XC50 System Application Appendix

Parallelization Efficiency

In order to make efficient use of limited computing resources, the applicants are asked to describe the parallelization efficiency as an indicator when applying for all the categories except for XC-Trial. This document explains how to measure the parallelization efficiency. **Important points are described in bold.**

There are two ways to measure the parallelization efficiency: strong scaling and weak scaling. The applicants are required to show the parallelization efficiency measured by either one of the scaling methods. **If there is no particular reason, please indicate the parallelization efficiency for strong scaling.**

Strong Scaling

Strong scaling is defined as how its execution time varies with the degree of parallelism[1] for a fixed total problem size. The execution time to complete a fixed problem with the degree of parallelism k is denoted by T_k. In order to measure the parallelization efficiency, the execution time with the degree of parallelism n is compared with that with a fixed m ($m \leq n/2$) as follows:

$$\alpha = \frac{mT_m}{nT_n}. \tag{1}$$

With the strong scaling method, if the degree of parallelism is doubled, the amount of work is halved for each processor.

Weak Scaling

Weak scaling is a method of comparing execution times by changing the degree of parallelism while keeping the amount of work per processor the same. The parallelization efficiency α is expressed by the following equation:

$$\beta = \frac{T_m}{T_n}. \tag{2}$$

With the weak scaling method, if the degree of parallelism is doubled, the total problem size increases by a factor of two.

[1]The degree of parallelism is the number of divisional processes, which is basically the product of the number of MPI processes and the number of OpenMP (or auto-parallel) threads.

Notice

- The problem size is the sum of the number of grids in fluid calculations and the number of particles in N-body calculations.
- The execution time is the time it took to calculate the main loop in your code, rather than the entire time to complete your job. You may skip the data I/O process during the measurement.
- The degree of parallelism is the number of divisional processes, which is basically the product of the number of MPI processes and the number of OpenMP (or auto-parallel) threads. With modern computers, the degree of parallelism can be changed even when the same number of nodes are used. For the sake of fairness, **please include the hardware used and the number of nodes corresponding to the degree of parallelism you describe.**
- In the above measurement, **the degree of parallelism m should be set to $m \leq n/2$ and then compared.**

(Date of last update of this document, Dec. 17th 2020)