

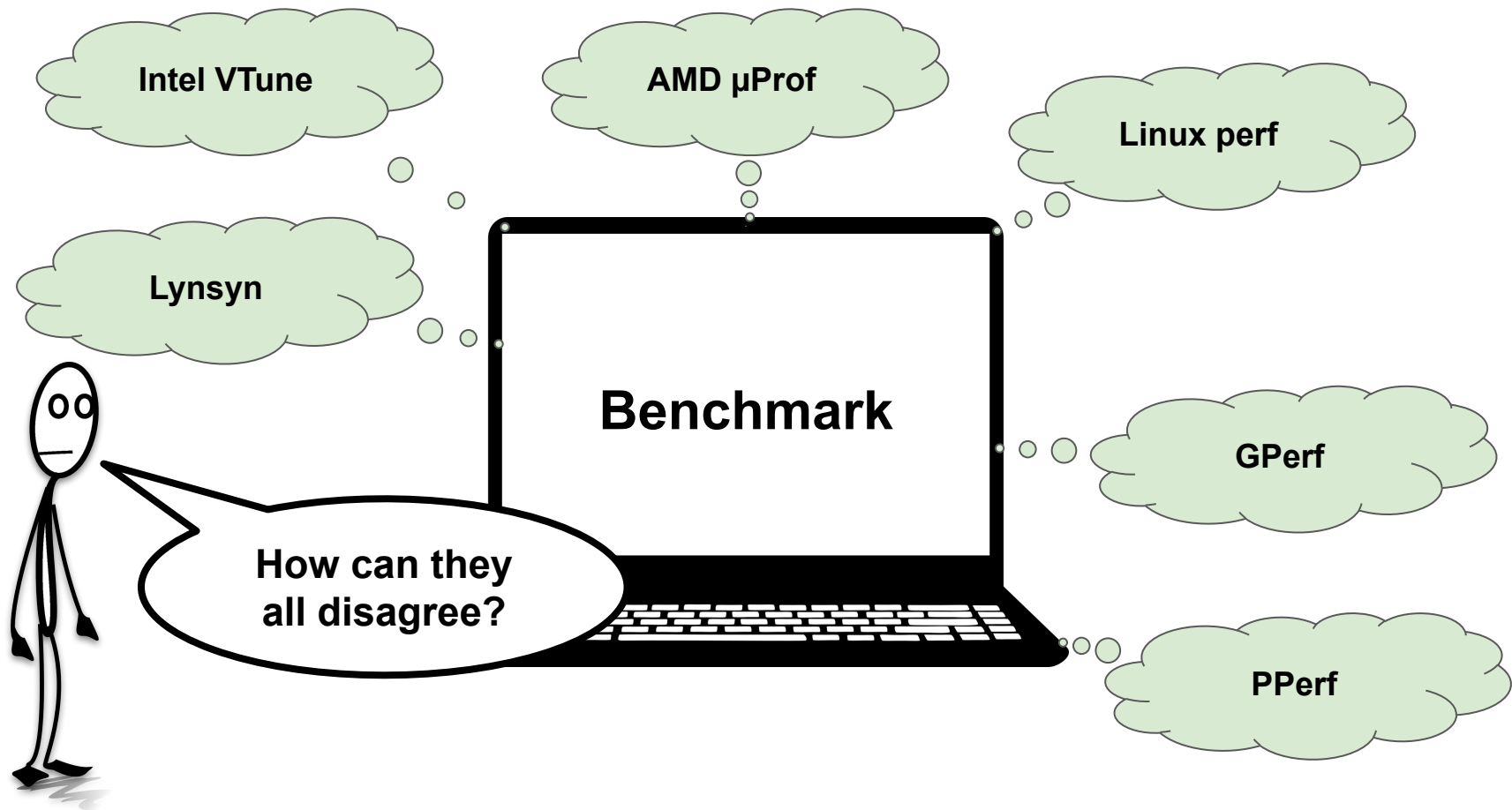


Norwegian University of  
Science and Technology

# **TraceDoctor: Versatile High-Performance Tracing for FireSim**

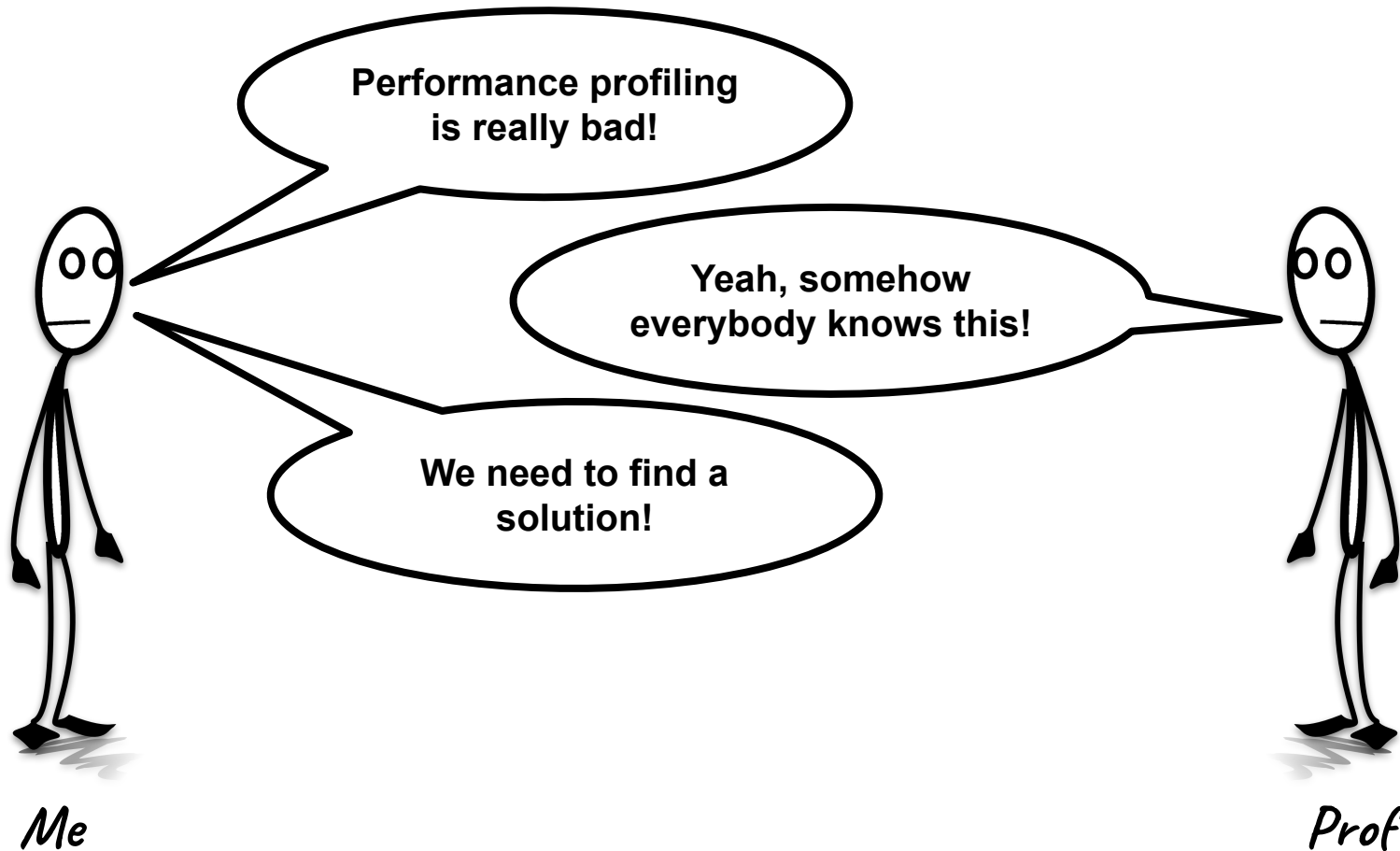
**FireSim and Chipyard User and Developer Workshop**  
ASPLOS 2023, Vancouver, BC, Canada

Björn Gottschall and Magnus Jahre



*Me*

**Spoiler:** They are all wrong! "TIP: Time-Proportional Instruction Profiling" [Micro21]



Performance profiling  
is really bad!

Yeah, somehow  
everybody knows this!

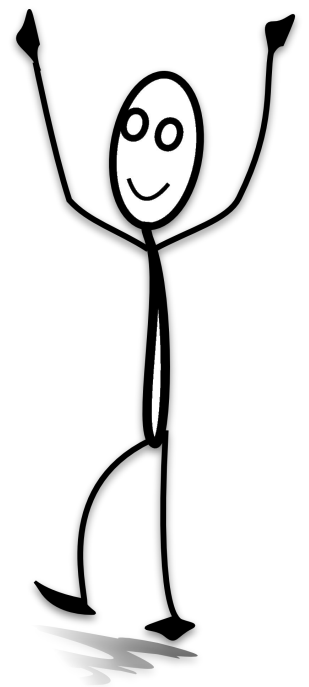
We need to find a  
solution!

*Me*

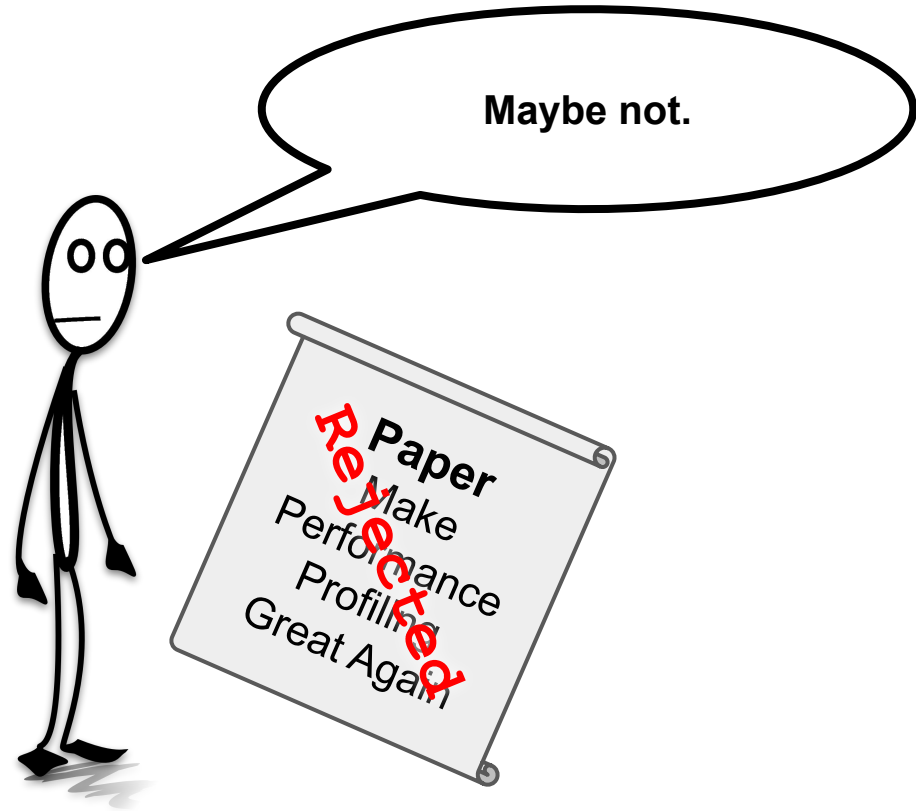
*Prof*



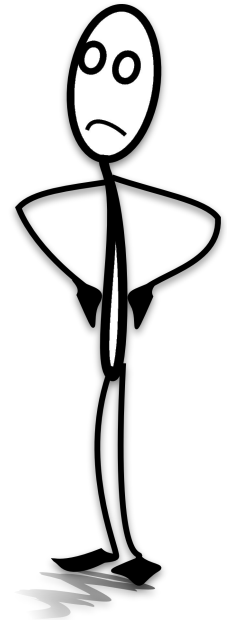
*Me*



*Prof*



*Me*



*Prof*

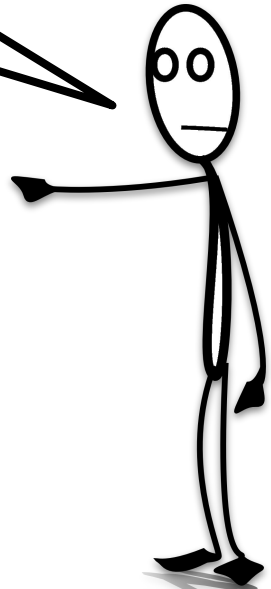


*Me*

**News**  
FireSim  
1.9.0  
released

Hold my beer!

We need a **Golden Reference!**  
We need to know about every instruction and every cycle!

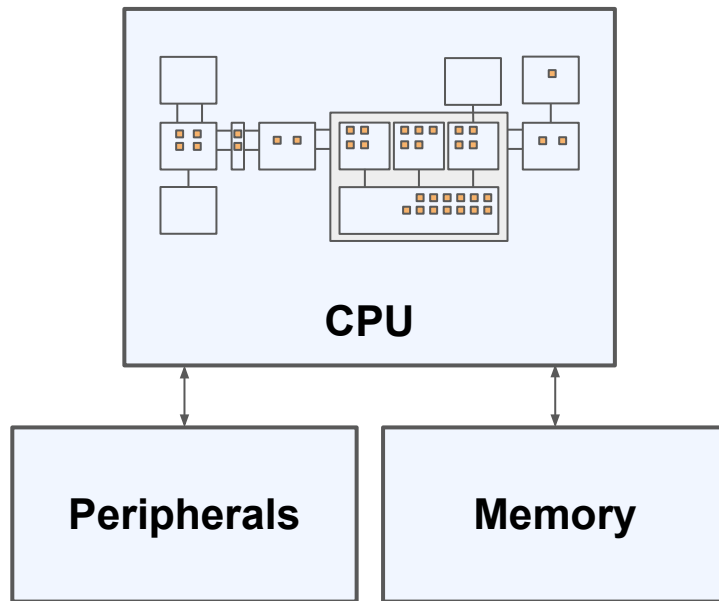


*Prof*

# FireSim

## Cycle-Accurate Full System Simulator

- Out-of-order, high performance, application class processor
- Full-stack Linux environment
  - block devices
  - networking
- Industry standard benchmarks
  - SPEC CPU2017
  - PARSEC



**We want to know about every single instruction and cycle!**

→ No functional models    → No statistical sampling    → No SimPoints

# Tracing in FireSim

## AutoCounter

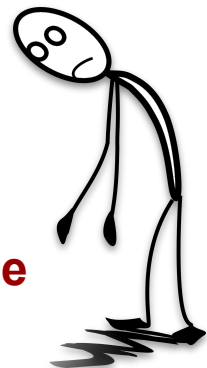
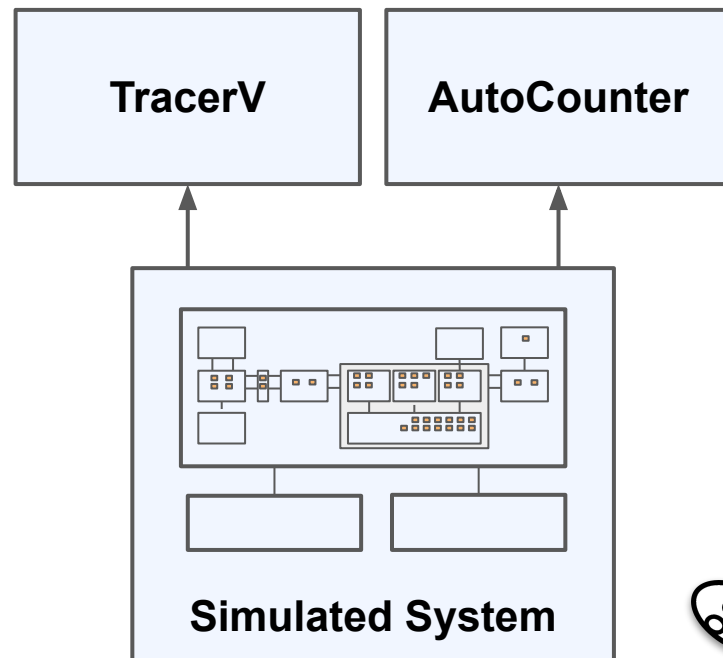
- Can provide statistical data from selected signals & registers

## TracerV

- Traces out all retiring instructions
  - Binary, text or flamegraph format
- Triggers can limit tracing scope
- 64 bytes per cycle (binary)

## SPEC CPU2017 bwaves on the SonicBoom

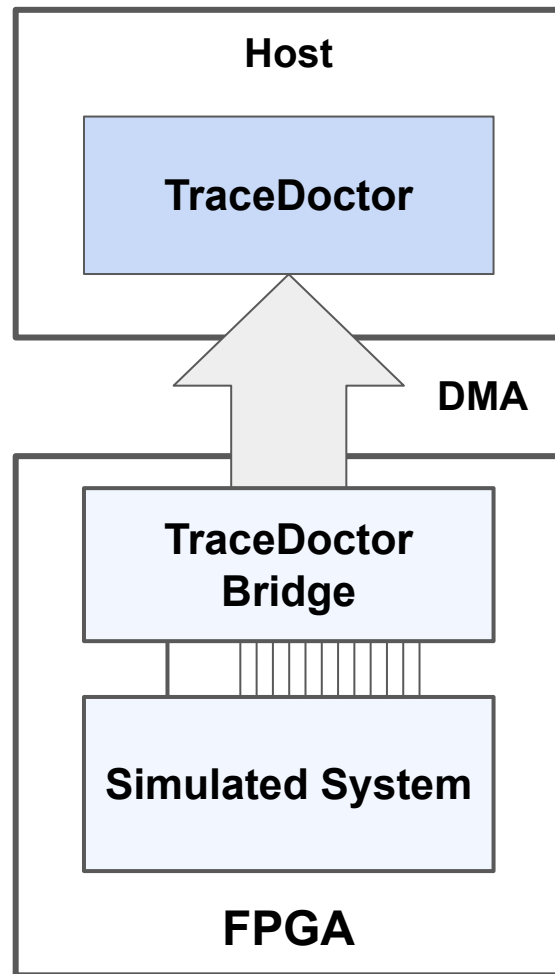
→ 2.8 trillion instructions → 6.8 trillion cycles → **433 terabyte trace**



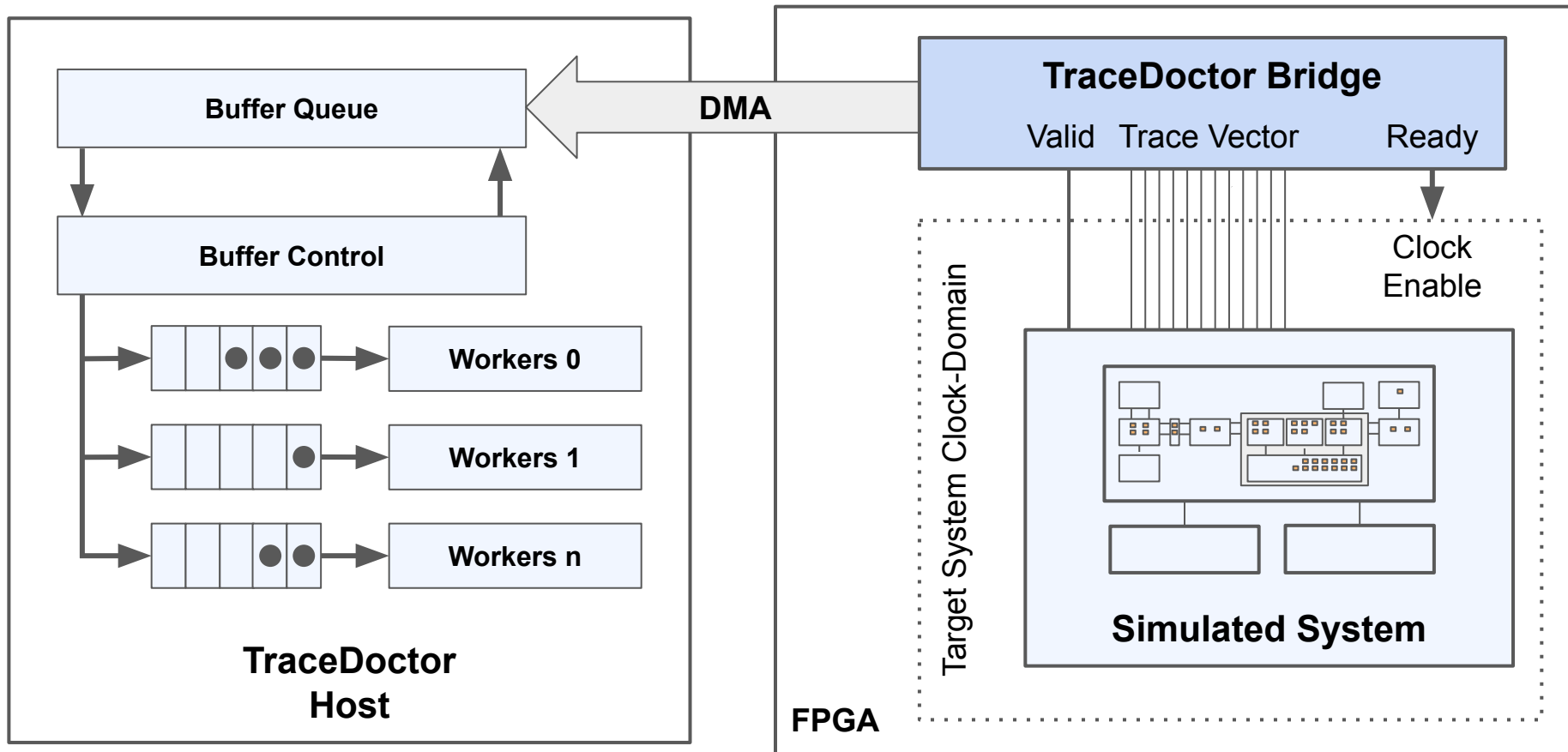


# TraceDoctor

- Can attach to **any signals** within the simulated design and trace them out **every cycle**
- Utilizes a **buffered & parallel host framework** to process traced data
- A **flexible** worker design enables **efficient** and highly **customizable** trace processing
- We analyzed more than **300 trillion cycles** over **292 trillion dynamic instructions** amounting to **4.3 petabytes of data**



# TraceDoctor Overview



# TraceDoctor - Writing a worker

How easy can it be?

```
td_binary::td_binary(  
    std::vector<std::string> const args,  
    struct traceInfo const info  
) : tracedoctor_worker("TD Binary", args, info, 1) {}
```

```
void td_binary::tick(char const * const data, unsigned int const tokens)  
{  
    fwrite(data, tokens, info.tokenBytes,  
           std::get<freg_descriptor> fileRegister[0]);  
}
```

→ That's all! This produces the same output as TracerV in binary mode.

# TraceDoctor - Using our worker

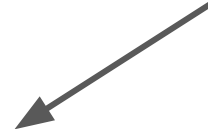
Command line  
parameter



Worker name



Output file



```
+tracedoctor-worker=td_binary,file:binary-%id.bin
```

How about automatic output compression?

```
+tracedoctor-worker=td_binary,file:binary-%id.bin.zst
```

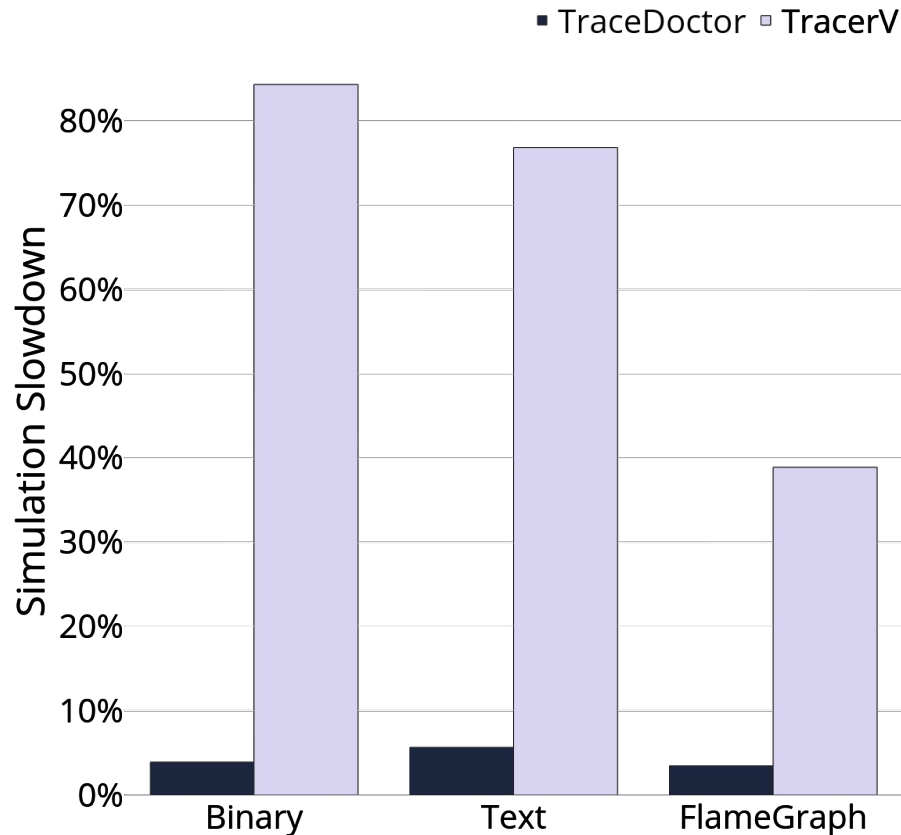


Just add a file extension  
e.g. .gz, .bz2, .xz or .zst

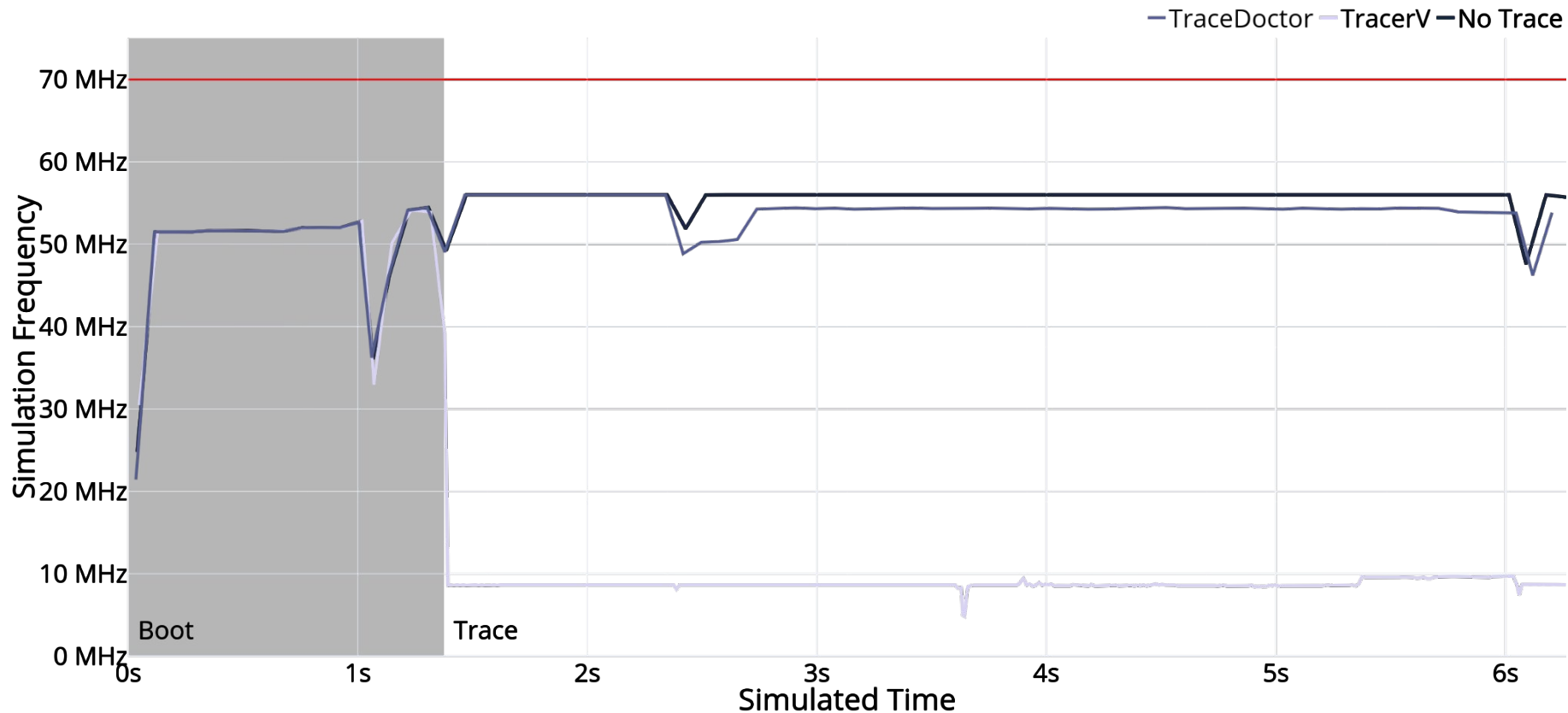
# Trace Performance Evaluation

FireSim 1.15.1

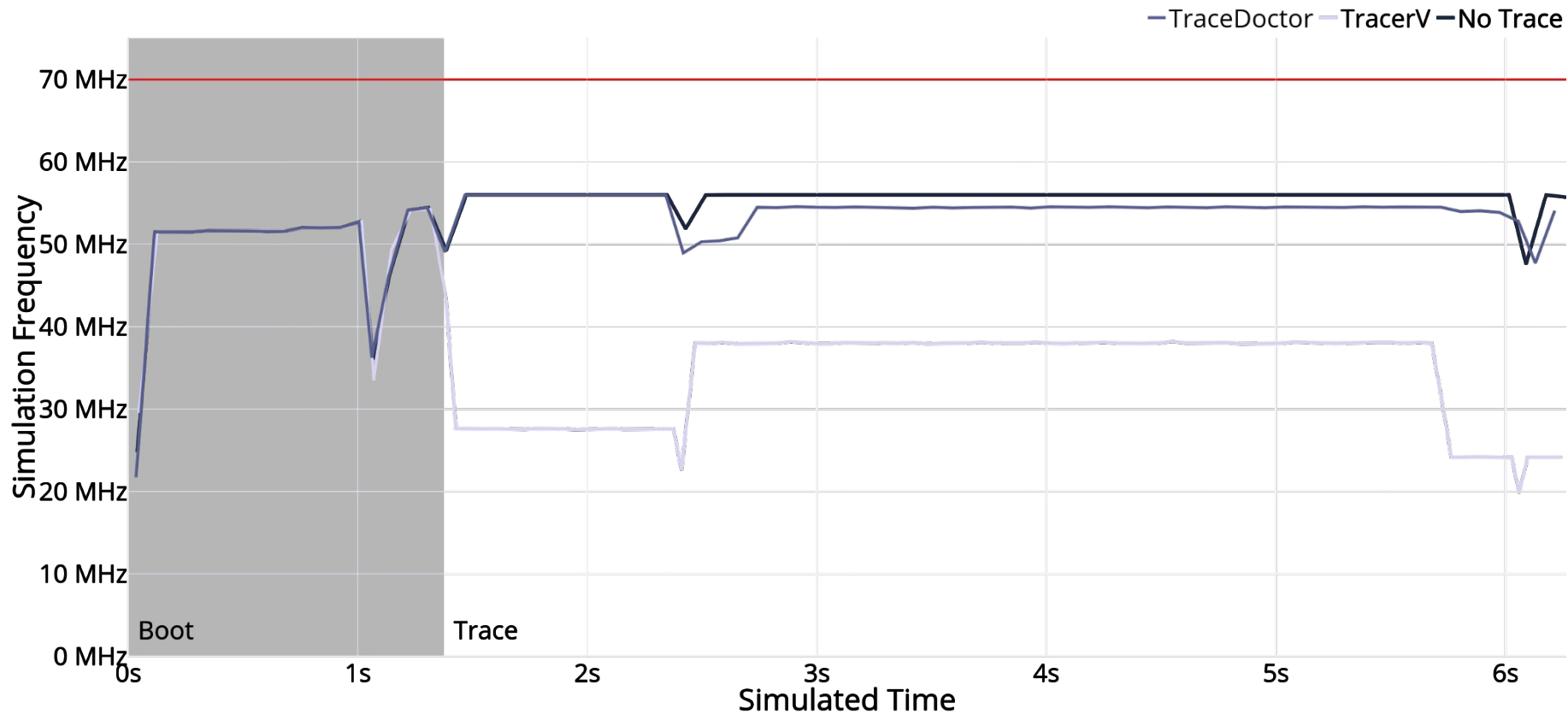
- SonicBoom (4-wide) @ 3.2 GHz
- Xilinx Alveo u250 @ 70 MHz
- Buildroot Linux 5.7
- SPEC CPU2017 nab
  - Test input set
- TracerV & TraceDoctor
  - Tracing of 7 billion dynamic instructions over 16 billion cycles
- No file I/O latencies
  - Outputs are written to /dev/null



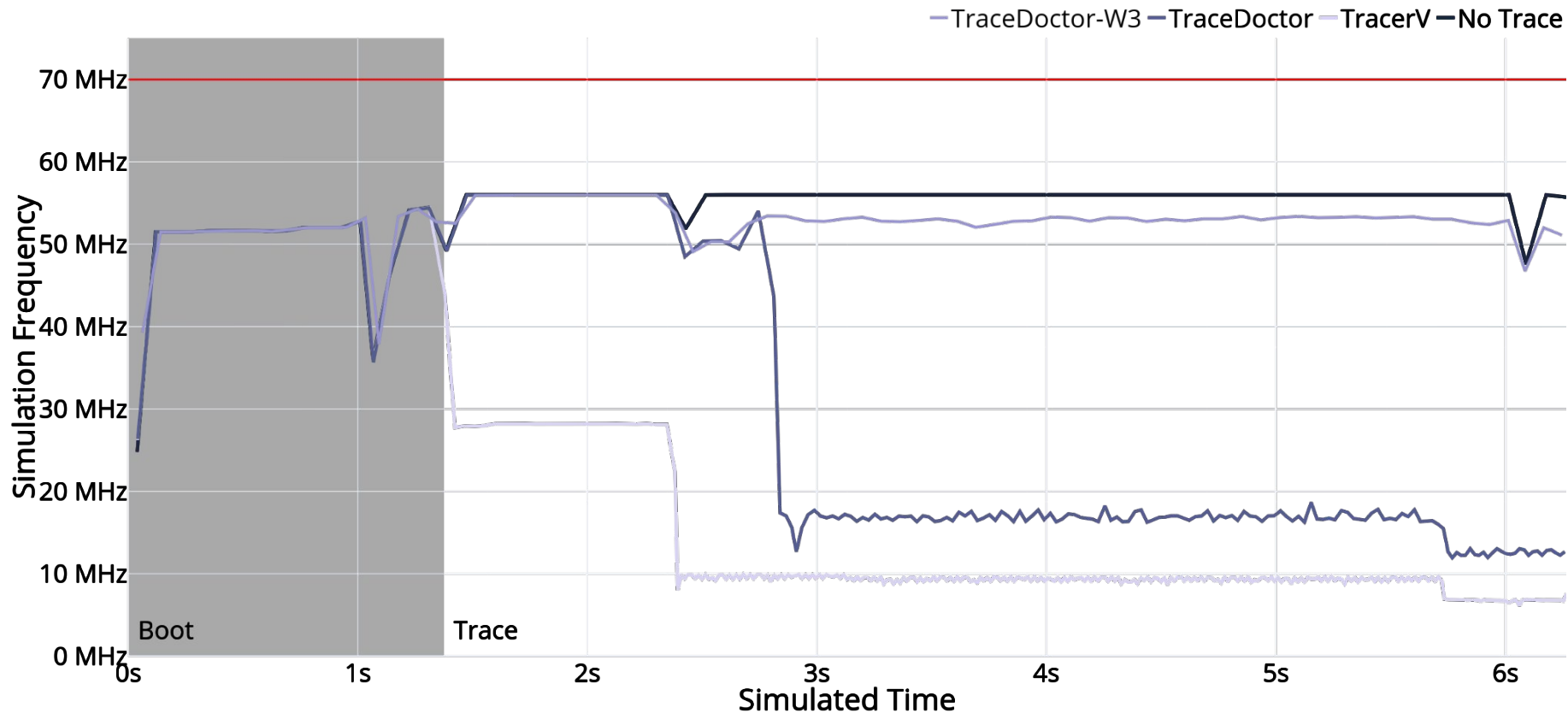
# Trace Performance - Binary Output



# Trace Performance - FlameGraph Output



# Trace Performance - Text Output





# Golden Reference

- 52 trillion dynamic instructions
  - SPEC CPU2017 \w reference inputs
- Tracing data still on the order of petabytes of storage

→ \$21.000 per petabyte per month

→ Performance alone is not enough

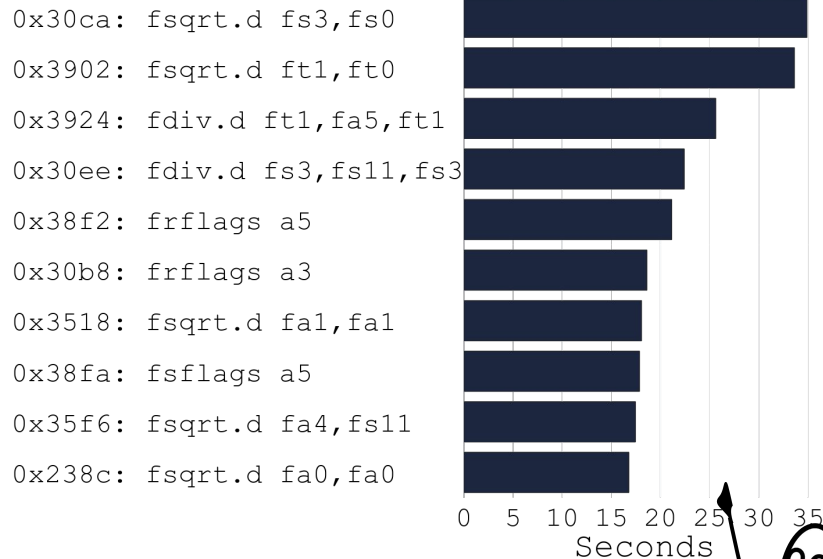
Amazon S3 pricing	
First 50 TB / Month	\$0.023 per GB
Next 450 TB / Month	\$0.022 per GB
Over 500 TB / Month	\$0.021 per GB



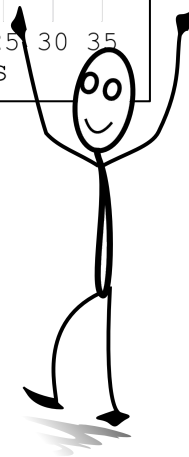
# Golden Reference

- Do we need a trace of every dynamic instruction?
- Performance profiling is about which instruction, function and application consumes time
- We don't need the exact order of retirement

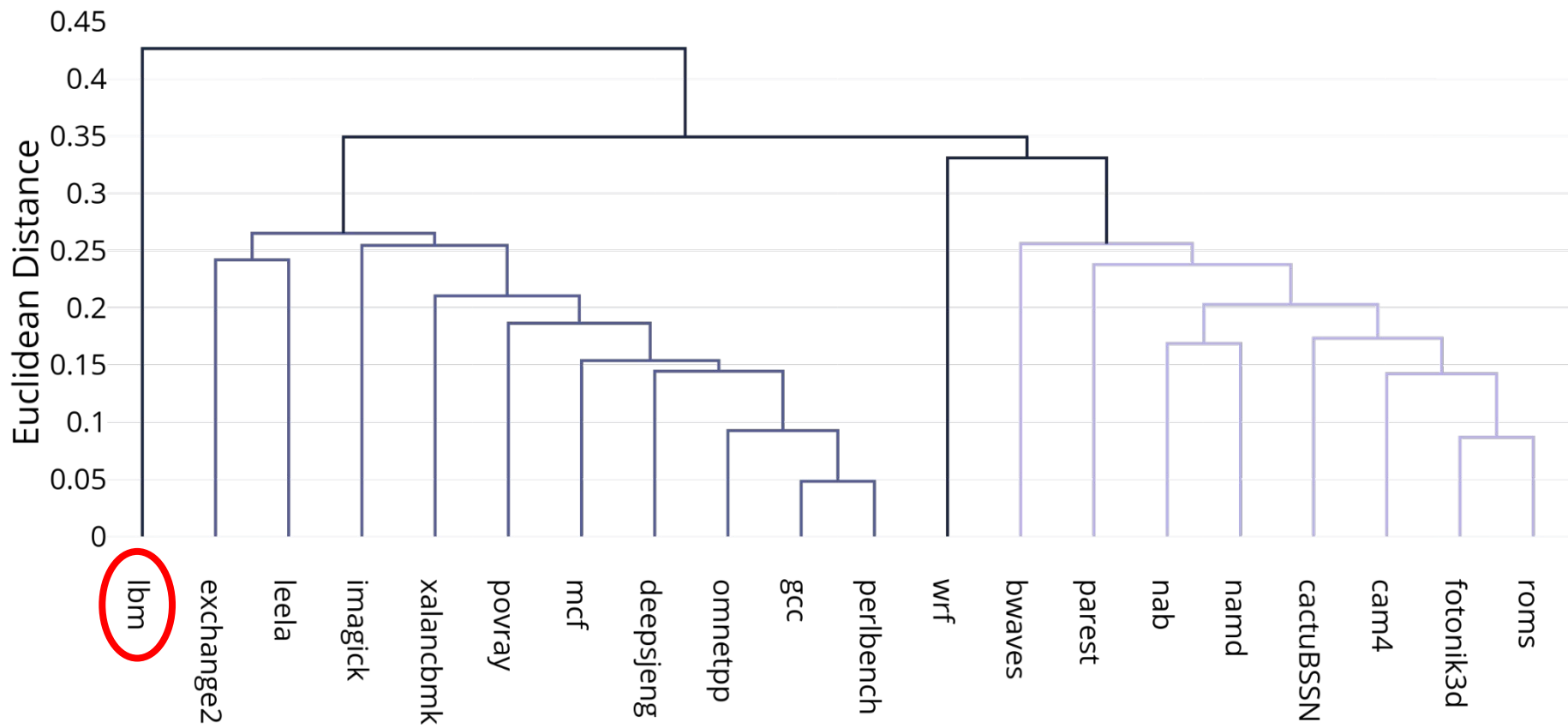
Top-10 Instructions of SPEC CPU2017 nab



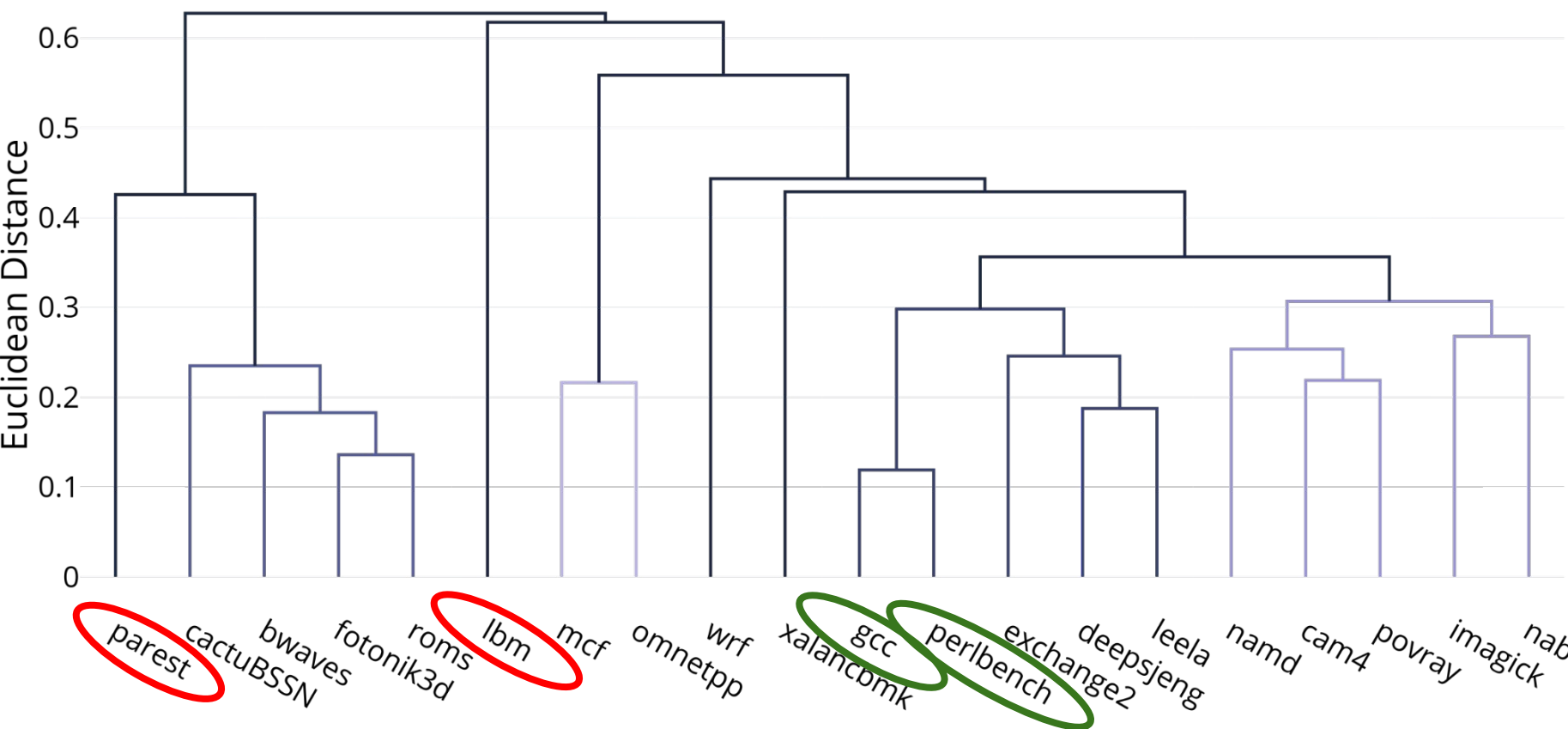
- Aggregate traces on-the-fly for all instruction addresses
- The **Golden Reference** for Performance Profiling



# Clustering SPEC CPU2017 Relative RISC-V Instruction Count

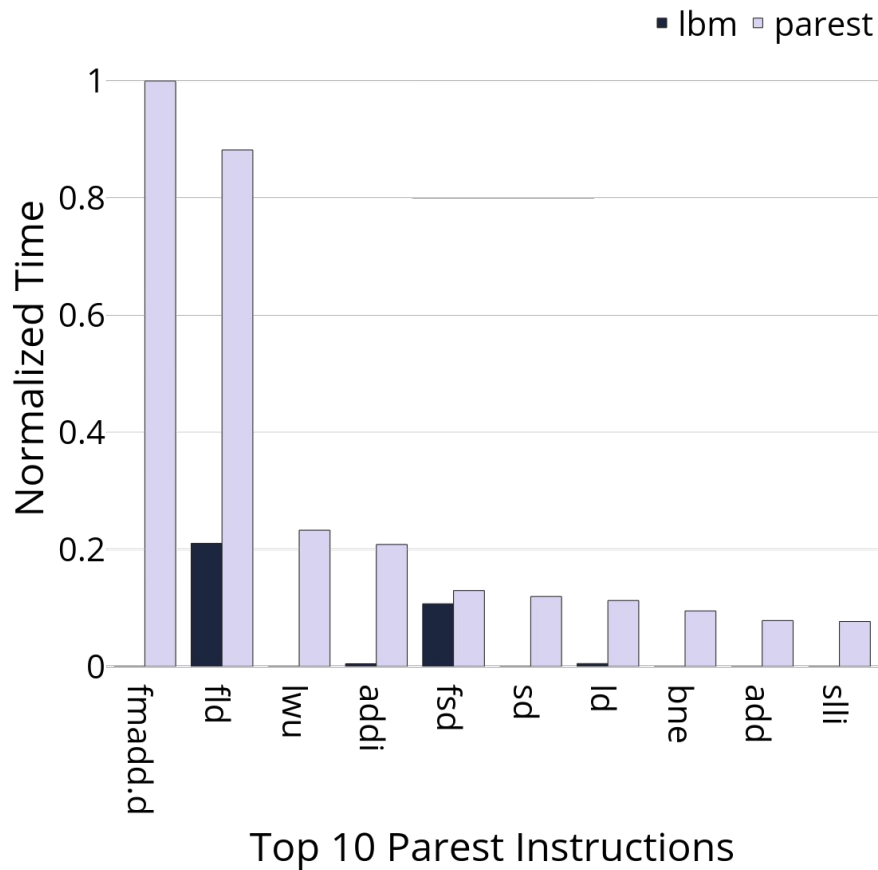
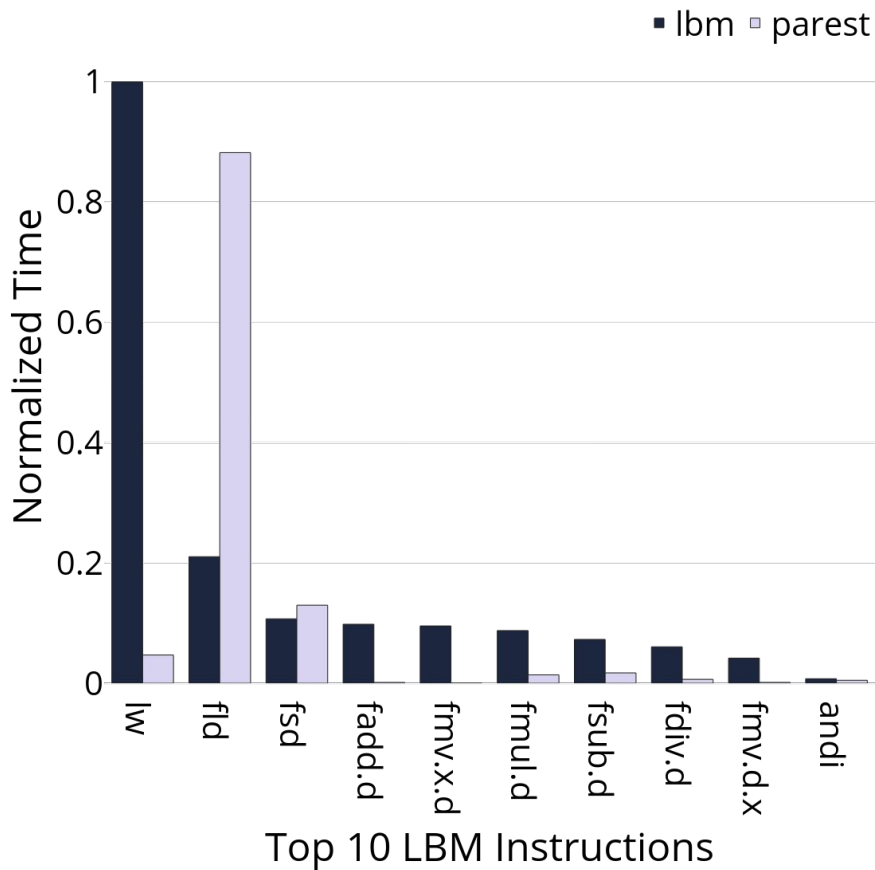


# Clustering SPEC CPU2017 Relative RISC-V Instruction Time

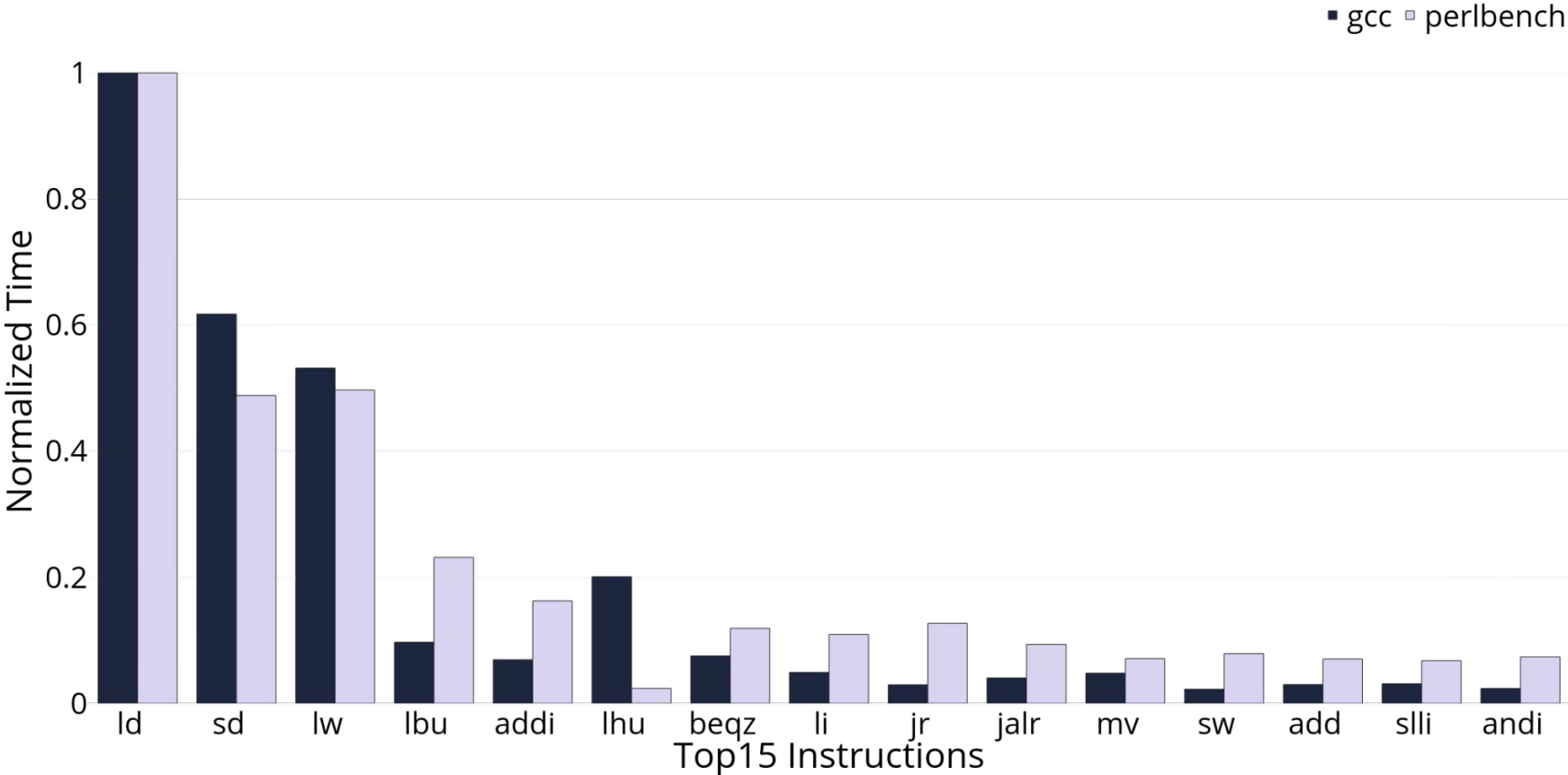


# Why is LBM such a lone wanderer?

- Top 10 Instructions with closest neighbor Parest



# Performance siblings GCC and Perlbench

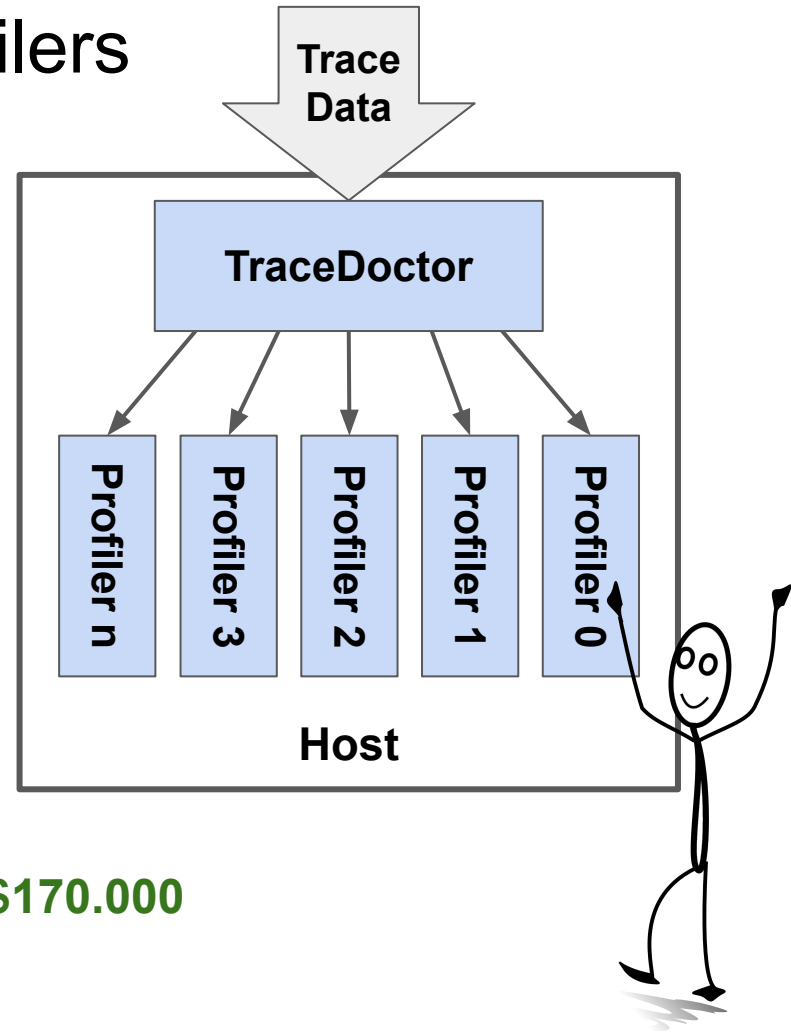


# Evaluation of Performance Profilers

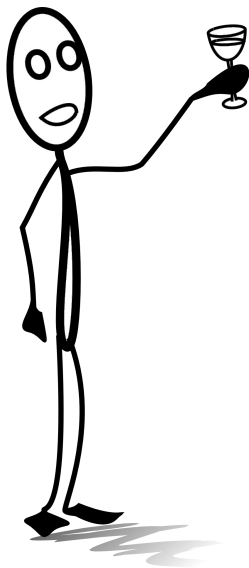
- By tracing the necessary information we can model different performance profilers out-of-band
- TraceDoctor's parallel workers allowed us to run all in many different configurations at once
- E.g. evaluation of different sampling strategies and frequencies

→ Saved us many many simulation runs

→ **A serial evaluation would have costed us \$170.000**



# Conclusion



- TraceDoctor is a high performance tracing interface for FireSim
  - It is flexible, versatile and saves time and money!
  - The first to enable end-to-end tracing in full-stack environments running industry benchmarks
- Ready to be adopted by you and FireSim!

<https://github.com/EECS-NTNU/chipyard/tree/tracedoctor>



# Thank You!



Björn Gottschall  
*[bjorn.gottschall@ntnu.no](mailto:bjorn.gottschall@ntnu.no)*



Magnus Jahre  
*[magnus.jahre@ntnu.no](mailto:magnus.jahre@ntnu.no)*