LCLS2 Data Analysis: Discussion with UEC

Christopher O’Grady
SLAC National Accelerator Laboratory
Aug. 25, 2023
The LCLS-II Data System is the focus.
Data System Overview

DAQ Readout Nodes

Data Reduction Pipeline

Software Trigger Nodes

Fast Feedback Layer

Data Cache

Fast Feedback Analysis Farm

Offline

Offline 10-100 PB

Lustre

HPC (SLAC, NERSC, LCF)

250 GB/s - 1 TB/s

Hutch

Experimental Hall

250 GB/s - 1 TB/s

10 GB/s - 1000 GB/s
Analysis/DAQ Information

Analysis/DAQ user documentation:
https://confluence.slac.stanford.edu/display/LCLSIIData/LCLS-II+Data+Acquisition+and+Analysis

New Shared Computing Facility (“S3DF”):
https://confluence.slac.stanford.edu/display/PCDS/Running+at+S3DF
Analysis ("psana") Big Ideas

- **"Perfectly Parallel"**: different events given to different cores using world-standard MPI parallelization. Scalable to 1MHz.
- Detector **calibrations stored in MongoDB/GridFS** for read-only access at HPC centers around the world
- **Same scripts works everywhere**: real-time and offline analysis (and embedded in DAQ!)
- **Hide complexity** that users don’t need to see: MPI parallelization, HDF5 production, detector corrections…
- Raw data (messy) in XTC2 format, Calibrated data (neat) in **much smaller** HDF5
- **psana is the fundamental LCLS analysis tool**. other tools exist on top: e.g. crystfel/cctbx, OM, smalldata_tools…

(https://github.com/slac-lcls/smalldata_tools)
from psana import DataSource
import numpy as np
import os
def my_smalldata(data_dict):
    print(data_dict)
    os.environ['PS_SRV_NODES']='1'
ds = DataSource(exp='tmoc00118', run=222)
smd = ds.smalldata(filename='mysmallh5.h5', callbacks=[my_smalldata])
for run in ds.runs():
    opal = run.Detector('tmo_opal1')
ebeam = run.Detector('ebeam')
runsum = 0
for evt in run.events():
    img = opal.raw.image(evt)
    photonEnergy = ebeam.raw.ebeamPhotonEnergy(evt)
    if img is None or photonEnergy is None: continue
    evtsum = np.sum(img)
    # pass either dictionary or kwargs
    smd.event(evt, evtsum=evtsum, photonEnergy=photonEnergy)
    runsum += evtsum # beware of datatypes when summing: can overflow
if smd.summary:
    tot_runsum = smd.sum(runsum) # sum (or max/min) across all mpi cores
    # pass either dictionary or kwargs
    smd.save_summary({'sum_over_run' : tot_runsum}, summary_int=1)
smd.done()
psana MPI Task Structure

Small XTC data
Big XTC data
Small user data

SRV0

SRV1

BD0

BD1

BD2

BD3

EB0

EB1

SMD0

Env variable PS_SRV_NODES (default 0) defines number of server nodes, which receive small user “processed” data from BD nodes, and optionally archives them to h5 files on disk.

Big Data nodes: use fseek offsets in xtc small data to fetch large data.

Optional user-selected small data to be used for the “filter” or “destination” callback, specified with “small_xtc” kwarg.

Env variable PS_SMD_NODES (default: 1) defines number of event builder nodes receiving small data. User can veto big-data event-fetching at this stage using xtc small data.

One small data MPI rank (multithreaded) reading all .smd.xtc files to send time-associated “blocks” of events to event-builder (EB) nodes.
AMI Big Ideas

- **Graphical LabView-style** analysis
- **Scalable**: multi-node/multi-core
- **Based on psana-Python**
  - As soon as detector is supported in psana, it works immediately in AMI
- **Easily extendable by users** with Python
- Can be **run online/offline** (with MPI parallelization)
- **8-minute video tutorial** and many offline examples that anybody can run here:
  
  https://confluence.slac.stanford.edu/display/LCLSIIData/ami#ami-HowToLearnAMI
Real-time Analysis with AMI
Real-time (~1s latency) TMO Analysis With AMI
Data Selection and Retention

- **Data selection:**
  - DAQ: aiming for 200GB/s real-time reduction to 20GB/s
    - start simple: thresholding/triggering
    - studying lossy compression for a variety of experiments ([https://confluence.slac.stanford.edu/display/PSDMInternal/SZ+Compression](https://confluence.slac.stanford.edu/display/PSDMInternal/SZ+Compression))
    - save a fraction of events with un-reduced data
  - second reduction step: users select which data they want in hdf5
    - per-event hdf5 data may not work as well at high rates?
    - dask is an option for larger-scale hdf5 analysis ([https://confluence.slac.stanford.edu/display/LCLSIIData/DASK](https://confluence.slac.stanford.edu/display/LCLSIIData/DASK))

- **Data retention:**
  - [https://confluence.slac.stanford.edu/display/PCDS/Data+Retention+Policy](https://confluence.slac.stanford.edu/display/PCDS/Data+Retention+Policy)
<table>
<thead>
<tr>
<th>Undulator</th>
<th>Instrument</th>
<th>Endstation</th>
<th>Technique</th>
<th>Detector</th>
<th>Detector Size</th>
<th>Detector Rate (Hz)</th>
<th>Data Rate (aggregate) (GB/s)</th>
<th>Utilization Factor (0-1)</th>
<th>Data Reduction Type (1st Cut)</th>
<th>DR Factor (1st cut)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SXU</td>
<td>NEH 1.1</td>
<td>DREAM</td>
<td>COLTRIMS</td>
<td>Digitizer</td>
<td>800000</td>
<td>100000</td>
<td>160.0</td>
<td>0.75</td>
<td>Zero suppression</td>
<td>0.020</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 1.1</td>
<td>DREAM</td>
<td>Time of Flight</td>
<td>Digitizer</td>
<td>1000000</td>
<td>100000</td>
<td>200.0</td>
<td>0.75</td>
<td>Zero suppression</td>
<td>0.020</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 1.1</td>
<td>LAMP</td>
<td>Time of Flight</td>
<td>Digitizer</td>
<td>1000000</td>
<td>100000</td>
<td>200.0</td>
<td>0.75</td>
<td>Zero suppression</td>
<td>0.020</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 1.1</td>
<td>LAMP</td>
<td>Imaging</td>
<td>SXR Imag. + Digi.</td>
<td>4000000</td>
<td>10000</td>
<td>82.0</td>
<td>0.45</td>
<td>Veto</td>
<td>0.100</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 2.2</td>
<td>LJE</td>
<td>XAS / XES</td>
<td>TES</td>
<td>1000</td>
<td>100000</td>
<td>20.0</td>
<td>0.60</td>
<td>Zero suppression</td>
<td>0.100</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 2.2</td>
<td>LJE</td>
<td>XAS / XES</td>
<td>TES</td>
<td>10000</td>
<td>100000</td>
<td>200.0</td>
<td>0.60</td>
<td>Zero suppression</td>
<td>0.100</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 2.2</td>
<td>LJE</td>
<td>XAS / XES</td>
<td>RIXS-ccd</td>
<td>4096</td>
<td>1000</td>
<td>0.0</td>
<td>0.60</td>
<td>N.A.</td>
<td>1.000</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 2.2</td>
<td>RIXS</td>
<td>IXS / RIXS</td>
<td>RIXS-ccd</td>
<td>4096</td>
<td>1000</td>
<td>0.0</td>
<td>0.60</td>
<td>N.A.</td>
<td>1.000</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 2.2</td>
<td>RIXS</td>
<td>XRD / RXRD</td>
<td>SXR Imaging</td>
<td>1000000</td>
<td>10000</td>
<td>20.0</td>
<td>0.60</td>
<td>ROI</td>
<td>0.100</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 2.2</td>
<td>RIXS</td>
<td>XPCS</td>
<td>SXR Imaging</td>
<td>1000000</td>
<td>10000</td>
<td>20.0</td>
<td>0.60</td>
<td>Compression</td>
<td>0.500</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 1.2</td>
<td>---</td>
<td>X-ray/X-ray</td>
<td>SXR Imaging</td>
<td>1000000</td>
<td>10000</td>
<td>20.0</td>
<td>0.30</td>
<td>ROI</td>
<td>0.100</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 1.2</td>
<td>---</td>
<td>Imaging</td>
<td>epix100-HR + Digi.</td>
<td>4000000</td>
<td>5000</td>
<td>42.0</td>
<td>0.45</td>
<td>Veto</td>
<td>0.100</td>
</tr>
<tr>
<td>SXU</td>
<td>NEH 1.2</td>
<td>---</td>
<td>XAS / XES</td>
<td>RIXS-ccd</td>
<td>4096</td>
<td>1000</td>
<td>0.0</td>
<td>0.60</td>
<td>N.A.</td>
<td>1.000</td>
</tr>
<tr>
<td>HXU</td>
<td>NEH 1.2</td>
<td>---</td>
<td>X-ray/X-ray</td>
<td>SXR Imaging</td>
<td>1000000</td>
<td>10000</td>
<td>20.0</td>
<td>0.30</td>
<td>ROI</td>
<td>0.100</td>
</tr>
<tr>
<td>HXU</td>
<td>NEH 1.2</td>
<td>---</td>
<td>Imaging</td>
<td>epix100-HR + Digi.</td>
<td>4000000</td>
<td>5000</td>
<td>42.0</td>
<td>0.45</td>
<td>Veto</td>
<td>0.100</td>
</tr>
<tr>
<td>HXU</td>
<td>NEH 1.2</td>
<td>---</td>
<td>XAS / XES</td>
<td>RIXS-ccd</td>
<td>4096</td>
<td>1000</td>
<td>0.0</td>
<td>0.60</td>
<td>N.A.</td>
<td>1.000</td>
</tr>
<tr>
<td>HXU</td>
<td>NEH 1.2</td>
<td>---</td>
<td>Imaging</td>
<td>ePixUHR</td>
<td>4000000</td>
<td>40000</td>
<td>336.0</td>
<td>0.45</td>
<td>Veto</td>
<td>0.100</td>
</tr>
</tbody>
</table>