Serverless? RISC more!

TCSS562
Group Members: Aaron Chen ,Mingzhi Ma, Derry Cheng

BE BOUNDLESS

1

Outline

- Introduction
- Related work
- Overview of approaches
- Key contributions
- Experiment results
- Critiques
- Q&A

UNIVERSITY of WASHINGTON

2

Introduction: Paper overview

Challenges in Serverless CPU Design:

- Complexity of cloud stacks and hardware simulation
- Limitations of modern server CPUs.

UNIVERSITY of WASHINGTON

3

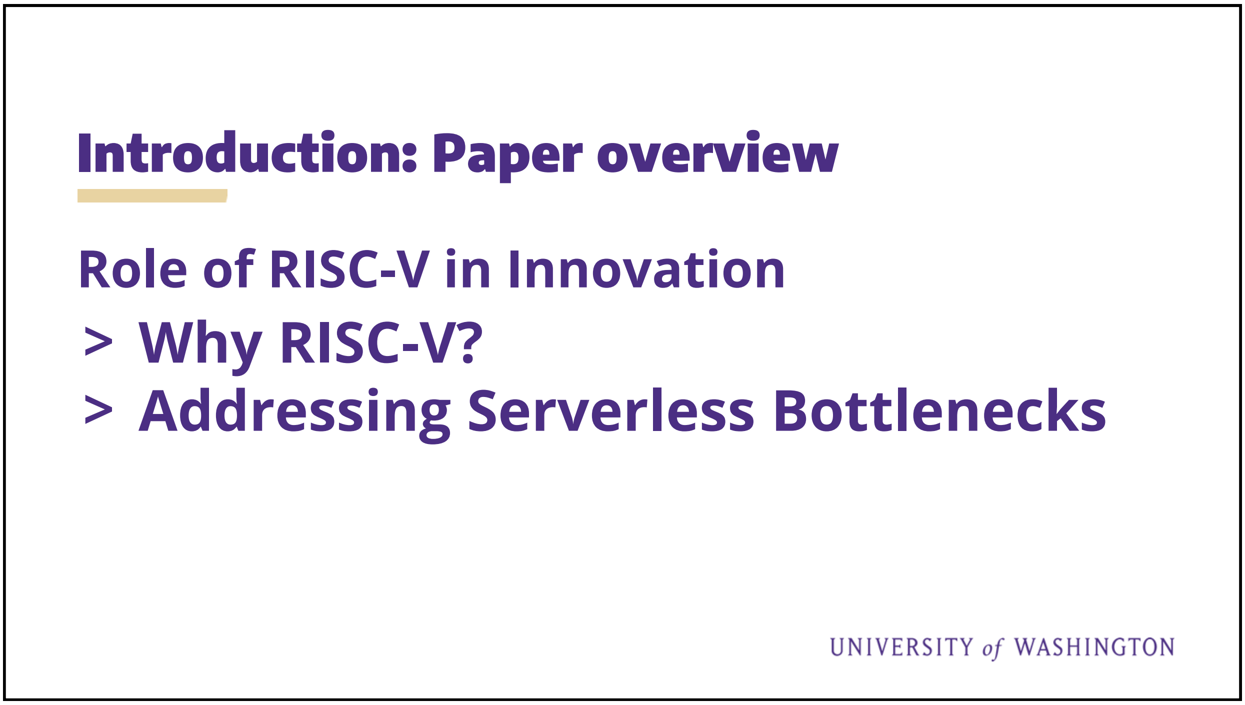
Introduction: Paper overview

Need for Co-Design and New Architectures

- Co-Design Tools for Realistic Workloads
- Shift to Manycore Architectures

UNIVERSITY of WASHINGTON

4



Introduction: Paper overview

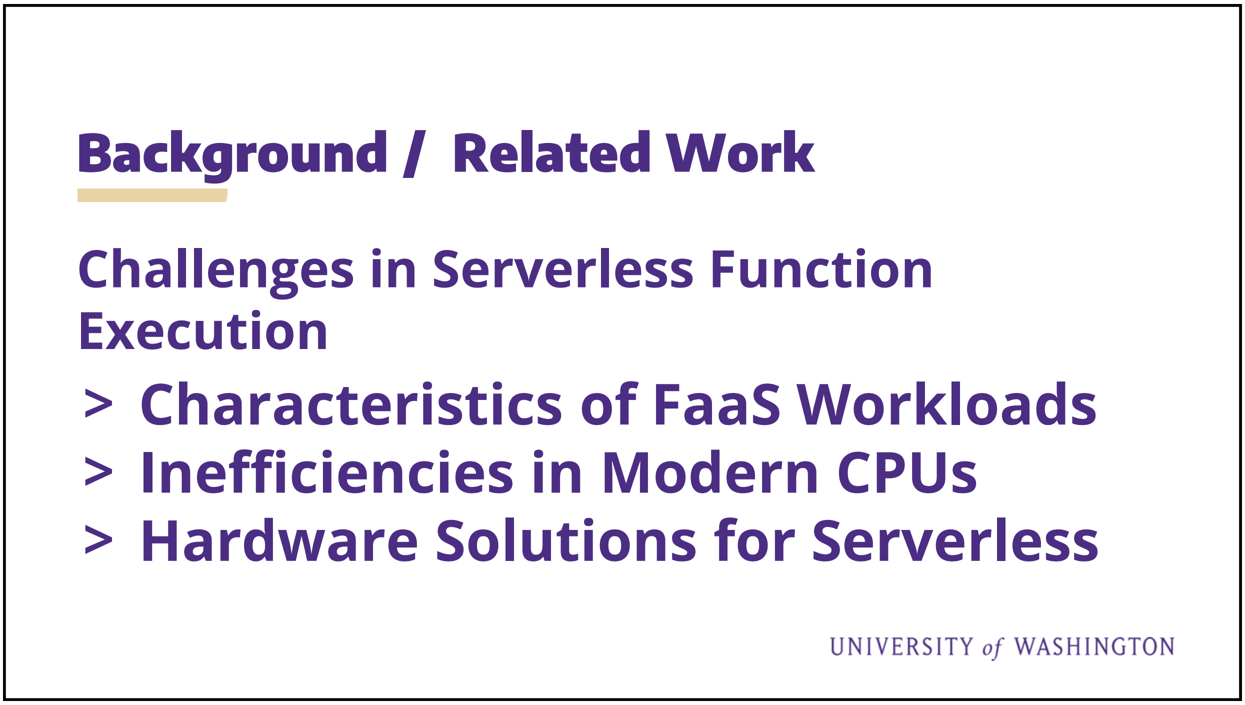
Role of RISC-V in Innovation

> Why RISC-V?

> Addressing Serverless Bottlenecks

UNIVERSITY of WASHINGTON

5



Background / Related Work

Challenges in Serverless Function Execution

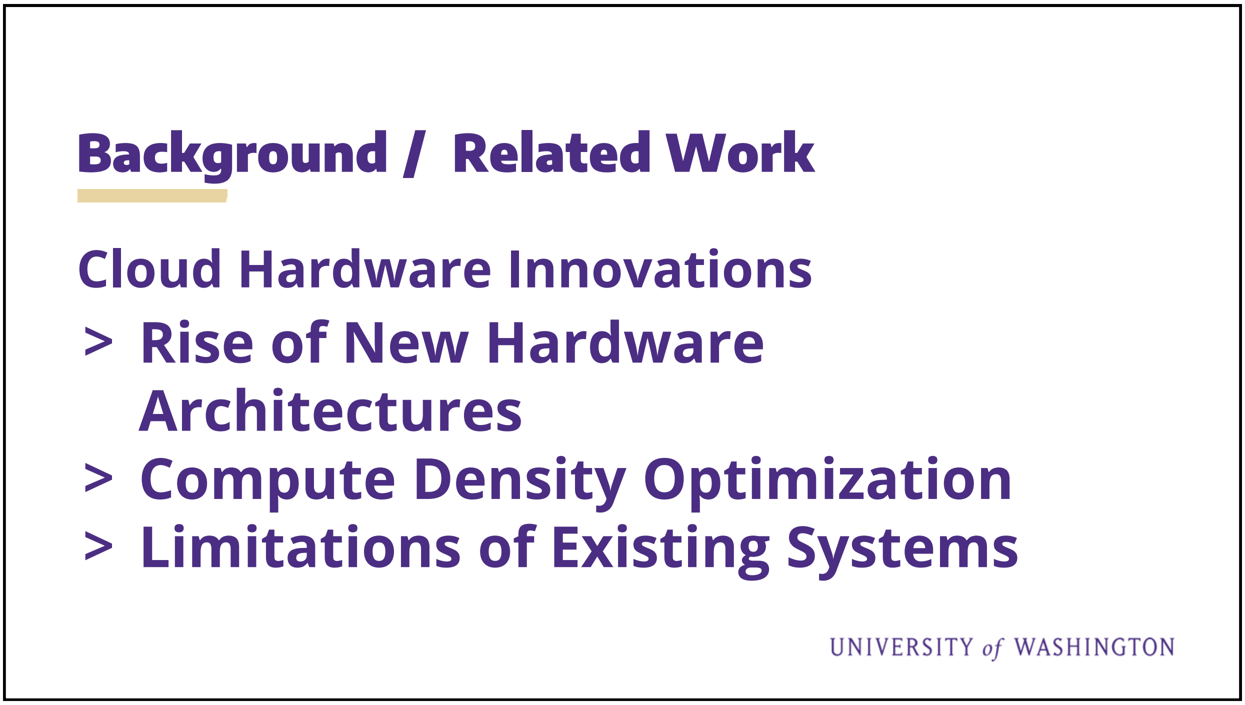
> Characteristics of FaaS Workloads

> Inefficiencies in Modern CPUs

> Hardware Solutions for Serverless

UNIVERSITY of WASHINGTON

6



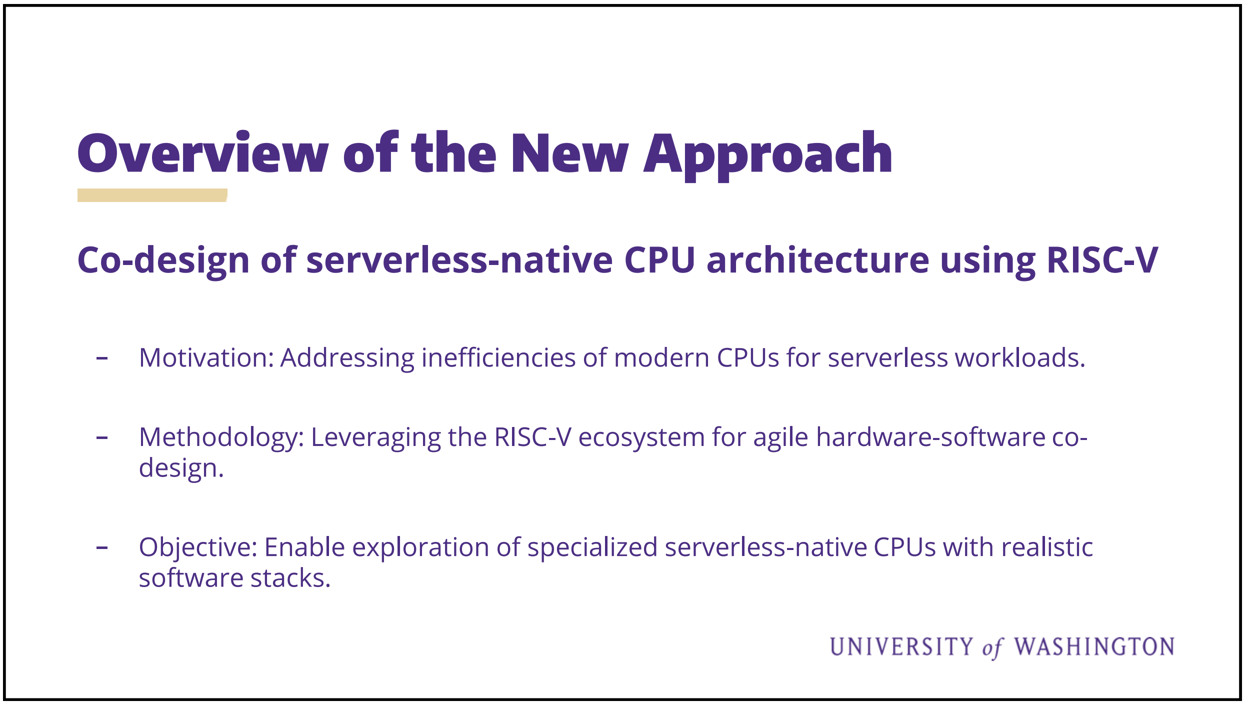
Background / Related Work

Cloud Hardware Innovations

- > Rise of New Hardware Architectures
- > Compute Density Optimization
- > Limitations of Existing Systems

UNIVERSITY of WASHINGTON

7



Overview of the New Approach

Co-design of serverless-native CPU architecture using RISC-V

- Motivation: Addressing inefficiencies of modern CPUs for serverless workloads.
- Methodology: Leveraging the RISC-V ecosystem for agile hardware-software co-design.
- Objective: Enable exploration of specialized serverless-native CPUs with realistic software stacks.

UNIVERSITY of WASHINGTON

8

Proposed Methodology

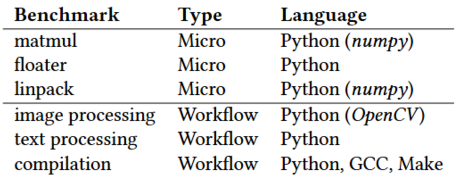
RISC-V Platform Advantages:

- Open-source and parameterizable architecture.
- Supports full-system stacks for accurate workload simulations.

Benchmark Approach:

Combination of microbenchmarks (e.g., matrix operations) and workflows (e.g., image and text processing).

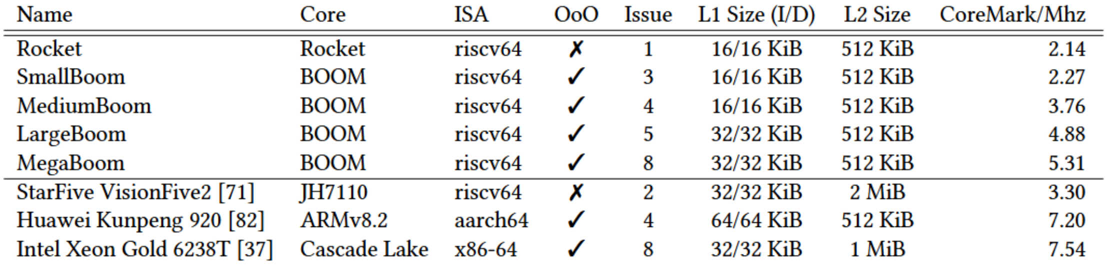
Table 2: Benchmarks Evaluated



Benchmark	Type	Language
matmul	Micro	Python (*numpy*)
floater	Micro	Python
linpack	Micro	Python (*numpy*)
image processing	Workflow	Python (*OpenCV*)
text processing	Workflow	Python
compilation	Workflow	Python, GCC, Make

RISC-V Configurations and Comparative Platforms

Table 1: Per-Core Configurations



Name	Core	ISA	OoO	Issue	L1 Size (I/D)	L2 Size	CoreMark/Mhz
Rocket	Rocket	riscv64	✗	1	16/16 KiB	512 KiB	2.14
SmallBoom	BOOM	riscv64	✓	3	16/16 KiB	512 KiB	2.27
MediumBoom	BOOM	riscv64	✓	4	16/16 KiB	512 KiB	3.76
LargeBoom	BOOM	riscv64	✓	5	32/32 KiB	512 KiB	4.88
MegaBoom	BOOM	riscv64	✓	8	32/32 KiB	512 KiB	5.31
StarFive VisionFive2 [71]	JH7110	riscv64	✗	2	32/32 KiB	2 MiB	3.30
Huawei Kunpeng 920 [82]	ARMv8.2	aarch64	✓	4	64/64 KiB	512 KiB	7.20
Intel Xeon Gold 6238T [37]	Cascade Lake	x86-64	✓	8	32/32 KiB	1 MiB	7.54

UNIVERSITY *of* WASHINGTON

Key contributions

1. First study leveraging RISC-V for serverless workload exploration with realistic system software.
2. Demonstrates the potential of RISC-V cores in achieving comparable instructions-per-cycle (IPC) to ARM.
3. Opens new avenues for evaluating microarchitectural features tailored for cloud-native workloads.

What's New?

- Comprehensive exploration of specialized CPU designs for serverless scenarios.
- Identification of bottlenecks in current CPU designs for short-lived, bursty workloads.

UNIVERSITY *of* WASHINGTON

Baseline Methods

Comparison Platforms: RISC-V vs. VisionFive2 vs. Huawei Kunpeng vs. Intel Xeon.

Metrics Evaluated: instruction count, IPC (instruction per cycle)

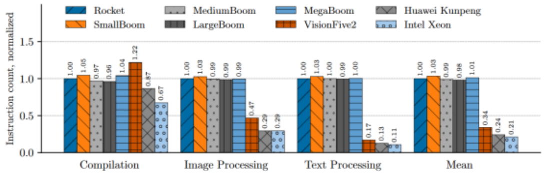


Figure 3: FaaS pipeline relative (to Rocket) instructions. The instruction count difference is most pronounced for the text processing pipeline. This workload consists mostly of compression and encryption operations, which benefit from highly specialized instructions present in the ARM and x86 ISAs. Note the mean presented is the harmonic mean.

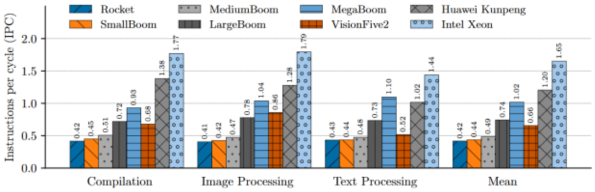


Figure 4: We compare IPC across all of the experimental platforms for each of the FaaS pipelines described in Section 4.1. The bar height indicates the IPC for each evaluated platform. As BOOM core issue width grows, the IPC grows as well, approaching in some cases the commercial processors. Note the presented mean is the harmonic mean.

UNIVERSITY *of* WASHINGTON

Key Results

- MegaBoom core achieves competitive IPC relative to ARM Kunpeng.
- RISC-V designs show promise for modular, efficient serverless processing.
- Extensions like RV64GCB significantly improve workload performance.

Author's Conclusions

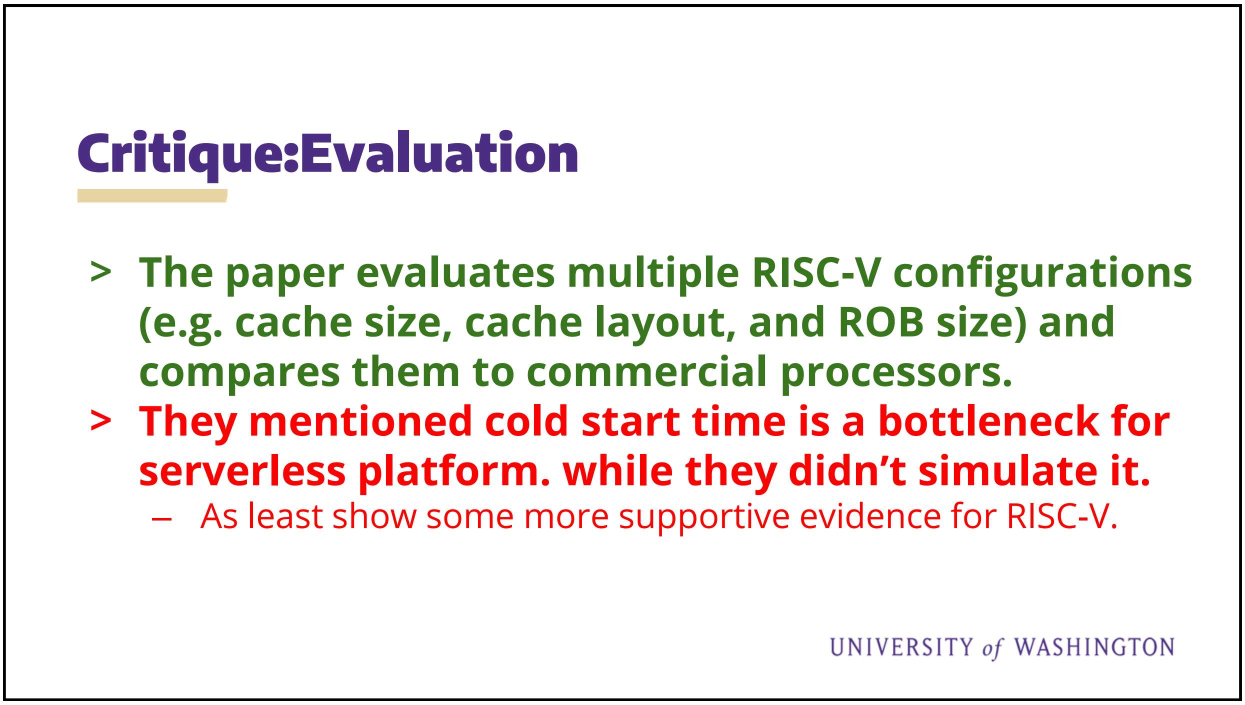
1. **Leveraging the RISC-V Ecosystem**
2. **Comprehensive Performance Evaluation**
3. **Identified Strengths and Weaknesses of RISC-V**

Critique:Strengths

- **Convincing Simulation:**
  - They applied FPGA to simulate real RISC-V hardware routing.
  - They loaded Linux to DDR memory to better emulate actual workload situations.
- **Diverse hardware configurations:**
  - Benchmarking results from the settings could help successors to design hardware and software specifications.
- **Decentralization:**
  - Based on RISC-V, industries can develop their own chips.

Critique:Weakness

- An open-source could be **unstable** and industries might not upload their new architectures.
  - **Stability is stepstone for cloud computing**, this paper should add some more statements to support it.
- The paper says “**RISC more!**” but in the end they haven’t surpassed mainstream products.

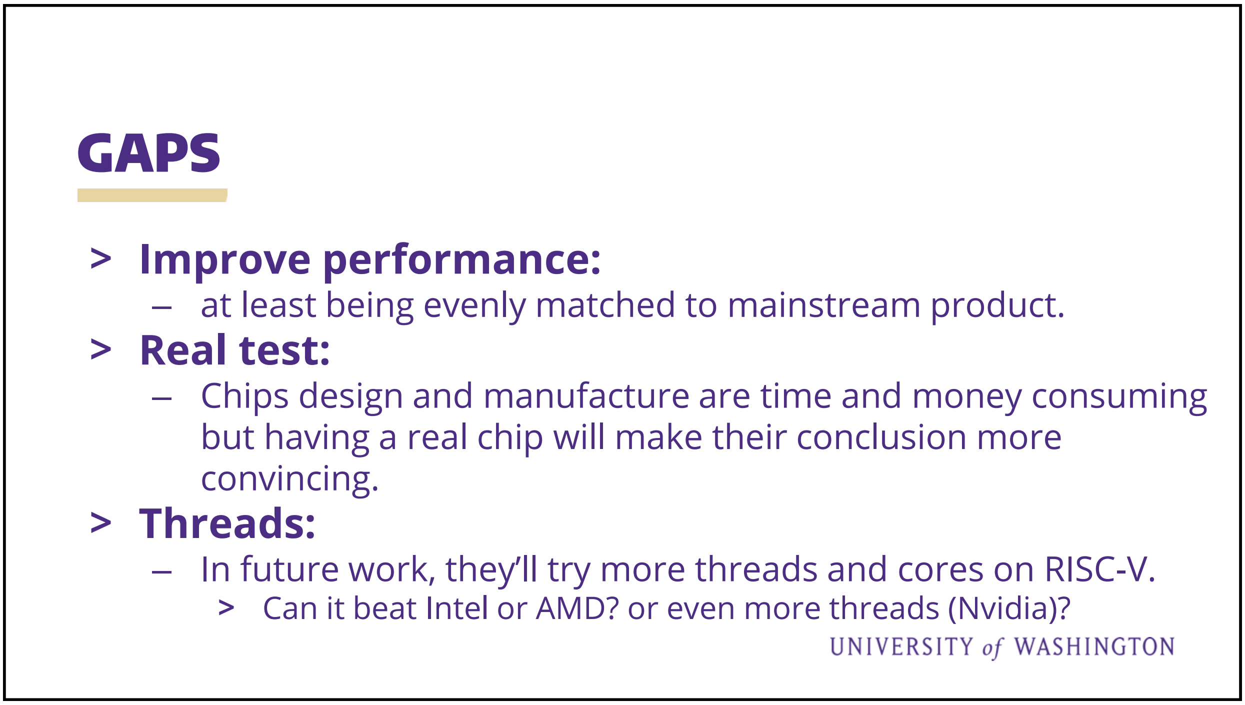


Critique:Evaluation

- The paper evaluates multiple RISC-V configurations (e.g. cache size, cache layout, and ROB size) and compares them to commercial processors.
- They mentioned cold start time is a bottleneck for serverless platform. while they didn't simulate it.
  - As least show some more supportive evidence for RISC-V.

UNIVERSITY *of* WASHINGTON

17



GAPS

- **Improve performance:**
  - at least being evenly matched to mainstream product.
- **Real test:**
  - Chips design and manufacture are time and money consuming but having a real chip will make their conclusion more convincing.
- **Threads:**
  - In future work, they'll try more threads and cores on RISC-V.
    - Can it beat Intel or AMD? or even more threads (Nvidia)?

UNIVERSITY *of* WASHINGTON

18

Questions?

UNIVERSITY *of* WASHINGTON

19