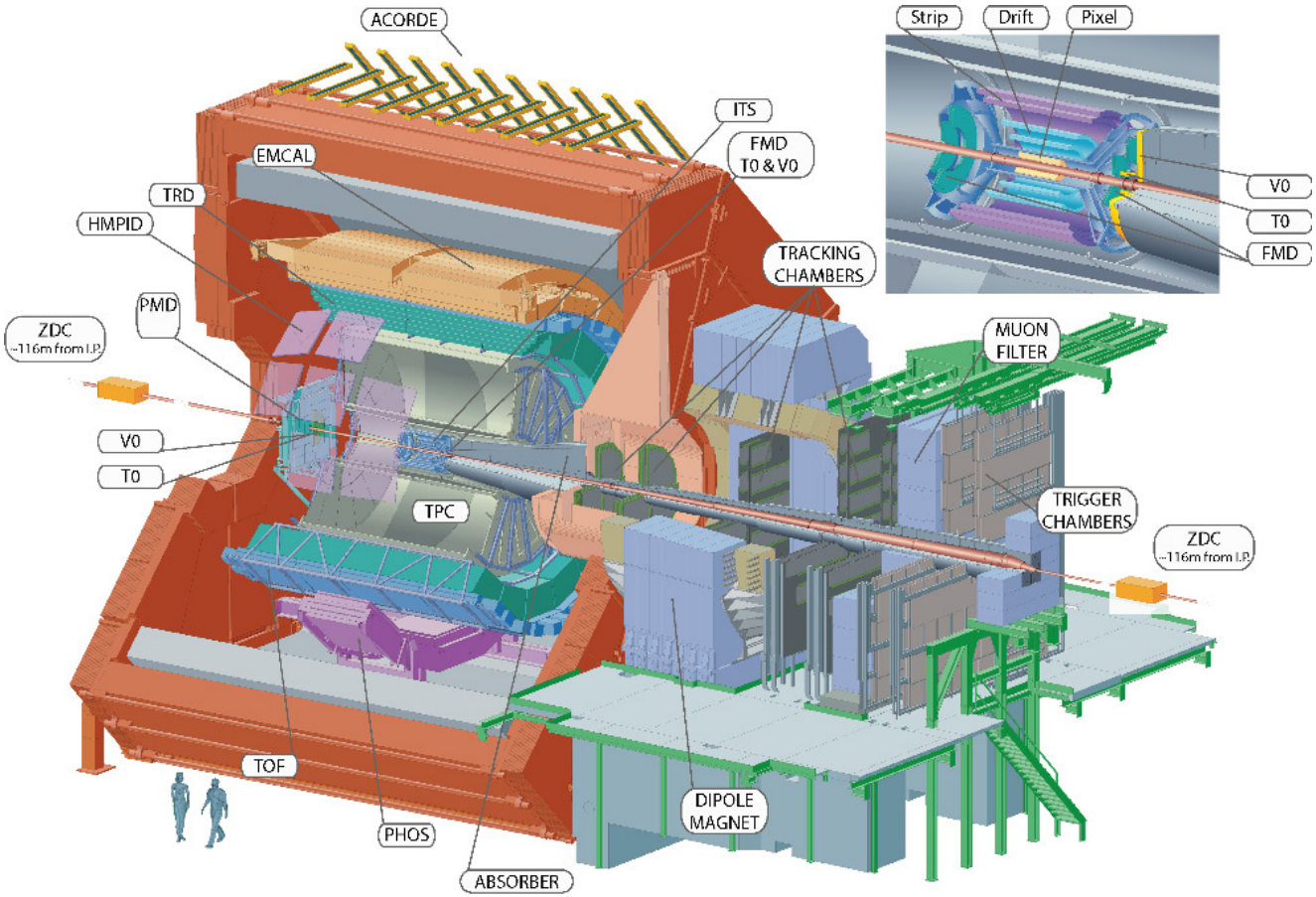
Test Driven Development for Device Drivers

and Rapid Hardware Prototyping

Dominic Eschweiler

13. April 2015

The ALICE Experiment

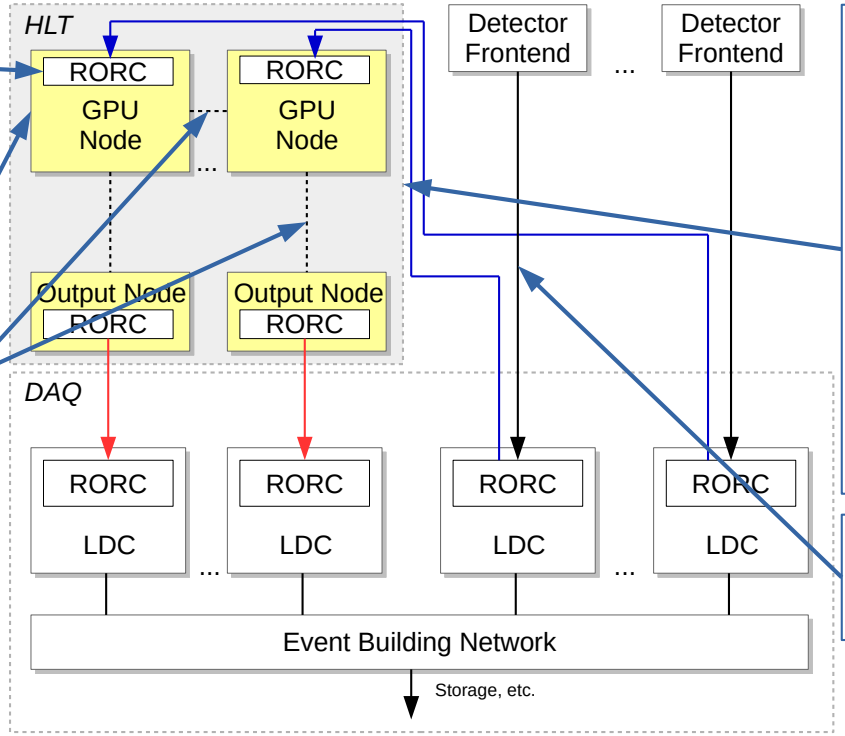


High Level Trigger

First analysis (cluster finding) directly at the Onboard FPGA

~80 Input Nodes
- Commodity Server
- Equipped with CRORCs and GPUs

FDR InfiniBand

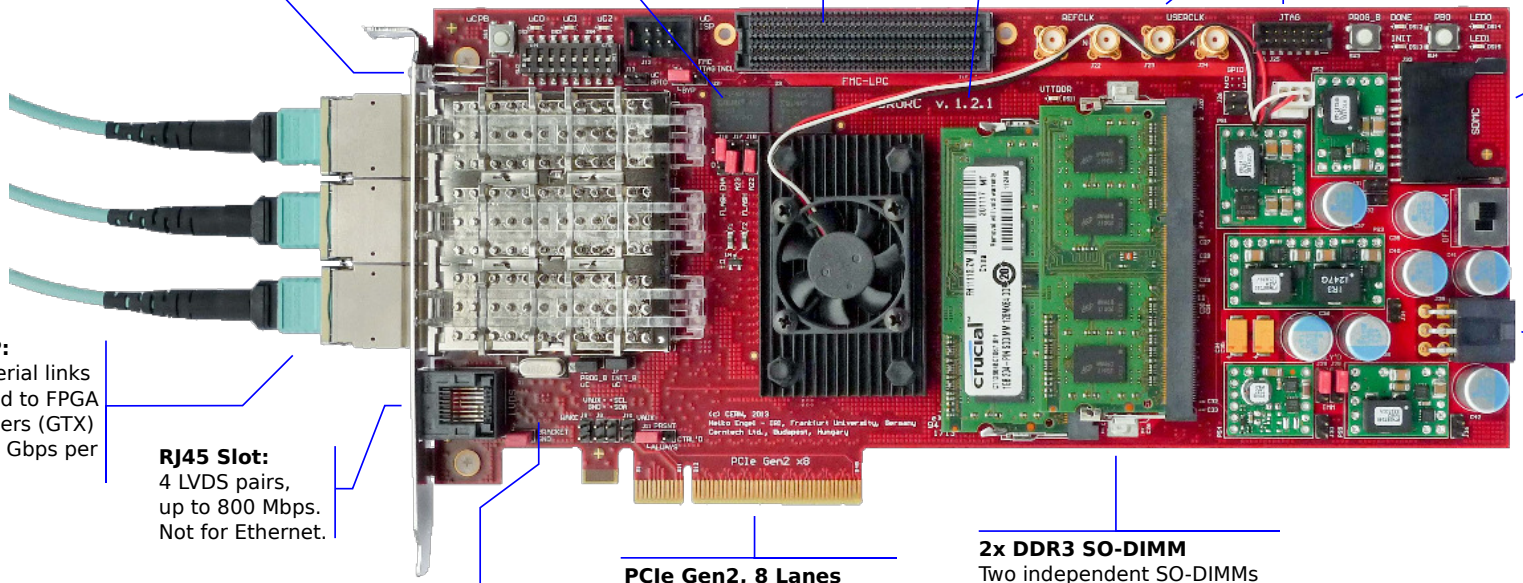


High Level Trigger
- **Trigger** Accept or reject events on the basis of full online reconstruction and analysis
- **Select** physics regions of interest within the events
- **Compress** the event size without loss of physics information.

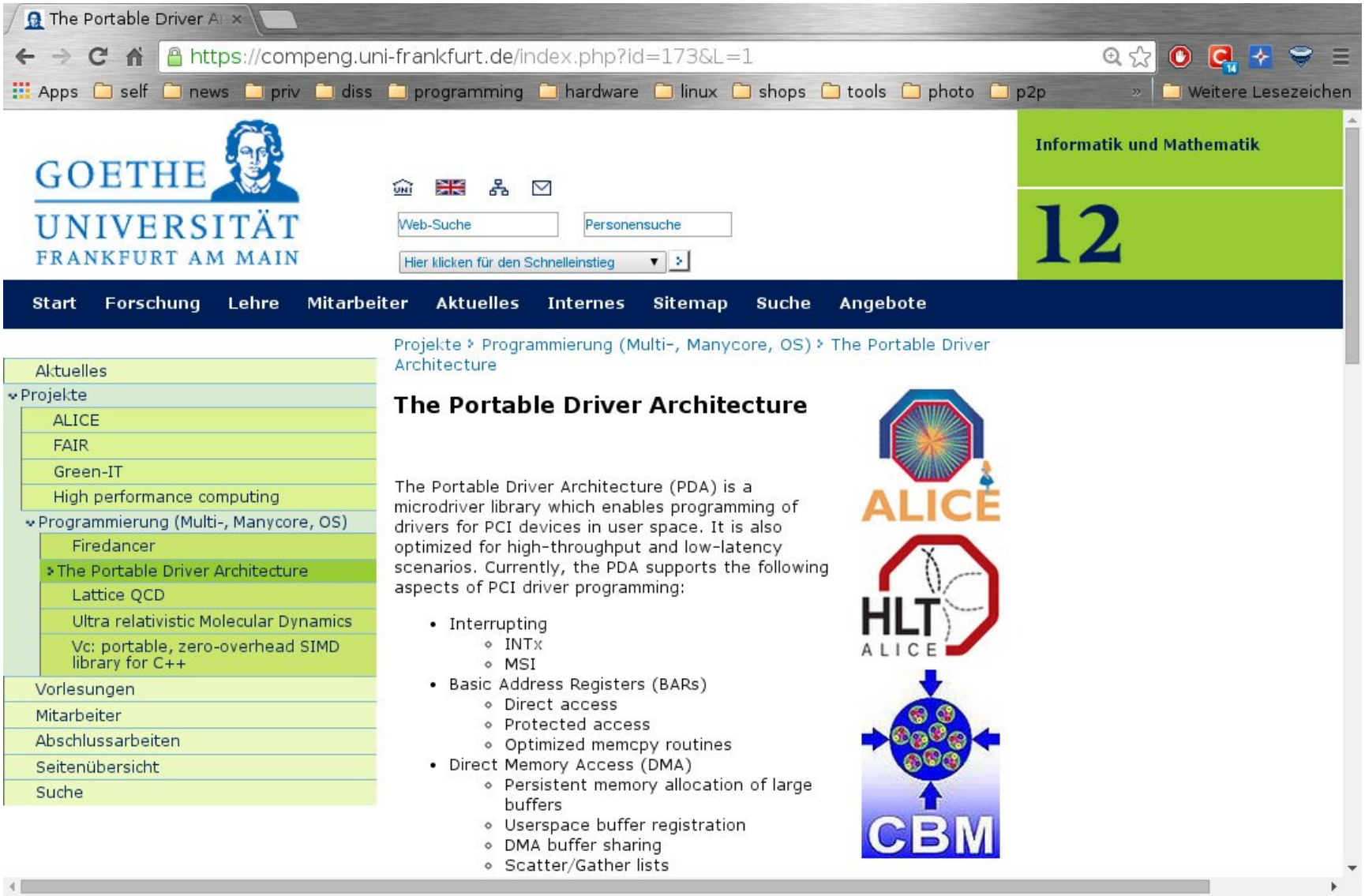
300 GiB/s Combined Peak Input Data Rate
- 500 optical links

The Common Read Out Receiver Card

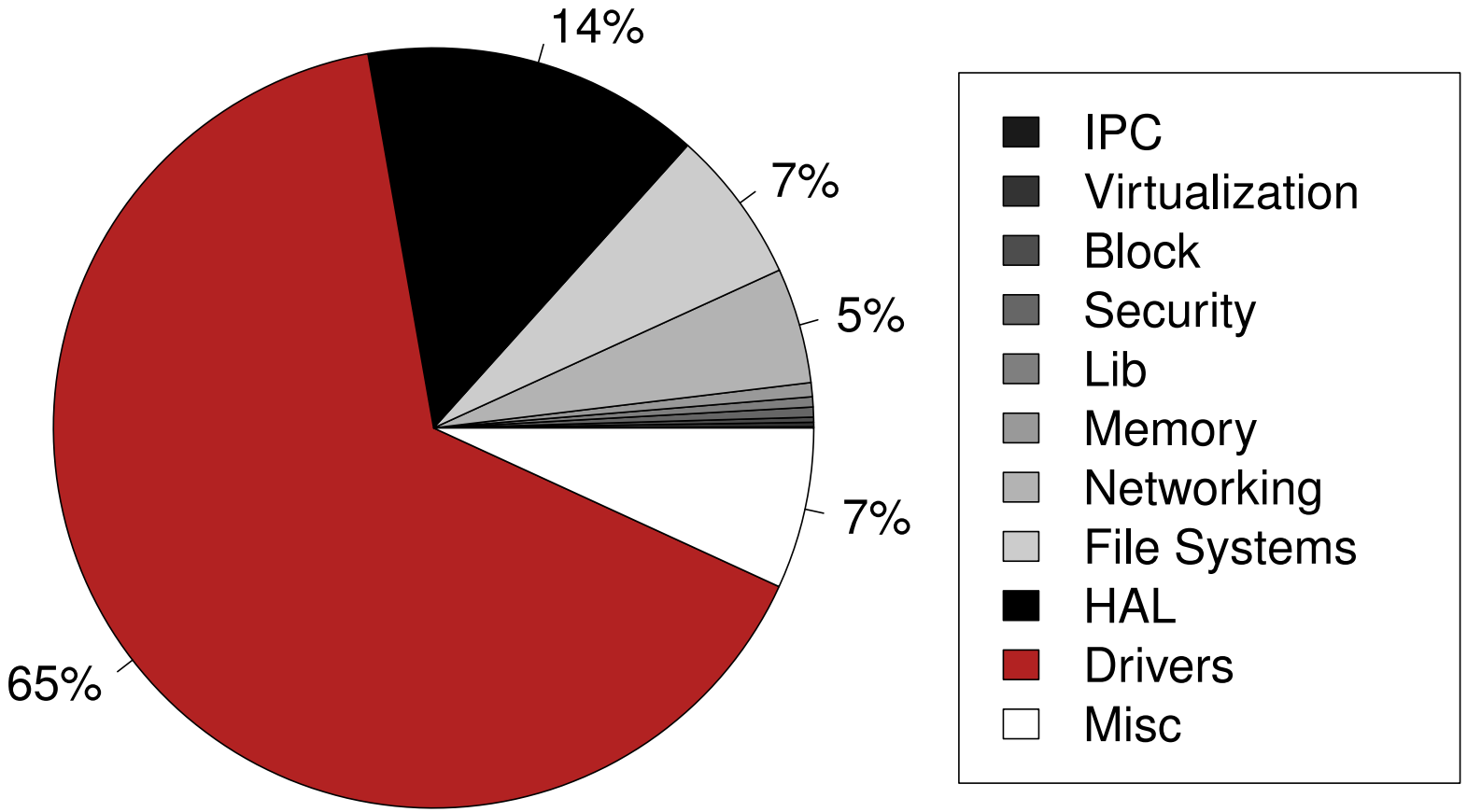
FIAS Frankfurt Institute for Advanced Studies



Author: Heiko Engel

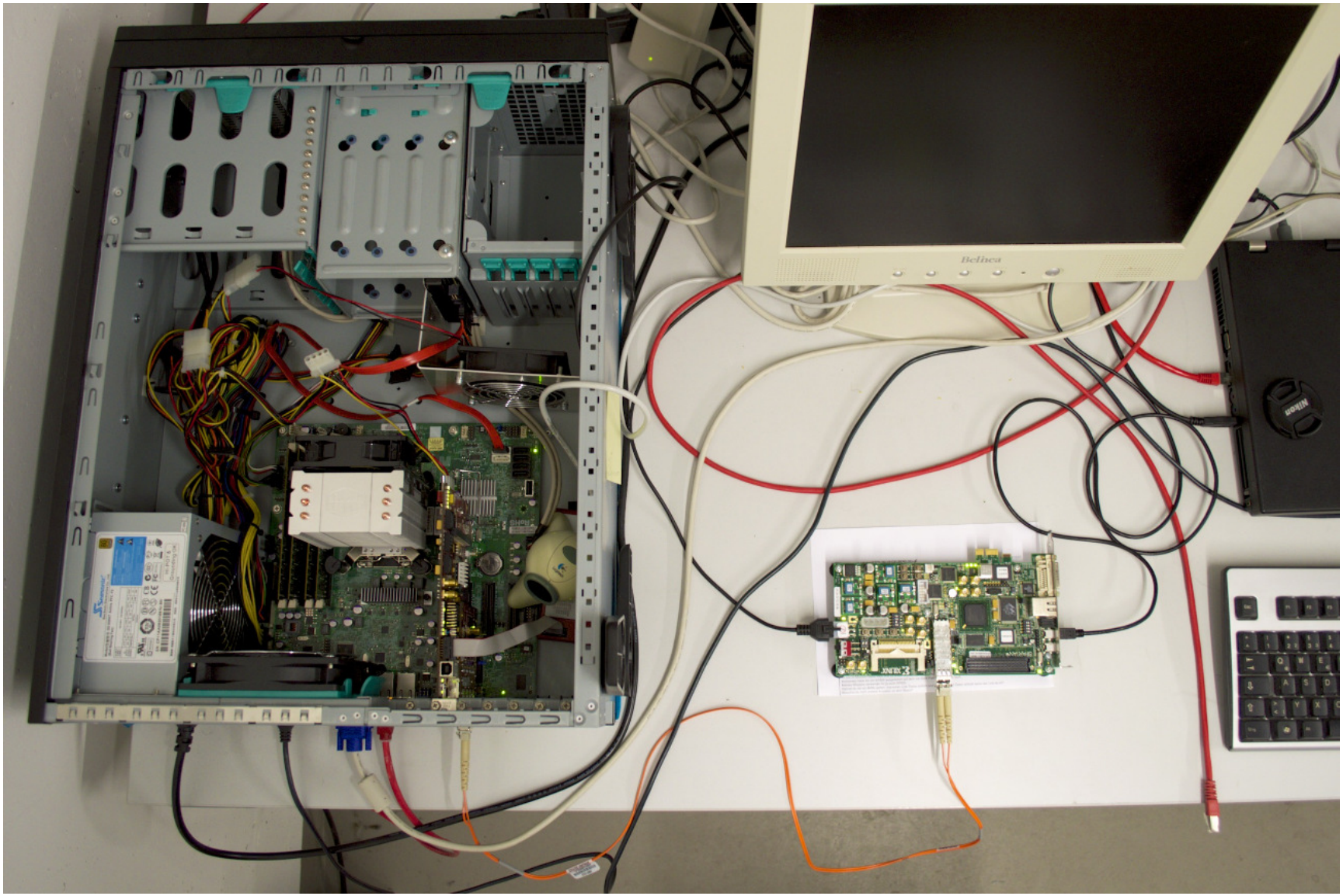

The Portable Driver A
https://compeng.uni-frankfurt.de/index.php?id=173&L=1
Apps self news priv diss programming hardware linux shops tools photo p2p Weitere Lesezeichen
GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Web-Suche
Personensuche
Hier klicken für den Schnelleinstieg
Informatik und Mathematik
12
Start Forschung Lehre Mitarbeiter Aktuelles Internes Sitemap Suche Angebote
Aktuelles
Projekte
ALICE
FAIR
Green-IT
High performance computing
Programmierung (Multi-, Manycore, OS)
Firedancer
The Portable Driver Architecture
Lattice QCD
Ultra relativistic Molecular Dynamics
Vc: portable, zero-overhead SIMD library for C++
Vorlesungen
Mitarbeiter
Abschlussarbeiten
Seitenübersicht
Suche
Projekte › Programmierung (Multi-, Manycore, OS) › The Portable Driver Architecture
The Portable Driver Architecture
The Portable Driver Architecture (PDA) is a microdriver library which enables programming of drivers for PCI devices in user space. It is also optimized for high-throughput and low-latency scenarios. Currently, the PDA supports the following aspects of PCI driver programming:
Interrupting
INTx
MSI
Basic Address Registers (BARs)
Direct access
Protected access
Optimized memcpy routines
Direct Memory Access (DMA)
Persistent memory allocation of large buffers
Userspace buffer registration
DMA buffer sharing
Scatter/Gather lists
ALICE
HLT ALICE
CBM

Device Drivers

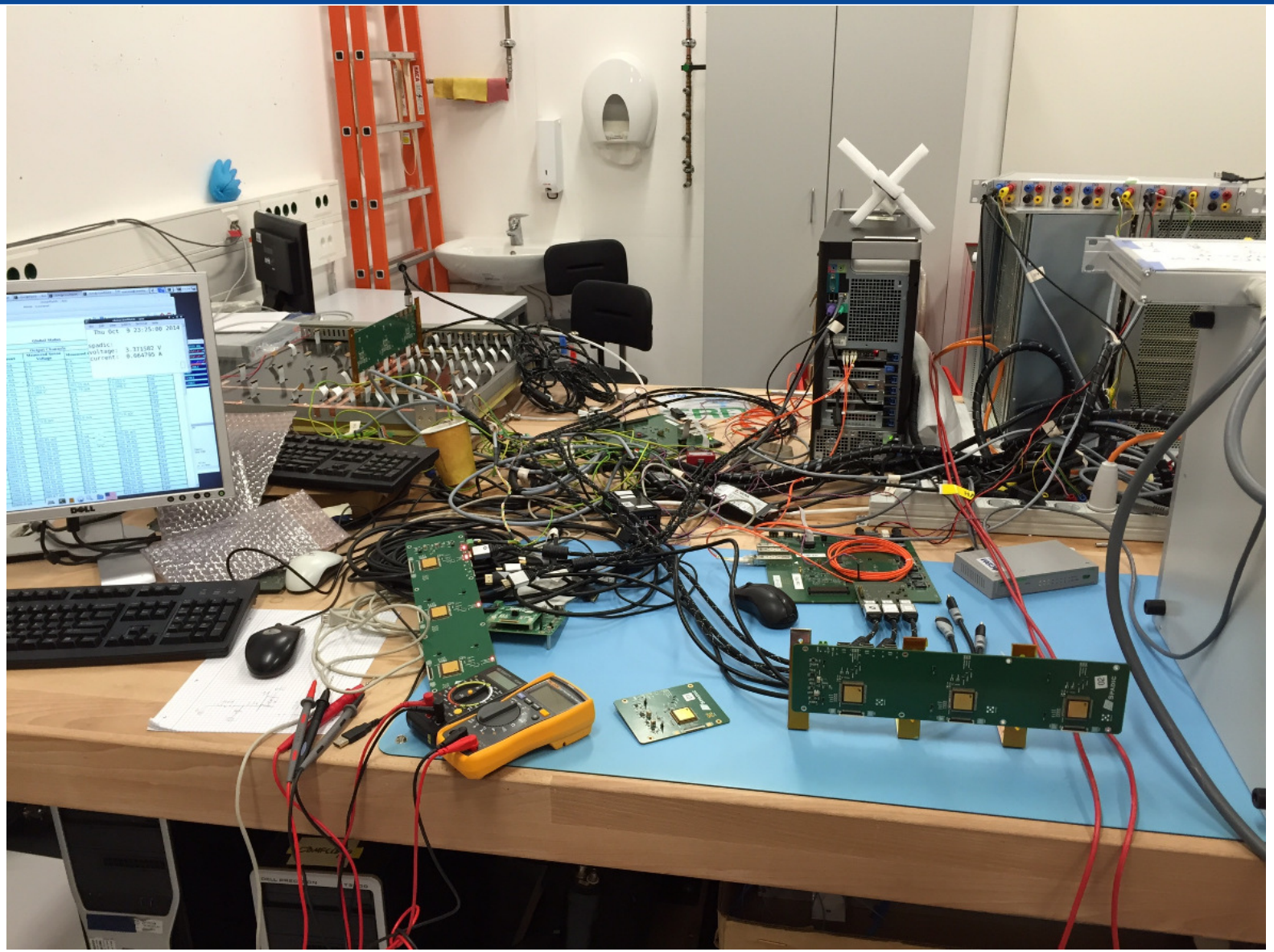


Device Drivers Development

FIAS Frankfurt Institute for Advanced Studies



Device Drivers Development



Test Driven Development

Application Code:

1. Write a test
2. Implement application until test complies
3. goto 1

Test Driven Development

Application Code:

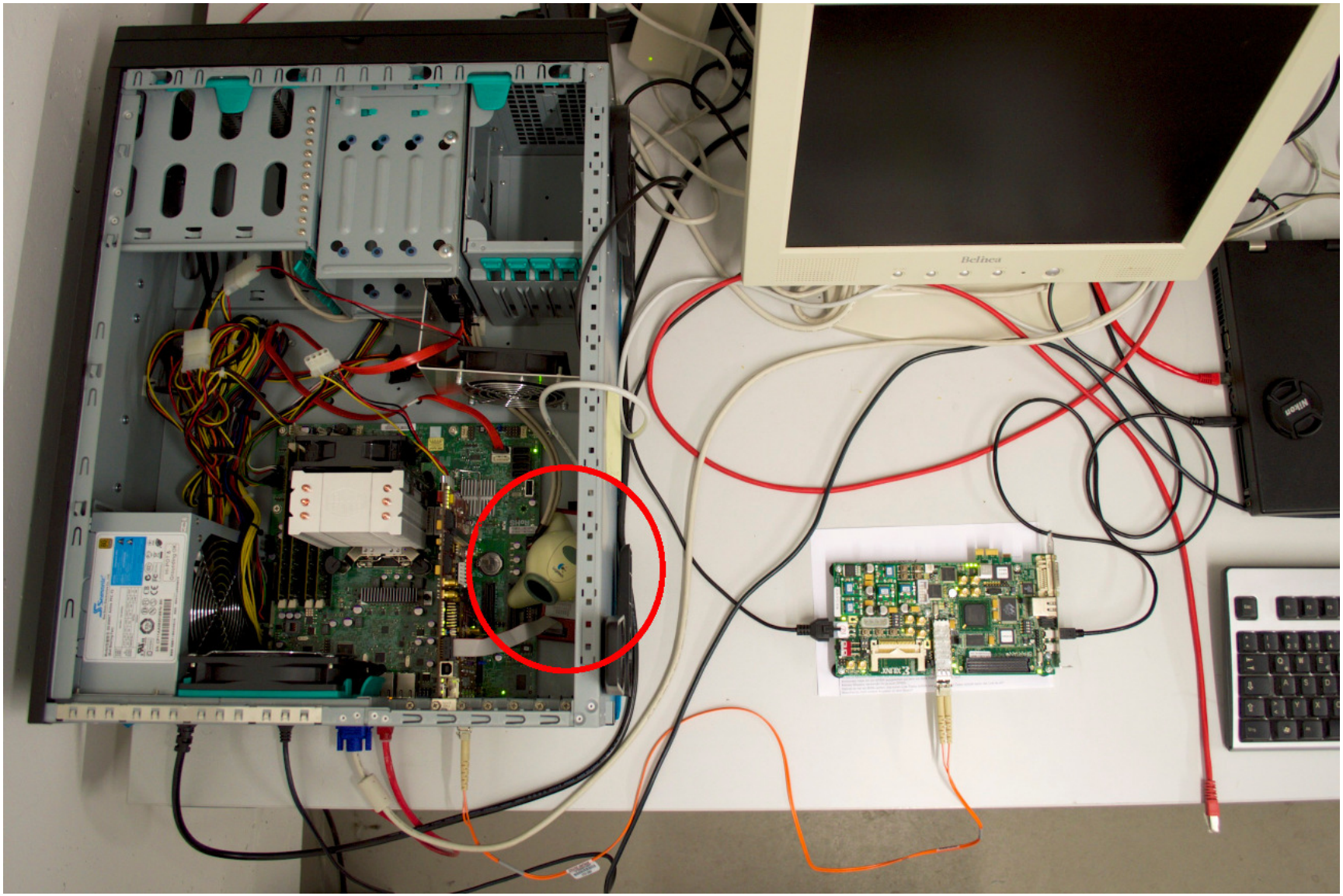
1. Write a test
2. Implement application until test complies
3. goto 1

Driver Code:

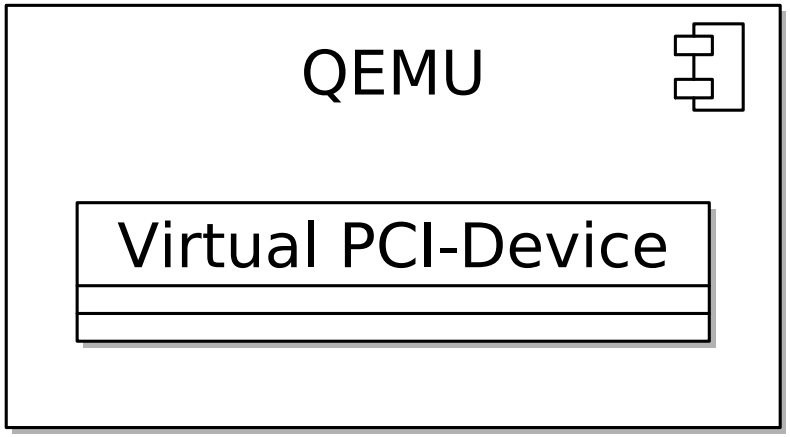
1. Write a test
2. Implement driver until you think that the test could comply
3. Run the test
4. Run into the basement for reseting your test machine
5. goto 2

Check Test Results?

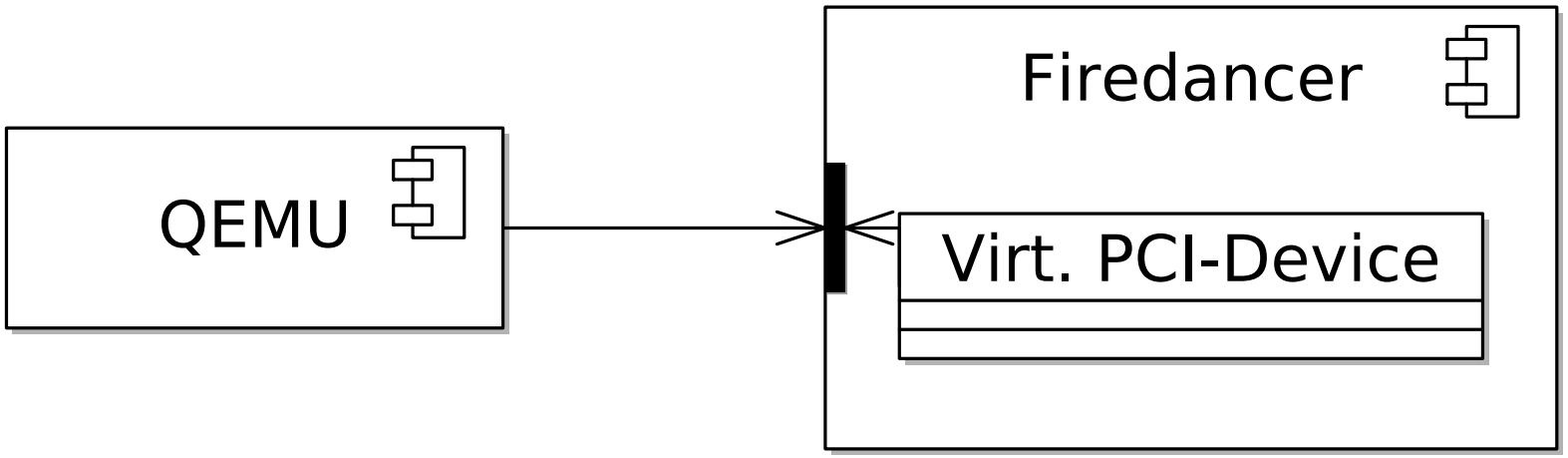
FIAS Frankfurt Institute for Advanced Studies



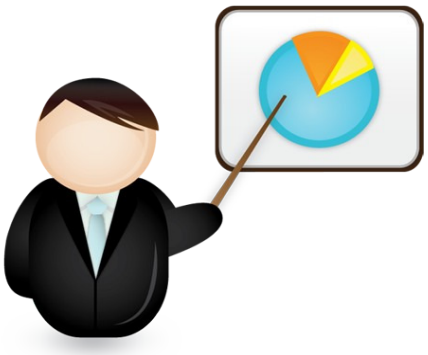
QEMU/KVM



QEMU/KVM



Use-Cases

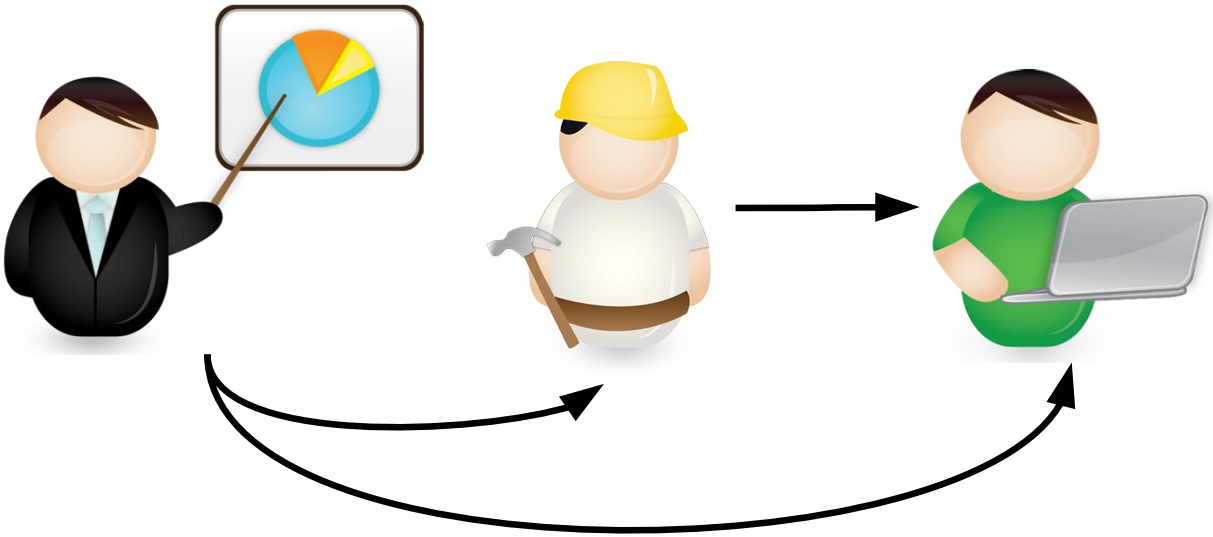


Hardware Software Codesigner

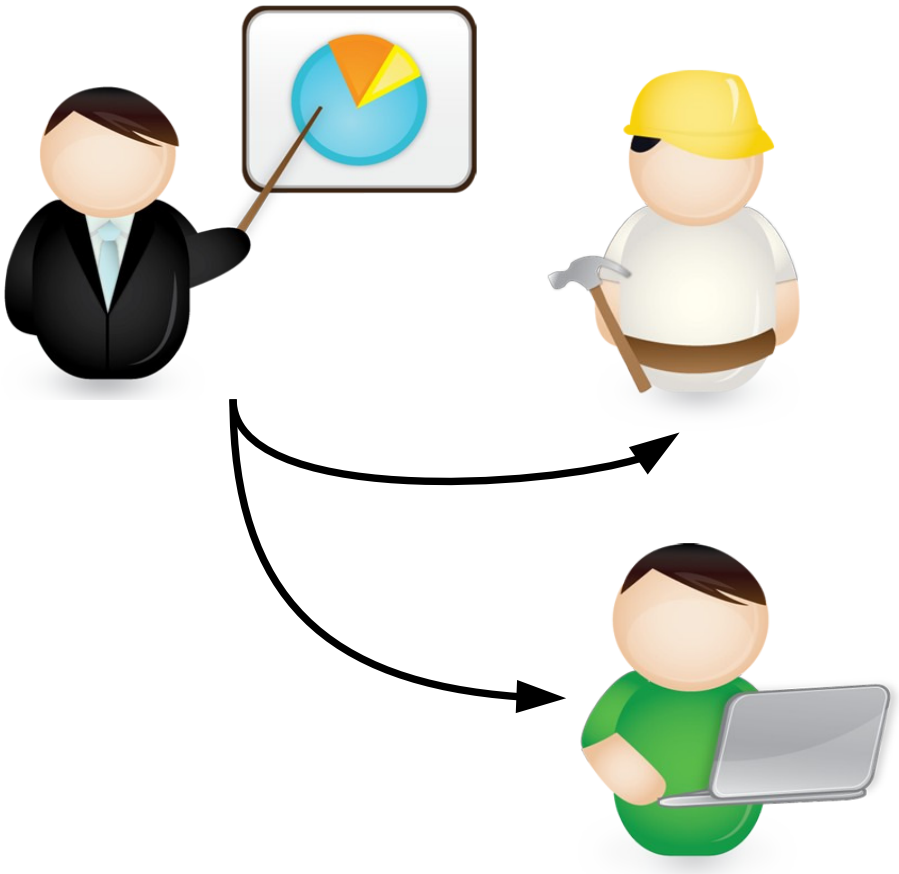
Hardware Engineer

Driver Developer

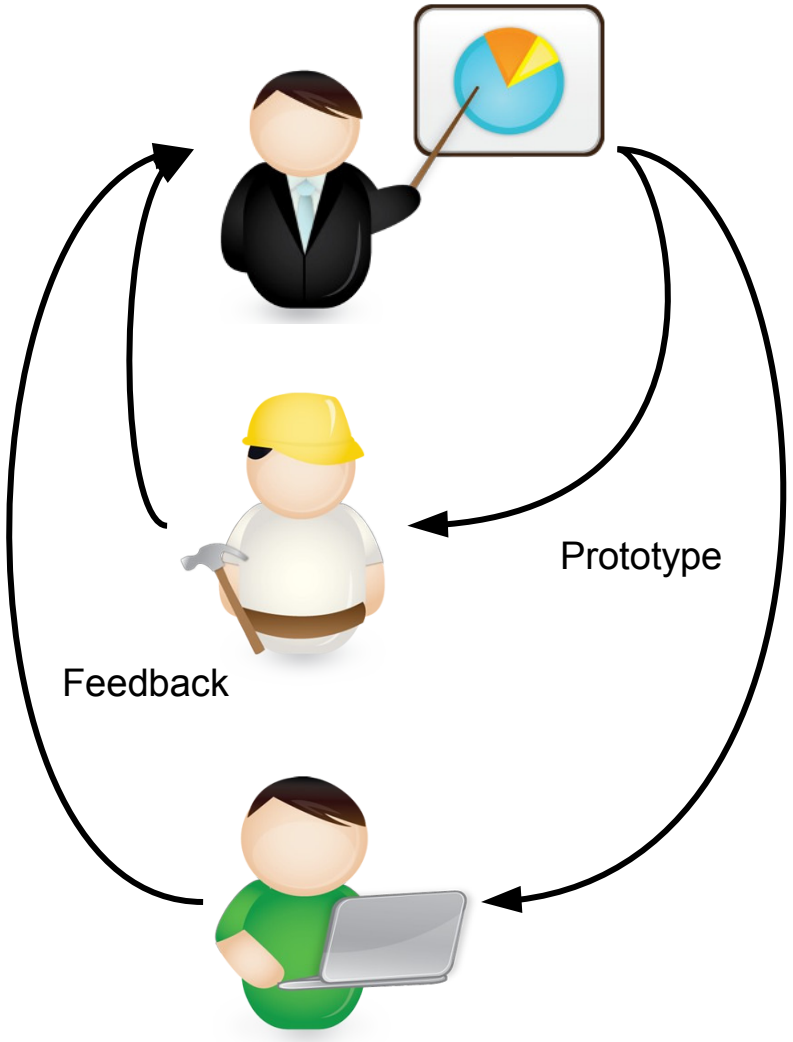
Use-Cases



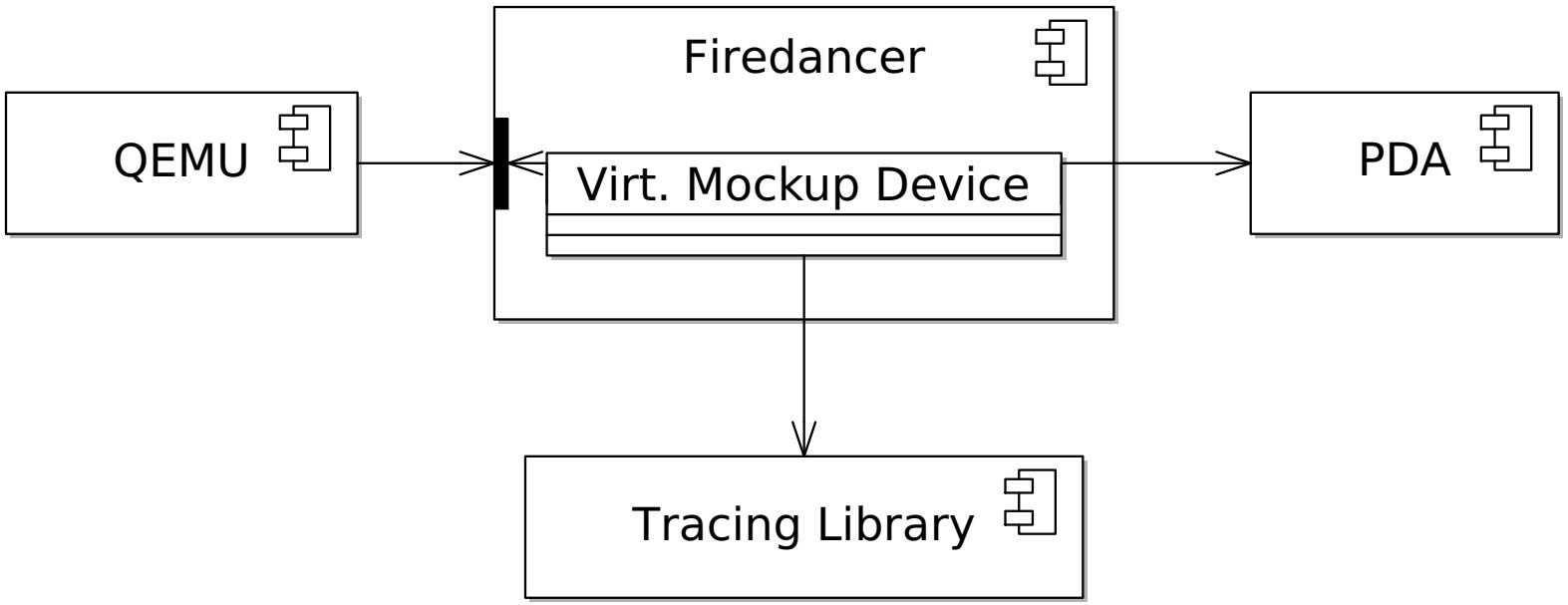
Use-Cases: Independent Deployability



Use-Cases: Rapid Prototyping



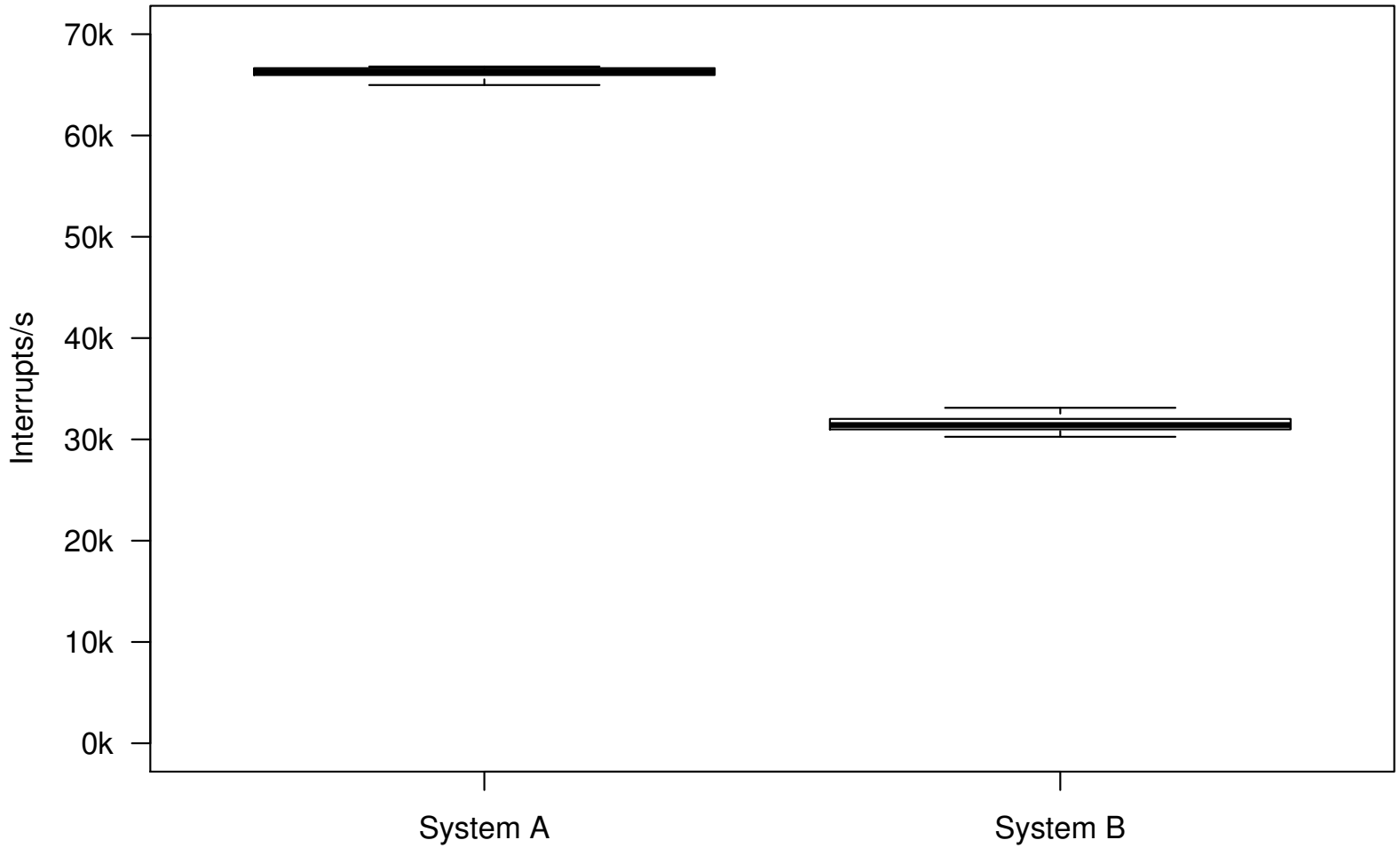
Use-Case: Low Level Debugging



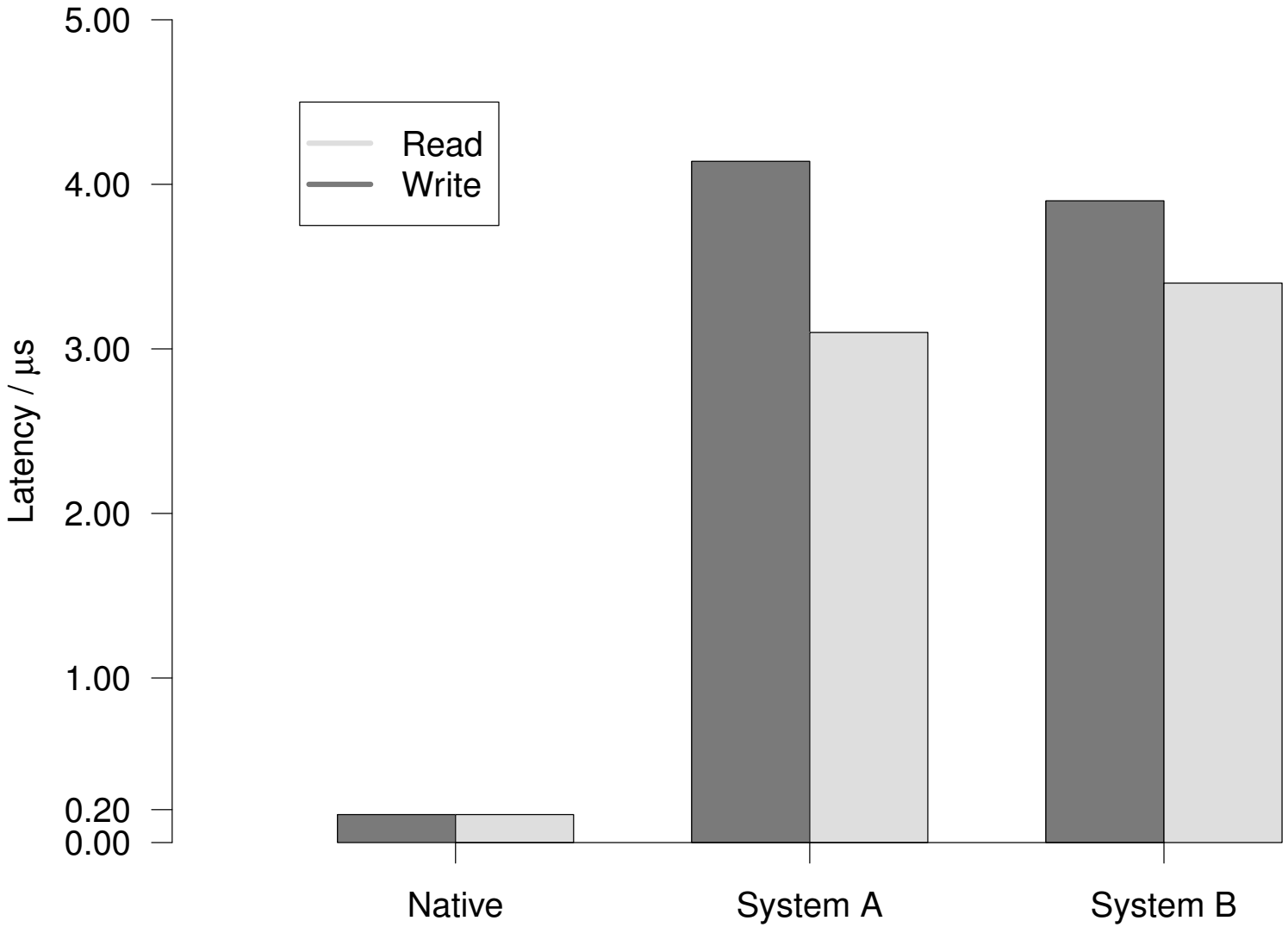
Test Systems

	System A	**System B**
CPU	Intel Xeon E5-2640	AMD FX-8320
CPU Clock	2.50 GHz	3.50 GHz
# Sockets	2	1
# Cores	12	8
Mainboard	SM X9DRG-QF	GA-78LMT
RAM Type	DDR 3 1333 MHz	DDR 3 1600 MHz
RAM Size	128 GiB	16 GiB
RAM Channels	4 (16 slots)	2 (4 slots)

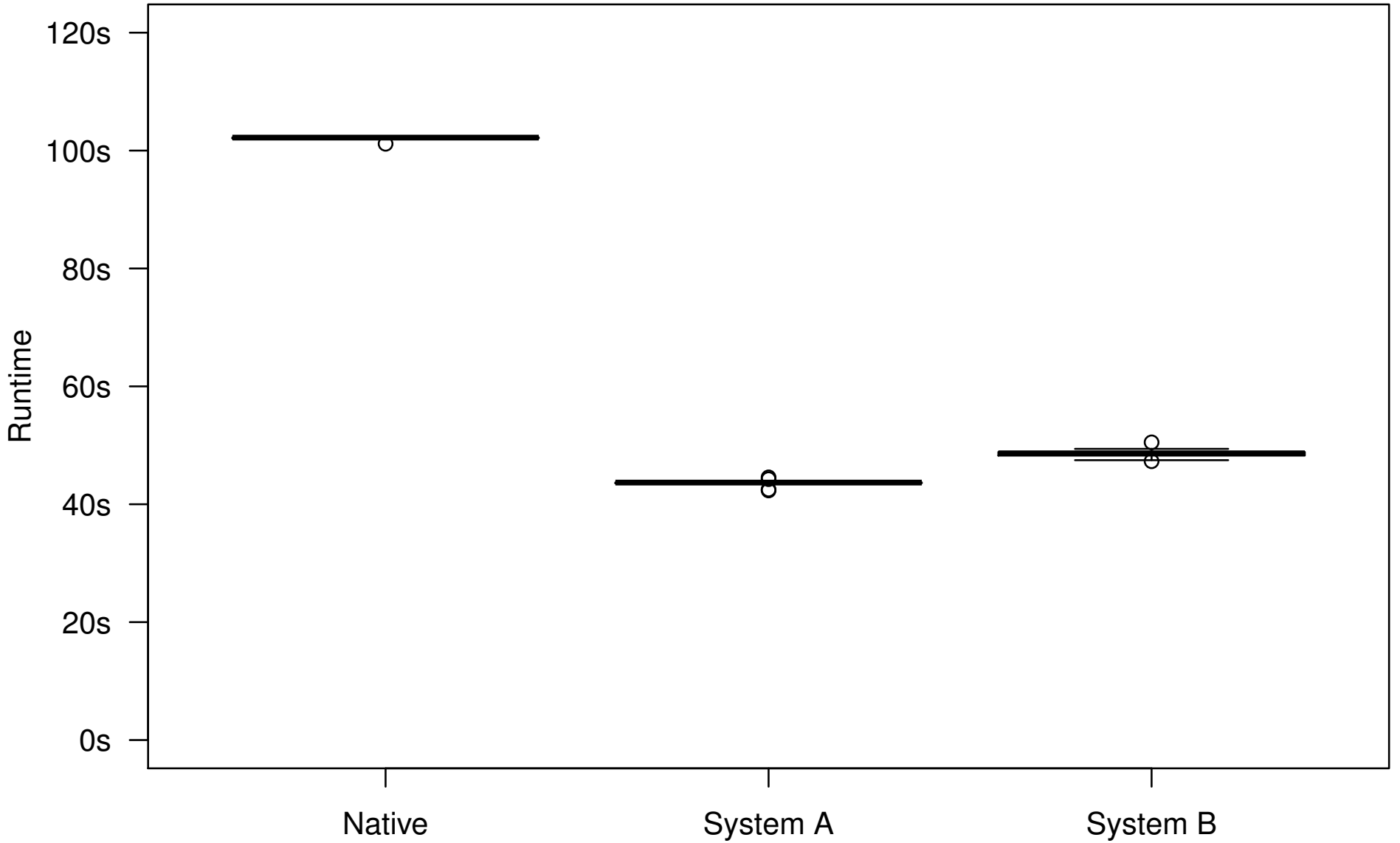
Interrupt Performance



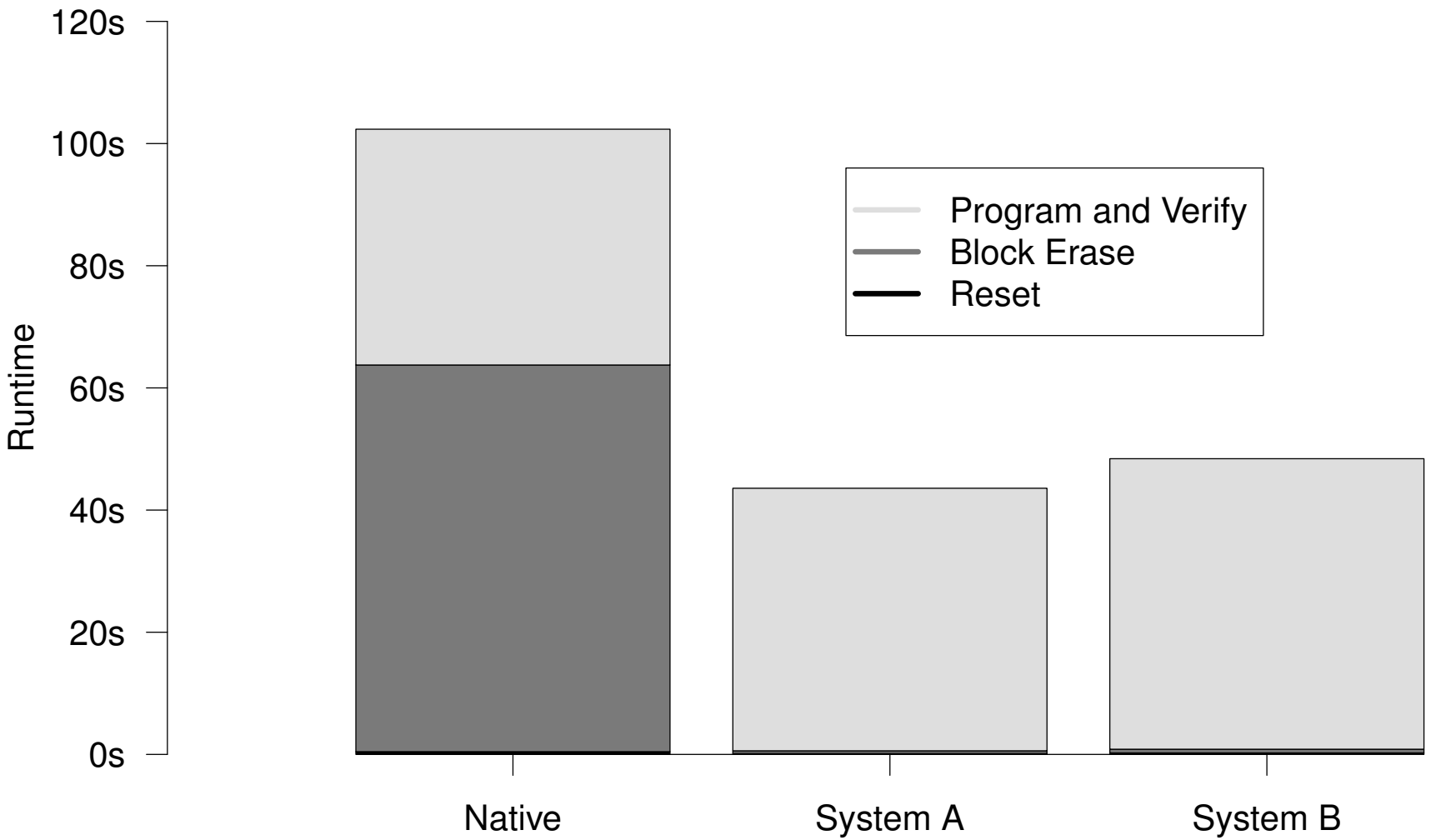
BAR Access Latencies



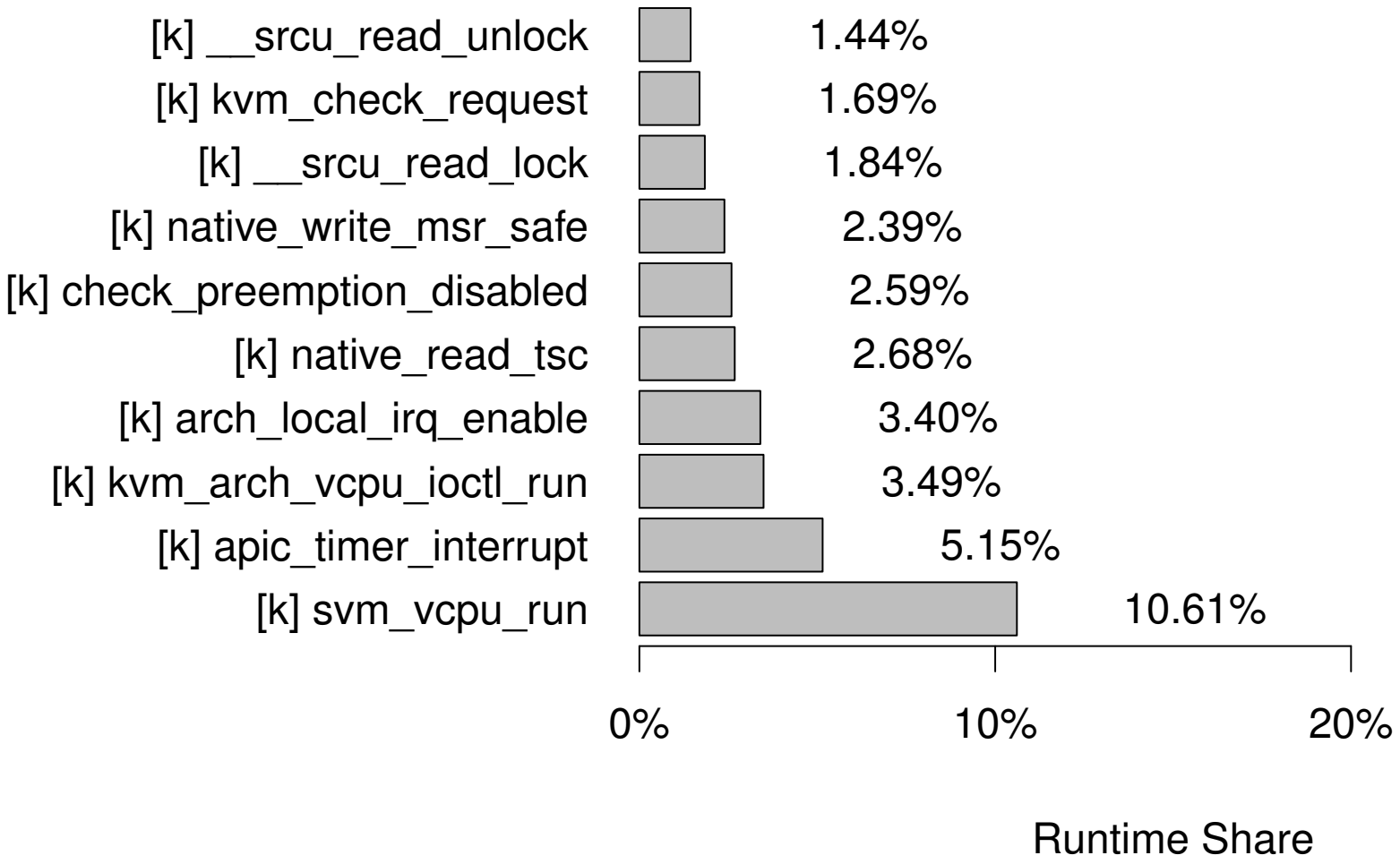
Flash Speed



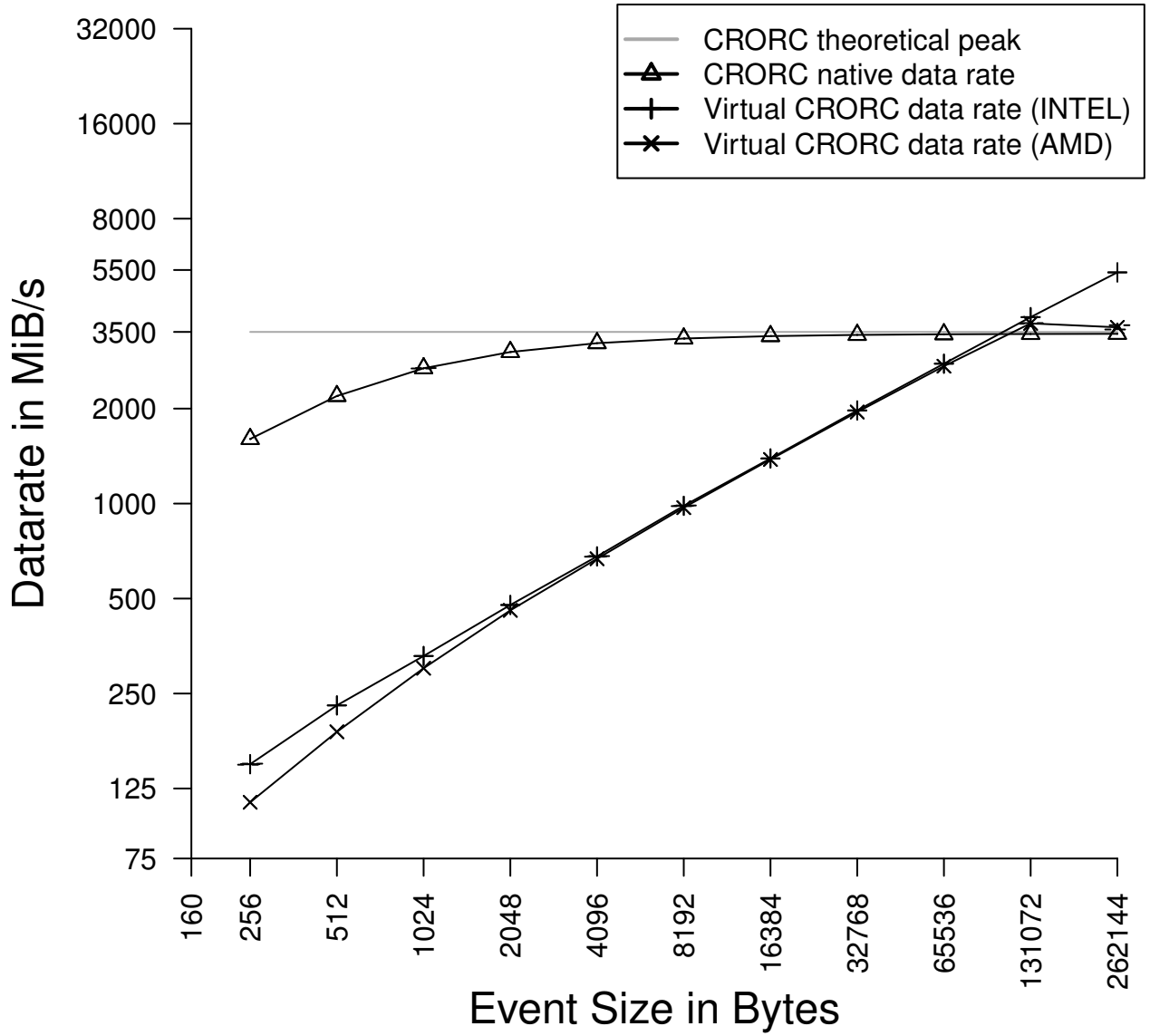
Flash Speed


120s
100s
80s
60s
40s
20s
0s
Runtime
Program and Verify
Block Erase
Reset
Native
System A
System B

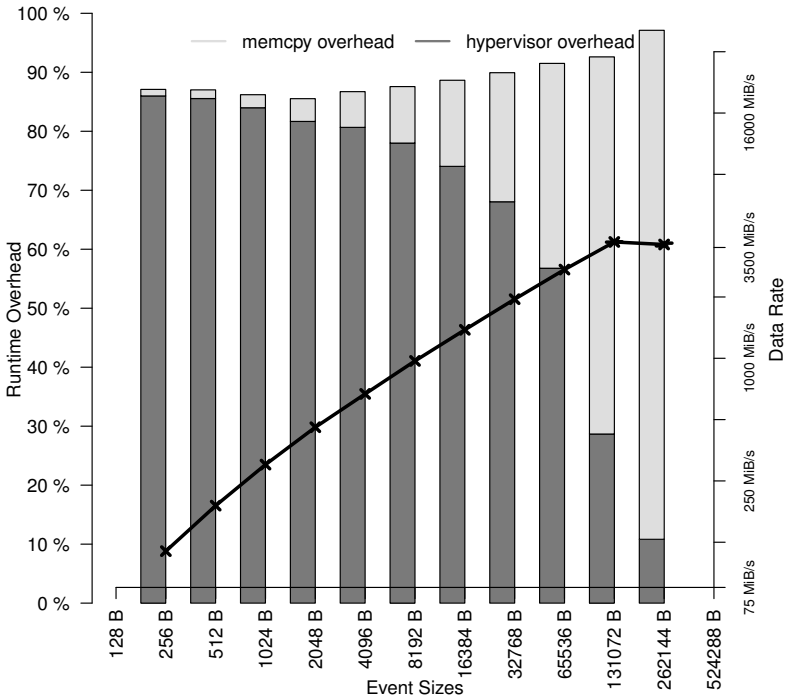
BAR Access Overhead

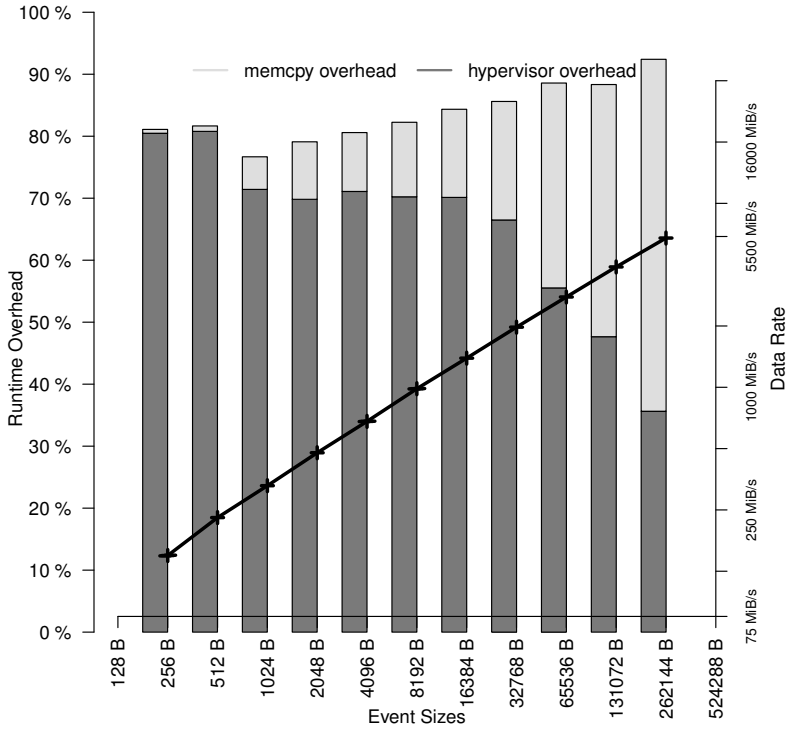


DMA Performance



DMA Overhead





The End

FIAS Frankfurt Institute for Advanced Studies

Thanks for the attention!

Get the code under:
`http://tinyurl.com/lew6yjp`
or
goto `http://www.compeng.de`
click Projects → Programming → Firedancer