FireSim and Chipyard Tutorial: Intro

Sagar Karandikar
UC Berkeley
sagark@eecs.berkeley.edu

1. Fill out the form at **[REMOVED FOR PDF]** now for EC2 instance access

Then

2. You’ll receive two emails. Follow insts to login, then wait.

Berkeley Architecture Research

FireSim

CHIPYARD

FireSim and Chipyard Tutorial: Intro

Sagar Karandikar

UC Berkeley

sagark@eecs.berkeley.edu

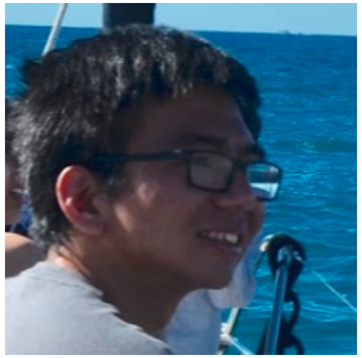
Berkeley
Architecture
Research

FireSim

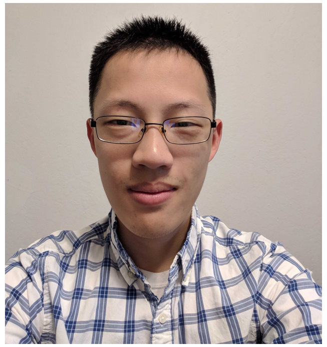
CHIPYARD

Presenters/Organizers

Sagar
Karandikar



Jerry
Zhao



Howard
Mao

Abraham
Gonzalez

John
Wright

David
Biancolin

Nathan
Pemberton

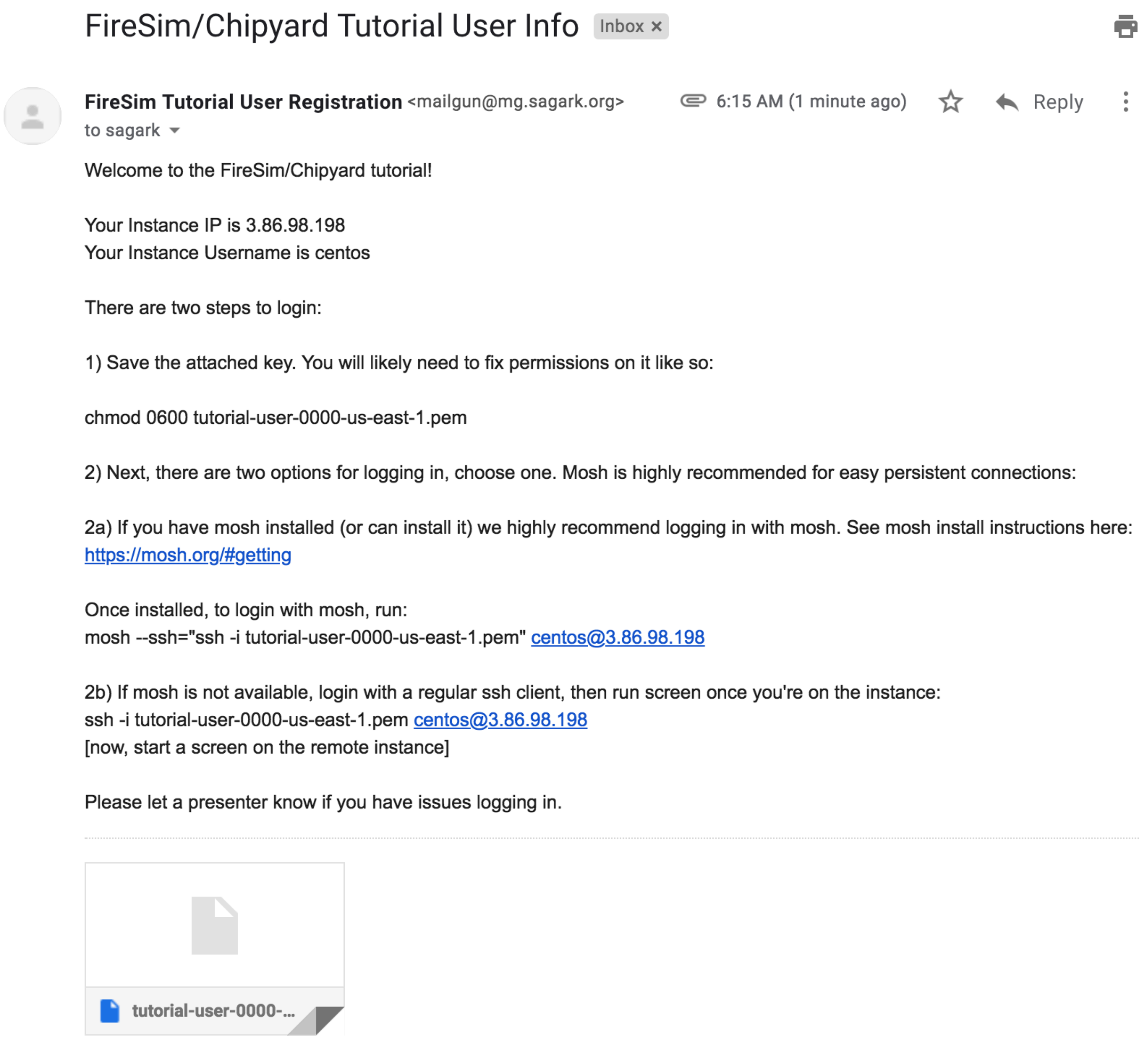
Albert
Ou

Alon
Amid

Krste
Asanović

Getting Started/Logistics (recap)

- Fill out the form at [REMOVED FROM PDF] now for EC2 instance access
- You'll receive two emails. One from Google Forms and one that looks like →
- Follow the instructions in this one to login to your FireSim manager instance, then wait



FireSim/Chipyard Tutorial User Info

FireSim Tutorial User Registration <mailgun@mg.sagark.org>

to sagark

Welcome to the FireSim/Chipyard tutorial!

Your Instance IP is 3.86.98.198
Your Instance Username is centos

There are two steps to login:

1) Save the attached key. You will likely need to fix permissions on it like so:

```
chmod 0600 tutorial-user-0000-us-east-1.pem
```

2) Next, there are two options for logging in, choose one. Mosh is highly recommended for easy persistent connections:

2a) If you have mosh installed (or can install it) we highly recommend logging in with mosh. See mosh install instructions here: https://mosh.org/#getting

Once installed, to login with mosh, run:

```
mosh --ssh="ssh -i tutorial-user-0000-us-east-1.pem" centos@3.86.98.198
```

2b) If mosh is not available, login with a regular ssh client, then run screen once you're on the instance:

```
ssh -i tutorial-user-0000-us-east-1.pem centos@3.86.98.198
```

[now, start a screen on the remote instance]

Please let a presenter know if you have issues logging in.

A *Golden Age* in Computer Architecture

- No more traditional scaling…
- An architect's dream: everyone wants custom microarchitectures and HW/SW co-designed systems
- Also, a golden age to have *direct impact* as researchers
  - Exploding open-source hardware environment
  - An open-ISA that can run software we care about

https://cacm.acm.org/magazines/2019/2/234352-a-new-golden-age-for-computer-architecture/fulltext

A *Dark Age* in Computer Architecture tools

- What do we need to do good architecture research?
  - Need tools that let us evaluate designs on a variety of metrics:
    - Functionality
    - Performance
    - Power
    - Area
    - Frequency
  - Especially in small teams (grad students, startups), these tools need to be *agile*
  - Historically, without good open IP, had to build abstract arch/uarch simulators out of necessity
    - But now, we have much better IP and software compatibility, so what's stopping us?

A *Dark Age* in Computer Architecture tools

- Designed to be operated by hundreds of engineers
- Not, 10s of engineers or 1s-10s of grad students
- Two hard questions:
  - Where do I get a collection of well-tested hardware IP + complex software stacks that run on it?
  - How do I quickly obtain performance measurements for a novel HW/SW system?

Two hard questions, answered!

- Where do I get a collection of well-tested hardware IP + complex software stacks that run on it?

CHIPYARD

- How do I quickly obtain performance measurements for a novel HW/SW system?

FireSim

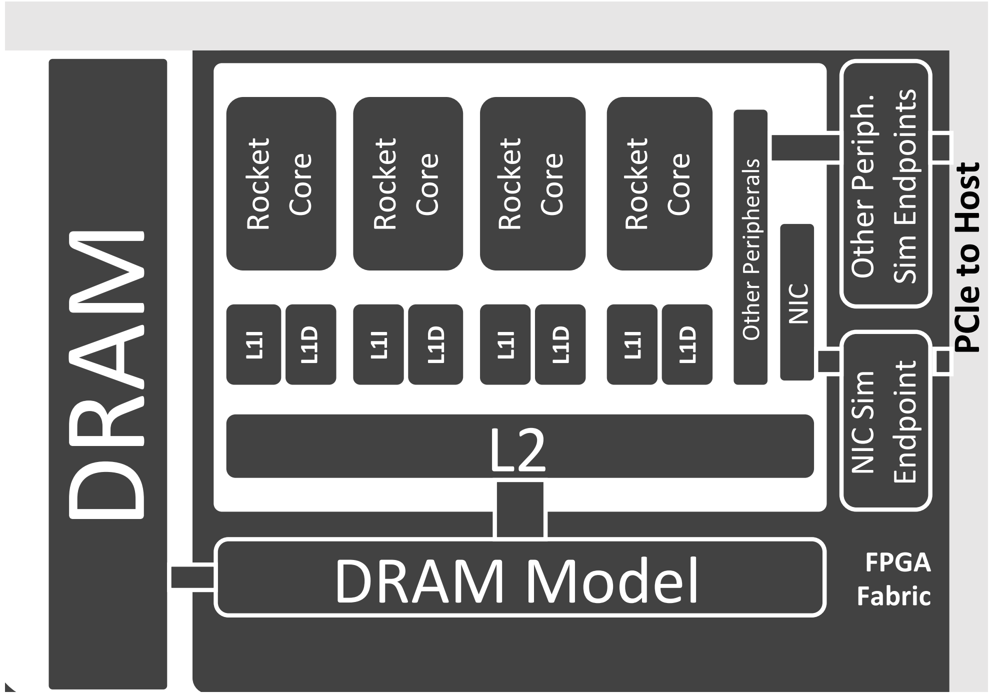
What can I do with these tools?

CHIPYARD + FireSim

Measure Functionality, Performance, Power, Area, Frequency *for real HW/SW systems*, quickly and easily, with small teams of engineers

What kinds of designs can I work with?

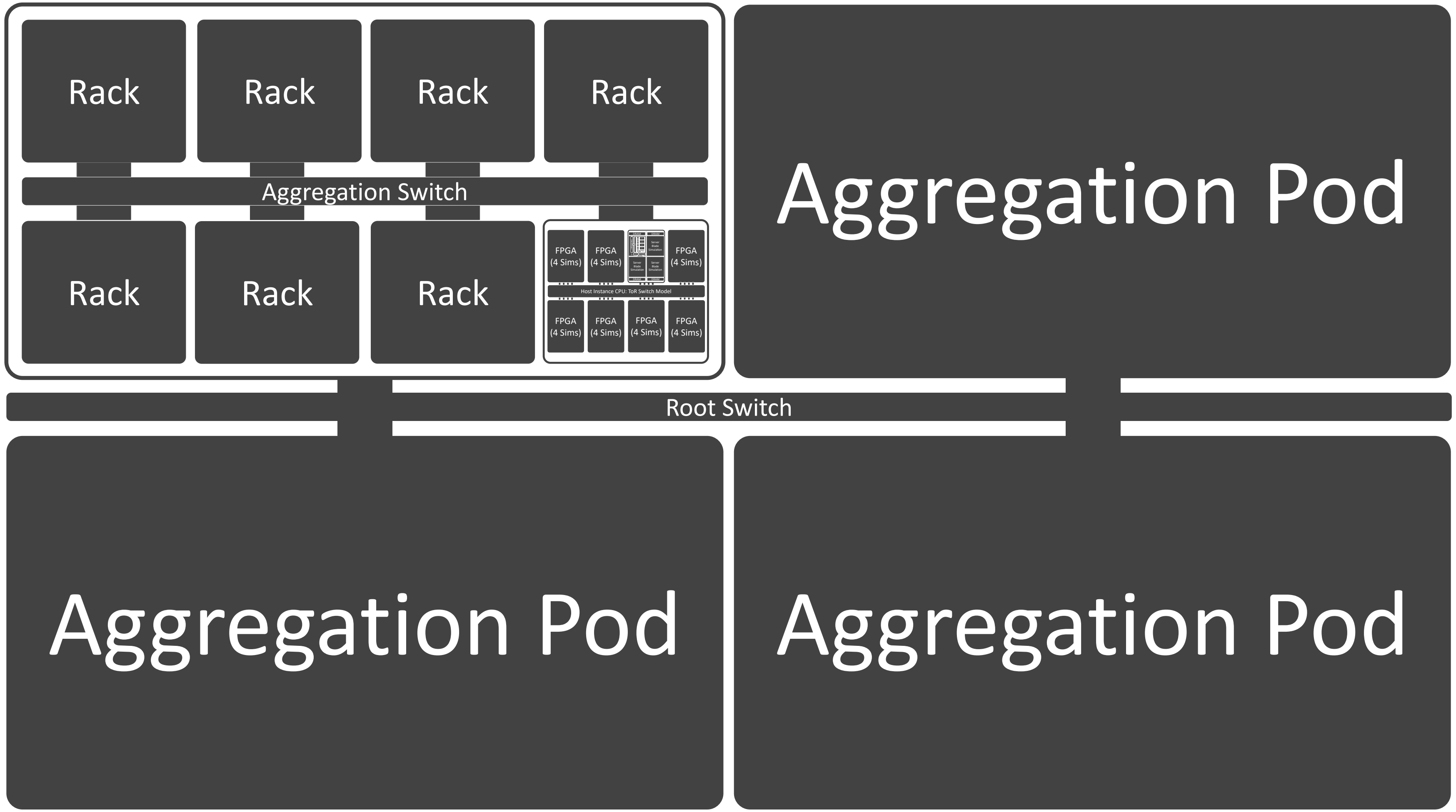
- RISC-V Cores:
  - Rocket Chip In-Order core, industry proven
  - BOOM Out-of-Order Superscalar core
- Accelerators
  - Hwacha Vector Accelerator
  - sha3 accelerator
  - NVDLA (NVIDIA Deep Learning Accelerator)
  - ML Accelerators (Berkeley Systolic Array, coming soon)
- Peripherals/other IP
  - L2 Cache, UART, Disk, Ethernet NIC, etc.
- FPGA-Simulation Models
  - Large LLCs, large DDR3 memory systems



Single SoC System

Berkeley Architecture Research

What kinds of designs can I work with?



Ethernet-Networked 1024 SoC System

Growing FireSim Community!

- Companies publicly announced using FireSim
  - Esperanto Maxion ET
  - Intensivate IntenCore
- Projects with public FireSim support
  - Rocket Chip, BOOM
  - Hwacha Vector Accelerator [11]
  - Keystone Secure Enclave [12]
    - https://github.com/keystone-enclave/keystone-firesim
  - NVIDIA Deep Learning Accelerator (NVDLA) [9]
    - https://github.com/nvdla/firesim-nvdla
    - https://devblogs.nvidia.com/nvdla/
  - BOOM Spectre replication/mitigation [10]
  - More in-progress! PR yours!

- First academic users
  - ISCA ‘18: Maas et. al. HW-GC Accelerator (Berkeley)
  - MICRO ‘18: Zhang et. al. “Composable Building Blocks to Open up Processor Design” (MIT)
  - Latest list @ https://fires.im/publications/#userpapers
- CCC/RV Summit tutorials
  - > 200 attendees
- Used in Berkeley’s CS152/252 Sp. 19
- More than 80 mailing list members
- More than 130 unique cloners per week

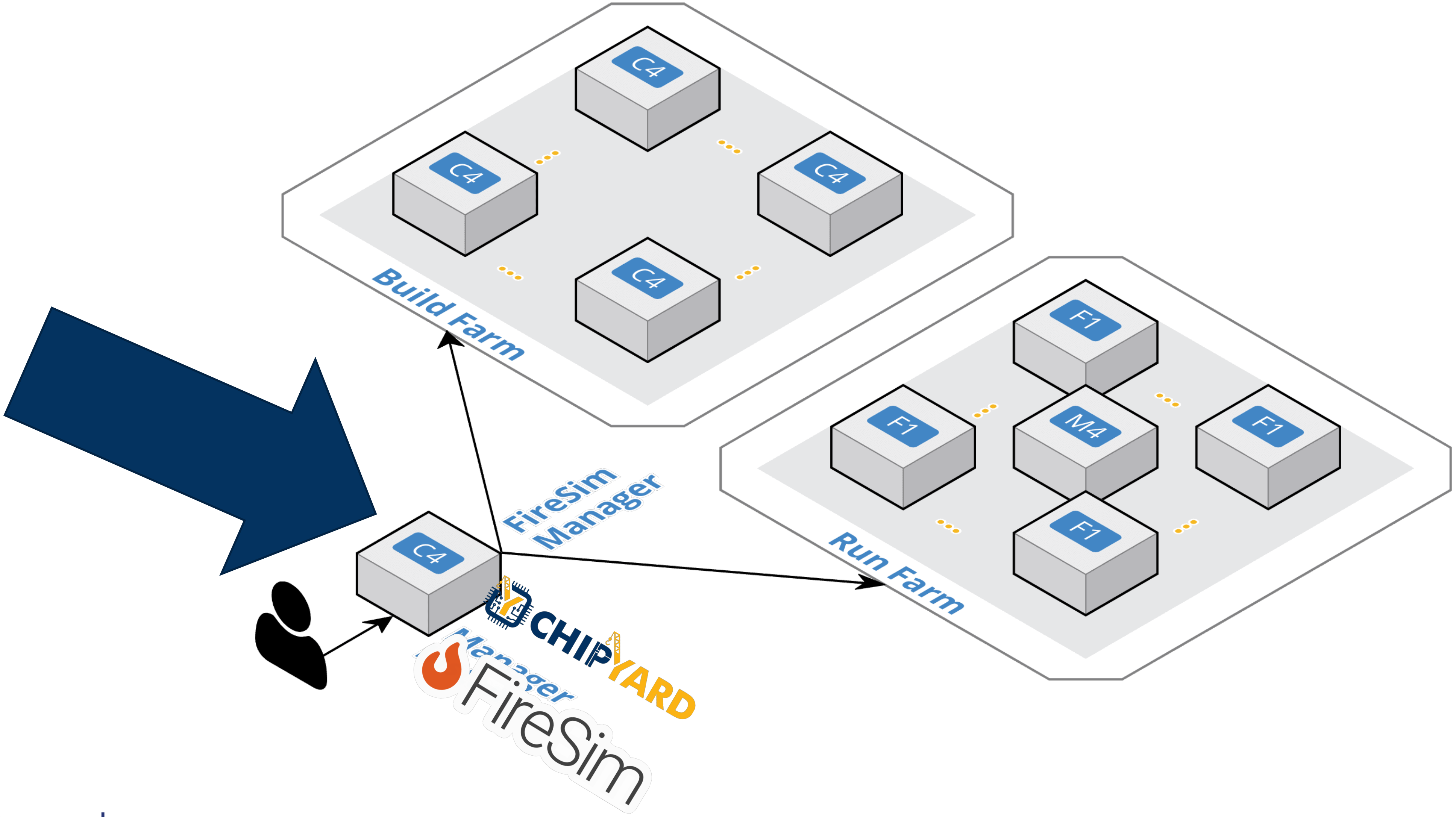
*FireSim ISCA’18 paper selected as an **IEEE Micro Top Pick of 2018 Arch. Confs** and as **the CACM Research Highlights Nominee from ISCA’18***

Chipyard 1.0 Released Yesterday!

https://github.com/ucb-bar/chipyard

Today's Logistics

You are here



Running a FireSim FPGA Build

- This will take a while, so we will run this in the background:

```
tmux new -s fpgabuild  # this will give you a persistent
                       # session you can reattach to
firesim managerinit
```

[When prompted, enter your email address to get a build completion notification]

```
# runs the HW build, all the way to AGFI
firesim buildafi
```

[Lastly, detach from tmux with “ctrl-b d”. We will return to this build later.]

[this will build a design called firesim-singlecore-no-nic-l2-lbp]

Today’s Agenda - AM

08:30 – 08:50: Introduction/Overview, Amazon EC2 Instance Setup, Logistics - Sagar

08:50 – 10:00: Chipyard Basics – Jerry, Howie

10:00 – 10:30: Building Custom RISC-V SoCs in Chipyard - Abe

10:30 – 11:00: Coffee break

11:00 – 11:20: Building Custom RISC-V SoCs in Chipyard (continued) - Abe

11:20 – 11:50: Hammer VLSI flow - John

11:50 – 12:00: Afternoon FireSim Session Preview - Sagar

Today’s Agenda - PM

12:00 - 13:00: Lunch

13:00 - 13:30: FireSim Introduction - Sagar

13:30 – 14:00: Building Hardware Designs in FireSim - David

14:00 – 14:30: Building Software Workloads in FireSim - Nathan

14:30 – 15:00: Running a FireSim Simulation: Password Cracking on a RISC-V SoC with SHA-3 Accelerators and Linux - Albert

15:00 – 15:30: Coffee break

15:30 – 16:15: Instrumenting and Debugging FireSim-Simulated Designs - Alon

16:15 – 16:55: FireSim Multi-FPGA Networked Simulation - Alon

16:55 – 17:00: Conclusion - Alon

Thanks to AWS, Xilinx, and ADEPT Sponsors

aws

XILINX®

ADEPT Lab Sponsors:

intel®

FUTUREWEI

Google

SEAGATE

SIEMENS

SK hynix