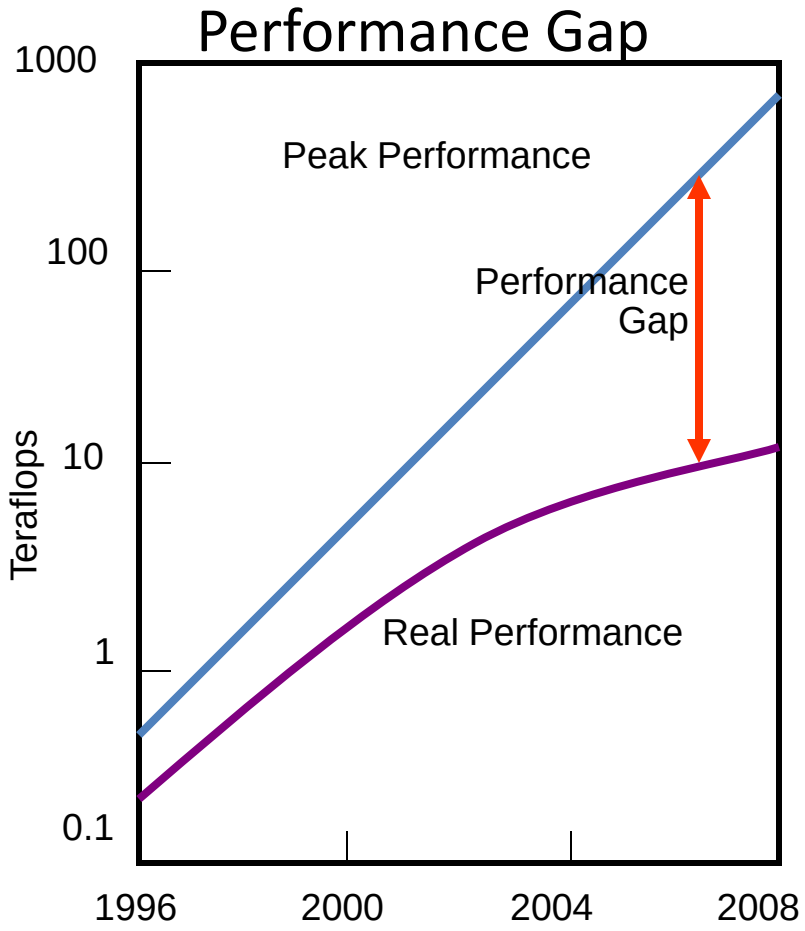


Knoll & Keyes, 2004

Iteration count scaling of Schwarz-preconditioned Krylov methods

Preconditioning problem size N processor number P	Iteration count	
	2D	3D
Point Jacobi	$\mathcal{O}(N^{1/2})$	$\mathcal{O}(N^{1/3})$
Subdomain Jacobi	$\mathcal{O}((NP)^{1/4})$	$\mathcal{O}((NP)^{1/6})$
One-level additive Schwarz	$\mathcal{O}(P^{1/2})$	$\mathcal{O}(P^{1/3})$
Two-level additive Schwarz	$\mathcal{O}(1)$	$\mathcal{O}(1)$

BOTTLENECK – PEAK PERFORMANCE



EULAG: standard peak performance

3-10% depending on system

Efficiency for many science applications declined from ~50% on vector supercomputers of 1990s to below 10% on parallel supercomputers today

OPEN QUESTION: to be efficient or to be accurate: i.e. how to improve peak performance on single processor (a key factor to achieve sustained Peta performance) but not degrade model accuracy?

- Profiling tools
- Automatic compiler optimizations
- Code restructurization
- Efficient parallel libraries

BOTTLENECK – PEAK PERFORMANCE

Scalar and Vector MASS (Math Acceleration Subroutine System)

Approximate clock cycle-counts per evaluation on IBM BG/L

function	libm.a	libmass.a	libmassv.a
Sqrt	159	40	11
Exp	177	65	19
Log	306	95	20
Sin	217	75	32
Cos	200	73	32
pow	460-627	171	29-48
Div	29	11	5
1/X	30	11	4/5

EULAG: increase up to 15% of peak on IBM BG/L system (optimizations in microphysics and advection), but differences in results may be expected

BOTTLENECK – LOAD BALLANCING

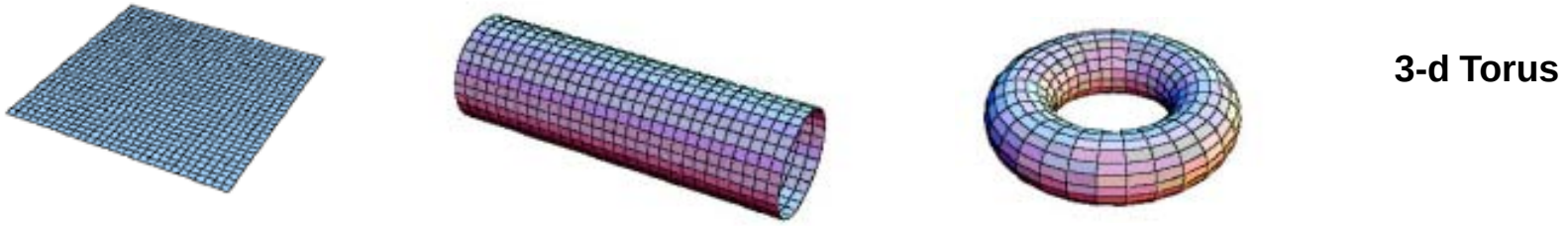
Balanced work loads:

small imbalances result in many wasted processors! (e.g. 100,000 processors with one processor 5% over average workload equivalent to ~5000 *idle processors)*

- No noticed balancing problems in Cartesian model
- Unbalancing in spherical code during communication over the poles
- Problem with grid partitioning in unstructured mesh model: proper criterion of efficient load balancing (e.g. geometric methods) vs workload of numerical algorithms used

BOTTLENECK – PROCESSOR MAPPING

Blue Gene / Cray's XT – torus geometry



3-d Torus

Torus topology instead of crossbar (e.g 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-

Good mapping ->

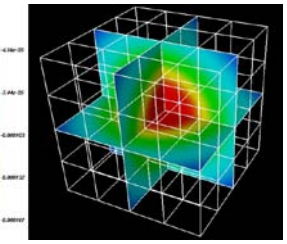
- **reducing message latency,**
- **smaller communication costs,**
- **better scalability and performance**

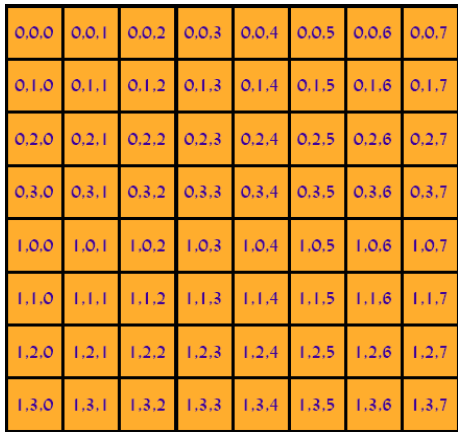
BOTTLENECK – PROCESSOR MAPPING

The mapping is performed by the system, matching physical topology

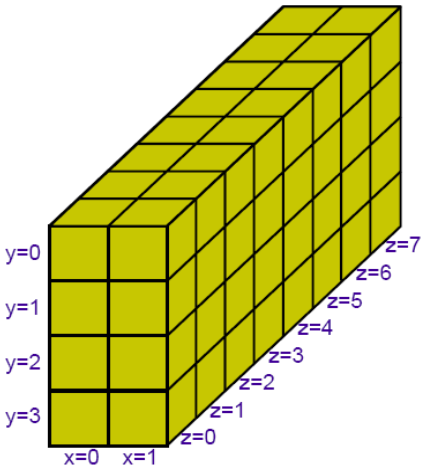
Node partitions are created when jobs are scheduled for execution

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





EULAG 2D grid decomposition



A contiguous, rectangular subsection of the 64 cores on compute node with shape 2x4x8

Torus topology for connecting nodes

Alternate and sophisticated user defined mappings are possible

BOTTLENECK - I/O

Requirements of I/O Infrastructure

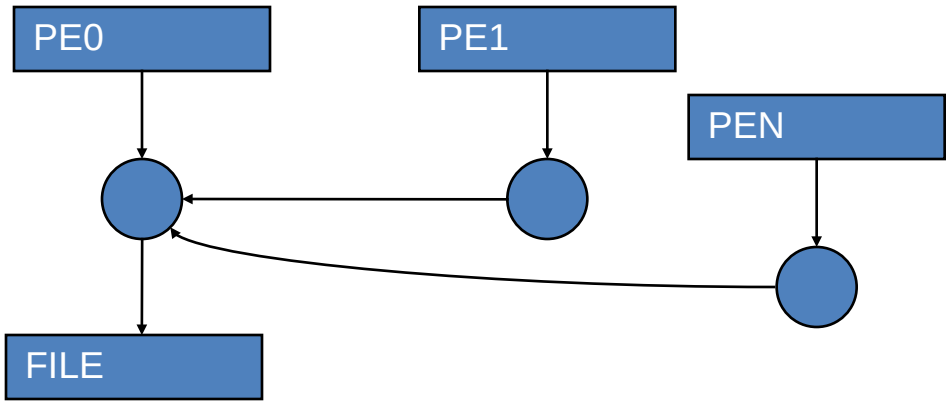
- Efficiency
- Flexibility
- Portability

I/O in EULAG

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

BOTTLENECK - I/O

Sequential I/O: all processes send data to rank 0, PE0 writes it to the file
… memory constrains, single node bottleneck, limits scalability

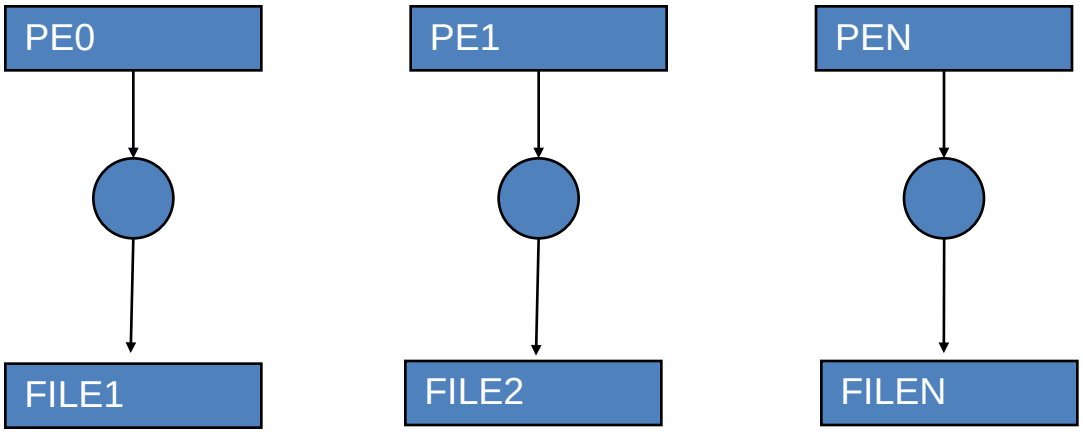


Memory optimization

- sub-domains are sequentially saved without creating single serial domain (require reconstruction of the full domain in post processing mode)

BOTTLENECK - I/O

Different way: Each process writes to a separate file (e.g Netcdf)

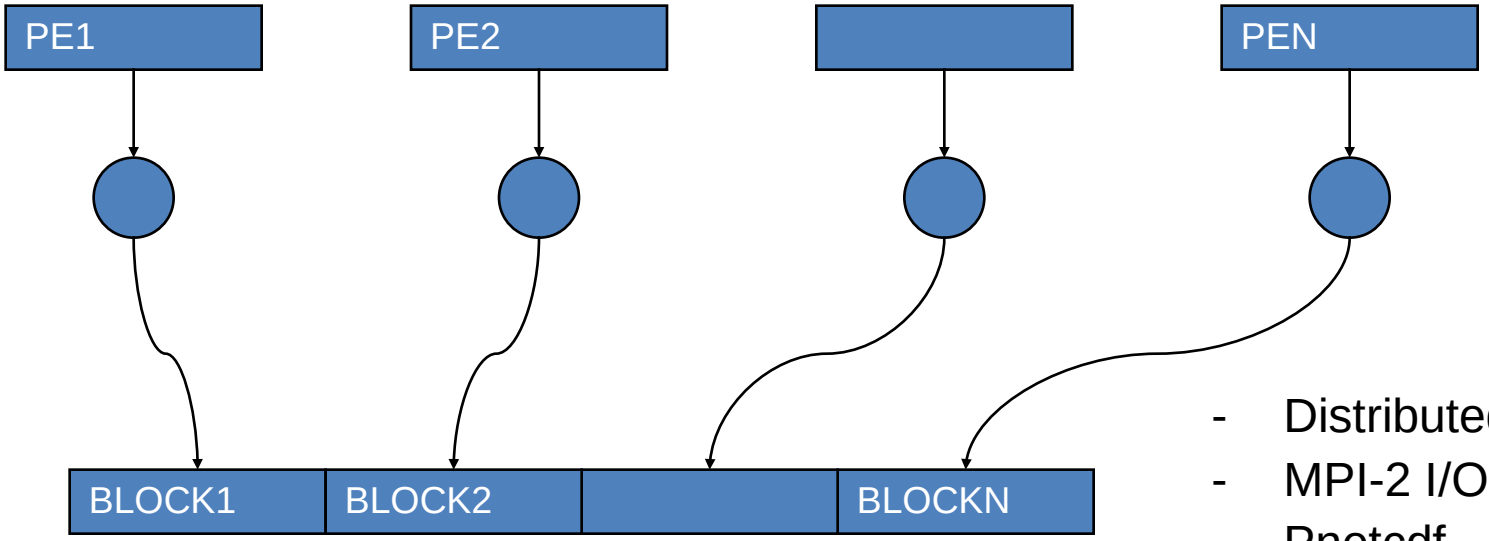


… high performance, scalability

… awkward: lots of small files to manage, difficult to read data from different number of processes

BOTTLENECK - I/O

Need for true scalable parallel I/O: multiple processes accessing data (reading or writing) from a *common* file at the same time



- Distributed File Systems
- MPI-2 I/O
- Pnetcdf

PROBLEMS:

Network bandwidth
Extra coordination required on shared file pointers
Some cluster parallel file systems do not support shared file pointers
Portability: advanced functions in MPI-IO are not supported by all file systems

OPEN QUESTIONS

- How to deal with new Petascale technologies:
  - GPU
  - millions of cores (threads)
- Solutions for scalable/efficient I/O
- Methods to increase peak performance on single node
- Efficient domain decomposition methods on parallel systems