YARR

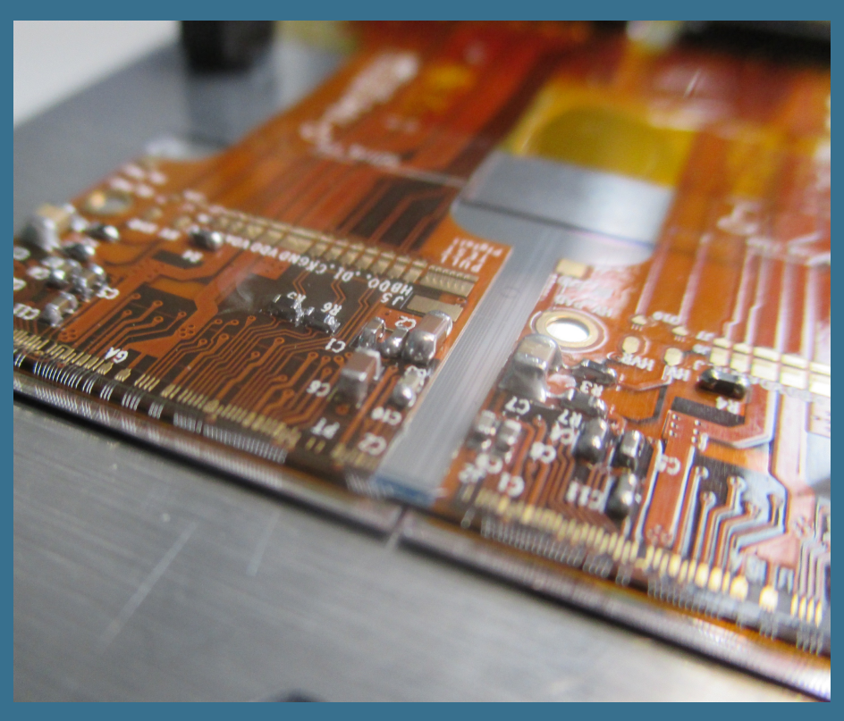
Yet Another Rapid Readout

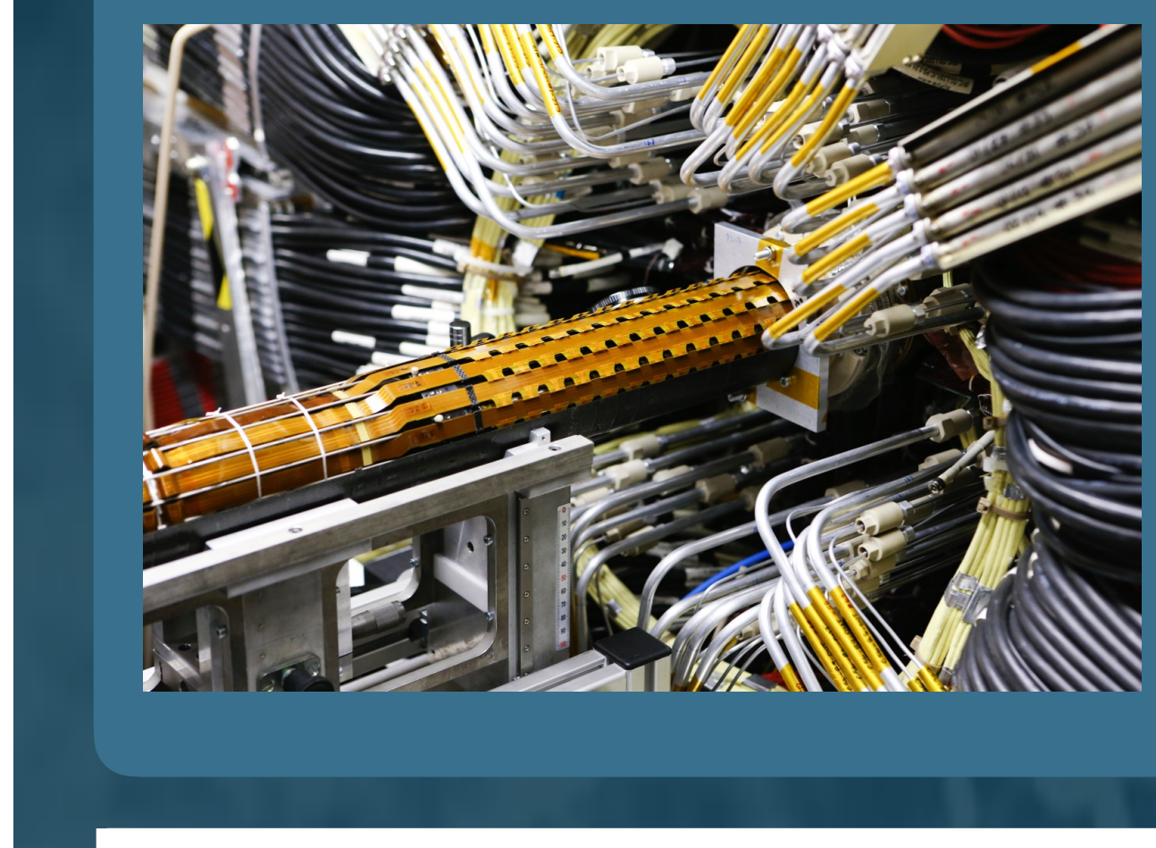
A PCIe based readout for current and future ATLAS Pixel modules

Timon Heim - Lawrence Berkeley National Lab

Introduction

The ATLAS Pixel detector uses custom ASICs to amplify and digitise the signals of the 92 million pixels that cover 1.75 m² of active area. To control and read out these ASICs custom readout electronics, typically utilising FPGAs, are used. The latest generation of readout chips, the FE-I4 (160Mb/s), has been integrated into YARR.





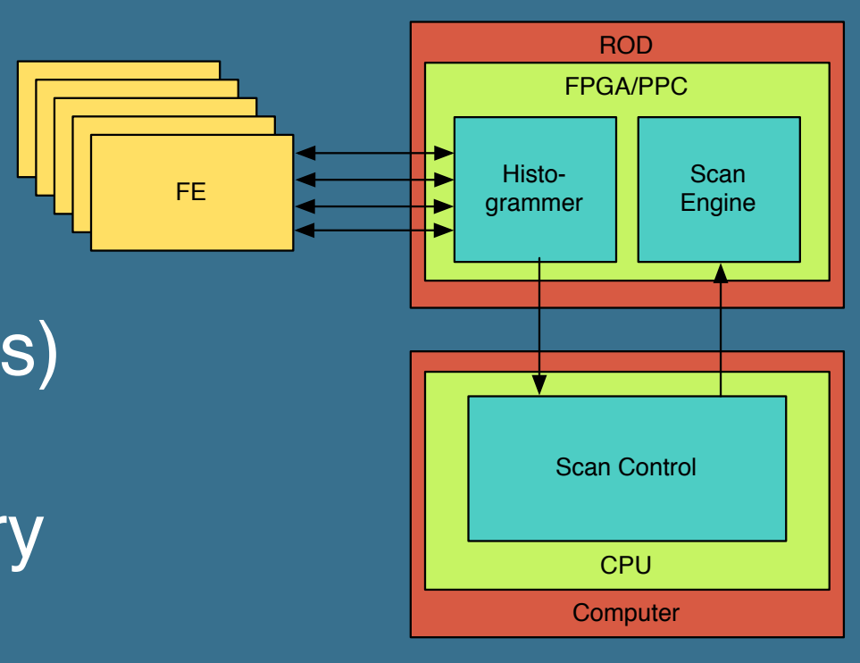
Readout requirements:

- High bandwidth & high channel-count serial communication
- Execution of complex calibration loops
- Local processing of calibration data

Concept

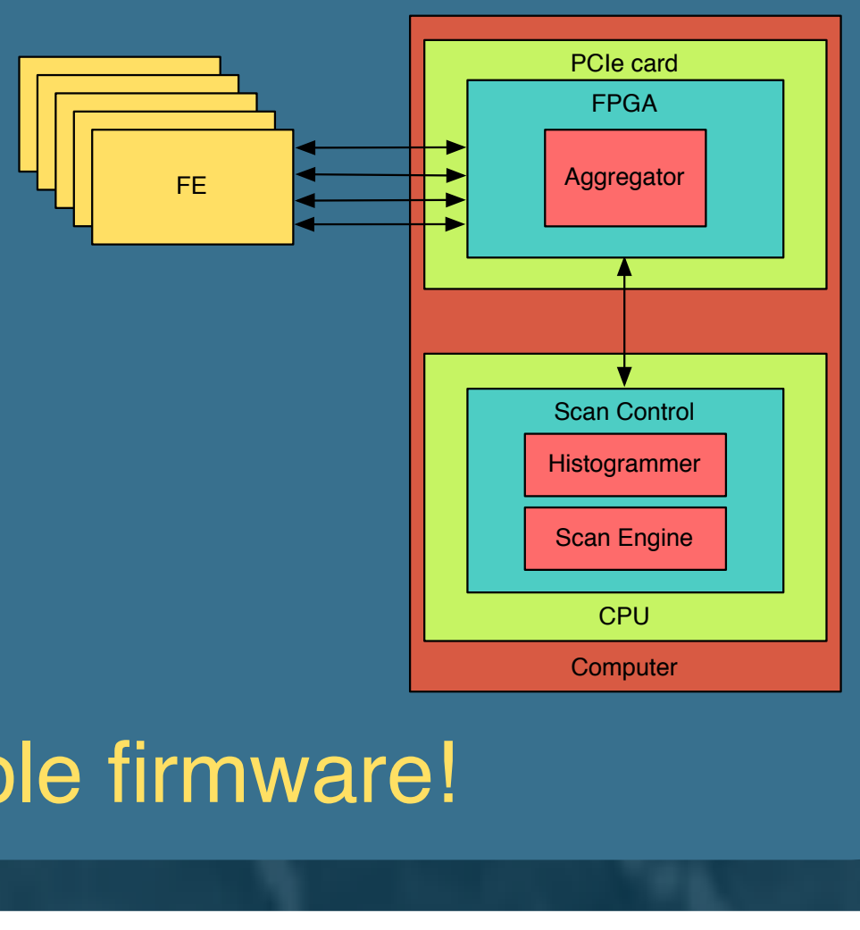
Traditional concept

- Low bandwidth link between host computer and Read-Out-Driver
- Requires in-FPGA preprocessing and lossy compression (histograms)
- Due to concept very complex firmware and software → high entry level for new developers
- Requires custom hardware



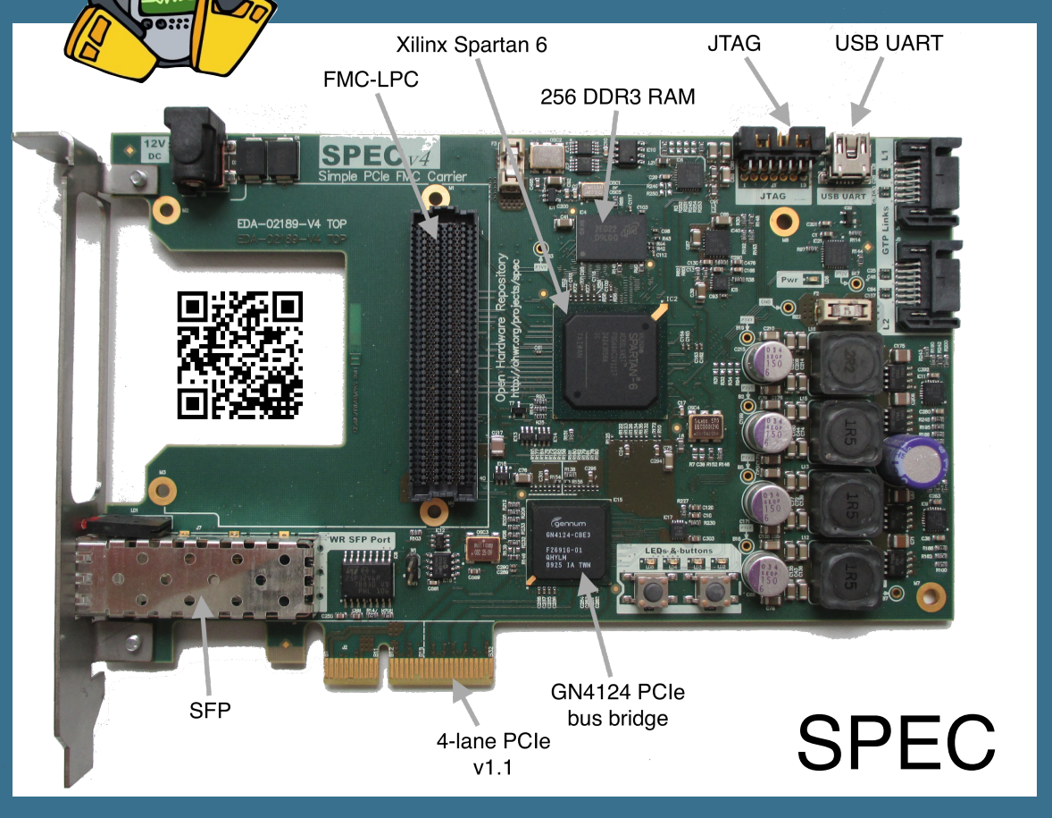
YARR concept

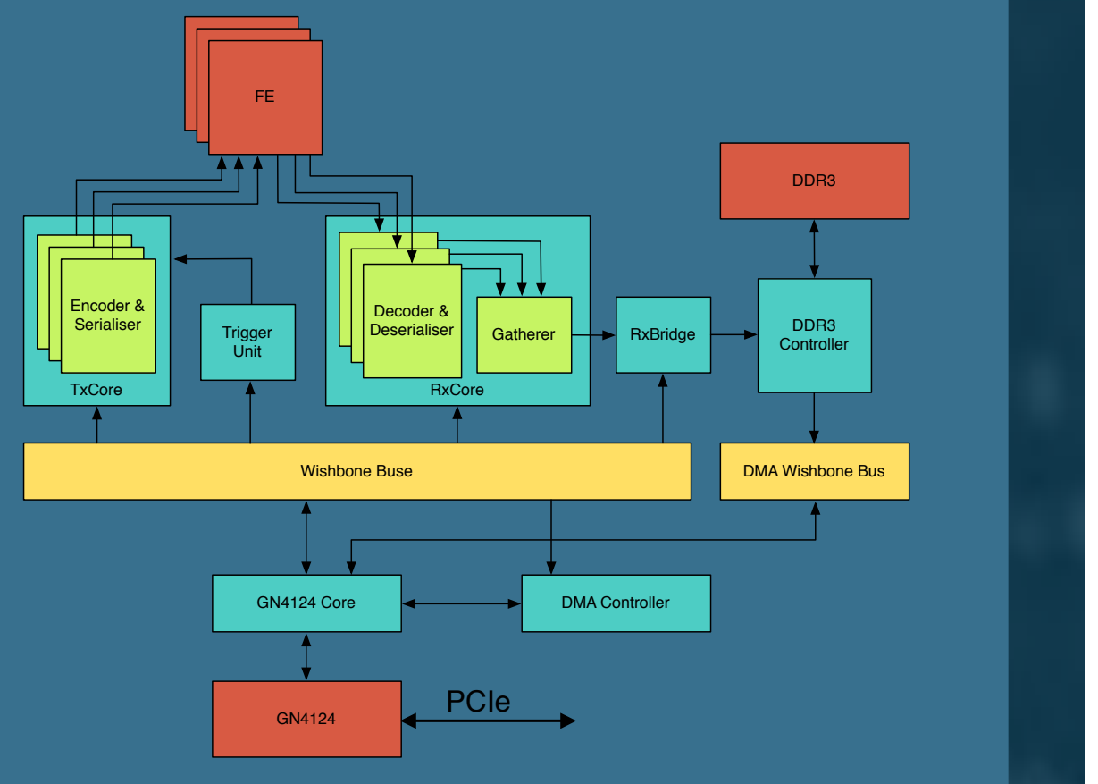
- Use high-speed link (PCIe) to transfer unprocessed data
- Perform all processing in software
- Lower entry level for new developers
- Can utilise COTS hardware

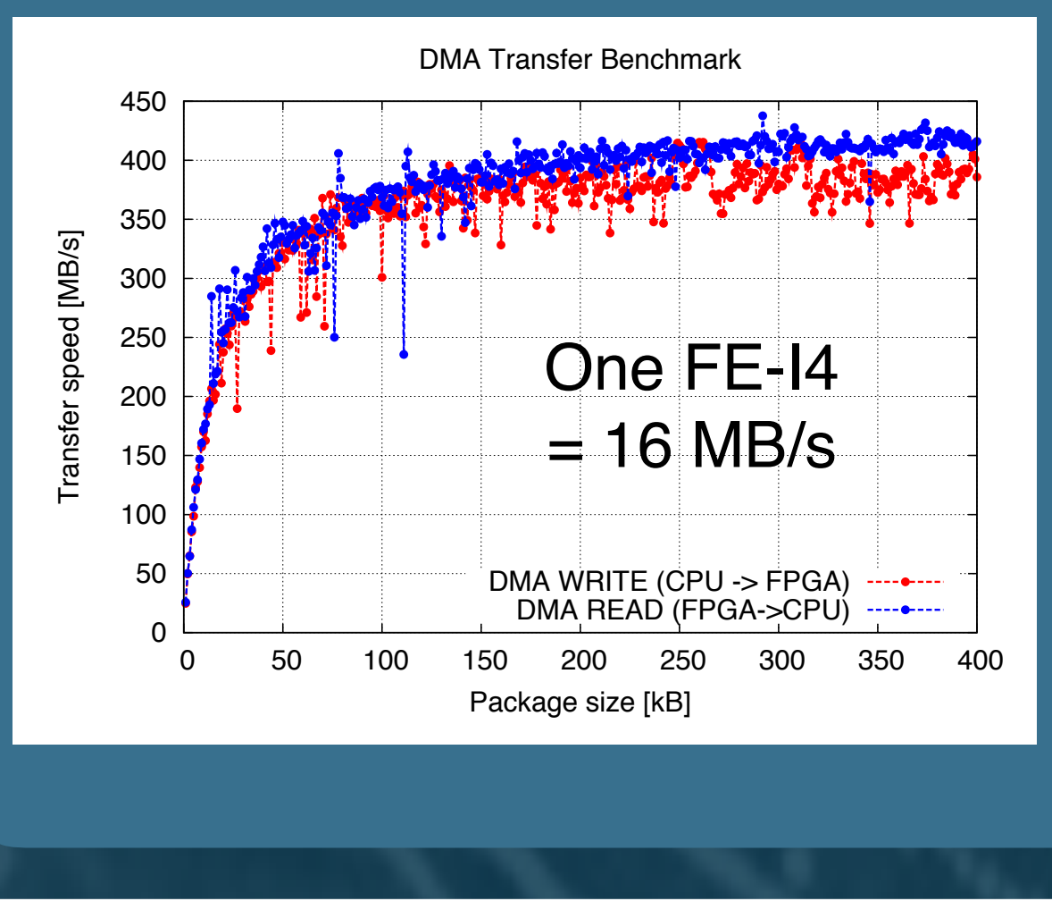


Smart software, simple firmware!

Hardware & Firmware







- COTS hardware with Xilinx Spartan 6 FPGA → cheap and available
- FMC port used for custom detector interface
- Buffer data in DRAM to achieve higher transfer speeds
- Keep it simple!

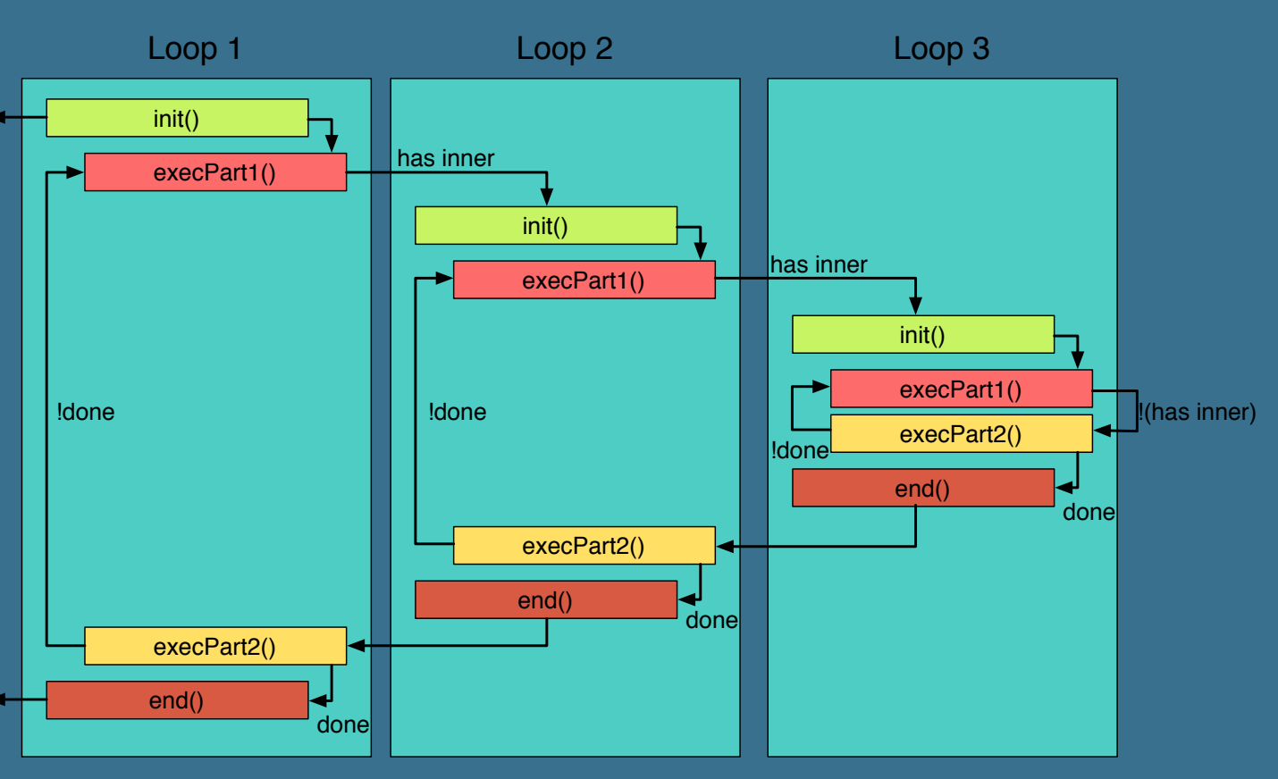
Scan Loop

Why loops?

- Inject test charge repeatedly to test response of pixels
- Cannot test all pixels at once: need to loop over portions of the pixel matrix
- Many scans involve looping over parameter range

Scan Engine

- Loops define actions performed on the Pixel module
- Nest loop actions in modular structure → High flexibility in constructing scans
- Scan engine executes nested loops



Do one thing and do it well

- Use search algorithms to speed up calibration and not brute force
- Search algorithms require input from higher analysis
- Don't perform analysis inside of scan loop, but rather feature interface for higher analysis to give feedback

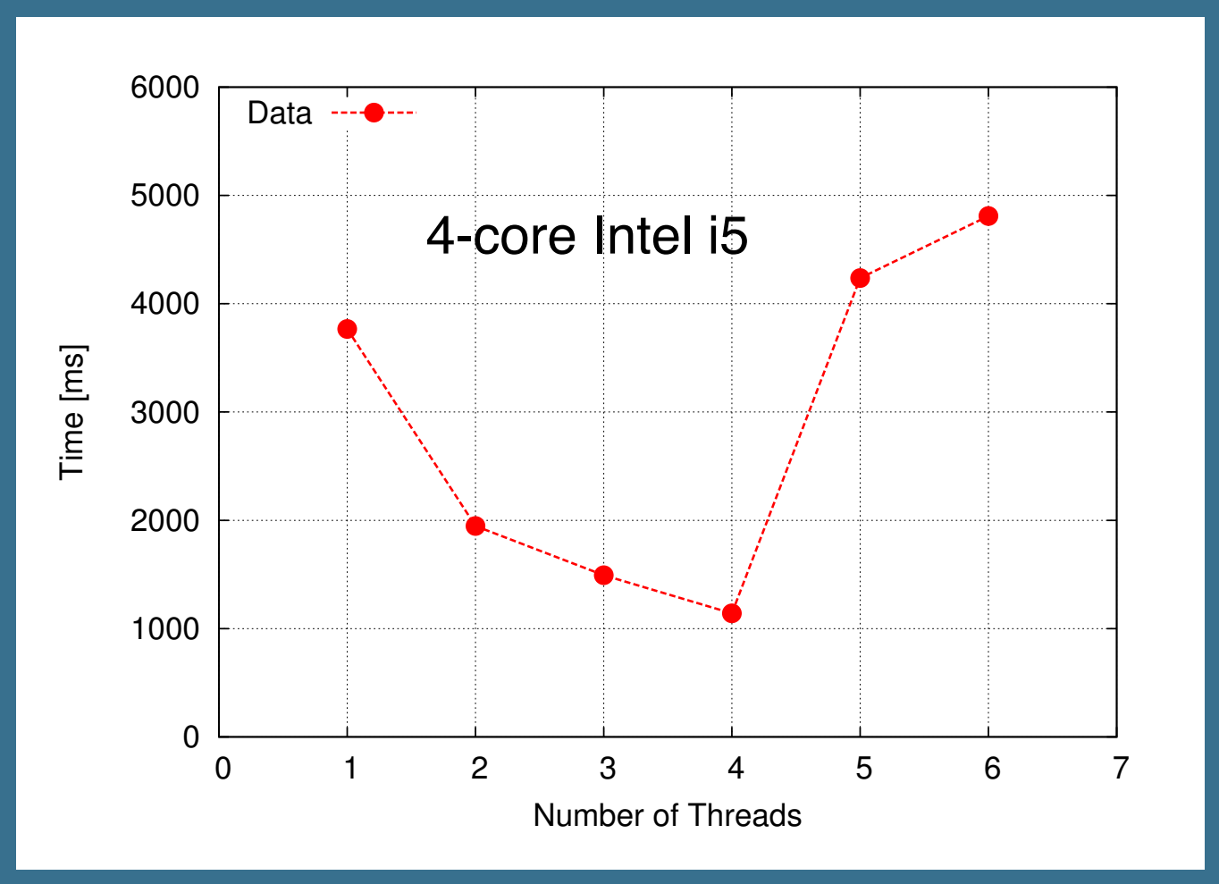
Feedback

Data Processing

Compete with FPGAs

- Convert binary data format from Pixel modules into standard data structures
- FPGAs excel at this job, but would need more bandwidth between FPGA and CPU
- Conversion → Perform same task many times on small data segments

→ Utilise modern multi core CPU architecture and parallelise data processing



Effect of parallel processing

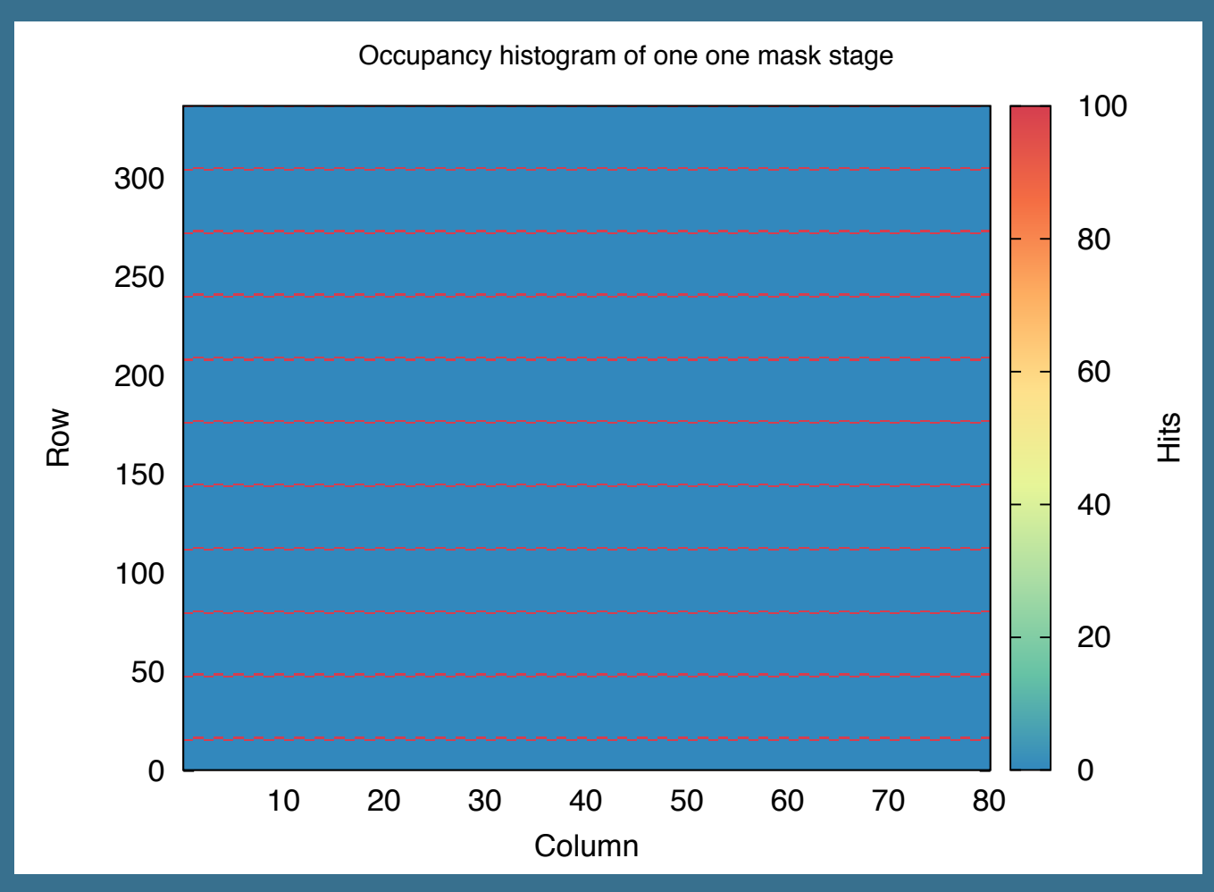
- Inter-thread communication lowers performance → Contain all necessary information in data containers
- This enables a processing step to be performed anywhere easily, e.g. FPGA, GPU or on a different computer

Split up data by module e.g. 3, as shown here

Histogramming

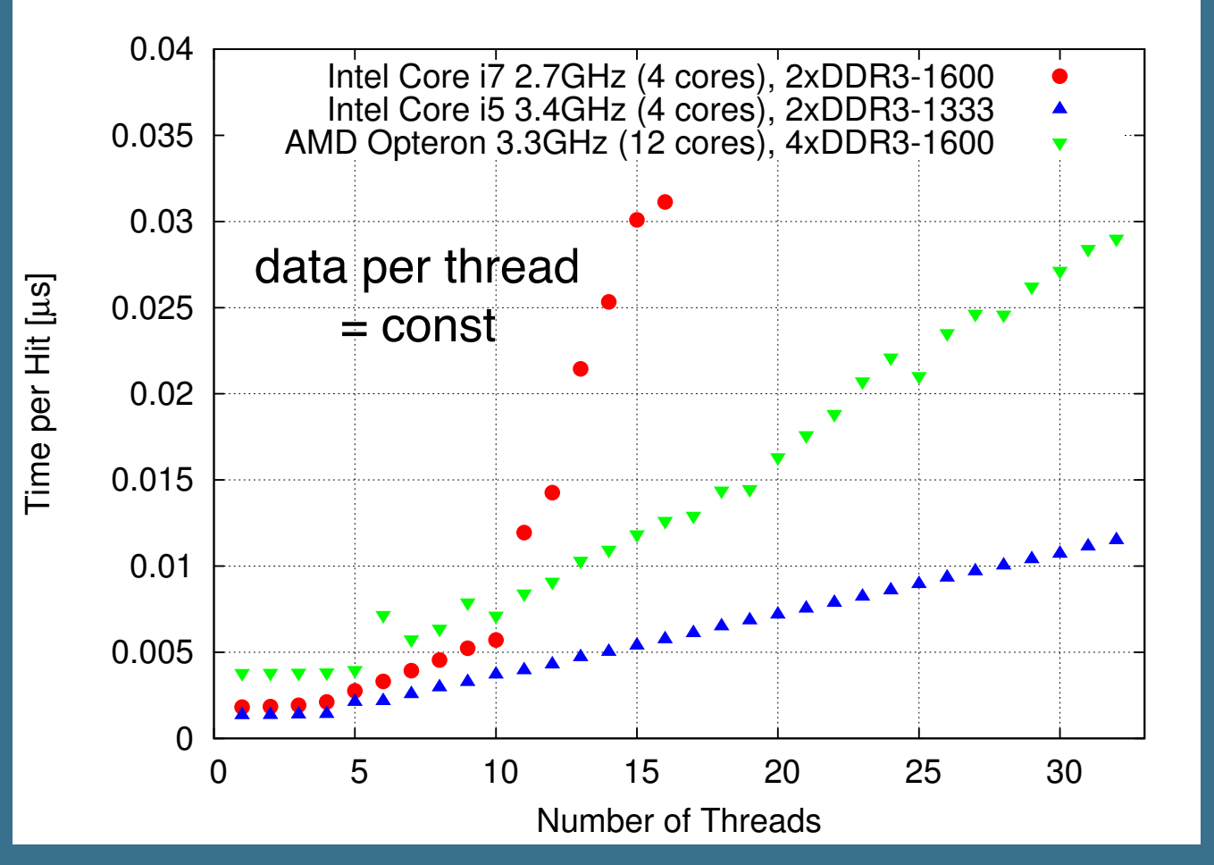
Why histograms?

- Lossy data compression → scan specific
- Plethora of memory in modern computers → not confined to certain number of histograms



Histogramming in DRAM

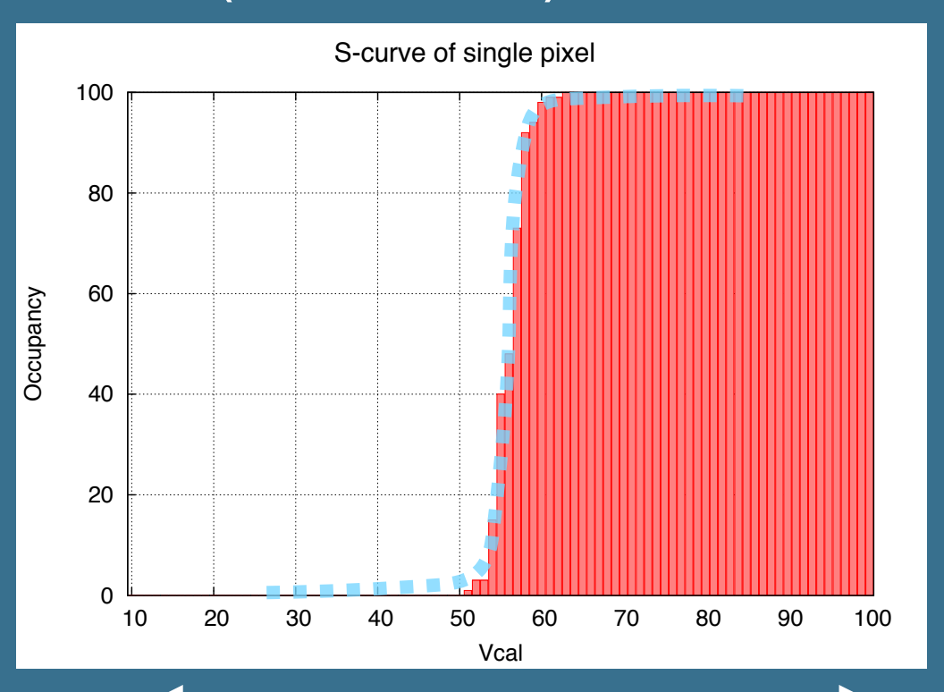
- Histogramming consists of many read-increment-write operations at random addresses → Usually done in SRAM
- Non-consecutive address switching slow in DRAM → Is it fast enough? Yes!



Analysis

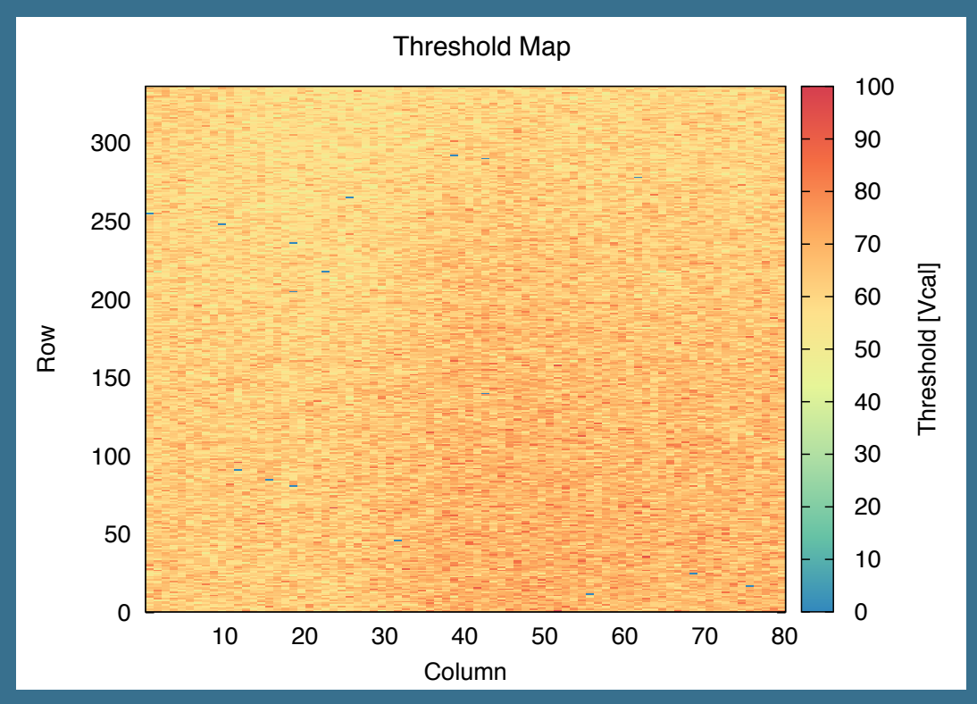
Example: FE-I4 Threshold Scan

$\mathcal{O}$(10k-100k) times



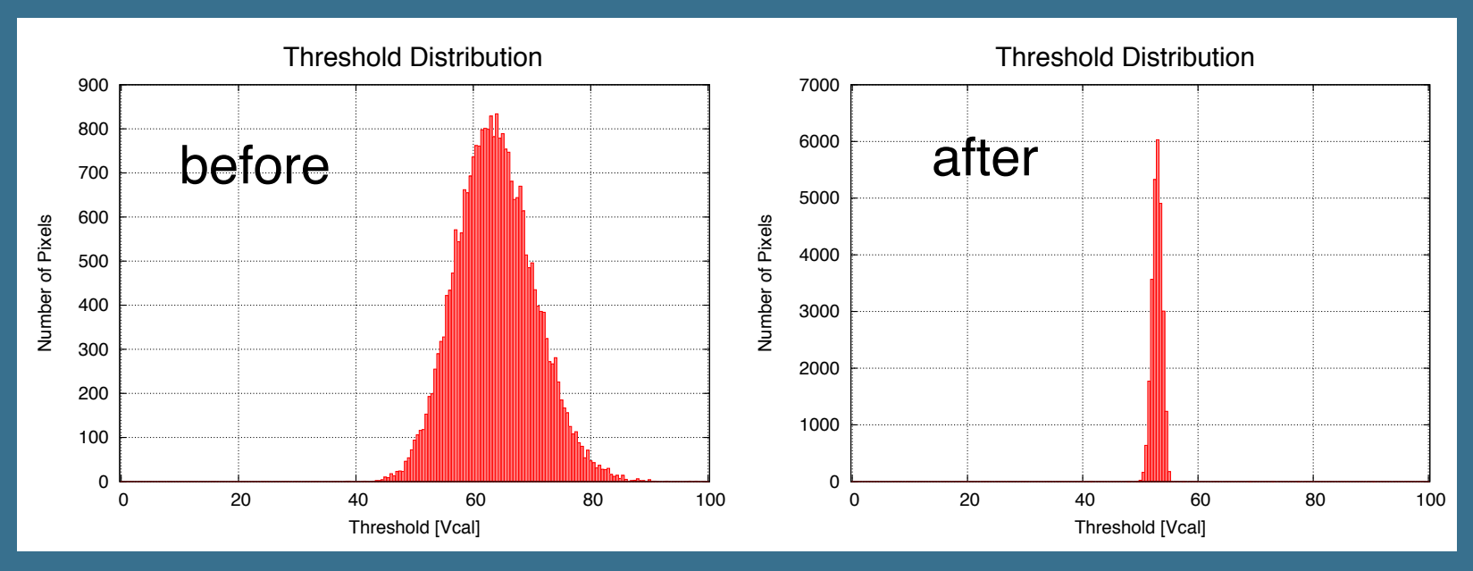
S-curve fit

Parameter range

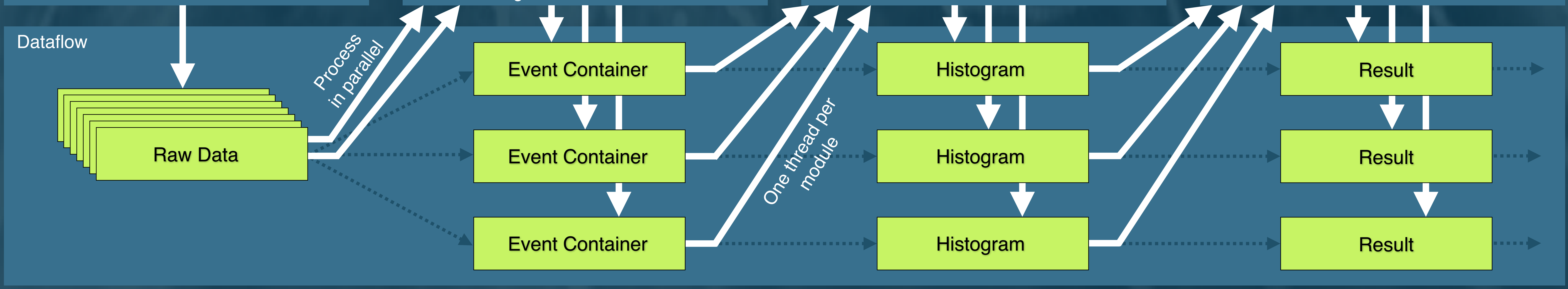


Calibration

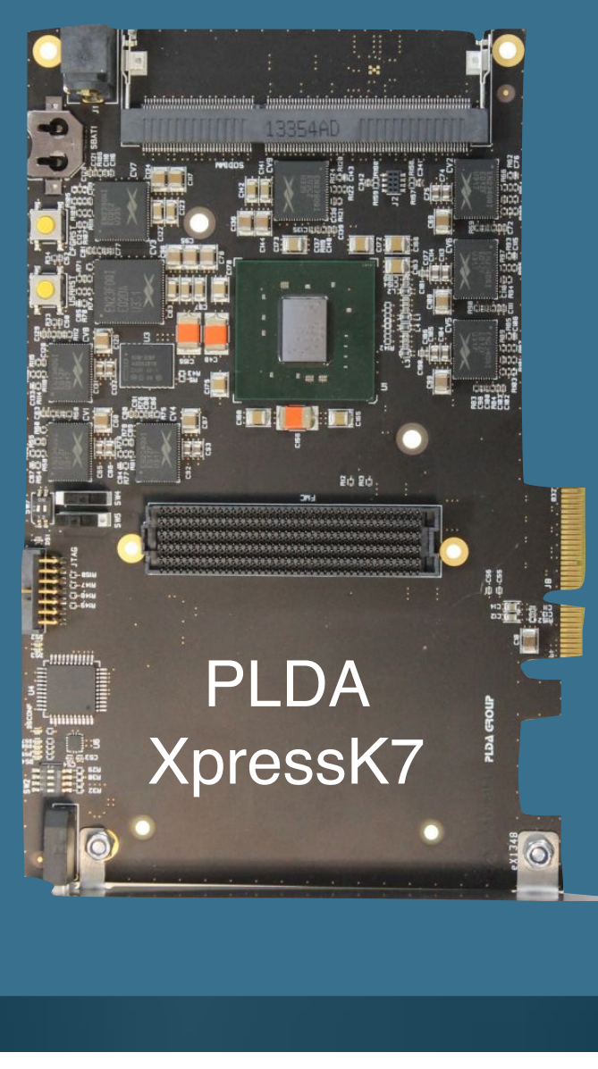
- Measure threshold
- Feedback to scan engine (too high/low)
- Analysis only measures threshold and does not adjust!



Feedback



In the future



New detector → New readout

- New detector modules will need readout links of up to 5Gbit/s
  - → Move to Xilinx Series 7 FPGAs to enable readout at 5Gbit/s
  - → PCIe 3.0 gives 16Gbit/s per lane with up to 16 lanes
- Is it still possible to perform all processing in software?

Smart software → Hardware agnostic?

- Traditionally different hardware solutions have been developed for lab testing and detector operation → Community and expertise split between two systems, creates manpower issues!
- Deploy YARR software for already existing Pixel detector hardware to prove that concept is hardware agnostic

GitHub

Email: theim@lbl.gov

ATLAS EXPERIMENT

U.S. DEPARTMENT OF ENERGY

UNIVERSITY OF CALIFORNIA

BERKELEY LAB

CERN

CHEP 2016 - San Francisco