Chapter 1-20181119

1. Since the area of China is 9.6 million square kilometers, we’ll need at least

9.6×106km2×25km×103cubes per km3 = 2.4×1011 grid points

With compute capacity of one billion calculations per second, we’ll need

2.4×1011×100calculations×24hours/109calculations per second = 5.76×105seconds

Roughly 7 days to predict the weather over China for the next one day.

1. We concern about high computational speed, low power consumption and reliability of processors. Developing greater speed is not enough, we should take greater storage that matches strong computational power into consideration as well. Otherwise, the access delay between the CPU and the memory will greatly reduce the performance improvement of high speed. The speed of memory can't keep up with the calculation speed, the processor has to stop and wait for the data to arrive. In addition to increasing the capacity of the cache, there is another way to increase the high-bandwidth near-core memory.

In the global high-performance system, Tianhe No. 2, which ranked first for three consecutive years, used KNC. KNC looks like a GPU and plugs into a PCIe slot. A KNC has achieved 1 TFlops floating-point performance, which means that 1T double-precision floating-point arithmetic instructions can be executed per second. Now KNL can reach 3 TFlops with one CPU. Its maximum configuration reaches 72 cores, 4 cores per core, and each core has two 512-bit vector processing units. In addition, its package has a 16G near-core memory, read and write bandwidth of up to 400GB per second, system DDR4 memory bandwidth of 90GB per second.

1. Data parallelism means dividing the data among diverse processors and each processor apply the whole process for its respective partition data, or rather, do the same for different data; While task parallelism means dividing the whole target into different tasks, each task responsible for one special task, in this case, all processors deal with the whole data. Assume an airport security system, different staff correspond to different parts, check-in, body search and so on, this is so-called task parallesim; On the other hand, each part contains not only one staff member (service window), such that passengers can perform security checks at the same time——data parallesim.
2. **1.1**

qua = n / p;

remain = n % p;

if (my\_rank < remain){

my\_num = qua+1;

my\_first\_i = my\_rank\*my\_num;}

else{

my\_num = qua;

my\_first\_i = my\_rank\*my\_num+remain;}

my\_last\_i = my\_first\_i+my\_num;

**1.3**

divisor = 2;

core\_difference = 1;

sum = my\_value;

while(divisor <= the num of cores){

if (my\_rank % divisor == 0){

partner = my\_rank + core\_difference;

receive data from partner;

sum += received data;}

else{

partner = my\_rank – core\_difference;

send sum to partner;}

divisor \*= 2;

core\_difference \*= 2;}

if (my\_rank == 0){

printf(“the sum is %d\n”, sum);}

**1.4**

bitmask = 1;

divisor = 2;

sum = my\_value;

while ( bitmask < the number of cores){

partner = my\_rank ^ bitmask;

if (my\_rank % divisor = =0){

receive data from partner;

sum += received data}

else{

send sum to partner;}

bitmask <<= 1;

divisor \*=2;}

if (my\_rank == 0){

printf(“the sum is %d\n”, sum);}

1. Problems:

Initially, the gcc version is not specified and cannot be compiled; the source file named test conflicts with the queue name.

Here is screenshots:



