

Report - Cache Awareness

Machine Configurations:

```
Architecture:           x86_64
CPU op-mode(s):         32-bit, 64-bit
Byte Order:             Little Endian
Address sizes:          39 bits physical, 48 bits virtual
CPU(s):                 8
On-line CPU(s) list:    0-7
Thread(s) per core:     2
Core(s) per socket:     4
Socket(s):              1
NUMA node(s):           1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  142
Model name:             Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz
Stepping:               10
CPU MHz:                1900.000
CPU max MHz:            3600.0000
CPU min MHz:            400.0000
BogoMIPS:               3799.90
Virtualization:         VT-x
L1d cache:              128 KiB
L1i cache:              128 KiB
L2 cache:               1 MiB
L3 cache:               6 MiB
NUMA node0 CPU(s):      0-7
```

Observations, Views & Conclusions:

- Reverse sequential access is a little bit faster than sequential access.
- Random access is usually 2-3 times slower than both sequential/reverse access.
- When the stride becomes larger, the gap between sequential access and random access is also getting larger and larger.
- Prefetched block access is faster than normal block access.
- The experiments almost match my expectations, except that reverse access is faster than sequential access.
- Careful! Although prefetch can accelerate the process, but how to choose the correct step for prefetching is very important. In my experiment, if we just prefetch every next element in the block, this can impair access speed. One of the reasons is that modern CPU or compiler can already do some kind of "prediction" or "prefetch" for our program, so that if we manually do the prefetch with a wrong way, it will not improve the speed.
- To best speed up block access, we should take care of the address we want to do the prefetch. If we fetch not far enough ahead, then there might be overlap parts in cache. In

my experiments, when the program prefetching from roughly the next 80 elements, it can best accelerate the block access. I think this is because by doing so I can make the most of computer cache.