Turning Petabytes into Physics: The case for scalable, user-centric software for next-generation science at LCLS

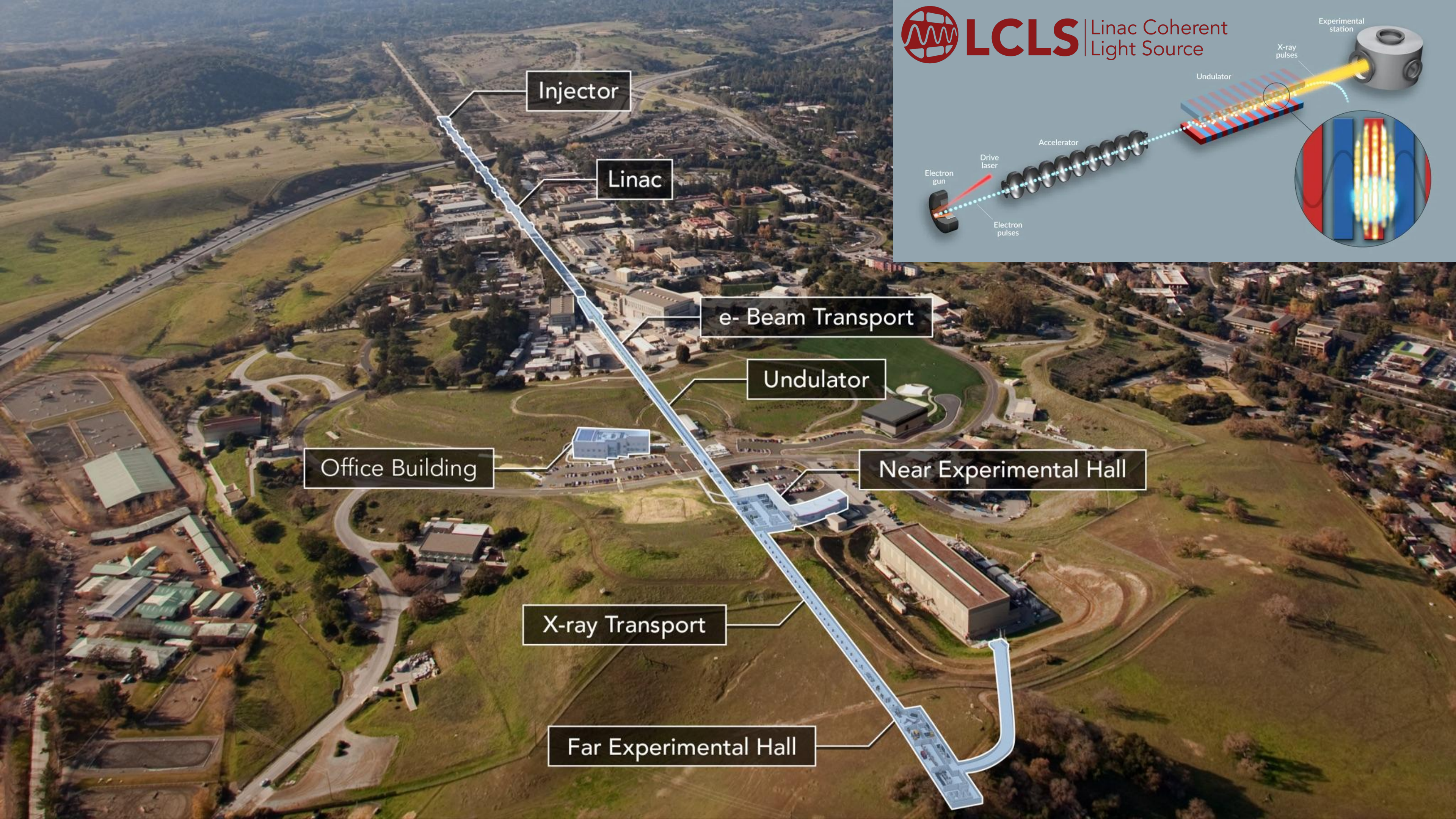
Jana Thayer/LCLS Data Systems Division Director on behalf of the Data Systems Team & friends
October 21st, 2025

Productive, Performant Software for Large Scale Scientific Data Analysis, SLAC

Many thanks to the people doing the honest work: Matthew Avaylon, Zhantao Chen, Ric Claus, Ryan Coffee, Dan Damiani, Dionisio Doering, Angelo Dragone, Gabriel Dohrliac, Vincent Esposito, Chris Ford, Kevin Dalton, Mikhail Dubrovin, Conny Hansson, Ryan Herbst, Wilko Kroeger, Xiang Li, Doris Mai, Stefano Marchesini, Valerio Mariani, Katalin Mecsesi, Riccardo Melchiorri, Silke Nelson, Chris O'Grady, Amedeo Perazzo, Frederic Poitevin, Omar Quijano, Lorenzo Rota, Amanda Shackelford, Thorsten Schwander, Murali Shankar, Monarin Uervirojnangkoorn, Matt Weaver, Seshu Yamajala, Cong Wang

Challenges of Large Experimental Facilities

SLAC

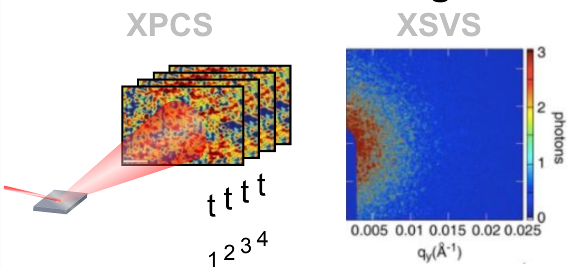


Injector
Linac
e- Beam Transport
Undulator
Office Building
Near Experimental Hall
X-ray Transport
Far Experimental Hall
LCLS Linac Coherent Light Source
Experimental station
X-ray pulses
Undulator
Accelerator
Drive laser
Electron gun
Electron pulses

Visualizing Fundamental Processes at Extreme Timescales

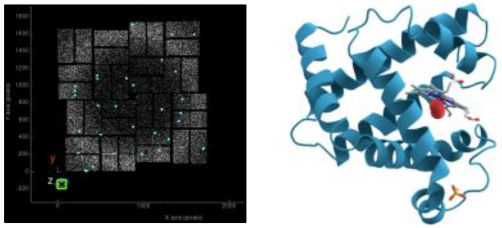
20+ experimental techniques with different workflows, & extreme throughput, and compute needs

Coherent Scattering



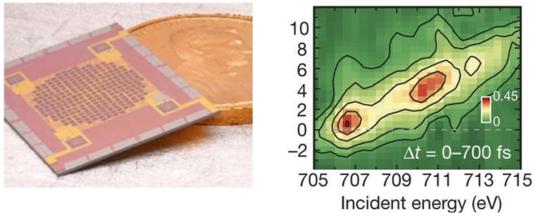
2024: 20 GB/s, 4 TF (reduction), 34 TF (analysis)
2027: 80 GB/s, 34 TF (reduction), 270 TF (analysis)

Nanocrystallography



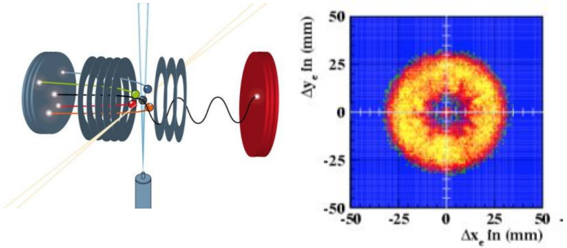
2024: 64 GB/s, 3 TF (reduction), 4 TF (analysis)
2028: 1.2 TB/s, 16 TF (reduction), 20 TF (analysis)

Resonant Inelastic Scattering



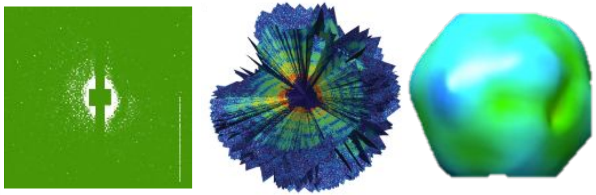
2023: 20 GB/s, 4 TF (reduction), 1 TF (analysis)
2025: 200 GB/s, 40 TF (reduction), 2 TF (analysis)

Coincidence Spectroscopy



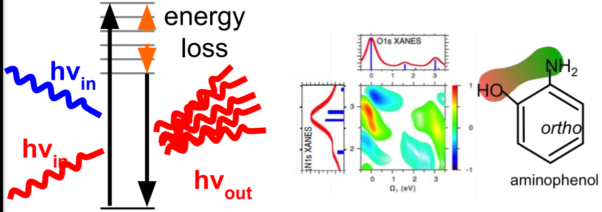
2024: 200 GB/s, <1TF (reduction), <1TF (analysis)

Coherent Imaging

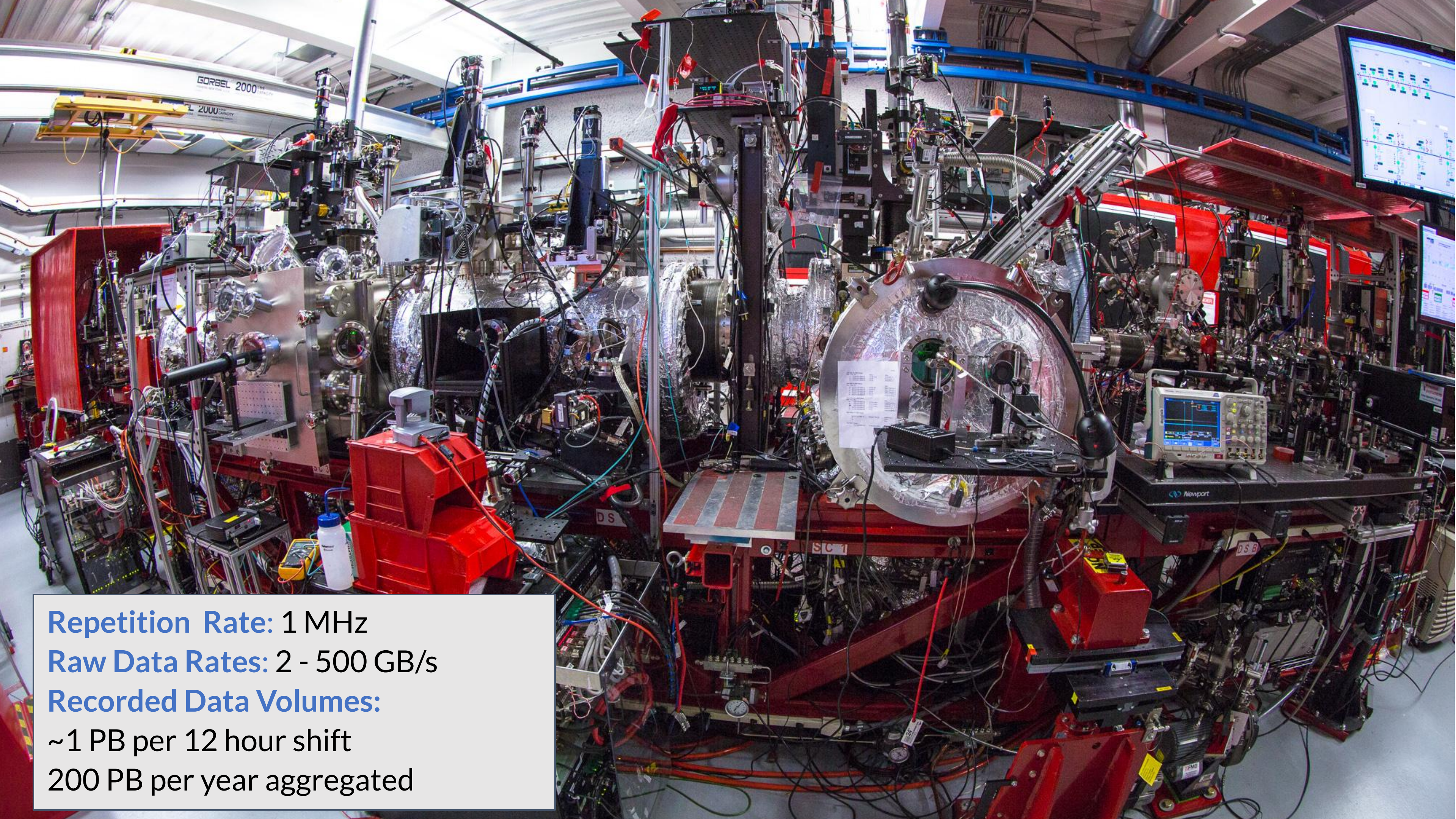


2024: 64 GB/s, 3 TF (reduction), 270 TF (analysis)
2027: 1.2 TB/s, 16 TF (reduction), 1340 TF (analysis)

Nonlinear Spectroscopy



2024: 20 GB/s, 3 TF (reduction), <1 TF (analysis)
2025: 80 GB/s, 16 TF (reduction), <1 TF (analysis)

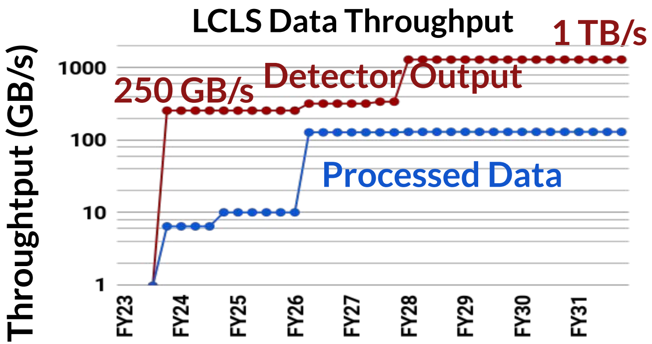


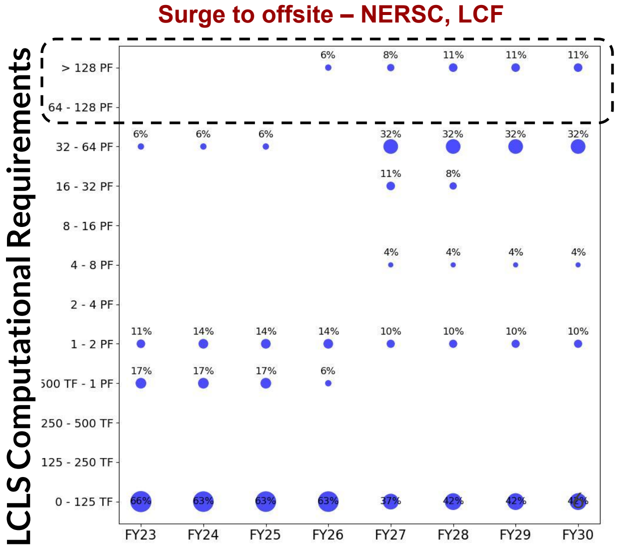
Repetition Rate: 1 MHz
Raw Data Rates: 2 - 500 GB/s
Recorded Data Volumes:
~1 PB per 12 hour shift
200 PB per year aggregated

LCLS-II Data System Drivers

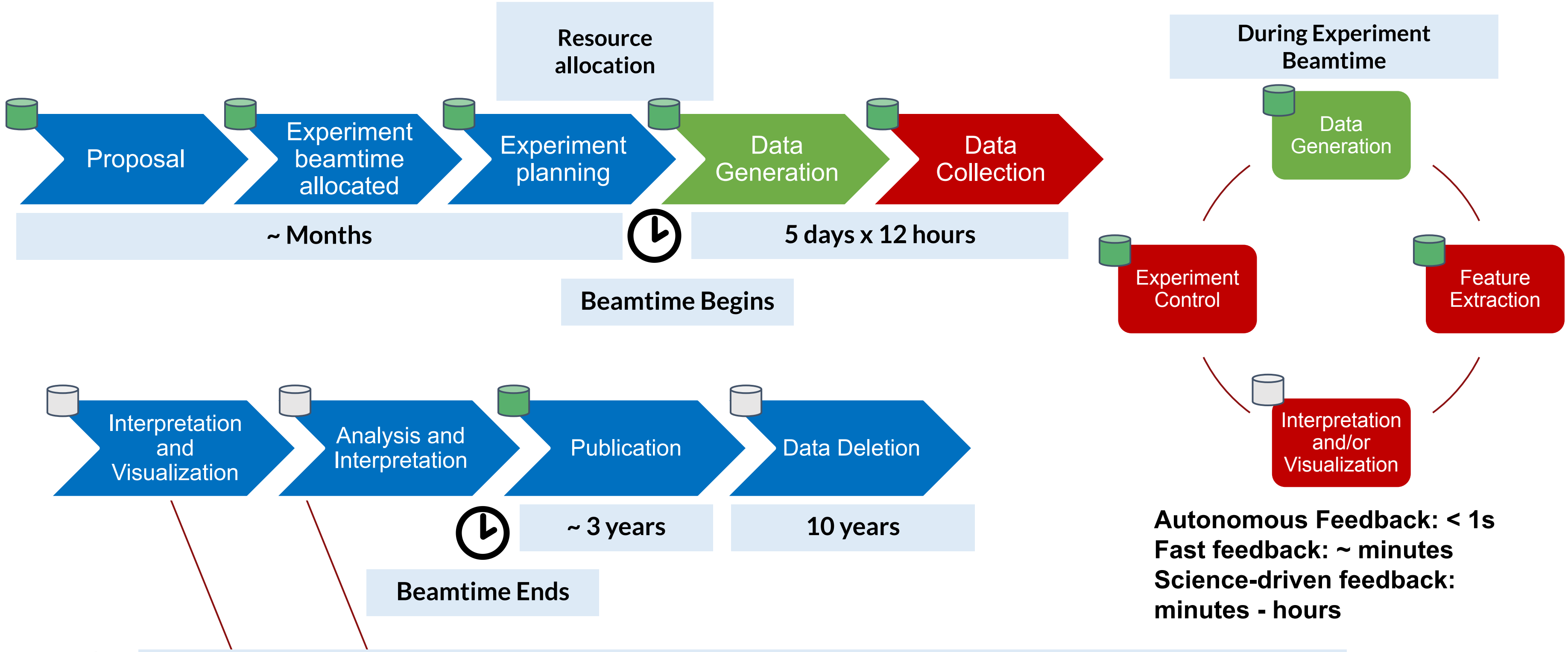
- **LCLS-II Upgrade:** greater data *velocity*, *volume*, and *complexity*
  - **Data Rates:** 120 Hz to 1 MHz (**10000x**)
  - **Raw Data Volumes:** 2 GB/s to 500 GB/s (**100x**)
  - **Recorded Data Volumes:** 2 GB/s to 50 GB/s (**10x**)
  - **Computational Requirements:** 80% ~1 PF, 20% ~1 ExaFLOP
- **Fast Feedback:** real-time analysis (sec/min) is essential to the users' ability to make informed decisions during experiments.
- **Variability:**
  - **Wide variety of experiments** with turnaround ~days
  - **Large dynamic range:** device readout 0.01 Hz - 1 MHz
  - **Data Complexity:** Variable length data (raw, compressed)
  - **Access patterns** to data vary by experiment and detector
  - Analysis is a mix of **tried-and-true** & **innovative techniques**
- **Time to Science:** **Development cycle** must be fast & flexible
- **No user left behind:** alleviate the pressure on users to gather resources to mount a significant computing effort.

Wide variety of experiments that need to modify analysis during experiments





Data Lifecycle Repeated for Hundreds of Experiments / Year

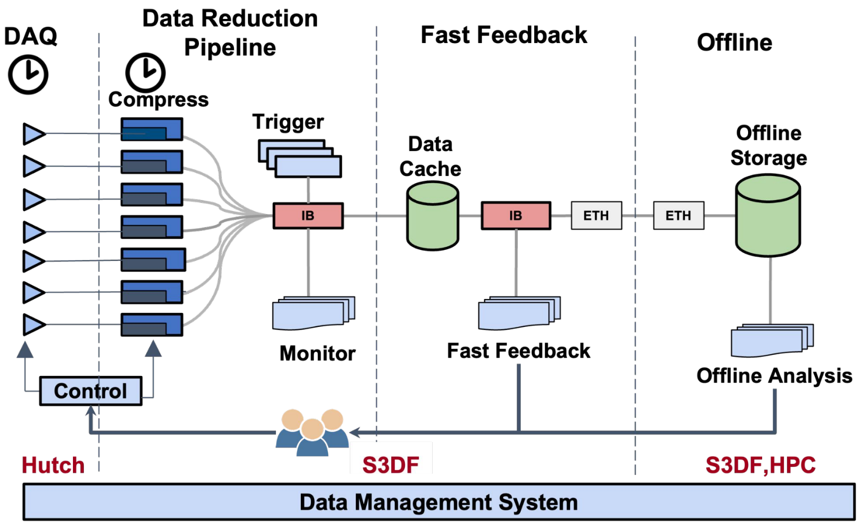


Autonomous Feedback: < 1s
Fast feedback: ~ minutes
Science-driven feedback: minutes - hours

Begin on shift analysis immediately for results within minutes.
Off shift: multiple passes of analysis over data on S3DF/HPC to determine next day's experiment plan.
Post-experiment analysis on S3DF/HPC continues for up to 10 years (or more) after beam time ends.

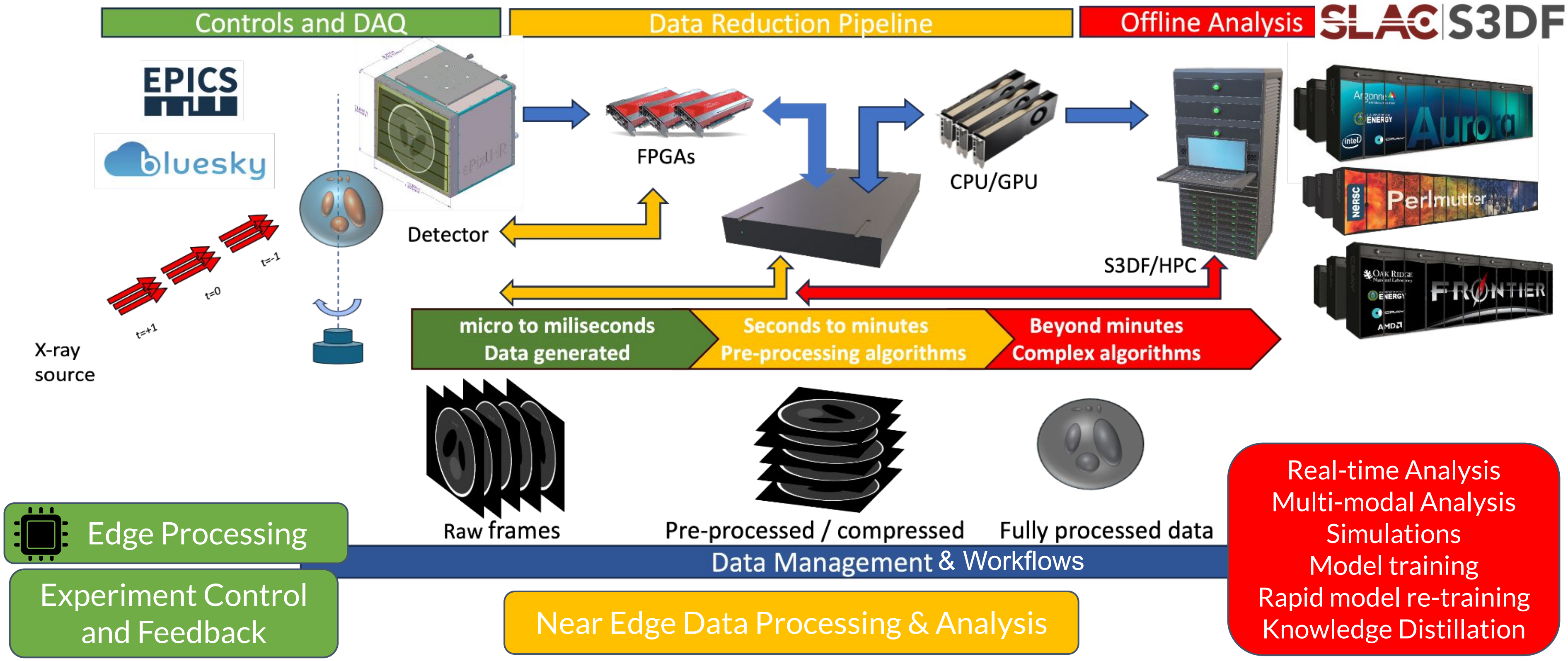
Big Data Handling Strategies for the Linac Coherent Light Source

- Do More With Less Data:
  - Data Reduction/Feature Extraction
  - Implications for Data Format
- Keep Up with the Data Rate
  - Analyze Data at the Rate of Production
  - Actively Monitor Data Quality
  - Performant, Scalable Software
- Seize the Means of Data Production
  - Use Actionable Info to Steer Experiments
  - AI assisted Decision-Making
  - Run algorithms at every layer of the pipeline
- Use Integrated Hardware and Software Infrastructure
  - Local resources and seamless access to remote HPC resources
  - Workflow Orchestration
- Data is a National Resource: Data curation and data management
  - Generate analysis pipeline-ready data products at the point of data collection
  - Automate metadata collection to render data findable and found data useable/reuseable



LCLS-II Data System infrastructure for big data handling strategies

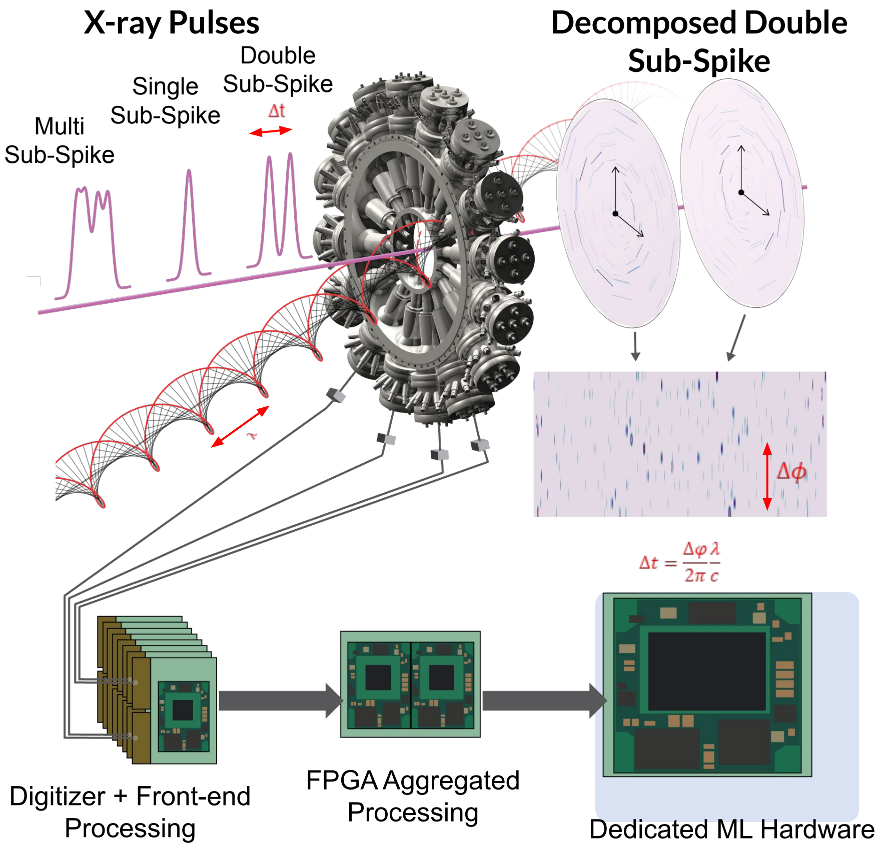
LCLS-II Data System is a heterogeneous pipeline with intelligence and computational power at the detector, the data reduction pipeline, local data center and remote HPC.



Life at the Edge

Credit: Amanda Shackelford, Ryan Coffee

EdgeML for Source to Sink Analysis and Steering



Multi-Resolution COokiebox (MRCO)

- Destination: S3DF
- Computing: EdgeML in FPGA
- Throughput: 200 GB/s → 2 GB/s

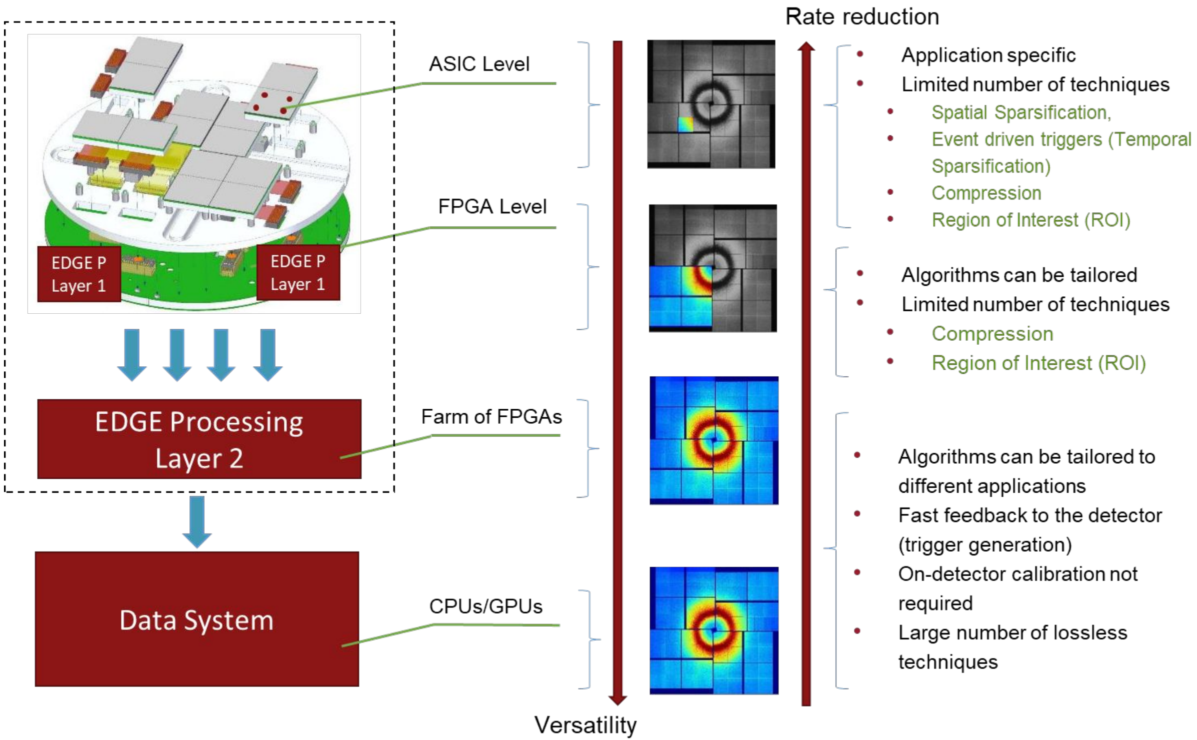
EdgeML for pulse reconstruction, closed-loop control, and AI enabled decision making feedback achieves 30 attosecond resolution

- EdgeML in FPGA differentiates # of sub-spikes/shot, enables veto and live data sorting
- Groq inference throughput > 10 kHz
- Experiment steering

In addition to ASIC/FPGA, novel accelerators possible for inference at edge, but need programmability

Edge Intelligence: push computing closer to the source

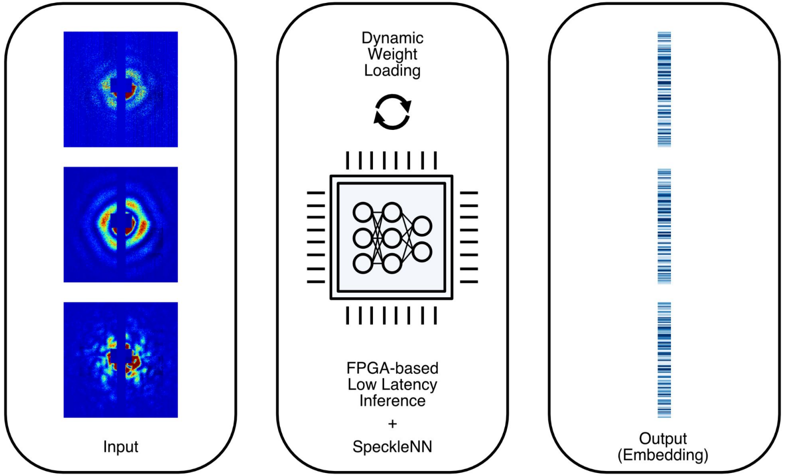
Distribute computing to extract information as data move through the network in the most efficient way.

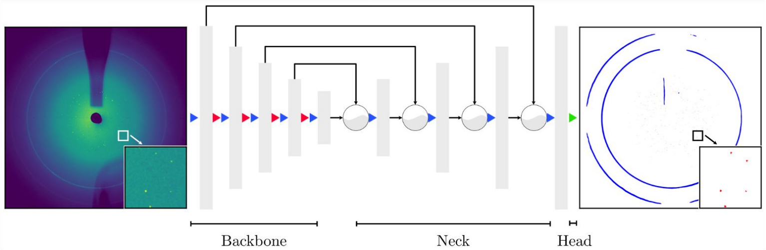


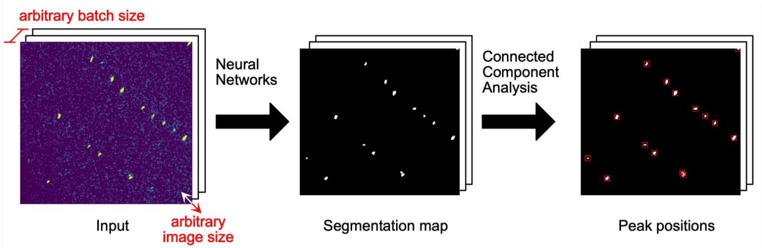
- Solve data transmission bottleneck using compression algorithms implemented in ASIC/FPGA
- Alleviate network, storage, and computing bottlenecks
- Implement adaptable, experiment-specific feature extraction.
- Enable low latency feedback for experiment steering and smart, autonomous expts
- Faster readout of desired information

Daily/weekly adaptability requirements imply the need for programmability & portability of algorithms

Produce Actionable Information for AI-enabled classification and feature extraction at the Edge







AI Classification of diffraction images in FPGA for real time vetoing

Wang, C. et al., 2023 (https://doi.org/10.48550/arXiv.2302.06895)
Herbst, R., et al. 2023 (https://doi.org/10.48550/arXiv.2305.19455)

1 MHz Autonomous Bragg Peak Finder for online feature extraction

Wang, C. et al., 2023 (https://doi.org/10.48550/arXiv.2303.15301)

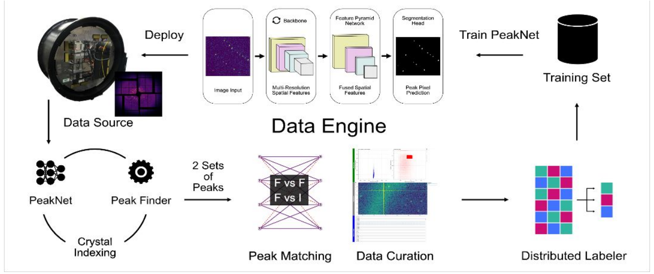
Cong Wang
Abhilasha Dave

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Award Number FWP-100643.

Connect Scientific Instruments to HPC to Create Smart Instruments

EdgeML creates new workflows that themselves require significant computing

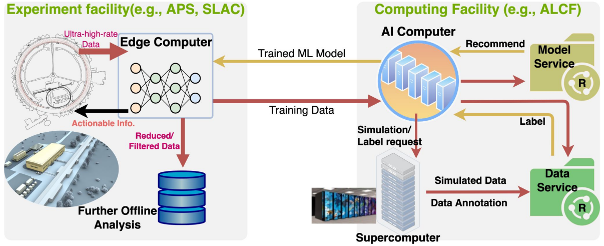
Large-scale Training



Models are too big to train on SLAC's Shared Science Data Facility (S3DF).

External computational resources are required.

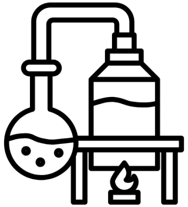
Rapid-Retraining



Full-Spectrum AI/ML Strategy must include the ability to

- develop a model offline
- move it to the Edge (online)
- retrain on-the-fly to adapt to changing experiment conditions

Knowledge Distillation



Knowledge Distillation makes models lighter without compromising their vision.

Model compression enables deployment of AI/ML on resource-constrained edge devices.

Heterogeneous pipelines support & introduce time-, data-, and compute-intensive workflows

DAQ/Data Reduction Pipeline

Thayer, et al. 10.1109/XLOOP49562.2019.00011

Data Reduction Pipeline - Do More with Less Data

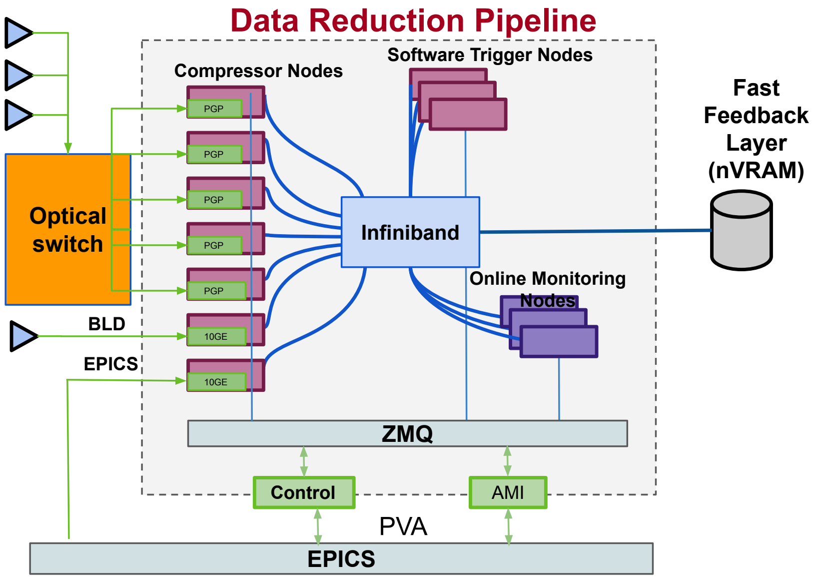
All data are equal, but some are more equal than others

On-the-fly data reduction

- Mitigates network, storage, computing bottlenecks
- Enables streaming to HPC
- Common algorithms, parameterized for adaptability

Software Trigger Nodes perform online event build collecting data from multiple detectors from same event.

Algorithms	TMO	RIX	XPP	TXI	DXS	MFX	CXI
Lossless	X	X	X	X	X	X	X
Veto					X	X	X
SZ Compression		X	X	X	X	X	X
Average image binning			X	X	X	X	
Pixel binning					X	X	X
ROI/Projection			X	X			
Angular integration and pie slicing			X	X		X	
Peak-finding/ threshold	X	X	X	X	X		



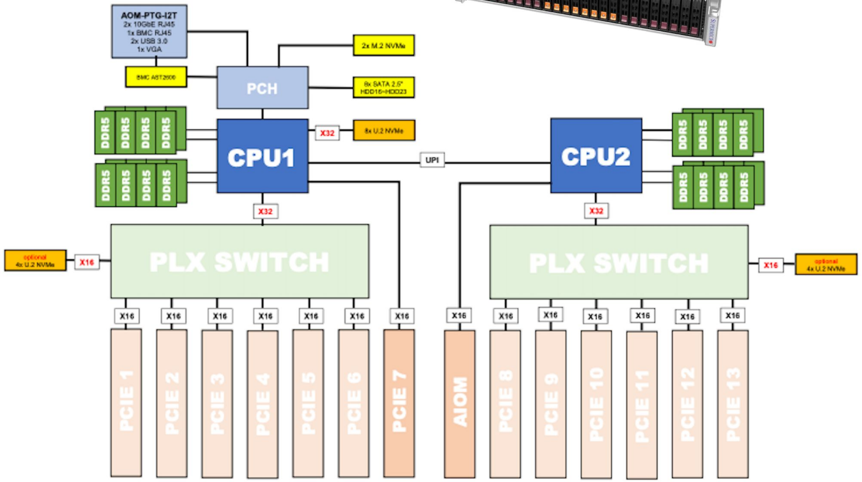
Save reduced data, and a pre-scaled fraction of raw data to allow us to verify that reduction does not affect physics results.

Data Reduction algorithms developed offline, then migrated online → portability is important

Data reduction for large, imaging high-rate area detectors

XPP ePixUHR 4 MP @ 35 kHz for LCLS-II-HE will produce ~ 280 GB/s on 240 fiber pairs

4x SuperServer chassis



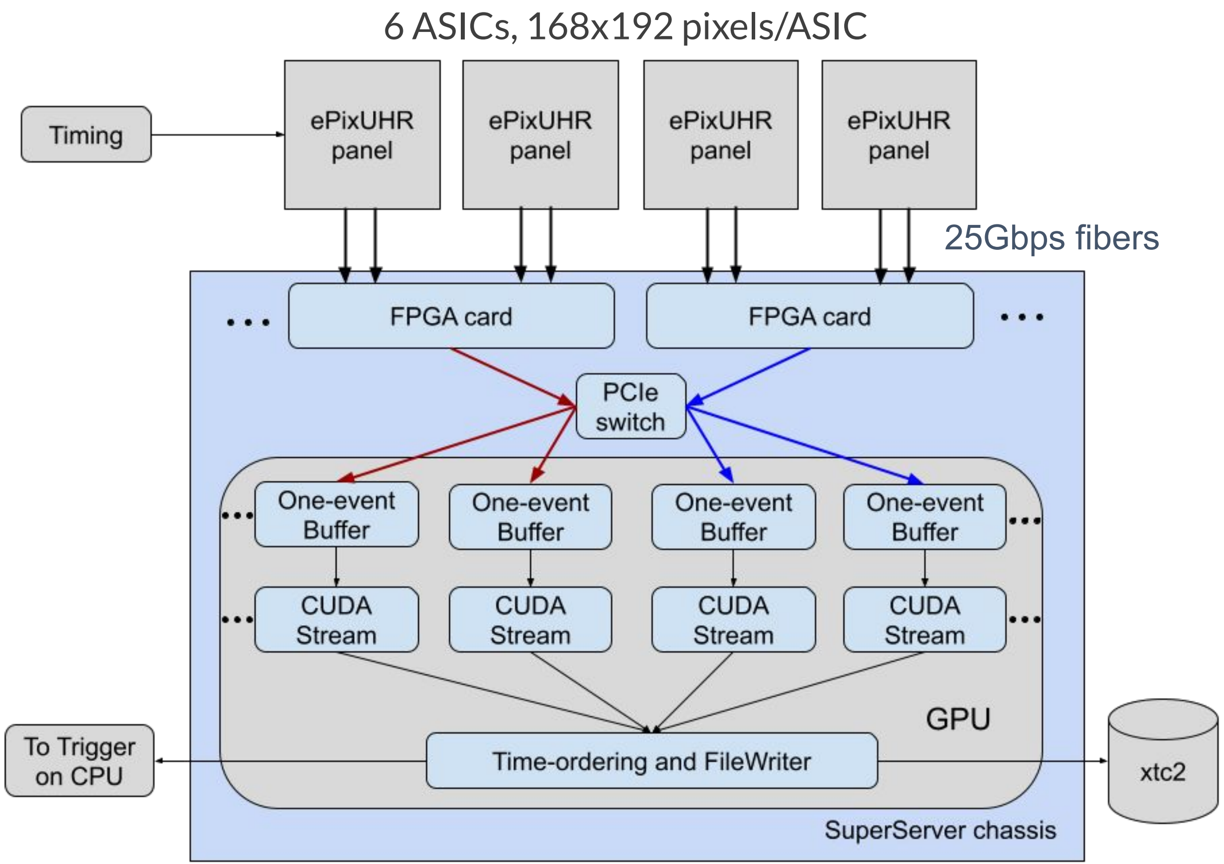
- **Detector protocols:**
  - PGP: SLAC high-speed FPGA protocol, based on Xilinx Aurora
  - UDP unicast/multicast
  - EPICS (TCP)
- Detectors with **widely varying readout rates** (1Hz thru 1MHz)
- **Firmware timestamping**, except for some detectors <120Hz
- **Deadtime:** disable triggers when buffers are full

4x chassis with 24x FPGA boards, 24x H200 GPUs

Worst case: 16 MP@35 kHz

LCLS-II-HE XPP ePixUHR @ 35 kHz - GPU Readout Dataflow

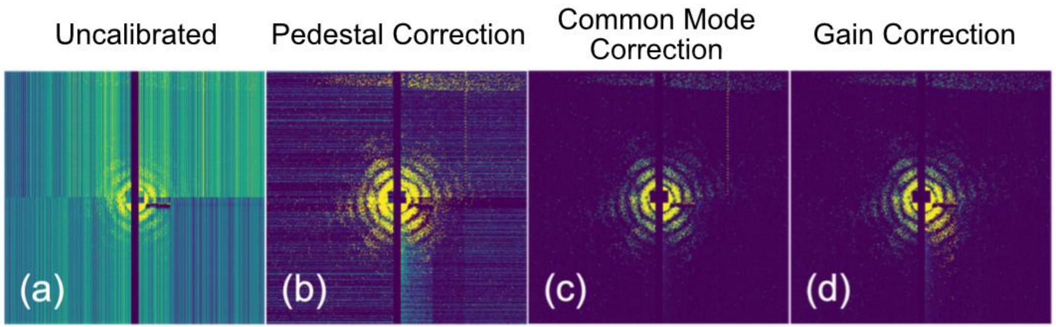
- **Minimal online event-build**: event build a small fraction of events to implement veto
- Use libfabric to **abstract out infiniband/ethernet differences**
- **Reuse buffers**: no malloc/free per-event, but do copy the data from the driver (non-ideal) to get standard format



Multiple GPUs per node (one shown)

GPU Data Reduction Algorithm Performance

Prerequisite to compression algorithms below is calibration and sometimes binning and shot-filtering.

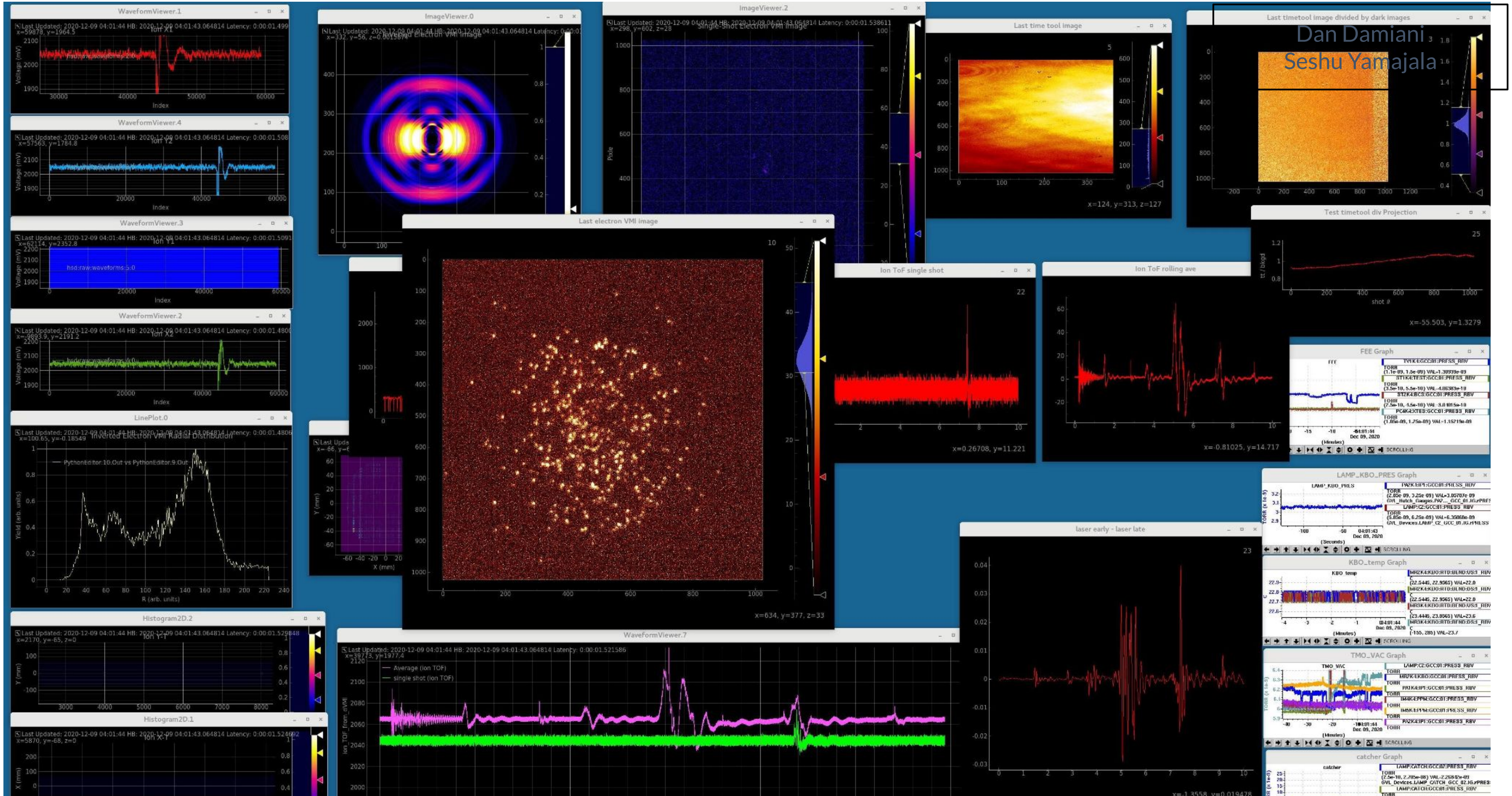


Algorithm	Rate	Notes
cuSZ lossy compression	91 GB/s	useful for SAXS/WAXS data
LC lossy compression	60 GB/s	4 GPU streams, useful for SAXS/WAXS data
pyFAI angular integration	1.7 GB/s	
Sparse-matrix angular integration	6.6 GB/s	
pyFAI peakfinder8	1.5 GB/s	Issue: requires bkgd calculation from all panels

Results on NVIDIA RTX A5000 GPU without DAQ overhead

Keep up with the Data Rate - Scalability and Parallelization

Dan Damiani
Seshu Yamajala



Data Format

xtc2: A Data Format for a Heterogeneous Pipeline

Data format is tightly packed and the same “on the wire” as it is in memory and in the file

- **In-memory format identical to persistent-storage format**
  - allows psana to run from shared memory and embedded in DAQ
- **No serialization required** when data is transmitted to remote machines
- **Natural support for fundamental types** (integers, floats…), arrays of fundamental types and variable-length data
- **Fields that describe the software/version** needed to interpret a particular block of data
- **Separate small metadata from large data** so it can be read more quickly. (e.g. fseek offsets of large data (useful for parallelization) and other small data that can be used to decide whether or not to pay the penalty of fetching the associated large data).
- **Easy python/C++ interface** (with no array copying when accessing)
- Natural support for **variable-length data**
- **Lightweight** (2500 lines of C++) with no dependencies

Raw Data Format: xtc2 allows for variable length data

xtc2 format is the same in-memory and file, enabling streaming

Datagram Header (per-event):

Timestamp (64-bit)

Datagram Type (32-bit)

Contains several nested (or concatenated) XTC headers (eXtensible Tagged Container):

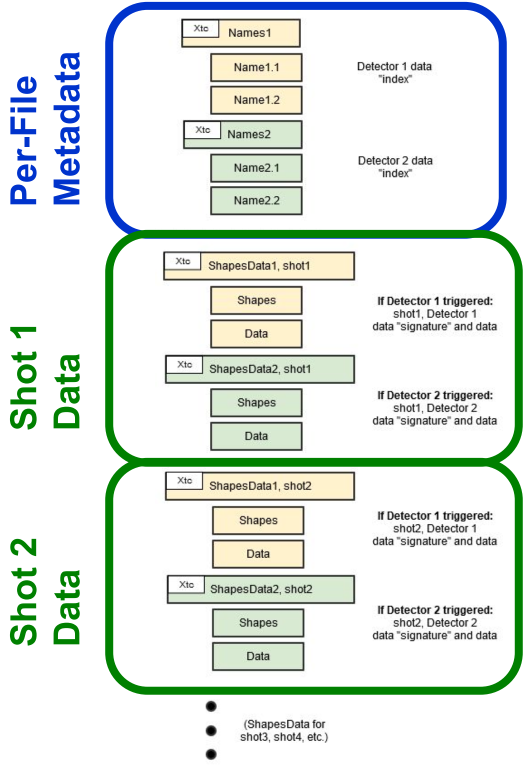
Damage (16-bit)

Src (32-bit)

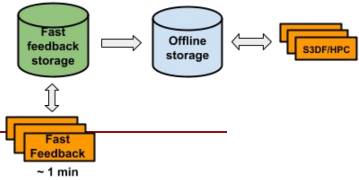
TypeId (16-bit)

Extent (32-bit)

Payload (as long as Extent)



Analysis Framework

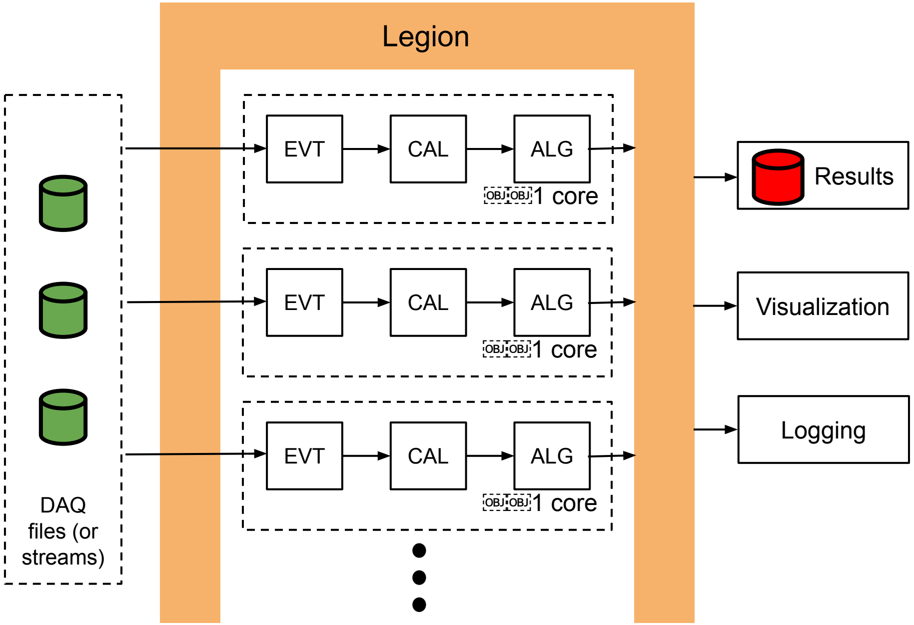


Offline Analysis Framework

psana - **P**hoton **S**cience **Ana**lysis framework for accessing data

psana provides parallelization, common algorithms, detector corrections, event-building, file format handling, visualization.

- Allows for real-time analysis, whether run online, in the fast feedback, or offline on 1 to 300,000+ cores
- psana uses MPI to provide:
  a. Overlap of I/O and compute
  b. Portable performance on new architectures
  c. "Perfectly parallel" pattern: applications can be scaled, limited by data distribution from filesystem or shared-memory
- LCLS uses conda/spack to create releases that run on S3DF and remote HPC resources



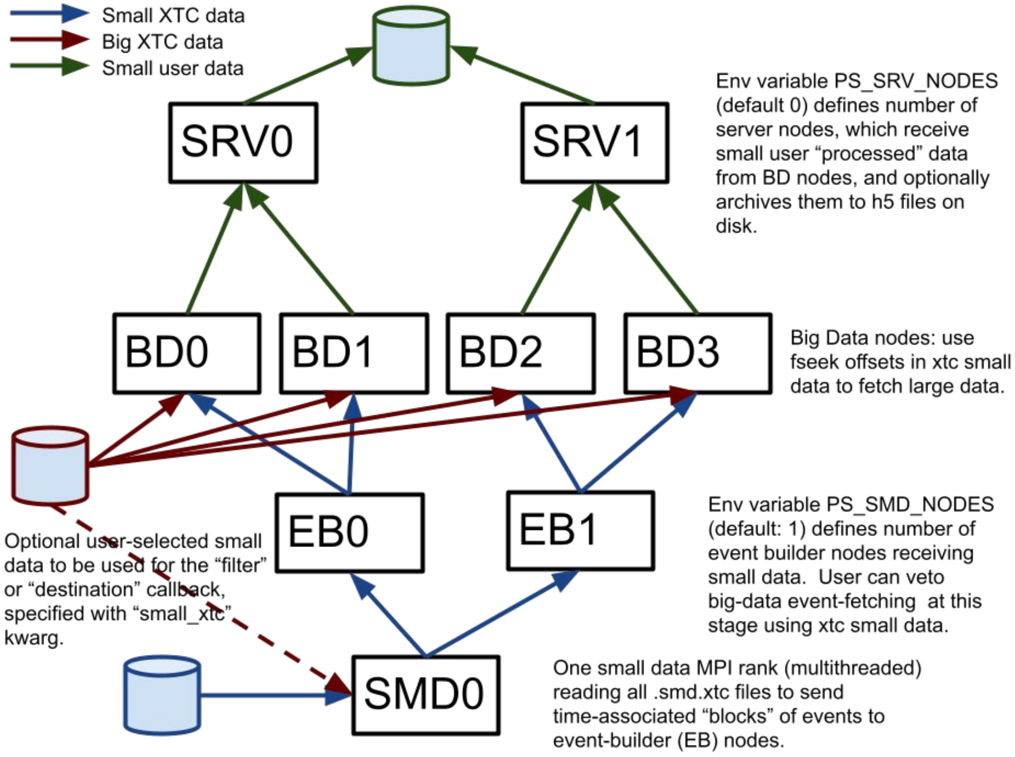
Reads science data from file or stream, distributes one event per core, performs detector calibrations and invokes science specific algorithms

Analysis ("psana") big ideas

- **Limited data types (float, int, arrays):** raw data translation to Python uses C++ Python extension
- "**Perfectly Parallel**": different events given to different cores
- **Memory management done with Python** reference counting with zero-copy of array data using PyArray_SimpleNewFromData()
- **Two-levels of Python interface**: messy raw data (uncorrected, hidden from user, for experts) and user-level "Detector" interface
- **Small data defined by users** at analysis time for filtering
- Analysis must work on individual area-detector panels, since **detectors have multiple segments**
- Detector **calibrations stored in MongoDB/GridFS** for read-only access at HPC centers
- **Same scripts work everywhere**: real-time and offline analysis
- **Hide unnecessary complexity** from users: MPI parallelization, HDF5 production, detector corrections
- **Integrating detector support** needed to correctly associate high-rate events with slow detectors
- **Live-mode** analyzes data while being written
- **Jump** to selected events (not available in live-mode)
- **Grafana** tools used **to understand/optimize high-rate performance** of analysis workflows

Generating calibrated, event-built HDF5 data products

psana uses MPI to distribute load; HDF5 product is the INPUT to user analysis pipeline



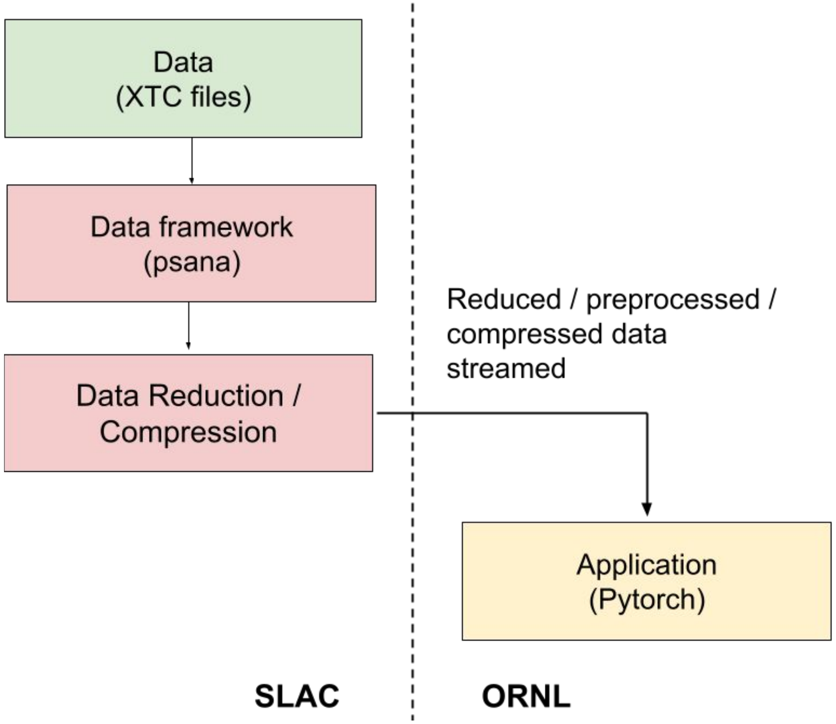
Note that this pattern hides complexity but uses MPI making psana more of an application than a library.

Any workflow built on top of it that thinks in can control distribution of tasks across nodes using MPI is going to have a bad time.

Remote-Location Data Processing: LCLStream

Data Streaming to Remote Facilities: LCLStream

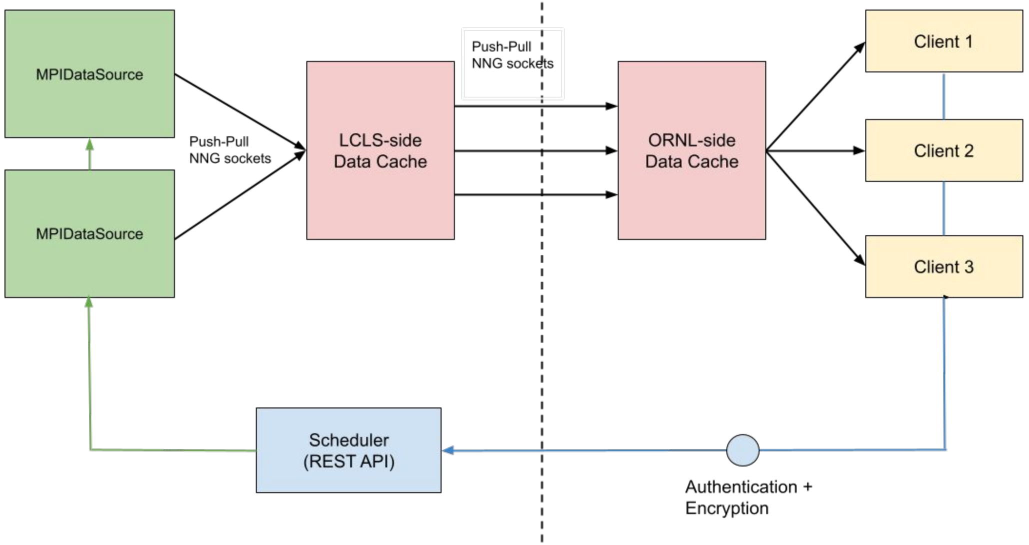
- Streamed data is reduced:
  - Only relevant data is transferred (detectors, hits)
- Streamed data is preprocessed:
  - Ready for scientific interpretation
- Streamed data is compressed:
  - Lossless compression



Remote-Location Data Processing: LCLStream

Data Streaming to Remote Facilities: LCLStream

- Streamed data is reduced:
  - Only relevant data is transferred (detectors, hits)
- Streamed data is preprocessed:
  - Ready for scientific interpretation
- Streamed data is compressed:
  - Lossless compression



Local Heterogeneous Computing using LCLStream

Separate:

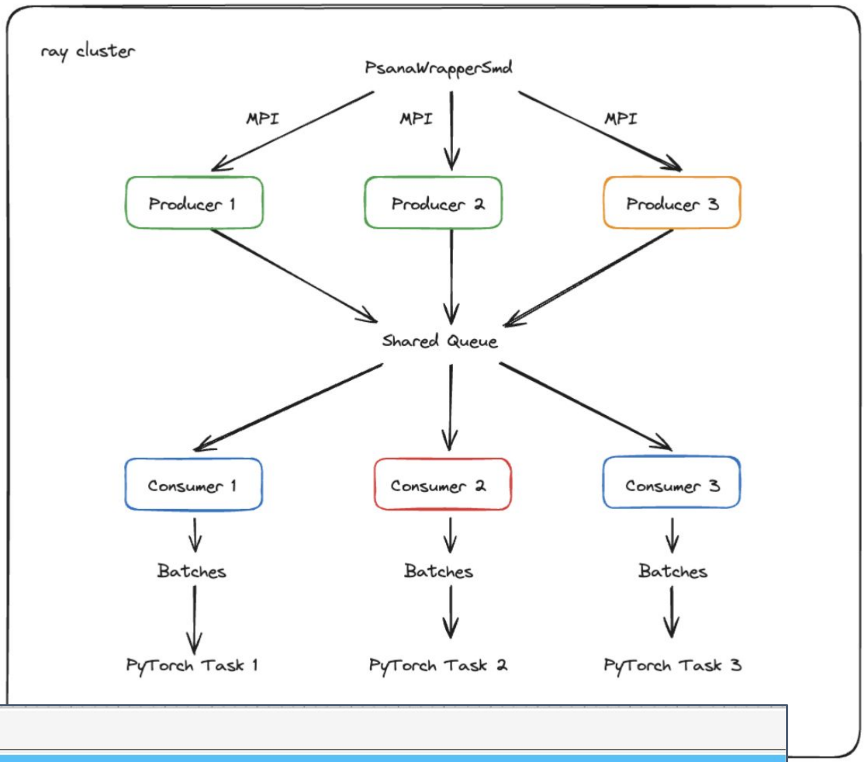
- Data reading (LCLStreamer - Psana)
- Data Processing (Processing code - Ray)

Psana: optimized for heavily parallelized processing of single events

GPU: Batches of events in contiguous memory

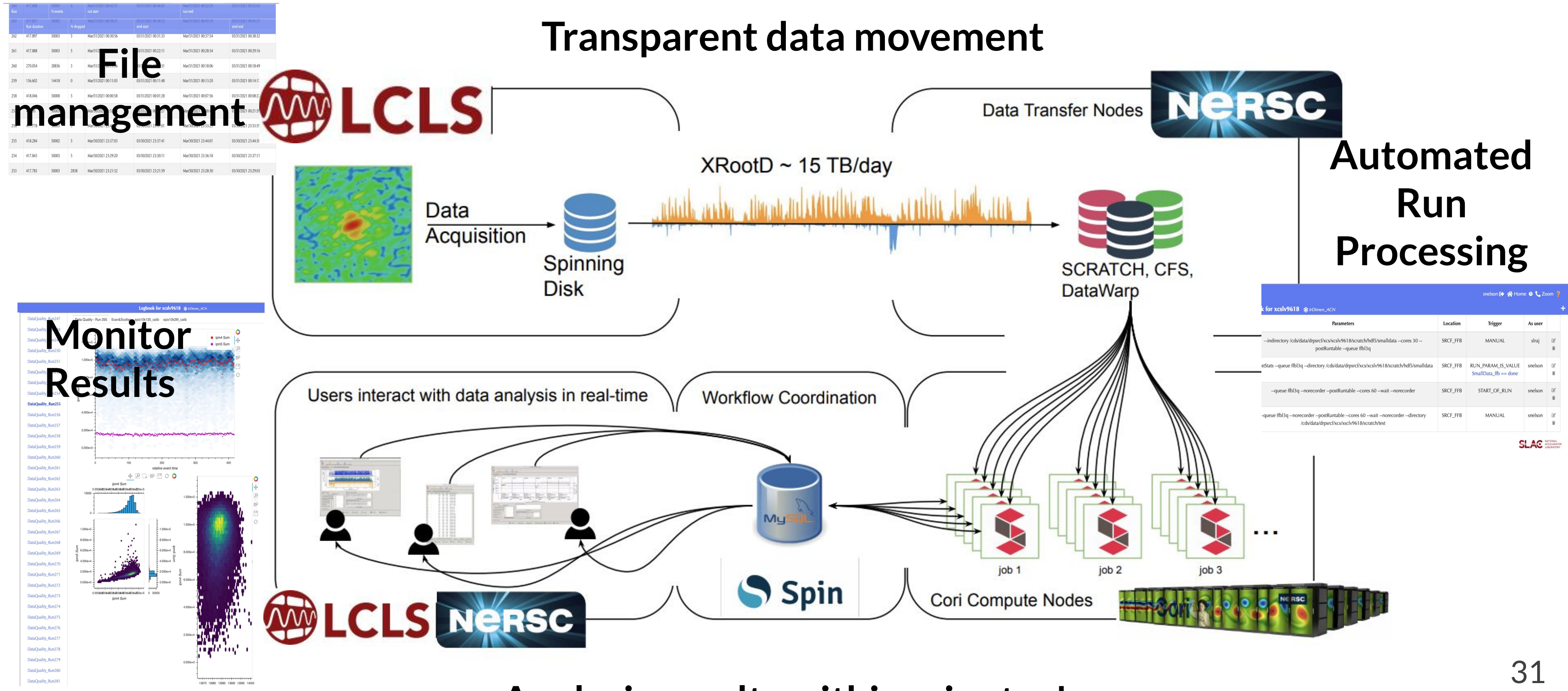
LCLStream can bridge the gap and optimize GPU usage

Cong Wang



https://doi.org/10.48550/arXiv.2206.11992

NERSC enables quasi-real time analysis for LCLS Experiments



Analysis results within minutes!

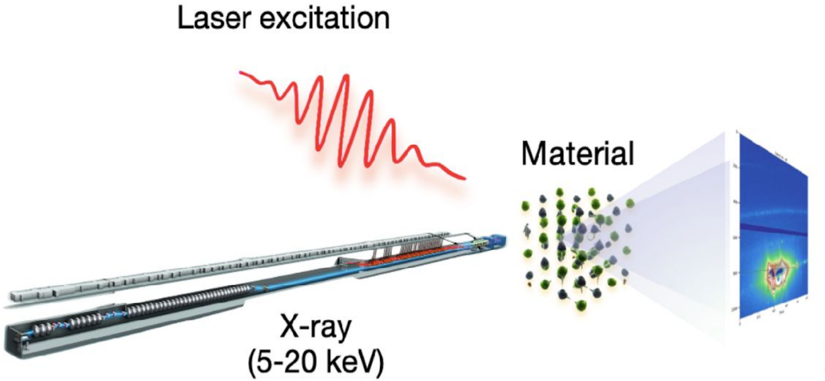
SLAC

Real-time Analytics:
Using HPC to accelerate analysis for time-resolved experiments

Credit: Quynh Nguyen

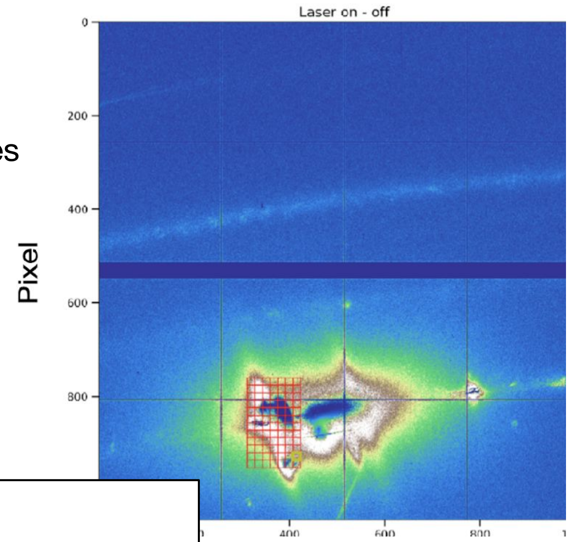
NERSC
National Energy Research Scientific Computing Center

cuPyNumeric for performant GPU-accelerated data analysis pipelines replaces numpy in user code



10+ hours to ~4 minutes
222x Faster

Terabytes/hour
→ 0.1 - 1 Terabytes/s



GPU needs for 33 kHz running

- EOS:
  90 nodes, 720 H100s
  1060x speedup
- Perlmutter:
  256 nodes, 1060 A100s
  939x speedup

  10+ hours → 90 s

Steps in the pipeline:

- Pedestal subtraction
- Gain correction
- ROI data reduction
- Event filtering (throw out bad events, select for desired qualities)
- Create HDF5/Data product input to user pipeline
- User-defined offline analysis (read from file/decompression penalty)

NVIDIA
Dr. Irina Demeshko
Malte Foerster

SLAC
Seshu Yamalaja
Dr. Quynh L Nguyen

Summary of Challenges and Opportunities

Every time hardware or software updates or a new opportunity (AI/ML) arises, the LCLS Data System platform, infrastructure, and hundreds of user workflows for data reduction, fast feedback, data quality monitoring, decision support, dat management and data processing must also change!

Challenges:

- Increased reliance on sophisticated workflows and HPC; some workflows are “canned”, some not
- Growing divide between user capabilities and computing sophistication required to use HPC to execute complex workflows
- Heterogeneous pipelines (ASIC, FPGA, CPU, GPU, TPU, accelerators)help efficiently distribute computing tasks, but complicate performance and portability
- Metadata and data management at high rate with AI/ML require new techniques for data wrangling

Opportunities:

- Wide range of intelligent operations and low code data processing opportunities: Adaptive tagging of code/data; Workflows specified by visual/data-flow languages and/or learned by example; Semantic search and auto-complete for code/data; Speech-directed operations
- Code acceleration methods; advanced heterogeneous memory/processing systems; data file formats; advanced job scheduling (eg to account for I/O, real time queues, etc)
- Cognitive engineering methods can be applied to optimize UI and workflow designs