Acceleration of Numerical Turbine using the Red–Black Method

TOHOKU UNIVERSITY

Yuta Hougi, Kazuhiko Komatsu, Osamu Watanabe, Masayuki Sato, Hiroaki Kobayashi

Background

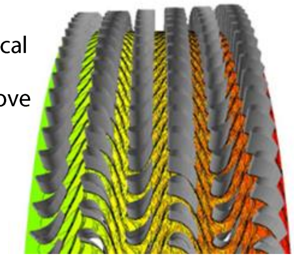
Numerical Turbine

- Steam turbines in thermal and nuclear power plants supply about 70% of the electricity in the world.
- Numerical Turbine is a program to simulate the physical state of the fluid around the turbine cascade [1].
- It is used to design efficient steam turbines and improve their maintainability.

Problem: Long execution time

- Expansion of the simulation scale of Numerical Turbine increases execution time.
- One of the time integration routines called "implicit" occupies about 35% of the total execution time.

→ It is needed to accelerate the implicit routine.



EX. droplet mass fraction distribution

Theoretical background of the implicit routine

- This routine calculates physical fields ΔQ^n of grid points derived from the vector of flux F_i, viscous term S, and source term H of the Navier-Stokes equation.

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}_i}{\partial \xi_i} + \boldsymbol{S} + \boldsymbol{H} = \boldsymbol{0} \quad (i = 1,2,3)$$

$$\Delta \boldsymbol{Q}^n = -\Delta t \left(\frac{\partial \boldsymbol{F}_i^n}{\partial \xi_i} + \boldsymbol{S}^n + \boldsymbol{H}^n \right)$$

- It is necessary to perform vector computing in the routine to exploit computational performance of modern high-performance computers.

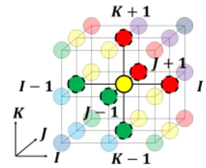
Calculation in the implicit routine

Data dependency of the implicit routine

- Calculating ΔQ^n of point (I, J, K) requires ΔQ^n of points $(I-1)$, $(J-1)$ and $(K-1)$ and ΔQ^{n-1} of points $(I+1)$, $(J+1)$ and $(K+1)$.
- Points being updated and their neighbors cannot be calculated at the same time.

Hyperplane method

- A group of grid points with the same color/number in the right figure is called a hyperplane [2].
- Adjacent points of the m-th hyperplane belong to the $m-1$-th and $m+1$-th ones.
- The grid points in one hyperplane can be updated at the same time.
- The hyperplane method allows vector calculations in the same hyperplane.



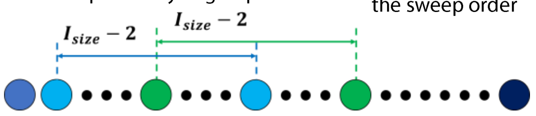
Data dependency of grid points

Hyperplane method and the sweep order

Bottleneck of the hyperplane method: Large stride memory accesses

- The vector calculations by the hyperplane method require stride memory accesses with a distance corresponding to the grid size.
- The stride memory accesses increase the number of loaded memory blocks.

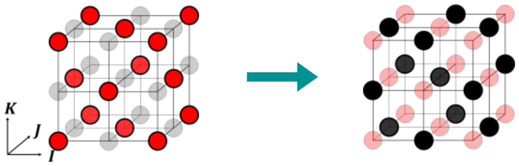
→ The memory access time becomes a bottleneck.



The stride distances among the same hyperplane points in the memory

Approach

Red-Black method



1st : Calculate all the red points referring ΔQ^{n-1} of black points

2nd : Calculate all the black points referring ΔQ^n of red points

- The colors of the neighboring points are different from each other.
- The Red-Black method avoids dependency among the adjacent points.
- The same color group can be updated at the same time.
- The calculations of the same color points can be vectorized.

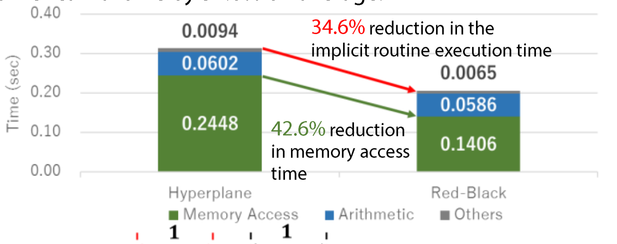
Memory access pattern

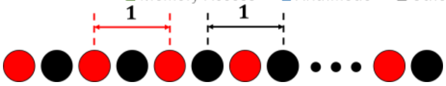
- The red and black points are arranged alternately in the memory, which performs unit stride accesses whose distances are 1.

→ The number of memory accesses can be reduced compared to the hyperplane method.

Evaluation

- We evaluate the implicit kernel with the hyperplane method and the Red-Black method using NEC SX-Aurora TSUBASA Vector Engine Type 10B.
- The Red-Black method can speedup the dominant routine of Numerical Turbine by 34.6% on average.





The unit stride distances among the same color points in the memory

Conclusions and future work

- We discuss applying the Red-Black method to the implicit routine to reduce the number of memory accesses.
- The Red-Black method can reduce the execution time of the implicit routine by 34.6% compared with that of the hyperplane method.
- Since the Red-Black method is not limited to conventional problem division, blocking for efficient use of cache is investigated.
- Further investigation is required for understanding the trade-off between performance and accuracy.

References

[1]S. Yamamoto et al., "Parallel computation of condensate flows through 2-d and 3-d multistage turbine cascades." In Proceeding International Gas Turbine Congress, 2007.

[2]H. Matsuoka et al., "Program optimization of Numerical Turbine for vector supercomputer SX-ACE." In Proceeding of Parallel CFD2016, p. 8, 2017.

Acknowledgements

This work was supported in part by MEXT as "Next Generation High-Performance Computing Infrastructures and Applications R&D Program," entitled "R&D of A Quantum-Annealing-Assisted Next Generation HPC Infrastructure and its Applications."

The authors are grateful to Professor Yamamoto and members of Yamamoto laboratory for providing the numerical turbine and their kind guidance, and to Mr. Watanabe who gave me technical comments.

(URL) https://www.cal.is.tohoku.ac.jp
(E-mail) yuta.hougi.t8@dc.tohoku.ac.jp