First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the K computer

Paper Discussion by Xiangyu Gao

Real Space DFT (RSDFT)

- Kohn-Sham Equation

$$\mathrm{H}_{KS}[\rho]\phi_i(\mathbf{r}) = \varepsilon_i \phi_i(\mathbf{r})$$

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} \left|\phi_i(\mathbf{r})\right|^2$$

Real Space DFT (RSDFT)

- Kohn-Sham Equation

$$\mathrm{H}_{KS}[\rho]\phi_i(\mathbf{r}) = \varepsilon_i \phi_i(\mathbf{r})$$

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} \left|\phi_i(\mathbf{r})\right|^2$$

Discretized
three
dimensionally

Real Space DFT (RSDFT)

- Kohn-Sham Equation

$$\mathrm{H}_{KS}[\rho]\phi_i(\mathbf{r}) = \varepsilon_i\phi_i(\mathbf{r})$$

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} \left|\phi_i(\mathbf{r})\right|^2$$

$\Rightarrow$ Discretized three dimensionally $\Rightarrow$ Solved numerically (Sparse matrix)

Real Space DFT (RSDFT)

- Kohn-Sham Equation

$$\mathrm{H}_{KS}[\rho]\phi_i(\mathbf{r}) = \varepsilon_i\phi_i(\mathbf{r})$$

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} \left|\phi_i(\mathbf{r})\right|^2$$

$\Rightarrow$ Discretized three dimensionally $\Rightarrow$ Solved numerically (Sparse matrix)

FFT

Real Space DFT (RSDFT)

- Kohn-Sham Equation

$$\mathrm{H}_{KS}[\rho]\phi_i(\mathbf{r}) = \varepsilon_i \phi_i(\mathbf{r})$$

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} \left|\phi_i(\mathbf{r})\right|^2$$

Discretized three dimensionally

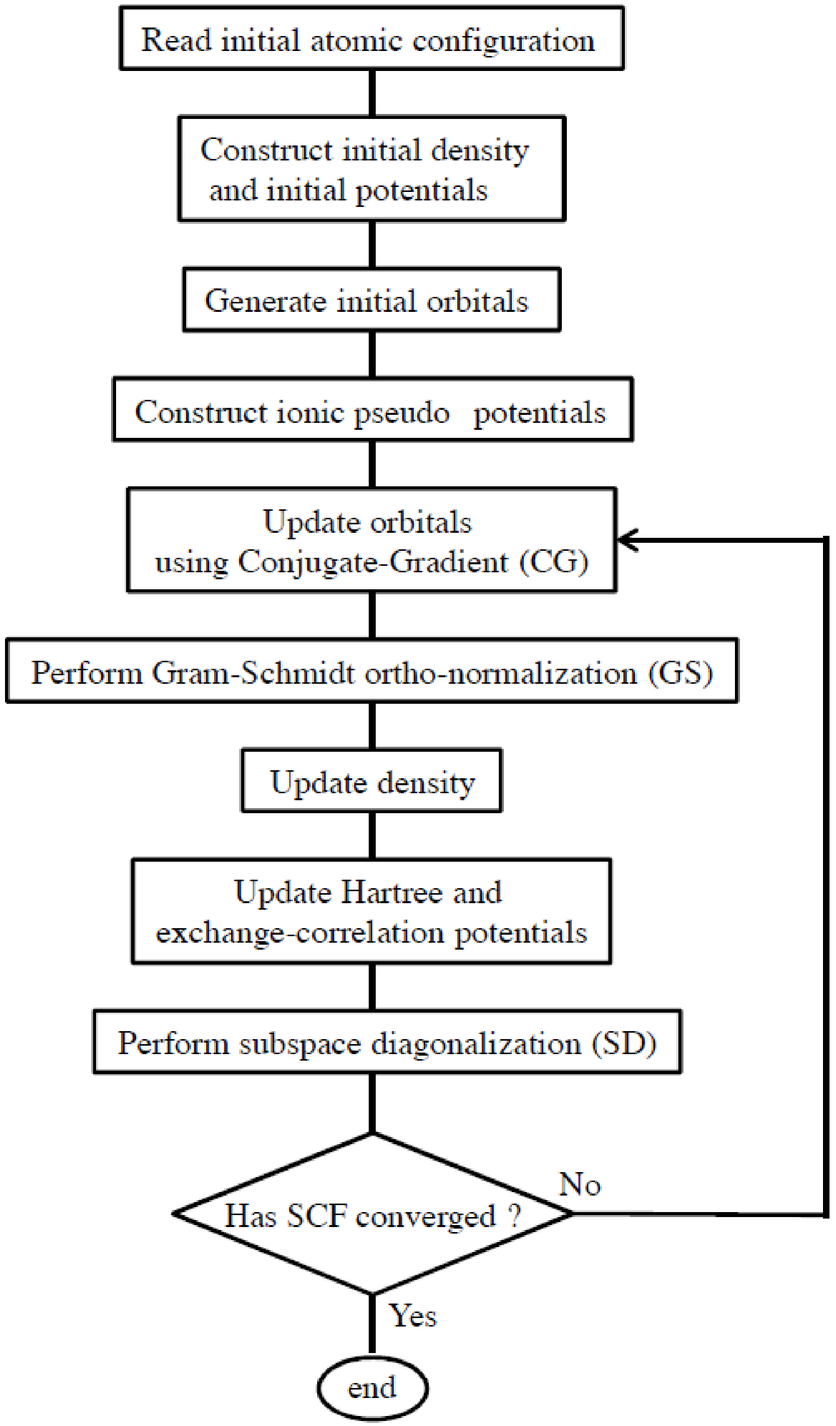
Solved numerically (Sparse matrix)

~~FFT~~

Better for massively parallel computer

Flow Chart

- Read initial atomic configuration (charge density, Hamiltonian)
- Initial electric orbitals are generated randomly.
- Optimize orbitals with Conjugate-Gradient (CG).
- Calculate charge density and Hamiltonian.
- Iterate this process until Hamiltonian and electric orbitals/wave functions are consistent with each other.



Code and K Computer

- RSDFT is developed by

筑波大学
University of Tsukuba

- Paper published in SC 11.
- Won Gordon Bell Prize.

Code and K Computer

- RSDFT is developed by

筑波大学
University of Tsukuba

- Paper published in SC 11.
- Won Gordon Bell Prize.

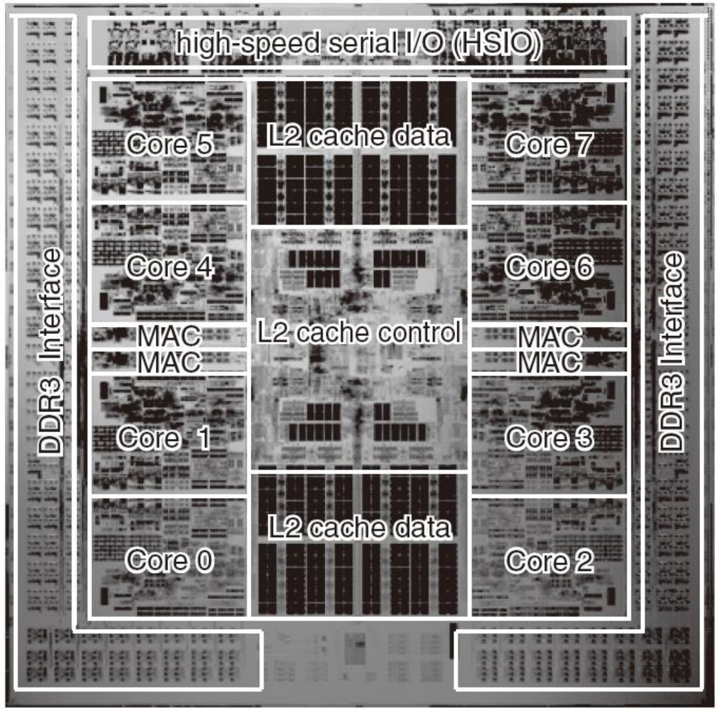
- 10 peta-flops sustained performance for the LINPACK benchmark (dense matrix calculation).
- LINPACK was introduced by Jack Dongarra

- CPU: SPARC64 VIIIfx.
- Interconnection: Tofu(six dimensional)
- SIMD+OpenMP+OpenMPI

SPARC64 VIIIfx

- Produced by

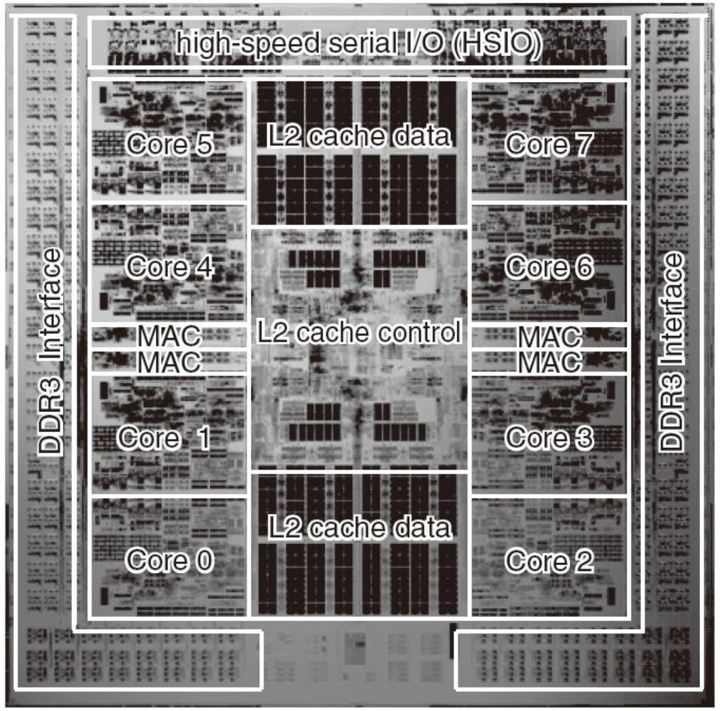
Fujitsu Ltd.
富士通株式会社



SPARC64 VIIIfx

- Produced by

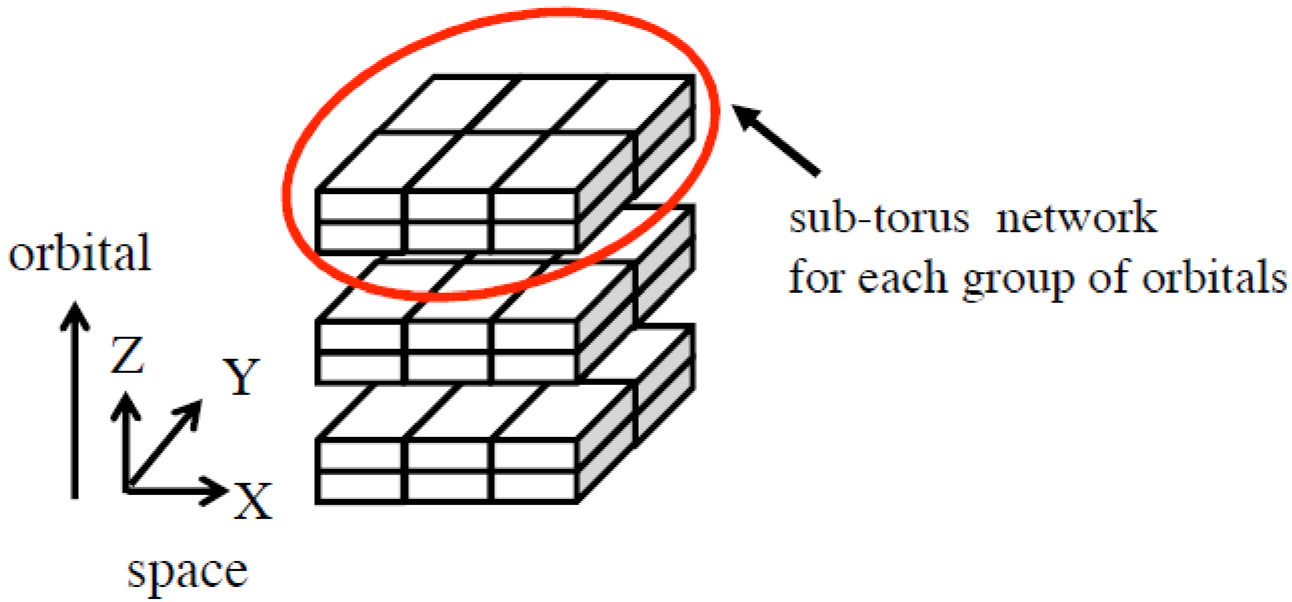
Fujitsu Ltd.
富士通株式会社



- 8 cores
- Shared L2 cache (virtual single processor)
- Hardware barrier/communication (much cheaper than software barrier/communication)
- SIMD: 2 operations per instruction
- Faster instructions for sine and cosine
- Sector cache mechanism
  - Splits cache into two sectors
  - One of them is used to register frequently used data

Parallelization in Orbitals

- Number of MPI tasks are limited to the number of grid points.
- Hard to fully utilize the full K computer
- Parallelization done in orbitals
- Global collectives done only within each group of orbitals, which are closer and cheaper.



Parallelization in Orbitals

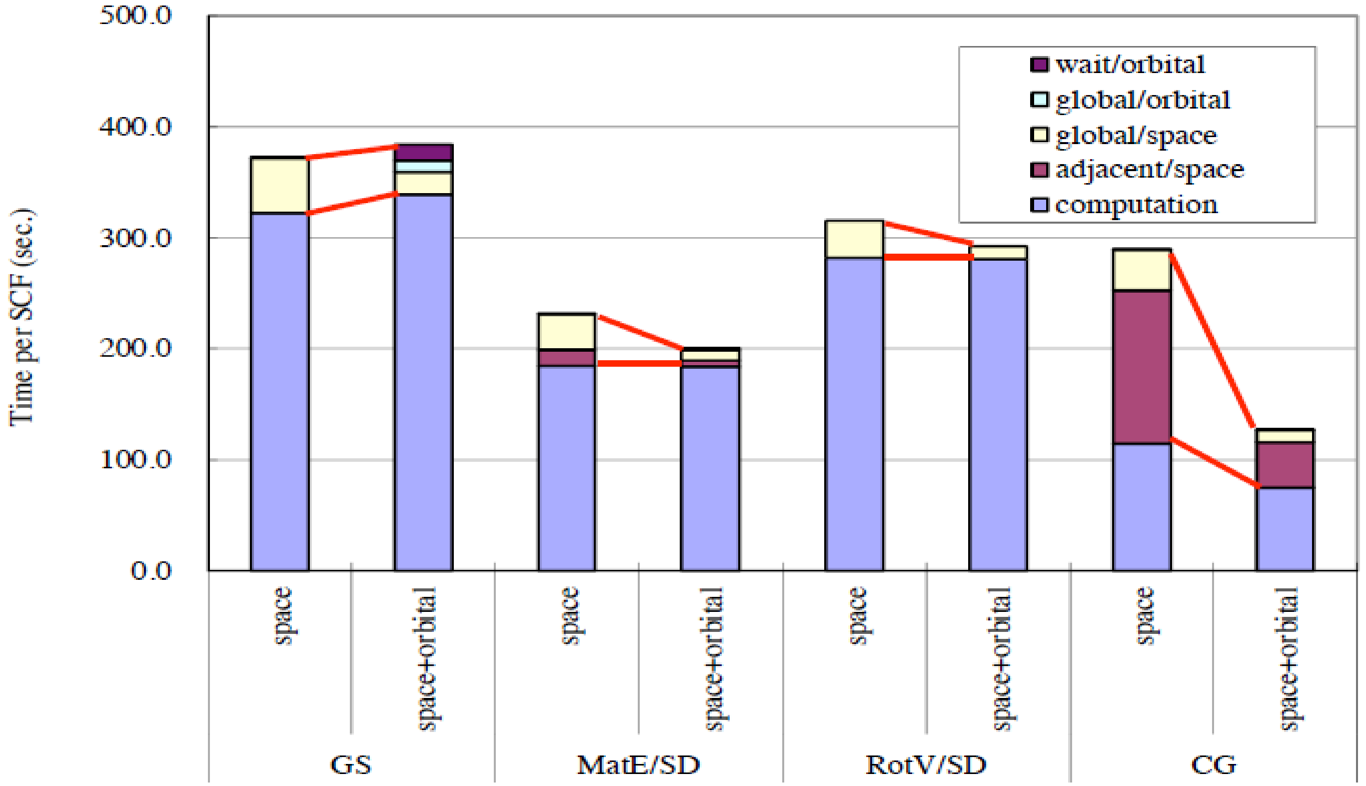


Figure 7. Comparison of execution time of GS, MatE/SD, RotV/SD and CG between spatial parallelization only and combination of spatial and orbital parallelization.

Implementation of Gram-Schmidt

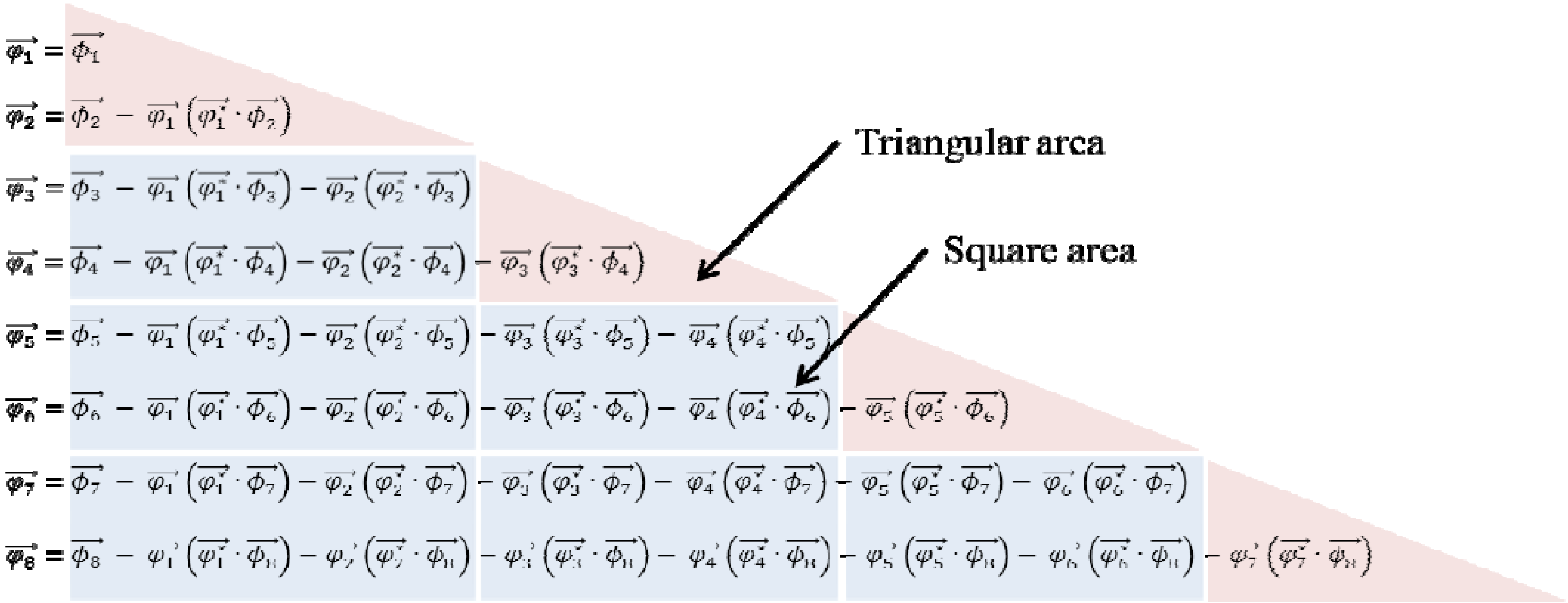
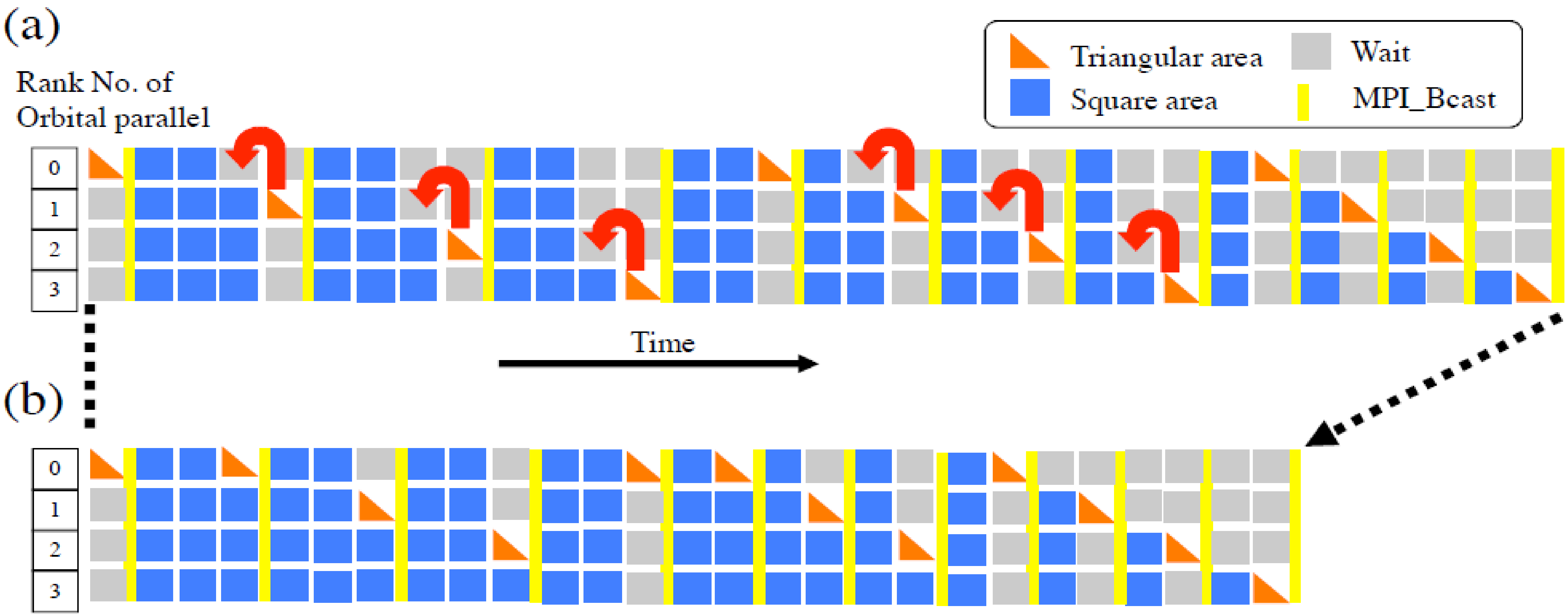


Figure 2. Example of partitioning the ortho-normalization process of orbitals. A triangular area in the first column partition is calculated first, and square areas in the same column partition below the triangular area are calculated concurrently, and so on.

Imbalanced load

Idle processes

Implementation of Gram-Schmidt



Imbalanced load

Idle processes

Implementation of Subspace Diagonalization

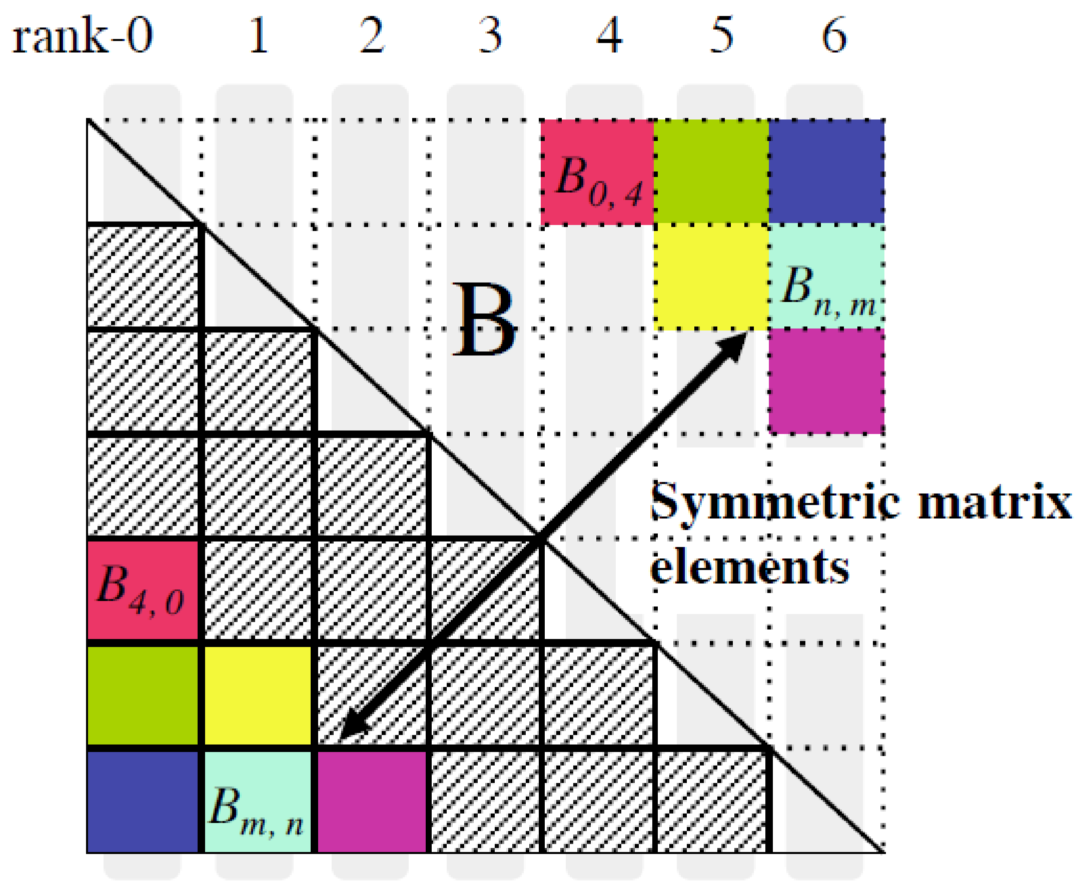


Figure 4. Balanced mapping of a Hermitian matrix. Block square matrices are evenly distributed to parallel tasks.

Overall Performance

Procedure block		Execution time (sec.)	Computation time (sec.)	Communication time (sec.)				Performance (PFLOPS/%)
				Adjacent /space	Global /space	Global /orbital	Wait /orbital	
SCF		5,456.21	4,417.152	83.18	899.05	15.87	40.93	3.08/ 43.63
SD		3,710.01	3,218.728	27.70	458.87	4.70	-	2.72/ 38.52
	MatE	1,084.70	717.45	27.70	337.85	4.70	-	3.09/ 43.72
	EigenSolve	1,322.16	879.61	-	442.39	-	-	0.04/ 0.61
	RotV	1,298.16	1,177.15	-	121.01	-	-	5.18/ 73.25
CG		209.29	57.66	55.48	96.14	0.01	-	0.05/ 0.74
GS		1,536.90	1,140.76	-	344.04	11.16	40.93	4.37/ 61.87

Quite good

Overall Performance

Procedure block	Execution time (sec.)	Computation time (sec.)	Communication time (sec.)				Performance (PFLOPS/%)
			Adjacent /space	Global /space	Global /orbital	Wait /orbital	Quite good
SCF	5,456.21	4,417.152	83.18	899.05	15.87	40.93	3.08/ 43.63
SD	3,710.01	3,218.728	27.70	458.87	4.70	-	2.72/ 38.52
MatE	1,084.70	717.45	27.70	337.85	4.70	-	3.09/ 43.72
EigenSolve	1,322.16	879.61	-	442.39	-	-	0.04/ 0.61
RotV	1,298.16	1,177.15	-	121.01	-	-	5.18/ 73.25
CG	209.29	57.66	55.48	96.14	0.01	-	0.05/ 0.74
GS	1,536.90	1,140.76	-	344.04	11.16	40.93	4.37/ 61.87

Overall Performance

Procedure block		Execution time (sec.)	Computation time (sec.)	Communication time (sec.)				Performance (PFLOPS/%)
				Adjacent /space	Global /space	Global /orbital	Wait /orbital	
SCF		5,456.21	4,417.152	83.18	899.05	15.87	40.93	3.08/ 43.63
SD		3,710.01	3,218.728	27.70	458.87	4.70	-	2.72/ 38.52
	MatE	1,084.70	717.45	27.70	337.85	4.70	-	3.09/ 43.72
	EigenSolve	1,322.16	879.61	-	442.39	-	-	0.04/ 0.61
	RotV	1,298.16	1,177.15	-	121.01	-	-	5.18/ 73.25
CG		209.29	57.66	55.48	96.14	0.01	-	0.05/ 0.74
GS		1,536.90	1,140.76	-	344.04	11.16	40.93	4.37/ 61.87

Quite good

DGEMM in BLAS