Començat el	dijous, 9 de desembre 2021, 16:27
Estat	Acabat
Completat el	dijous, 9 de desembre 2021, 16:46
Temps emprat	18 minuts 58 segons
Qualificació	**3,00** sobre 3,00 (**100**%)

Pregunta **1**

Correcte

Puntuació 1,00 sobre 1,00

Assume that the 128 elements of vectors a and b are distributed across the 4 nodes (M_{0-3}) of a NUMA multiprocessor system as shown in the following figure and explained below:

That the first block of 32 consecutive elements is stored in M0, the second block of 32 consecutive elements is stored in M1, the third block of 32 consecutive elements is stored in M2, and the last block of 32 consecutive elements is stored in M3.

Each NUMA node consists of a single processor with its own local cache hierarchy and a portion of the physically distributed but logically shared main memory. Caches are kept coherent across (NUMA) nodes by using a directory-based coherence protocol. Each cache and memory line has a number of bits sufficient to store 4 consecutive elements of these vectors. The first element of vectors a and b is aligned to the beginning of a cache line.

Given the following OpenMP parallel region (thread with OpenMP identifier i executes in node M_i):

```
#pragma omp parallel num_threads(4)
#pragma omp single
for (iter=0; i<99; iter++) {
    #pragma omp taskloop num_tasks(4)
    for (int i=0; i<128; i++)
        b[i] = foo1(a[i]);
    #pragma omp taskloop num_tasks(4)
    for (int i=0; i<128; i++)
        a[i] = foo2(b[i]);
}
```

Which of the following statements is correct?

Trieu-ne una:

- The execution of this parallel region does not cause coherence traffic; in other words, since each node has a portion of the vectors and the number of tasks generated in each taskloop equals the number of nodes, there will be no coherence commands interchanged between them.
- The execution of this parallel region only causes coherence traffic during the execution of the first iteration (iter=0) of the outer loop; in other words, once the elements are brought into cache there will be no coherence commands interchanged between nodes.
- ✔ The execution of this parallel region may cause coherence traffic during the execution of all iterations of the outer loop; in other words, it is highly probable that the processor in a node accesses (reads and/or writes) data cached in the cache of other nodes. Well done!

La teva resposta és correcta.

Pregunta **2**

Correcte

Puntuació 1,00 sobre 1,00

For the same architecture, if the parallel region is changed as follows (assuming thread with OpenMP identifier i executes in node M_i):

```
#pragma omp parallel num_threads(4)
{
  int myid = omp_get_thread_num();
  int BS = 128 / omp_get_num_threads();

  for (iter=0; i<99; iter++) {
    for (int i=myid*BS; i<(myid+1)*BS; i++)
        b[i] = foo1(a[i]);
    for (int i=myid*BS; i<(myid+1)*BS; i++)
        a[i] = foo2(b[i]);
  }
}
```

Which of the following statements is correct?

Trieu-ne una:

- The execution of this parallel region does not cause coherence traffic because each node has a portion of the vectors and the static assignment will always assign the same iterations to each thread. Thus, each thread will only access the portion of the vectors stored within the memory of a node. Consequently, there will be no coherence commands interchanged between nodes ✔ Well done!
- The execution of this parallel region only causes coherence traffic during the execution of the first iteration (iter=0) of the outer loop; in other words, once the elements are brought into cache there will be no coherence commands interchanged between nodes.
- The execution of this parallel region may cause coherence traffic during the execution of all iterations of the outer loop; in other words, it is highly probable that the processor in a node accesses (reads and/or writes) data cached in the cache of other nodes.

La teva resposta és correcta.

Pregunta **3**

Correcte

Puntuació 1,00 sobre 1,00

Assume now that the 128 elements of vectors a and b are allocated only in the node associated to M_0. This has happened because 1) the programmer simply forgot to parallelize the initialization loop:

```
for (int i=0; i<128; i++) {
    a[i] = random();
    b[i] = random();
}
```

and 2) because of the usual "first touch policy" that is applied in NUMA systems. As in the previous two questions, assume thread with OpenMP identifier i executes in node M_i. Compared to the previous two questions, in which vector elements were distributed among all the nodes, which of the following statements is true:

Trieu-ne una o més:

- The two previous versions of the parallel region (the one with taskloop and the other with static schedule) will not cause now any coherence traffic since vectors are stored in a single node (M_0) of the system.
- The version based on taskloop will now behave even worse since all coherence commands should be served by the same directory structure in node M_0. ✔ Correct!
- The code in Question 2, which does a static schedule of iterations to each thread, will have a similar behaviour after the execution of the first iteration (iter=0) of the outer loop. However, during the first iteration of the outer loop the behaviour will be worse now since all coherence commands should be served by the same directory structure in node M_0. ✔ Correct!

La teva resposta és correcta.

◄ Questions Unit 4: Recursive task decompositions, cut-off and dependences

Salta a...

Lab1 laboratory assignment ►