Caches, Prefetchers and their Impact on Column Scan Performance

Cache Overview

- Underlying Principal: Data Locality
- L1: Separated into Instruction and Data Cache
- L2: Unified Cache
- L3: Shared between all cores
- (L4: Between CPU and DRAM, DRAM itself)
- Inclusive: Everything in L1 is included in L2 is included in L3
- Exclusive: L1, L2 and L3 contain no duplicate data
- Remember cache coherence while writing

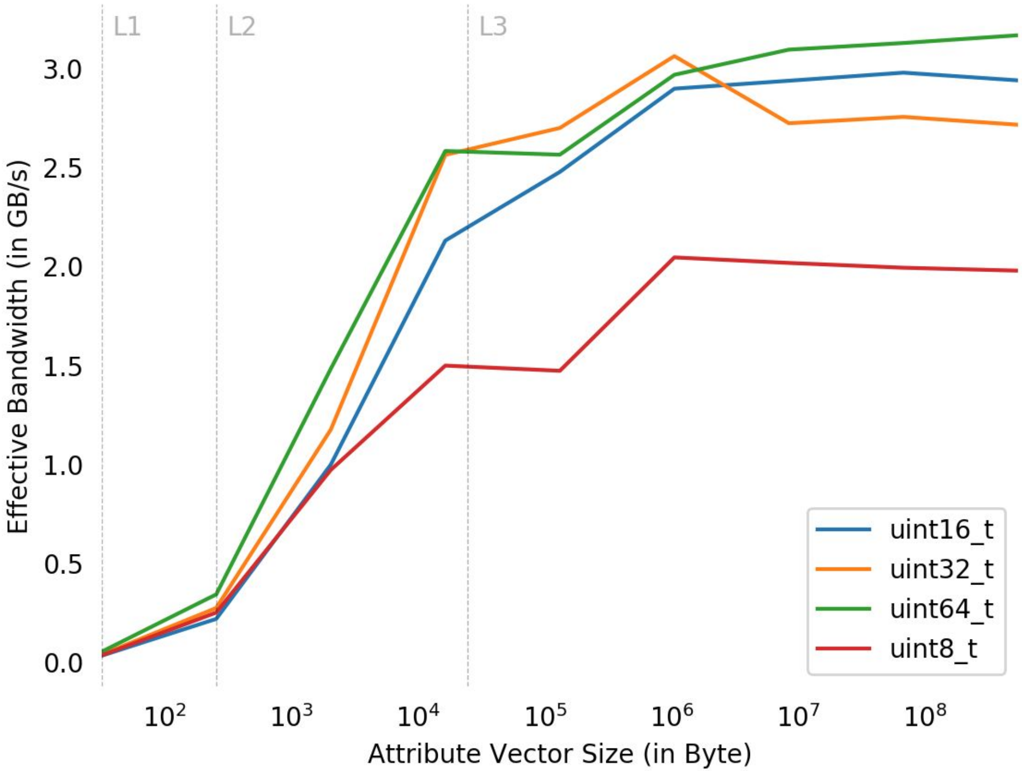
Intel H/W prefetchers

- L2 hardware prefetcher (normal prefetcher)
- L2 adjacent cache line prefetcher (loads an additional cache line)
- DCU prefetcher (L1 Data)
- DCU IP prefetcher (L1 Instructions)

Configuration of Prefetcher

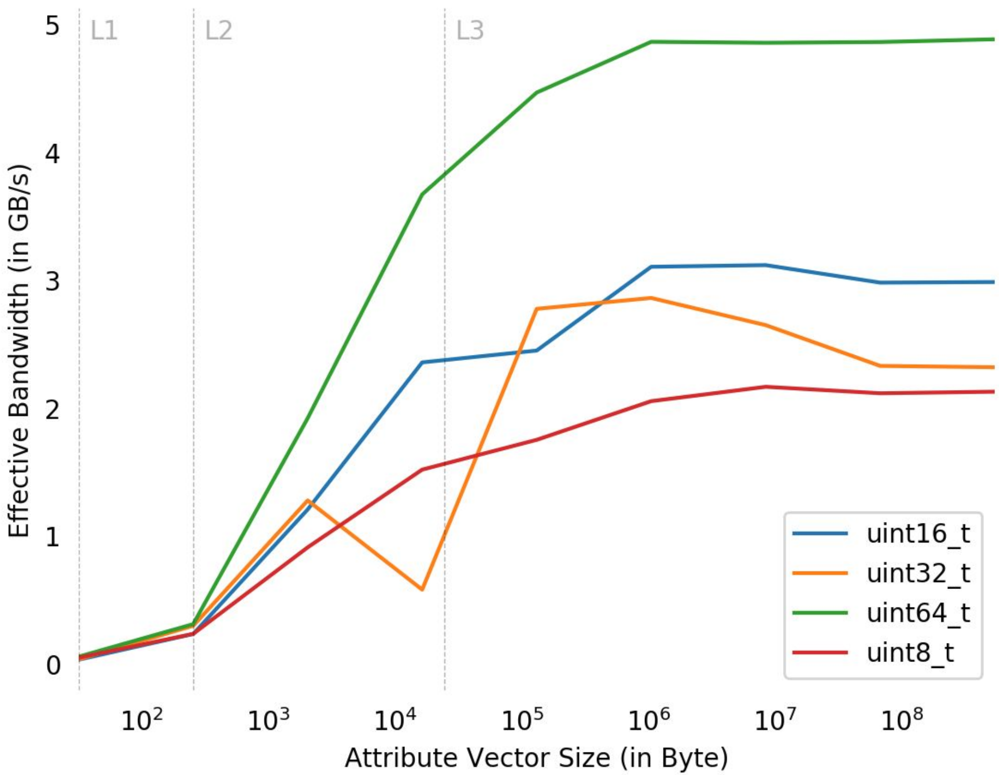
- Configuration in Model Specific Register (MSR)
- MSR-Tools to manipulate bits
    - Nehalem and later: bits 0-3 at 0x1A4
    - Before Nehalem: bits 9 and 19 at 0x1A0

Column Scan Bandwidth as a Function of the Column Size and the Bitcase



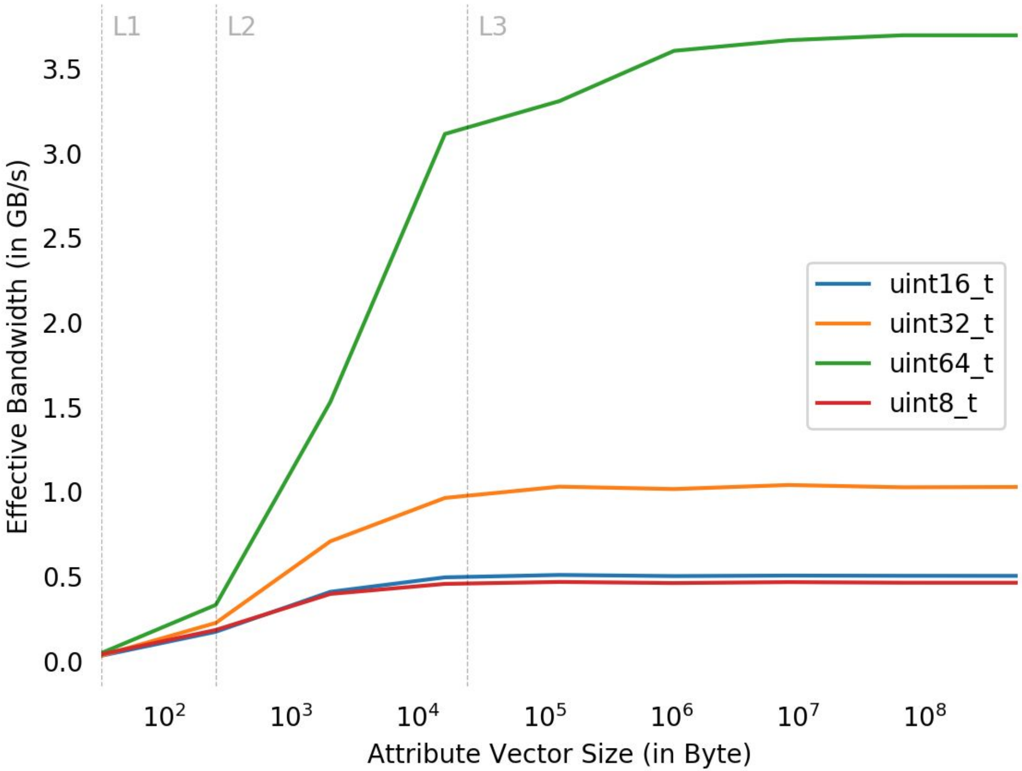
Intel(R) Xeon(R) X7560 @ 2.27GHz, w/ Prefetcher, 1 Thread

Aggregate Bandwidth as a Function of the Column Size and the Bitcase



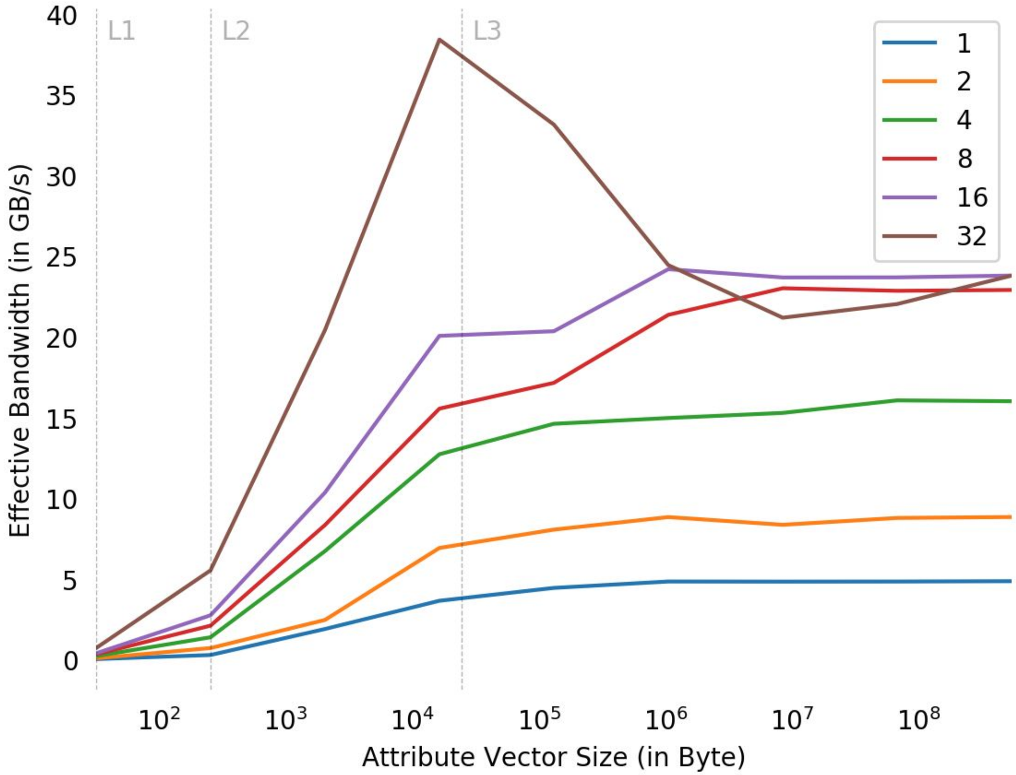
Intel(R) Xeon(R) X7560 @ 2.27GHz, w/ Prefetcher, 1 Thread

Aggregate Bandwidth as a Function of the Column Size and the Bitcase



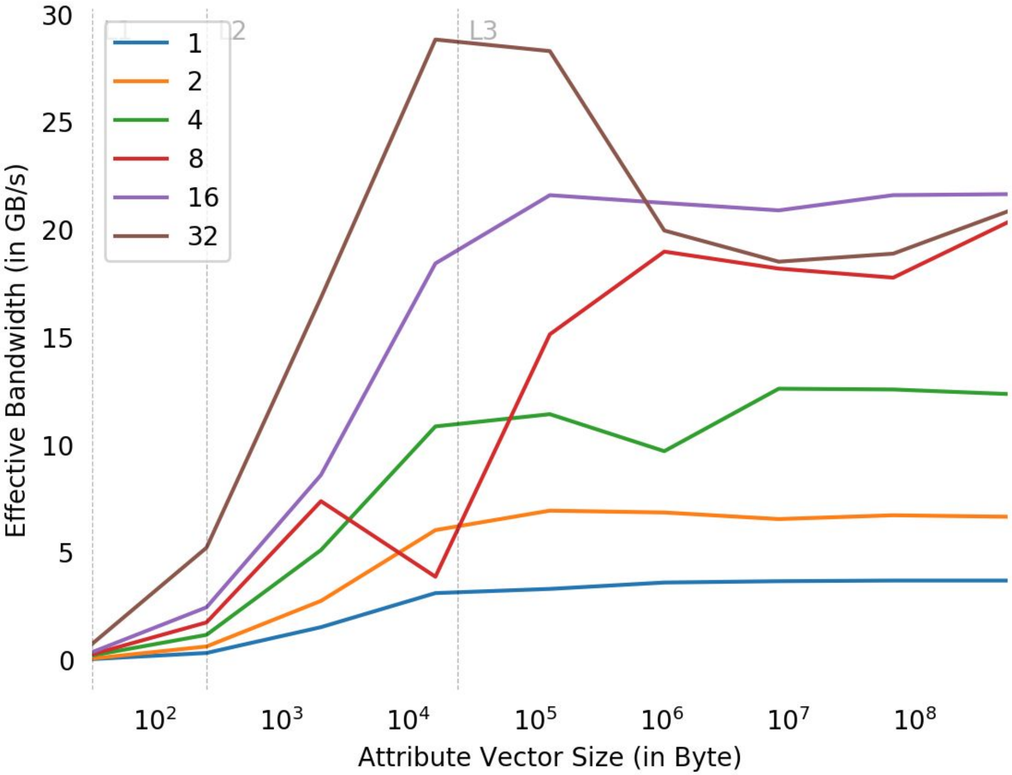
Intel(R) Xeon(R) X7560 @ 2.27GHz, w/o Prefetcher, 1 Thread

Aggregate Bandwidth as Function of Column Size and Thread Count



Intel(R) Xeon(R) X7560 @ 2.27GHz, w/ Prefetcher, uint64_t

Aggregate Bandwidth as Function of Column Size and Thread Count



Intel(R) Xeon(R) X7560 @ 2.27GHz, w/o Prefetcher, uint64_t

Questions

- What part of the CPU‘s available bandwidth can a core utilize for a column scan?
  - Limited by data processing in cpu, not by mem b/w, if prefetcher enabled
- How relevant is the hardware prefetcher for the performance of a column scan?
  - Quite, factor 2
- Does Multi-Threading help in achieving a higher scan bandwidth?
  - Yes, doubling #threads increases performance by 2/3
- How does bringing in SIMD (AVX) change this
  - Boosts processing in cpu and reduces cpu as bottleneck (Aggregate)