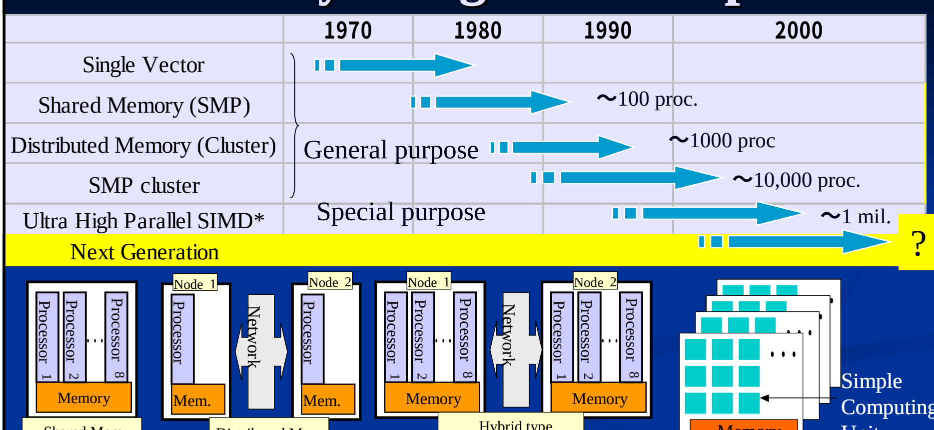
One approach of new parallel architecture for real space descretization methods

Noriyuki KUSHIDA

Japan Atomic Energy Agency

History of high end computers

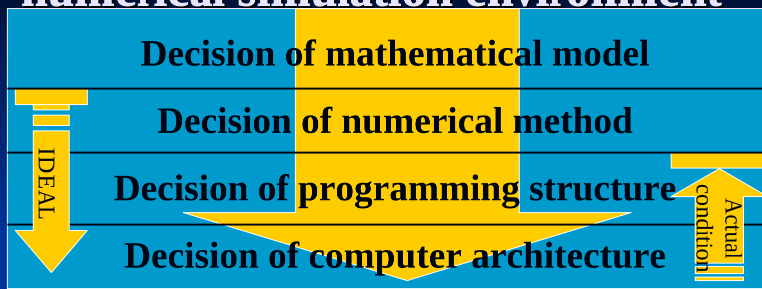


- **Issues for next generation supercomputer**
  - Performance of single processor → plateued progress
  - Increase of number of processors → limit of physical wiring

Bound of the speed-up of general purpose supercomputer

*SIMD：Single Instruction Multiple Data

Construction process of numerical simulation environment



- Ideal process for HPC environment construction
  - Problem decision → Computer architecture decision
- Plateued improvement of performance
  - Programming structure is designed with computer architecture
  - Limitation of development of computer performance

To obtain “Next Generation HPC”,
We must **re-design** new computer architecture
considering **the feature of numerical method**

Linear equation solvers

- Most time consuming part of numerical simulation
  - Static problem, Nonlinear Newton method,,,
- Several type of solvers has been developed for each matrix
  - Sparse, Dense, Positive definite, indefinite,,,,
- In this presentation, sparse matrix was focused on
  - Most “real space descritization” methods provide
  - Computational performance is strongly depends on the data (matrix) structure and solving method

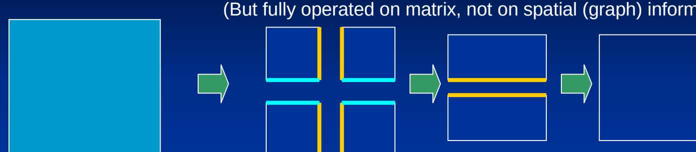
Performance evaluation on existing machine

- Machine
  - SGI Altix 3700@JAEA; 128CPU x 16 Nodes, NUMAlink 3.2Gbps
    - Can be considered as simple shared memory computer
  - SGI PC-cluster@JAEA; 2CPU x 32 Nodes, Infiniband 8Gbps
- Solvers
  - Parallel Jacobi method (in-house)
  - PERCEL (Conjugate Gradient type@JAEA, Japan)
  - PCP (Conjugate Gradient type@AIST, Japan)
  - SuperLU DIST(Direct type @ Berkeley CRD, USA)
  - MUMPS (Direct type@ENSHEEIHT, France)
  - All programs are parallelized with MPI
- Problem
  - Poisson's equation by 3D FEM(8-noded 1st order elements)
  - 1~64 parallel

Overview of Solvers

Direct (Multifrontal) method

Decomposition and DOF condensation
(But fully operated on matrix, not on spatial (graph) information)



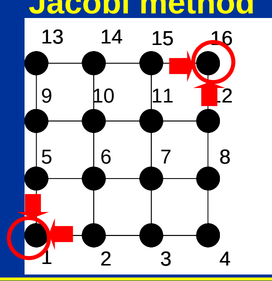
Entire domain

Blue: Unknowns

Green: Reduction

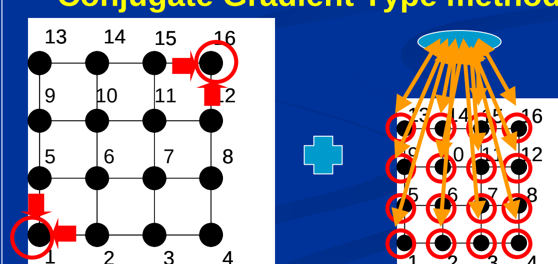
Spatially global operation

Jacobi method



Spatially local (mat-vec)

Conjugate Gradient Type method

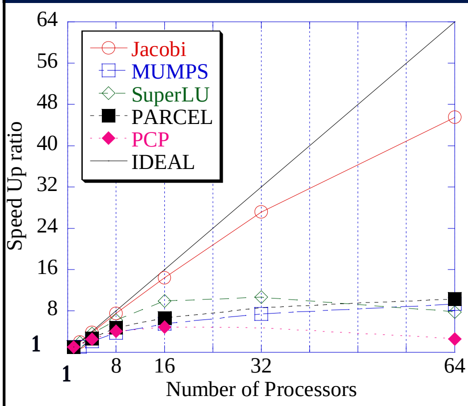


Local (mat-vec)

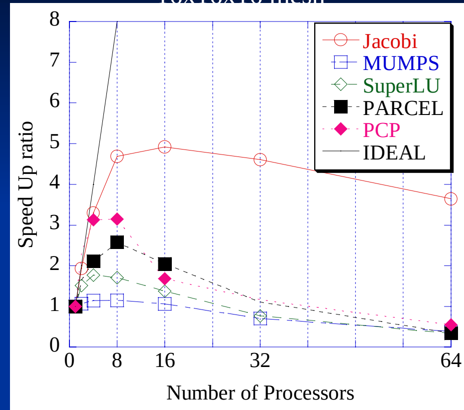
Global (dot product)

Parallel efficiency@Altix

64x64x64 mesh



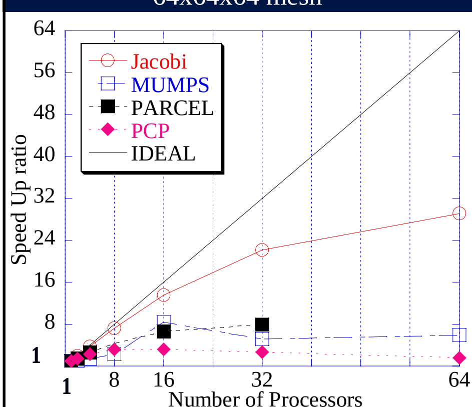
16x16x16 mesh



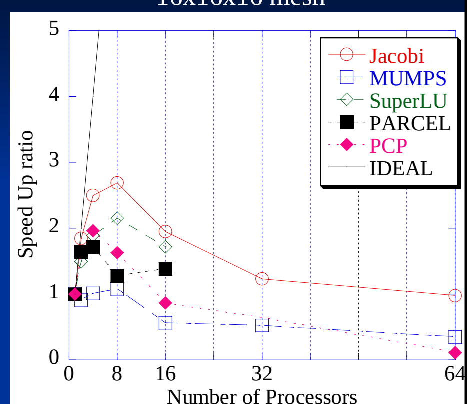
- Jacobi provides best parallel performance
- CG & Direct type lose parallel performance quickly

Parallel efficiency@PC-cluster

64x64x64 mesh



16x16x16 mesh



- Jacobi provides also best parallel performance
- However, loses parallel efficiency faster than Altix
- PARCEL(CG) is second best for both machine

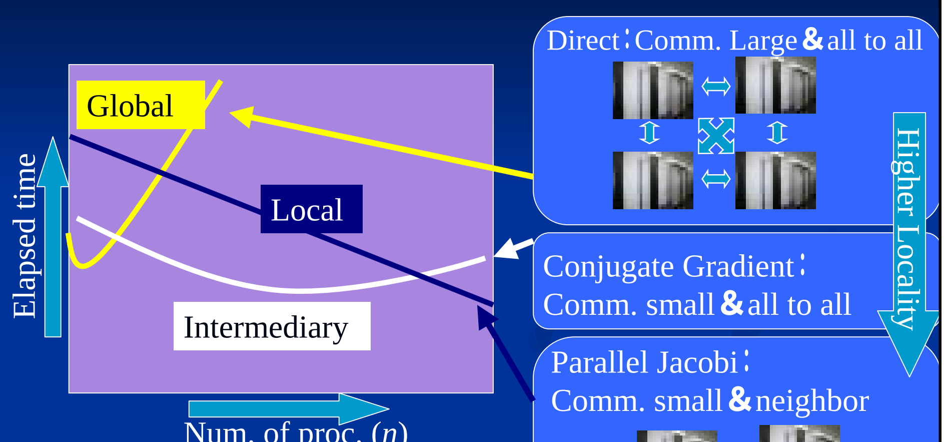
Discussion 1/2

- Parallel efficiency
  - Direct methods are relatively low
  - Parallel Jacobi method is extremely high
  - → Global operation decrease parallel performance

- Elapsed time
  - Conjugate gradient is the best (best choice for current computer)
    - PARCEL, 26sec @ 64CPU, Altix, 64x64x64
  - Parallel Jacobi is the worst while parallel efficiency is good
    - 105 sec @ 64CPU, Altix, 64x64x64

- Assumption
  - If we can construct “ULTRA HIGHLY PARALLEL” computer, Jacobi can become best?

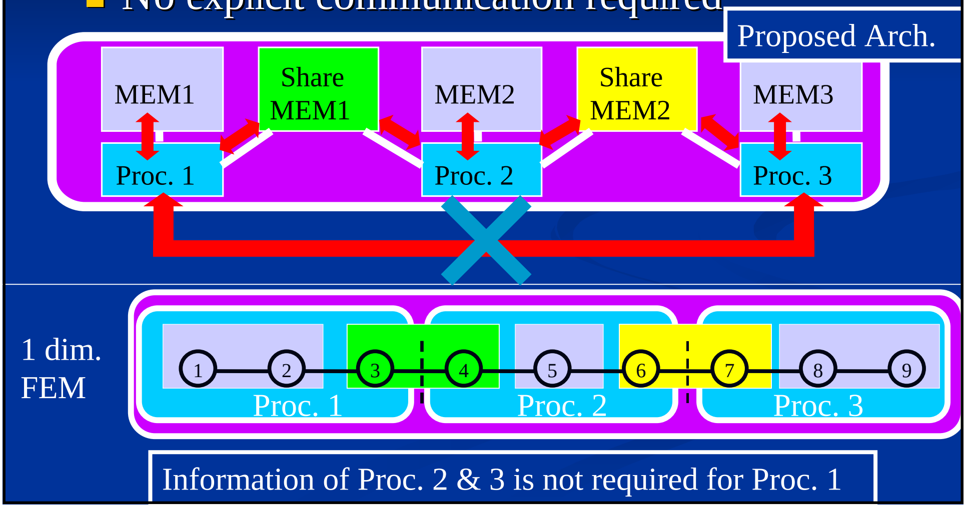
Discussion 2/2



Jacobi is scalable

Highly local operation architecture

- Connected only with neighbor PE
- Available for highly parallel
- No explicit communication required



Conclusions

- Parallel performance of several sparse linear equation solver is examined
  - Jacobi > CG >= Direct
  - The result is agree with operation locality
    - Operation of Jacobi is quite localized
- New computer architecture was proposed
  - Specialized for Jacobi type solver as first step
  - Spatially localized computer
  - It can be used for most explicit dynamic simulation
  - Higher integration degree, easy extension of num. of CPUs