Benchmarking the memory subsystem

Erik Saule

1 Analysis

How is the memory subsystem structured on your machine. On Linux system likwid-topology is a good tool to know how the memory system is architectured.
Question: How much DRAM memory? Memory speed? Memory Technology? (Not all is exposed unless you are root.)
Question: How much bandwidth can your processor draw from the bus? You will need to discover width of the memory bus and clock speed of the memory bus. Full duplex/half duplex? (On intel processor, ARK often indicates something.)
Question: What is the highest level of data cache? How big is it? How is it shared across core? Same question for all levels of data cache.

2 Bandwidth

The purpose of this section is to measure the maximum memory bandwidth of the different components of the system. The easiest way to ensure that is to do the minimum amount of arithmetic operation per byte of data.

For each level of the memory hierarchy we will measure read, write and read/write bandwidth. The easiest way of doing this is to have each core do a measurable number of memory transfer on a piece of data of a particular size. Plot each bandwidth as a function of the size of the data each core work on.

To measure read bandwidth, the easiest test is often to simply compute the sum of an array. To measure write bandwidth, the easiest test is often to set a memory region to zero. To measure read/write bandwidth, the easiest test is often to copy an array in an other one.

The measurement itself can be an issue because of the fill-in (what happens at the beginning) and flush-out (what happens at the end). A reasonable way to measure is to loop over your main operation multiple time and time only the middle loop iterations to make sure the measurement are carried out while all the cores are busy.
Question: Write a code to measure read bandwidth
Question: Write a code to measure write bandwidth
Question: Write a code to measure read/write bandwidth
Question: Measure read, write, read/write bandwidth at different size of data. (From 1KB to 200MB)

3 Latency

The best way to measure memory latency is to perform memory operations that are not easily predicted. Linked list are certainly king in that context. Write an element-less singly linked list. (To be clear it is simply an array of integer next that refers to itself so that you traverse it by doing `current = next[current];`.)

For different size of the list, measure the time it takes to follow a large number of links and report the time per link followed.

You will need to set the list in a particular way depending on what you want to see.
Question: Write a code that traverses some element of a linked list as described above.

Question: Write function to create a linked list which structure is aggreable to the core structure.
Question: What is the latency in that case?
Question: Write a function to create a linked list that will jump around the memory subsystem while avoiding to build short cycles.
Question: For different size (From 1kB to 200MB), what is the latency of the system?

4 Extra Credit

Question: Write a code that trigger associativity effects. What is the penalty when associativity impacts negatively performance?
Question: Write a code that measures the impact of the TLB. What is the penalty for TLB-trashing?

A Make sure bandwidth is memory bound

Look at the assembly to ensure the it is mostly IO. You can estimate a sufficient arithmetic to IO instruction by looking at flops/bandwidth ratio.

B Alignment

Be wary of of alignment. Look at the difference between `_mm256_load_ps` and `_mm256_loadu_ps`. Allocate memory with `posix_memalign`(3) if needed

C Other interesting things

`_mm256_stream_load_si256` makes a non-temporal read (won't be cached).

D Some remarks about likwid

git clone https://github.com/RRZE-HPC/likwid.git
cd likwid
Edit config.mk to change PREFIX to /users/esaule/local
make
make install
you can then run ~/local/bin/likwid-topology. Option -g of likwid topology is nice

E Details about RAM

"dmidecode –type 17" This command will give details about the DRAM. Such as memory technology, Speed etc. Need to be root to do that, so you can not do that on mamba.

On centaurus you get:

```
# dmidecode --type 17
# dmidecode 3.0
Getting SMBIOS data from sysfs.
SMBIOS 2.8 present.

Handle 0x1100, DMI type 17, 40 bytes
Memory Device
```

```
Array Handle: 0x1000
Error Information Handle: Not Provided
Total Width: 72 bits
Data Width: 64 bits
Size: 16384 MB
Form Factor: DIMM
Set: 1
Locator: A1
Bank Locator: Not Specified
Type: DDR4
Type Detail: Synchronous Registered (Buffered)
Speed: 2133 MHz
Manufacturer: 00AD063200AD
Serial Number: 261D8909
Asset Tag: 001607B1
Part Number: HMA42GR7AFR4N-TF
Rank: 2
Configured Clock Speed: 2133 MHz
Minimum Voltage: 1.2 V
Maximum Voltage: 1.2 V
Configured Voltage: 1.2 V
```