Disclaimer: "The contents of this document are scribe notes for The University of Texas at Austin EE382V Spring 2007, Computer Architecture: User System Interplay. The notes capture the class discussion and may contain erroneous and unverified information and comments.*

Performance of the VAX-11/780 Translation Buffer: Simulation and Measurement

Lecture #08: Wednesday, 21 February 2007
Lecturer: Bob and Carl
Scribe: Kyle Cullen
Reviewer: Min Kyu Jeong

1 Background information

This paper was more of an evaluation paper that shows the details of how the Translation Buffer in a VAX processor works.

VAX was one of the first machines to incorporate Paged Virtual Memory. This caused problems with memory usage. The Translation Buffer is a hardware mechanism that allows for the processor to use address space outside of the actual hardwares memory.

1.1 Reference

References

[1] D. W. Clark and J. S. Emer., "Performance of the VAX-11/780 translation buffer: simulation and measurement", ACM TOCS, 1985

1.2 What Problem is being Solved?

- How is the VAX Translation Buffer characterized?
- How does the TB help / hurt performance
- Long term objectives would be improvements in future VAX designs

1.3 Who are the intended users?

- Readers
  - PC arch who are dealing with Virtual Memory and Paging
  - Operating System Writers
  - More intended for readers than end users
- Users: End Users
  - Educational, Scientific / Engineering, and Commercial communities

1.4 What is Unique?

- Tracing Technique that allows for tracing in all operating modes: user, system
  - Traces only see what processor actually saw
  - Limitation: no speculative threads.
- The writers were able to use hardware and software simulation at the same time
- At the time (1985) the VAX virtual Address space was unique.
- The histogram approach was unique because it kept track of all the op-codes and the microcode instructions being ran on the processor.

1.5 How is the idea evaluated?

- Measured TB miss rate, TB hit time, TB miss fill time
- Moved from misses per access to misses per instruction because it helped to eliminate some of the dependence on the specific processor organization.
  - This metric is very VAX-specific; allowing fair comparisions between different VAX machines and future architectures, but not against different machines.
  - This metric will not be as sensitive to data-path access widths as misses per hardware reference.
  - It allows simple computation of the relative importance of the TB miss service time to processor performance.

	Hardware	SW - Simulation
Pros	Accurate Fast More Accurate	No special hardware Flexible Reproducible System simulator
Cons	Need instrument to attach to Can not measure hardware changes Not repeatable	Difficult to represent real work load Expensive in computation resources Slow Inaccurate if simulated hardware is inaccurate

- Three Tests
  - DynaProbe
    - Full Day - real process
    - Remote Emulator (5min)
    - Reproducible
  - Microcode Histogram
    - Real workload
    - Remove Emulator (1hr)
    - Average Behavior Characteristics
  - Trace Driven Simulator
    - System level
    - 7 benchmarks
- x = used

Test Number hline 1	Desc Miss Ratio	DynaProbe Histogram x	Microcode	Simulator
2	Miss / Instr	x	x	x
3	Types of Misses	x	x	x
4	I-Stream		x	
5	D-Stream		x	
6	M-bit setting		x	
7	Time to Service	x	x	
8	CPI	x	x	
9	Context Switching	x	x	x
10	Invalidations		x	
11	Intr Mix	x	x	x
12	TB Size(half)	x		
13	Associativity / TB Size			x
14	Split/Join			x

- 3% of instructions missed in TB
- Double misses don't occur that often because they are only $\tilde{3}$% of all misses
- 90% of misses in I-Stream are Branches and 10% are page crossings
- There are more misses in Reads because in normal program order a memory location is read from and then written to.
- M-bit Setting and Invalidations are insignificant.
- Avg CPI = 10 cycles; time to service TB miss = 22 cycles
- the instruction and Data TB can be joined with the sizes are small but when they get large then they should be split.

1.6 Was the evaluation in line with the stated user requirements?

- Yes
- The evaluation was extensive. It showed insight into almost every aspect of the Translation Buffer.
- It was nice to see benchmarks taken from actual programs from three distinct groups of VAX users.
- The writers did characterize the TB, so they met their requirements.

- They could have looked at source code to help identify the specific type of instructions the benchmarks were running.
  - This would have allowed the users to make better generalizations about why things happened the way they did.
- They talked about adding Process IDs to the cache lines but they never tried to simulate a processor using this technique.
  - This would require different infrastructure
  - This would be hard to simulate without building new hardware

1.7 Was technology a factor in the problem or solution?

- No not really

1.8 Were new tools or software techniques introduced?

- The tracing technology that allowed the writers to capture processor operations in modes other than user mode.

1.9 How may other users be affected?

- This paper was chosen because it shows the reader how to think about a topic and tear it down to the specifics. This allowed current PC architects and students to become extended users.

1.10 How can the discussion of this paper be generalized in the context of the class?

- The writers thought about users specifically and made their benchmarks resemble actual code that might be used on the machine.
- They demonstrated the difference between hardware and software.
  - Used trade-offs of HW and SW regarding evaluation methods and not to the virtual memory system.
  - To make their results more believable they used the pros of HW and SW when evaluating the performance of the Translation Buffer.
- As they showed their results they explained possible reasons for the results they were getting. All papers should strive to do this.

- This paper let its writers and readers fully understand what the VAX architects built. If an engineering group fully understands their product then they will be able to make it better that much easier.
- Shows extensive Evaluation
- Used real hardware

2 Next Time Prep

On Monday we will be reading a paper on Power and will be placed on the website later this week. On Wednesday we will be taking our quiz and reading a short paper that has not been determined yet.