Stream Benchmark

Nicole Orzan

February 9, 2018

The aim of today's exercise session was to run the STREAM benchmark (taken from https://www.cs.virginia.edu/stream), which is a benchmark program that measures sustainable memory bandwidth (in MB/s) and the corresponding computation rate for simple vector kernels. We had to use it to:

1. Estimate the bandwidth of one single core of ulysess/c3e;
2. Compare the bandwidth of two different cores, reading from memory associated to the socket of core 1 against the memory associated with the other socket;
3. Estimate the overall bandwidth of one node of ulysess/c3e;
4. Do it for all threads available in one socket and make a plot of the results.

I built stream.c using the provided Makefile, and I worked on Ulysses and Cosilt cluster.

Cosilt's results

Output of the numactl –hardware command:

```
[norzan@b15 stream]$ numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11
node 0 size: 32726 MB
node 0 free: 31172 MB
node 1 cpus: 12 13 14 15 16 17 18 19 20 21 22 23
node 1 size: 32768 MB
node 1 free: 31795 MB
node distances:
node   0   1
  0:  10  21
  1:  21  10
```

So we have available two sockets with 12 cores each (24 in total).

1 and 2 - Single Core

First of all I estimated the bandwidth of a single core of the node using the serial version of the code; I ran it 10 times to get a mean value from the sample. To run it I used the command:

```
(for i in `seq 1 10`; do OMP_NUM_THREADS=1 numactl --cpunodebind 0
--membind 0 ./stream.x; done) | grep "Triad:" | awk '{print␣$2}'
```

were I used numactl to lock the process to socket 0 and local memory 0. The average result I obtained from the runs is: 12300.18 ± 19.83 MB/s.
To compare the bandwidth reading from memory associated to the socket of core 1 against the memory associated with the other socket I changed the variable membind to 1. The average of the runs in this case is: 7562.11 ± 16.86 MB/s.

num threads	NUMA region 0 [Mb/s]	NUMA region 1 [Mb/s]
1	12447.1	7740.7
2	22428.1	14101.6
3	27263.4	20622.8
4	29870.3	22772.0
5	30549.5	23474.1
6	31613.7	23735.4
7	31740.7	23968.2
8	31990.4	23985.3
9	32045.4	23918.7
10	32250.4	23921.3
11	32296.0	24016.2
12	32372.3	24018.3

Table 1: Bandwidths I obtained on c3e cluster changing the number of threads from 1 up to 12.

3 - Node

I estimated the overall bandwidth of one node. To do this I used the openmp version of the code fixing the number of threads to 24 (cause we have 24 cores in total):

```
(for i in `seq 1 10`; do OMP_NUM_THREADS=24 numactl --cpunodebind 0
--membind 0 ./stream_omp.x; done) | grep "Triad:" | awk '{print␣$2}'
```

The average result of 10 runs is: 32162.14 ± 18.75 MB/s.

4 - All threads in one socket

I estimated the bandwidth of one socket. I used again the openmp version of the code, changing this time the number of threads used (from 1 to 12) and I distinguished again the 2 cases of memory association: the one in which the memory was binded to the same socket (membind –0) and the one in which the process was forced to use distant memory (membind –1). I used the following command line:

```
(for i in `seq 1 12`; do OMP_NUM_THREADS=$i numactl --cpunodebind 0
--membind 0 ./stream_omp.x; done) | grep "Triad:" | awk '{print␣$2}'
```

You can find the results of those runs in the table 1.

From the table and the figure we can clearly see that when a core has to use the memory associated to its socket it has an higher bandwidth if compared to the case in which it has to use the memory associated with the other socket.
In figure 1 you can see the two curves of the bandwidths obtained varying the "membind" variable.

Ulysses' results

Output of the numactl –hardware command:

```
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9
node 0 size: 81891 MB
node 0 free: 79785 MB
node 1 cpus: 10 11 12 13 14 15 16 17 18 19
node 1 size: 81920 MB
node 1 free: 79407 MB
node distances:
node   0   1
  0:  10  11
  1:  11  10
```

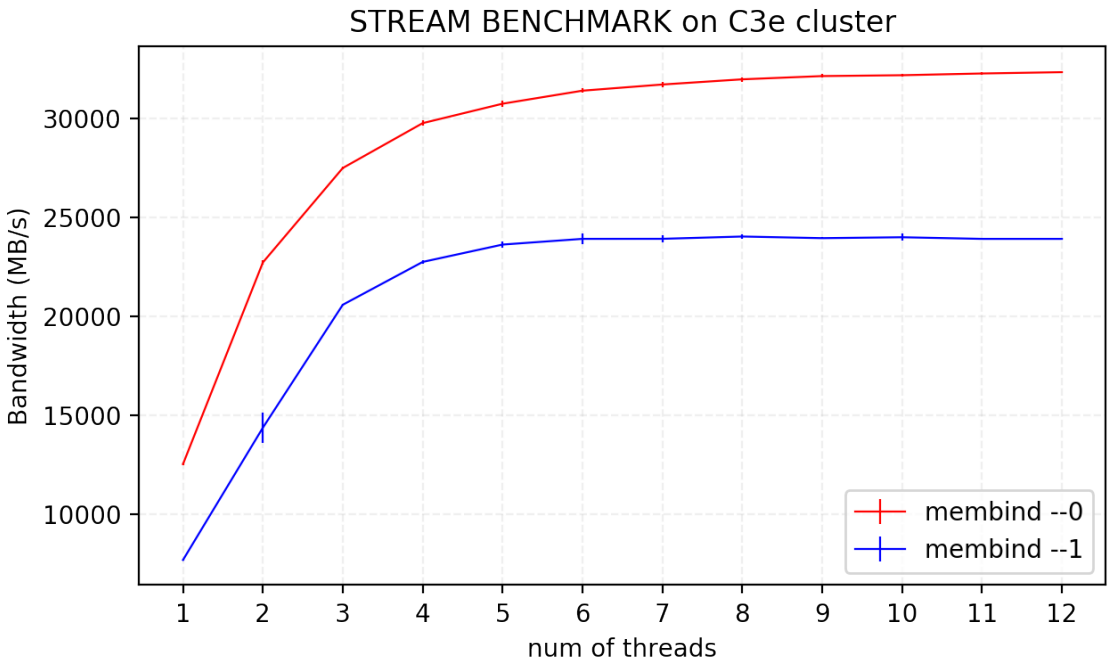

Figure 1: Bandwidths of the node b15 of c3e cluster. Each point is a mean of 10 runs.

So we have available two sockets with 10 cores each (20 in total).

1 and 2 - Single Core

I estimated again the bandwidth of a single core of the node using the serial version of the code and running it 10 times. Locking the process to local memory (–membind 0) the average result is: 13486.70 ± 9.22 MB/s.
Then I locked the process to the memory associated with the other socket (–membind 1), and the average of the runs is: 9824.84 ± 38.17 MB/s.

3 - Node

I estimated the overall bandwidth of one node using this time 20 threads, using again the omp version of the code; I ran the code 10 times and the average result I obtained is: 40252.38 ± 277.71 MB/s.

4 - All threads in one socket

I estimated the bandwidth of one socket changing the number of threads used (from 1 to 10) and I distinguished again the 2 cases of memory association: membind –0 and membind –1. You can find the results of those runs in the table 2.
In figure 2 you can see the two curves of the bandwidths obtained varying the "membind" variable.

We can clearly see that the Ulysses' node has a strange behavior: it doesn't seem that the bandwidth is bigger when we are using memory associated to the same socket (mem0).

num threads	NUMA region 0 [Mb/s]	NUMA region 1 [Mb/s]
1	13423.6	9762.3
2	20151.3	17245.4
3	22012.0	20235.7
4	22051.2	21564.8
5	22156.3	21739.4
6	22426.2	22001.9
7	22019.6	21992.4
8	22361.2	22006.5
9	22112.0	22298.9
10	22094.7	22026.0

Table 2: Bandwidths I obtained on ulysses cluster changing the number of threads from 1 up to 10.

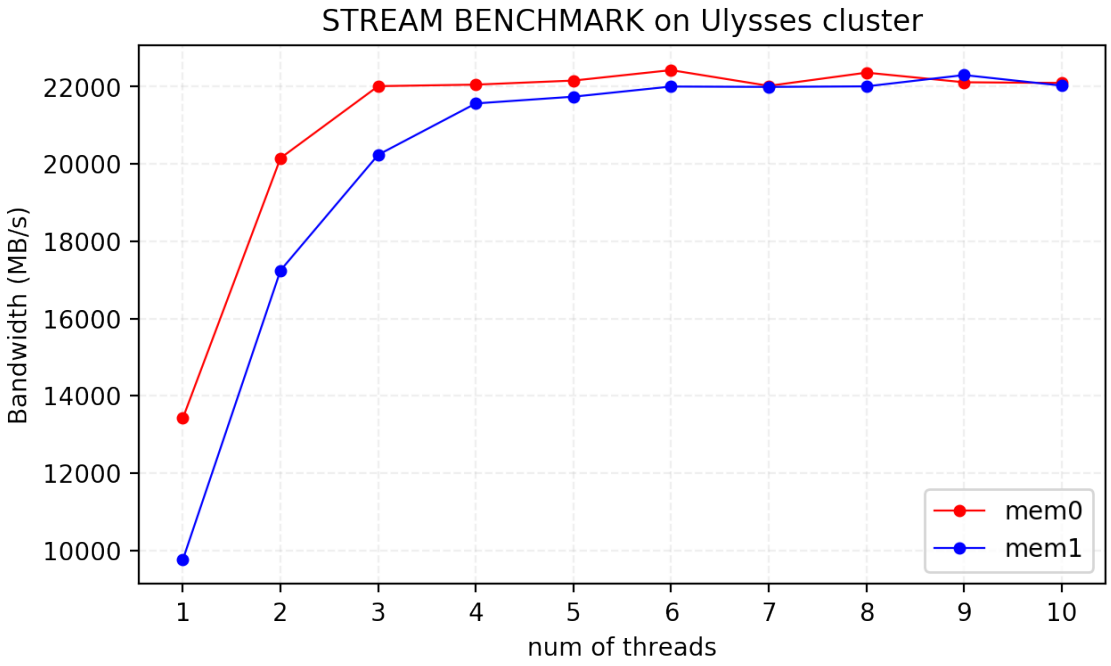

Figure 2: Bandwidths of the node c04-01 of ulysses cluster.