

Georgios Pournaras⁺

Vasileios Karakostas⁺

George Papadimitriou^{*}

Dimitris Gizopoulos⁺

⁺University of Athens, Greece

^{*}University of Patras, Greece

cal.di.uoa.gr

1. Motivation

- Serverless computing has gained a lot of interest recently
- The interest in RISC-V has also increased significantly
- Our Goal: Bridge the gap in benchmarking support between serverless computing and RISC-V across the stack

3. RISC-V Serverless Benchmarking Support

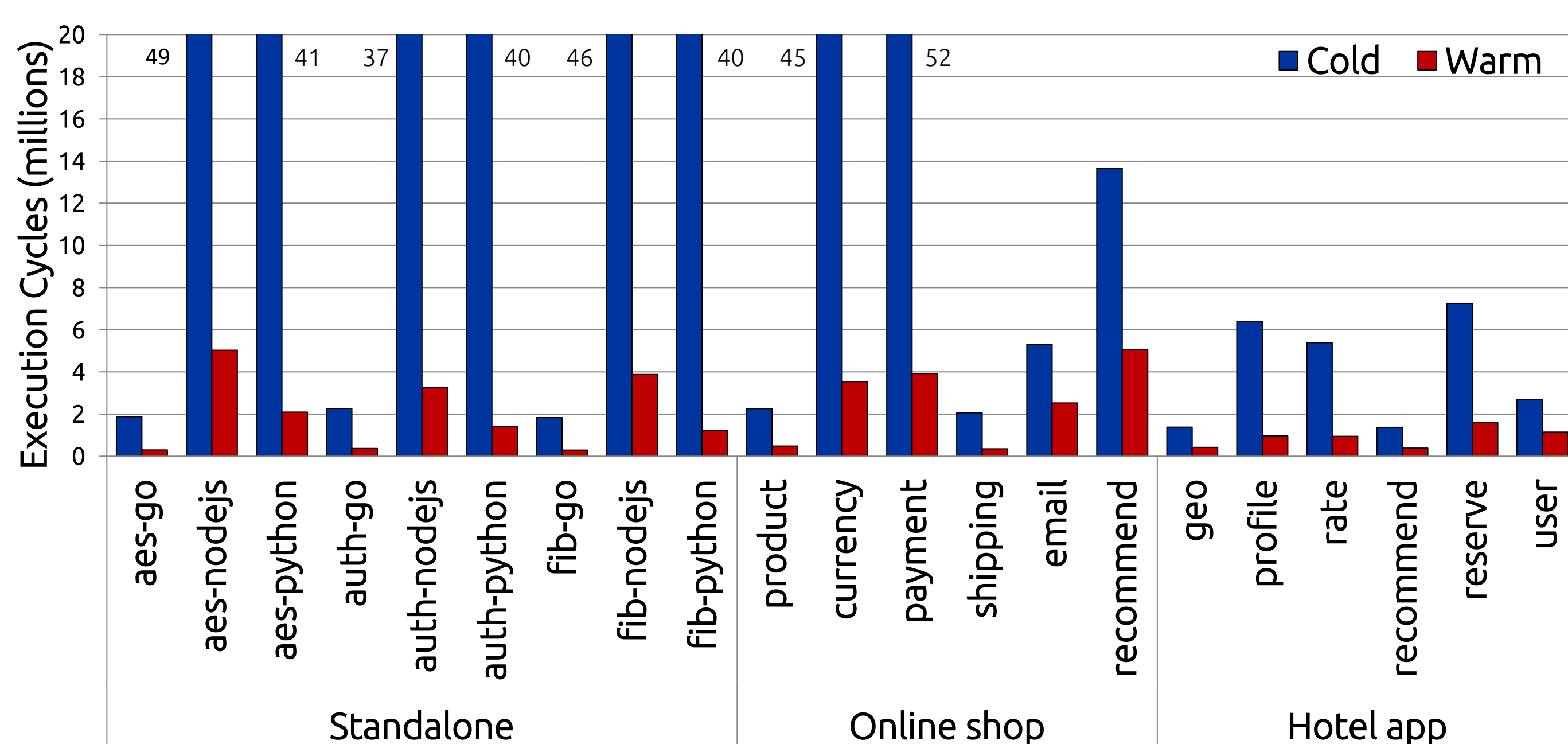
Porting the vSwarm Functions to the RISC-V ISA

- Generic methodology
 - (i) Identify base images, (ii) resolve dependencies, (iii) update dockerfiles, and (iv) build containers
- Case study: Hotel application
 - Dependency on MongoDB which has not been ported to RISC-V → used Apache Cassandra instead

Running the Functions

- Simulated systems: (i) QEMU to create disk images, (ii) custom Linux kernel to run containers, (iii) gem5 to create checkpoints and measure performance
- Real systems: scripts that ease experimentation via perf

4. Evaluation Results



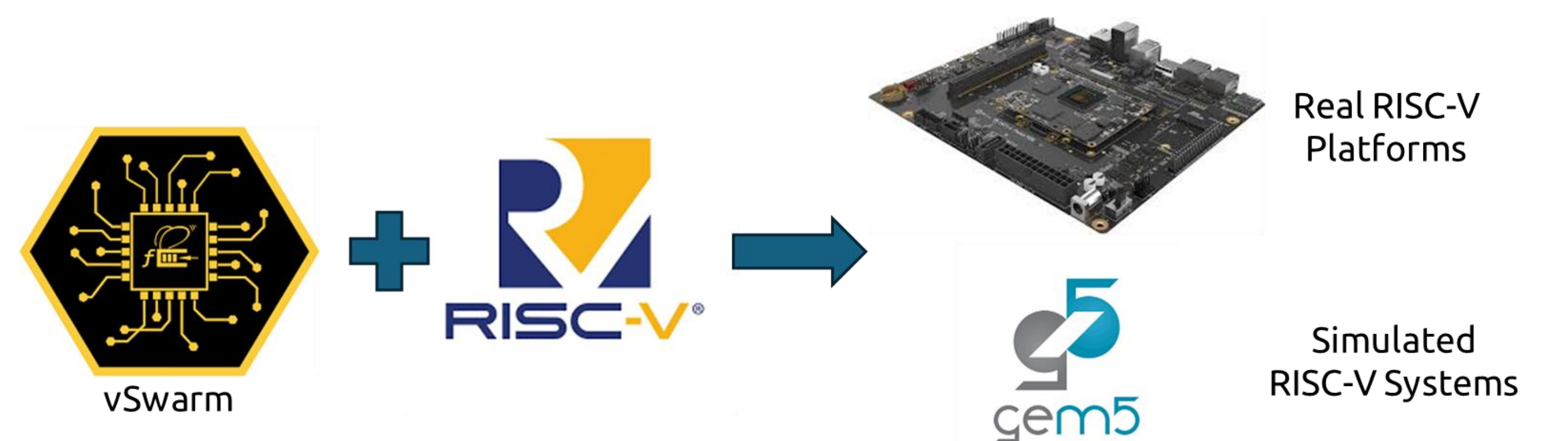
- Performance differs across functions and depends on the language & runtime
- The cold vs. warm performance trade-off is significant, ranging from 2.1x to 32.7x
- L1 I-cache misses are higher than L1 D-cache misses for most functions except for the Python-based ones
- Minor impact due to changing the database in Hotel

Infrastructure Availability

- The source code is publicly available at: <https://github.com/vitamin-v-software/vSwarm-RISCV>

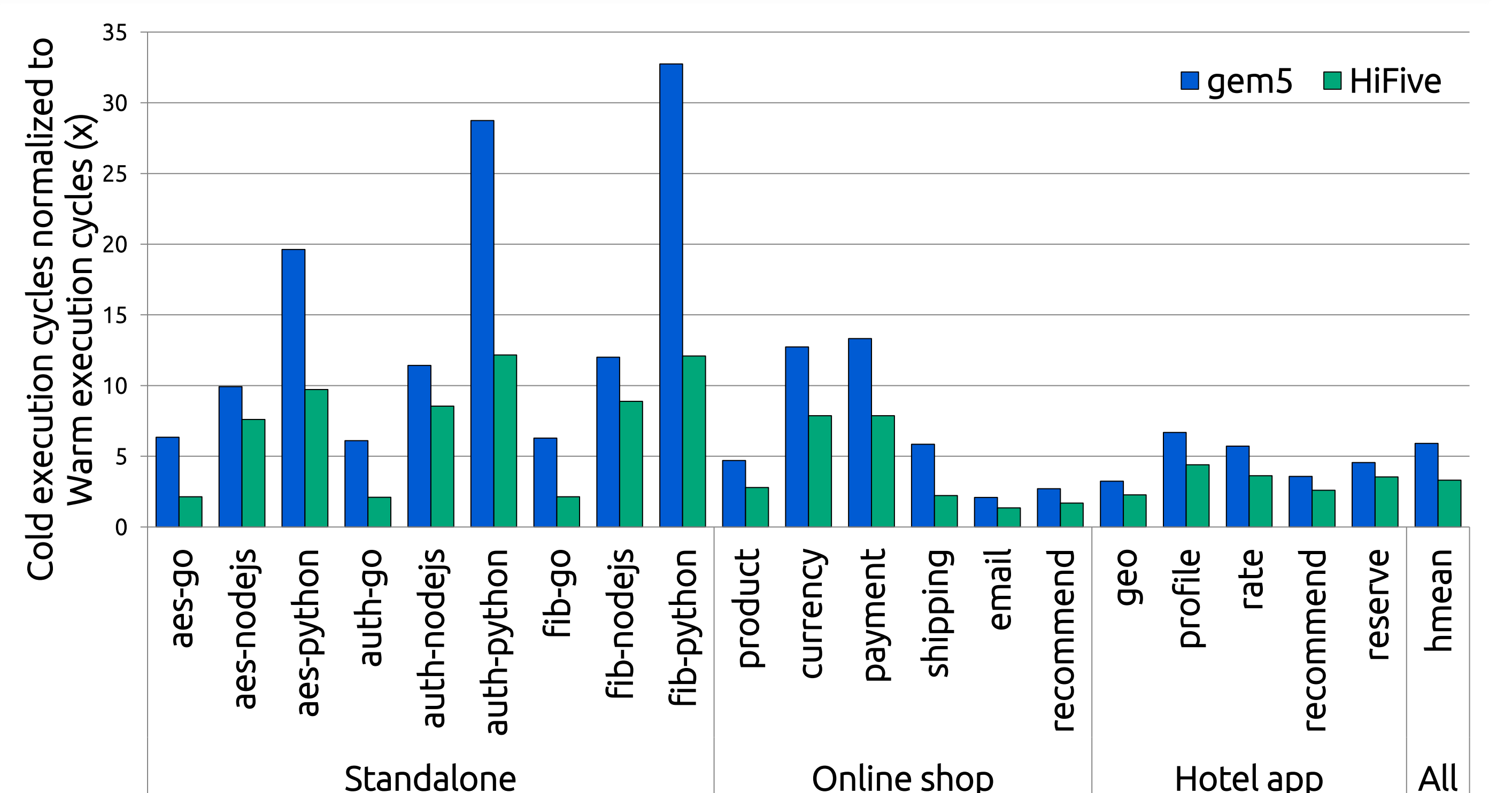
2. Our Approach

- Enhance the vSwarm benchmarking infrastructure ^[1, 2]
 - Why vSwarm? Variety of workloads | x86 & Arm | gem5
- Port the serverless benchmarks to RISC-V
 - Allows experimentation on real RISC-V platforms
- Provide support to run the RISC-V functions in gem5
 - Enables evaluation of various uarch components
 - Allows direct comparison of different ISAs



App	Functions	Runtime
	AES, Auth, Fibonacci	Go/Python/NodeJS
Online Shop	Product Catalog, Shipping	Go
	Recommendation, Email	Python
	Currency, Payment	NodeJS
Hotel	Geo, Profile	Go
	Rate, Recommendation	
	Reservation, User	

5. Validation



- Cold vs. warm execution on gem5 & HiFive Premier P550
- gem5 underestimates the execution time compared to the real platform
- Still, similar performance trade-off across functions

References

- [1] D. Ustiugov, P. Petrov, M. Kogias, E. Bugnion, and B. Grot. "Benchmarking, analysis, and optimization of serverless function snapshots". ASPLOS 2021
- [2] <https://github.com/vhive-serverless/vSwarm>