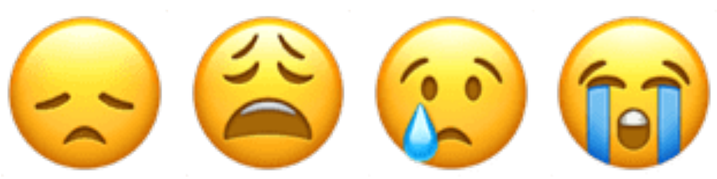
Lecture 22:

Course Wrap Up



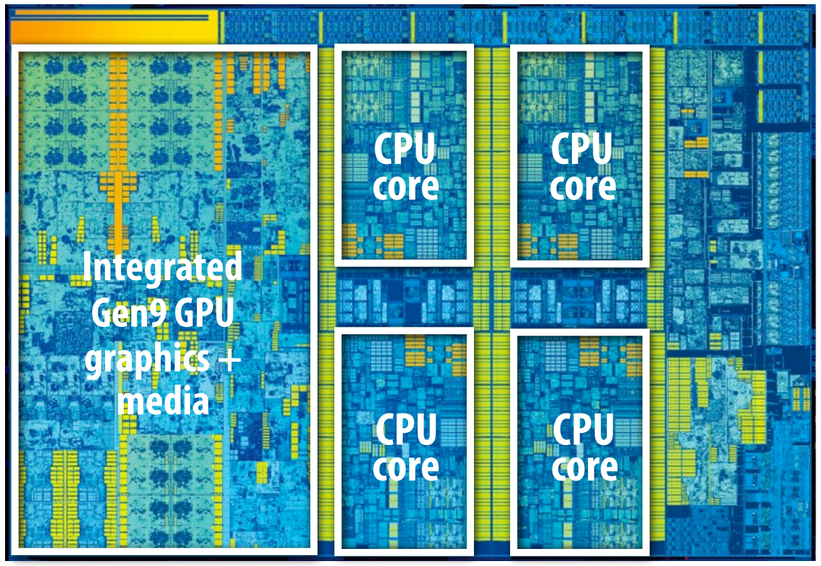
(Professor)

(Students)

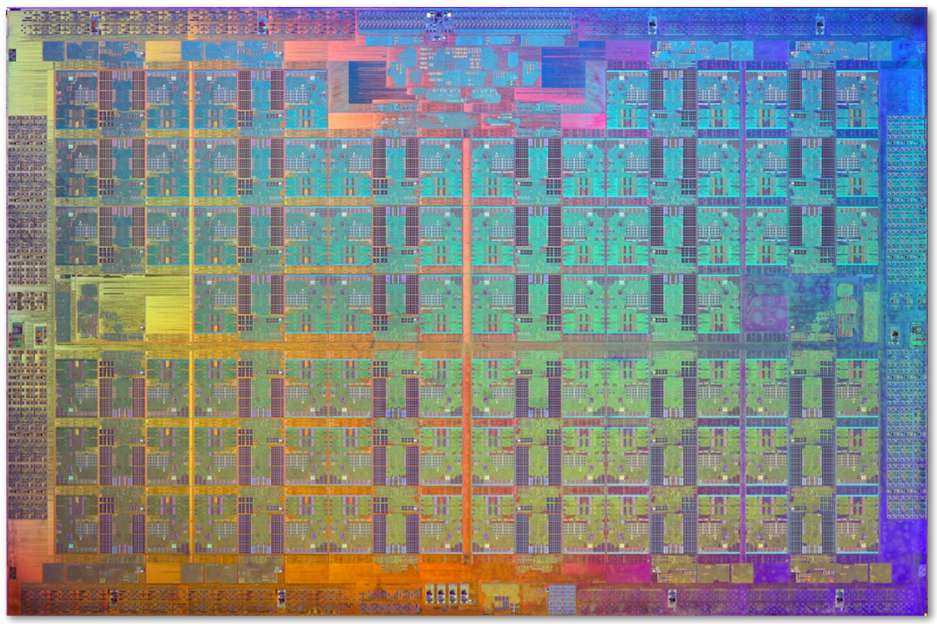
Parallel Computer Architecture and Programming

CMU / 清华大学, Summer 2017

For the foreseeable future, the primary way to obtain higher performance computing hardware is through a combination of increased parallelism and hardware specialization.



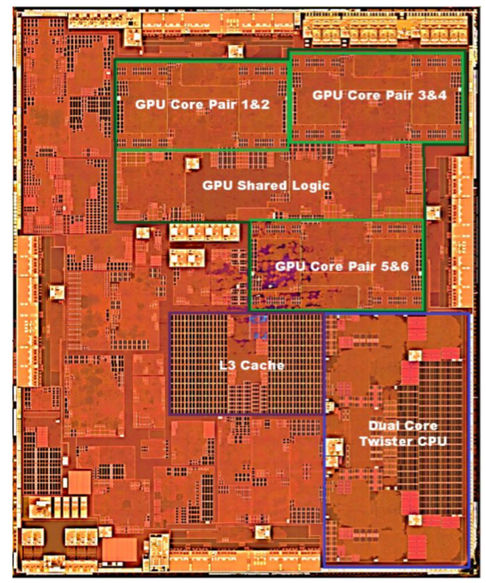
Intel Core i7 CPU + integrated GPU and media



Intel Xeon Phi
72 cores, 16-wide SIMD, 4-way multi-threading

NVIDIA Maxwell GPU
(single SMM core)
32 wide SIMD
2048 CUDA/core threads per SMM

FPGA
(reconfigurable logic)



Apple A9
Heterogeneous SoC
multi-core CPU + multi-core GPU + media ASICs

Today's software is surprisingly inefficient compared to the capability of modern machines

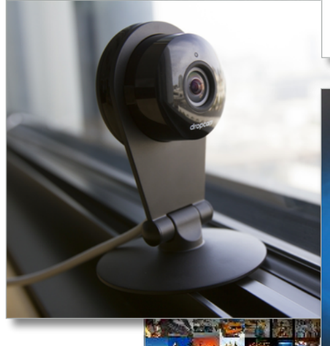
A lot of performance is currently left on the table (increasingly so as machines get more complex, and parallel processing capability grows)

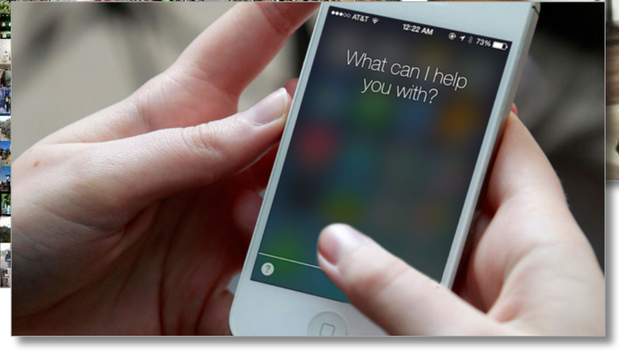
Extracting this performance stands to provide a notable impact on many compute-intensive fields (or, more importantly enable new applications of computing!)

Given current software programming systems and tools, understanding how a parallel machine works is important to achieving high performance.

A major challenge going forward is making it simpler for programmers to extract performance on these complex machines.

This is very important given how exciting (and efficiency-critical) the next generation of computing applications are likely to be.





Key issues we have addressed in this course

Identifying parallelism

(or conversely, identifying dependencies)

Efficiently scheduling parallelism

1. Achieving good workload balance

2. Overcoming communication constraints:

Bandwidth limits, dealing with latency, synchronization

Exploiting data/computation locality = efficiently managing state!

3. Scheduling under heterogeneity (using the right processor for the job)

We discussed these issues at many scales and in many contexts

Heterogeneous mobile SoC

Single chip, multi-core CPU

Multi-core GPU

CPU+GPU connected via bus

Clusters of machines

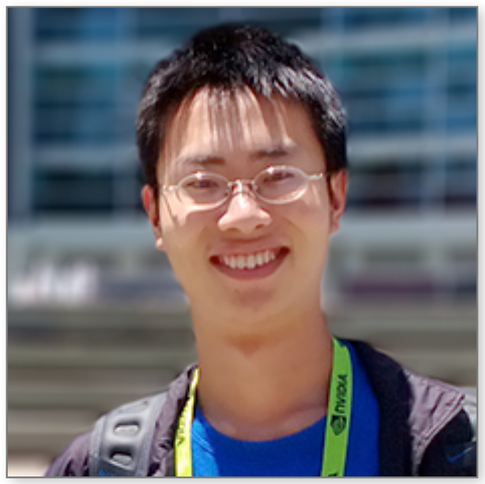
Large scale, multi-node supercomputers

Other classes with overlapping topics

- Distributed computing
- Computer architecture
- Operating systems
- Compilers
- Database systems

- Application areas that benefit from parallel computing ideas: scientific computing, machine learning, computer graphics, computer vision, etc…

Thank you to the course TAs!



Yong He
Ph.D. student
(CMU)

Xu Ji
Ph.D. student
(Tsinghua)

Yan Gu
Ph.D. student
(CMU)

Ping Xu
M.S. student
(Tsinghua)

SOME TIPS

After taking this course, you are ready to try undergraduate research in parallel computing!

Why research (or independent study)?

- You will learn way more about a topic than in any class.

- You think your undergrad friends are very smart? Come hang out with Ph.D. students! (you get to work side-by-side with them and with faculty). Imagine what level you might rise to.

- It's way more fun to be on the cutting edge. Industry might not even know about what you are working on. (imagine how much more valuable you are if you can teach them)

- It widens your mind as to what is possible.

There are many opportunities at Tsinghua

- For example, Prof. Wei Xue's group
  - Research focus: efficient parallel algorithms and applications for Taihu Light
  - Designing performance models to predict application performance
  - Group won 2016 Gordon Bell Prize for weather simulation on Taihu Light. (Your TA Ping Xu is one of the winners!)
- Do not be shy to talk to Ping Xu, Xu Ji, or Prof. Xue

What my own Ph.D. students are working on these days…

- Generating efficient code from image processing or deep learning DSLs (Halide Autoscheduler)
- Designing a new shading language for future real-time 3D graphics pipelines (collaboration with NVIDIA)
- Creating a parallel computing platform makes it simpler and more efficient to analyzing large video collections on large clusters (Scanner project: "Spark for video")
- Designing more efficient DNNs to accelerate image processing
- Computer graphics tools for animation and theatrical lighting design

Maybe you might like research and decide you want to go to grad school

Without question, the number one way to get into a top grad school is to receive a strong letter of recommendation from faculty members from Tsinghua and from other schools. You get that letter only from being part of a research team for an extended period of time.

DWIC letter: (“did well in class” letter) What you get when you ask for a letter from a faculty member who you didn’t do research with, but got an ‘A‘ in their class. This letter is essentially thrown out by the Ph.D. admissions committee at good schools.

A very good reference

CMU Professor Mor Harchol-Balter's writeup:

"Applying to Ph.D. Programs in Computer Science"

http://www.cs.cmu.edu/~harchol/gradschooltalk.pdf

Research is just one option…

(Despite what many of us biased faculty tell you,
there are other good ones too)

Why not start your own project?

Interested in applying computer science to a problem that excites you? Give it a shot!

I'm sure there are plenty of independent study opportunities at Tsinghua.

Like a topic enough to be your own boss? Consider starting your own company.

Why go work for Baidu or Alibaba when you can start a company that beats them?

Your professors encourage you to be brave and take risks.

You are lucky because you are extremely talented. The cost of "messing up" for you is actually much less than for other students because your backup plan is very good.

Be ambitious while at Tsinghua with opportunities beyond just classes. If it doesn't work out, you'll try something else and you'll probably succeed... or end up getting the good job you would have gotten anyway.

I've very much enjoyed my time at Tsinghua.

Good luck to you!

And thanks for being a great class!

Assignment 3 is handed in,
so now go relax and eat some…