

INFOMOV 2018 EXAM - November 6, 17.00 - 19.00 - BEATRIX 7th FLOOR

Answer these questions as elaborate as necessary. Don't be too elaborate; incorrect statements in your answer reduce your score. Negative scores for a question are not possible however. This exam consists of 9 questions on 1 page. Your grade is calculated as $(pts * 9. f / max_pts) + 1$.

1. Explain the following concepts in 30 words or less.

3x5 pts

- a) False sharing *When the same data sits in the caches of two cores and gets modified.*
- b) Prefetching *Getting data into a cache ahead of time.*
- c) Out-of-order execution *When the CPU reorders instructions to improve utilization of execution units.*

2. Your code contains the following snippet:

```
float a = table[20];  
a += b; a += c;  
a += d; a += e;  
a += f; a += g;  
table[20] = a;
```

The problem with this code is in the dependencies.

Sum b+c, d+e, ... separately to reduce this.

Also interesting: fetch table[20]; do the additions

in a zero-initialized variable, write back to table[20] at the last moment (decouple fetching / adding to it).

How would you modify this code to make it execute faster on a modern CPU? 10 pts

3. NVidia's Pascal GPU architecture uses a cache line width of 128 bytes.

- a) Why is it, you think, that it uses a wider cache line than a CPU? 10 pts
- b) Why, you think, is the cache line not even wider on the GPU? 10 pts

I was looking for two observations. 1: a GPU typically works on sequential data. Most of you realized this. And 2: a warp is 32 threads, 32 times an int or a float is 128 bytes. Larger types do occur (float2, float4), but a wider cache line would require either a wider bus to mem, or multiple transfers for a single line.

4. "A fully associative cache does not suffer from collisions and under-utilization and is, for a given size, the most efficient caching scheme."

- a) Is this statement true? Motivate your answer. 10 pts
- b) What is, in the context of caching, a 'collision'? 5 pts
- c) What is, in the context of caching, 'under-utilization'? 5 pts

Full points if you did not deny the lack of collisions / under-utilization in a fully associative cache, and generally correctly discussed this cache type. Collisions were often confused with evictions; however we really need to have multiple addresses fighting for the same slot for a collision.

5. List three reasons for using fixed point math on a modern CPU.

10 pts

Pick any three from: sometimes faster than float, "better for the pipes", saves on conversions, some CPUs don't have floats (dubious considering the modern CPU in the question, but ok), ...

6. "Object Oriented Programming, when used correctly, makes poor use of modern CPU architectures." Why? 10 pts

Most of you completely focused on the consequences for caching. I decided that at least two factors are needed for full points, so one of: caching / data layout problems, branch mispredictions, making data parallel processing harder.

7. 64-bit code typically runs somewhat faster than 32-bit code.

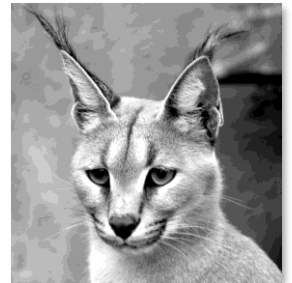
- a) Why? *Moar registers.* 10 pts
b) Describe a situation where a 64-bit build would be slower. 10 pts

That would be when we read / write tons of pointers. E.g. tank pointers in a grid. :)

8. One difference between debug and release mode compilation is that in debug builds, the compiler does not reorder instructions.

- a) Why does the compiler reorder instructions in release builds? 10 pts
b) Name one other important difference between debug and release compilation. 5 pts

Combined with 1c this was a poor question. Essentially 15 free points for everyone unless you really messed up. For completeness: reordering happens to improve execution unit utilization. Note that the CPU also does this, but for much shorter distances. Other differences between debug and release: variables are initialized, dead code is eliminated, ...



9. Identify the creature in the picture. 0 pts

This really got you confused. No, this is not a Hedgehog, nor an elephant. It's a Caracal. So go fill out the Enquete (which it also isn't).

May the Light be with you.

Thanks for participating, I really enjoyed teaching this year's class!

