Data-centric Profiling Working Group Outbrief

Basic Concept

- Associating performance data with data objects (arrays), beyond code contexts (loops, procedures)
  - PMU support
  - data-centric attribution
  - use of data-centric profiling

Data-centric Profiling WG

- Current PMU support
  - Intel PEBS, AMD IBS, IBM Mark events to sample memory accesses
    - effective address, latency, memory layers
  - monitoring loads only is not enough, but also stores/prefetching instructions
    - use L1D replacement event (https://software.intel.com/en-us/forums/intel-performance-bottleneck-analyzer/topic/326007)
  - better to monitor evicted cache lines
    - Jeff’s paper: http://www.cs.umd.edu/~hollings/papers/ijhpca06.pdf
  - LBR: use call stack mode (monitoring calls/returns) to reconstruct the call stack
    - 16 frames on average with 32 LBR slots
  - Intel PT
    - ptwrite (Goldmont), a lightweight printf triggers LBR. Call ptwrite inside malloc can obtain the call path from LBR
  - page fault events, a hardware event (Goldmont)
    - possible measure first touch location
  - limitation
    - no PID or TID. OS Kernel needs to get this information
    - PEBS latency_above_threshold may produce biased results
      - sample MEM_LOAD/MEM_RETIRED

- Handle attribution to data structures
  - static — easy to handle from symbol table
    - need Dyninst to extract allocation source lines from DWARF
  - heap
    - high overhead if malloc/free are frequently called
      - probably use ptwrite to reduce the overhead
    - call stack is important
      - merge the objects allocated in the same call path
      - (David) meaningful allocation site may a few frames above the "malloc"
  - stack
    - (Xiaozhu) Dyninst supports to extract the information from DWARF

- Use of data-centric profiling
  - locality optimization
    - data layout optimization
      - David has some work in helping developers change data layout
    - temporal locality
  - false sharing
    - HITM events for loads
      - may miss store-store false sharing
    - Intel PTU, toplev, Feather (Xu’s group) identify false sharing
  - NUMA optimization
    - lightweight pattern analysis across threads
  - structure splitting
    - identify how different fields of a data structure are accessed
      - structslim from Xu’s group: https://dl.acm.org/citation.cfm?id=2854053

Challenges

- Stephane: how to do data profiling offline
  - collect all raw data online with low overhead
  - perform data attribution offline
  - timestamp information
- Michael: automate the fix
  - Joseph (UPenn)'s approach of detecting and fixing false sharing
  - Intel PGO can improve a DB workload by 25% to guide global data reorganization on Itanium
- Stephane: compiler support to annotate each memory access instruction
  - which type accessed
  - the offset
- Michael: data-centric profiling on small cores
  - insights for temporal locality