Welcome!

Computer Systems' Seminiar

DArchR
DAVIS ARCHITECTURE RESEARCH

UCDAVIS
COMPUTER SCIENCE

Purpose & Plan

Get people interested in Computer Systems Research together

- Develop collaborations?
- Start working on grants?
- Provide some social connections!

Weekly (when we're not superseded)

Presenters (you all!)

- Students describing their work
- Practice for conference talks
- Cool ideas you're working on
- <Your ideas here>

Quarter plan

Most important: Thanks Zhantong!

Computer Systems Seminar - UC Davis Computer Architecture

10/17: No meeting

(Yinzhan Xu, “Shaving Logs Via Large Sieve Inequality: Foster Algorithms for Sparse Convolution and More”)

10/24: ???

10/31: Prof Tapti Palit?

Reach out to Zhantong if you want to present!

Today’s agenda

Let’s hear about some of the cool stuff going on at UCD!

- Prof. Tapti Palit
- Prof. Amanda Raybuck
- Prof. Caleb Stanford
- Prof. Dipak Ghosal
- Prof. Jason Lowe-Power

Discuss…

Take a picture!

Goals

- Learn a little about each others’ research
- Get ideas for masters/undergrad projects

Delving into the Darker Side

A look into the computer systems research of the DArchR group

arch.cs.ucdavis.edu

Desolation Wilderness

REVIEW

COMPUTER SCIENCE

There's plenty of room at the Top: What will drive computer performance after Moore's law?

Charles E. Leiserson[1], Neil C. Thompson[1,2]*, Joel S. Emer[1,3], Bradley C. Kuszmaul[1]†, Butler W. Lampson[1,4], Daniel Sanchez[1], Tao B. Schardl[1]

Table 1. Speedups from performance engineering a program that multiplies two 4096-by-4096 matrices. Each version represents a successive refinement of the original Python code. "Running time" is the running time of the version. "GFLOPS" is the billions of 64-bit floating-point operations per second that the version executes. "Absolute speedup" is time relative to Python, and "relative speedup," which we show with an additional digit of precision, is time relative to the preceding line. "Fraction of peak" is GFLOPS relative to the computer's peak 835 GFLOPS. See Methods for more details.

Version	Implementation	Running time (s)	GFLOPS	Absolute speedup	Relative speedup	Fraction of peak (%)
1	Python	25,552.48	0.005	1	—	0.00
2	Java	2,372.68	0.058	11	10.8	0.01
3	C	542.67	0.253	47	4.4	0.03
4	Parallel loops	69.80	1.969	366	7.8	0.24
5	Parallel divide and conquer	3.80	36.180	6,727	18.4	4.33
6	plus vectorization	1.10	124.914	23,224	3.5	14.96
7	plus AVX intrinsics	0.41	337.812	62,806	2.7	40.45

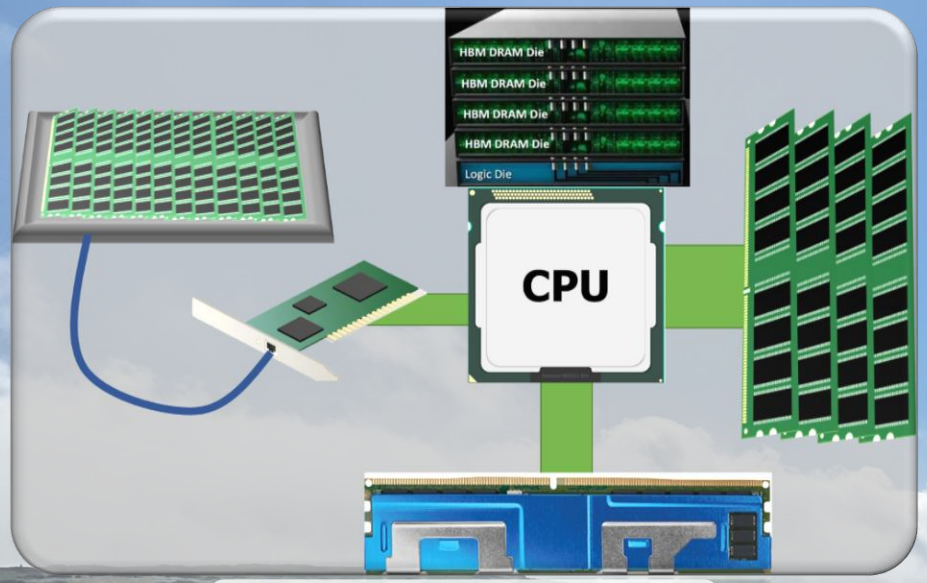
Tahoe National Forest

What we do in DArchR

Predict problems brought on by technology 5-10 years in the future

Develop *new* hardware and software mechanisms to solve these problem

Yosemite National Park



Data movement

Data security

gem5

open source initiative®

Building tools

Tomales Bay State Park

Memory Management Projects

If multiple **hosts** share a single memory (CXL)

How to model this in gem5?

How to provide coherence between host caches?

What granularity to share?

How to manage the interconnect?

How to modify workloads?

Remote memory controller optimizations?

How to ensure reliability?

Data Security

Idea: What if we separate isolation/protection from OS

- Don’t use page tables, processes, etc.

Similar to “confidential computing,” but more focus on data

Projects:

- Design OS extensions
- Design new hardware mechanisms
- Integrate with FPGAs, sensors, distributed systems, etc.

Eldorado National Forest

Open-source software
Used by 1000s
By academia and industry

gem5 Projects

Verifying gem5 RISC-V implementation

New methodology for area/power estimation for gem5

Support for sv32, sv48, sv57 for RISC-V

Absolute & Relative Accuracy of gem5

Testing DRAM caches with GPUs

Comparing accuracy of simulation methodologies

Developing accurate models for specific hardware

…. Many others

DArchR

DAVIS ARCHITECTURE RESEARCH

Pinnacles National Park