

Memory Framework and Conclusions

4-16-2003

Opening Discussion

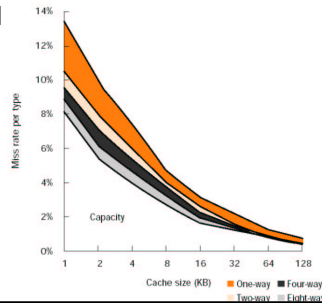
- What did we talk about last class?
- Have you seen anything interesting in the news?

Major Issues

- All of the things we have discussed with memory hierarchies have some significant similarities. Answering these four questions largely specifies the system.
 - Where can a block be placed?
 - How is a block found?
 - What block is replaced on a miss?
 - How are writes handled?

Origins of Misses

- Compulsory/cold-start misses
- Capacity misses
- Conflict/collision misses



Modern Touches

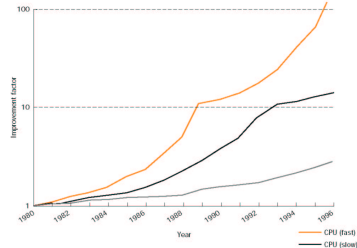
- Modern processors allow the processor to continue executing instructions while resolving a cache miss. With non-blocking caches those instructions can even access the cache. This requires out of order execution.
- The newest architectures are also adding prefetch instructions. These don't calculate anything, but bring a block of memory into cache before it is needed.

Fallacies

- Programmers can ignore details of the memory system of computers. As we have seen, poor memory performance can completely overshadow all the work we put into making a fast CPU with a pipelined datapath.
- The best thing you can do in general is to improve the locality of your code. Doing this requires knowing how your language lays out memory.

Speedy Memory?

- What we see here is that the problems are only going to get worse.



Intel and AMD Now

- If you really want to know the details of a processor, you can certainly find it. The manufacturers are more than happy to give programmers all the information they need to optimize programs for a specific architecture.
- Let's go look now at where we can find that type of documentation and see what it has to say.

Minute Essay

- Do you have any questions about the memory hierarchy material? What do you think we will have to do to prevent memory stalls from killing performance as processors get faster? We don't have class on Friday. On Monday we will be starting chapter 8.
- I should be posting assignment #7 shortly.
