Real-Time Data Analysis at NERSC: a Trial Run of Nascent Exascale Experimental Data Analysis

Johannes Blaschke\textsuperscript{1}, Aaron Brewster\textsuperscript{1}, Daniel Paley\textsuperscript{1}, Derek Mendez\textsuperscript{1}, Nicholas Sauter\textsuperscript{1}, Wilko Kroeger\textsuperscript{2}, Murali Shankar\textsuperscript{2}, Bjoern Enders\textsuperscript{1}, Deborah Bard\textsuperscript{1}

\textsuperscript{1}Lawrence Berkeley National Lab
\textsuperscript{2}SLAC National Accelerator Lab

CUG, May 5, 2021
Serial Crystallography

Kern et al., 2018 Nature 563:421
When should I move onto the Next Sample?

• Beamtime is scarce!

• Critical live feedback:
  o Does the beam hit the sample?
  o Do we see crystals?
  o Does the data make sense?
  o What is the quality of the data?

• Can I move on to the next sample?
Experimentalists Are In The Driver's Seat
Live Data Analysis for Experiments in 2020, and Beyond!
Click anywhere on this screen to play video
Deploying CCTBX at NERSC
Data Movement XRootD clusters

- **eLog**
  - LCLS metadata manager

- **datamvr**
  - xrdcp process

- **ana-filesystems**
  - Lustre

- **SLAC/LCLS**
  - NERSC

- **ESNET-SENSE API**
  - used to reserve bandwidth

- **XRootD redirectors**
  - XRootD data servers

- **ESNET-SENSE API**
  - used to reserve bandwidth

- **XRootD redirectors**
  - XRootD data servers

- **ESNET-SENSE API**
  - used to reserve bandwidth
Data Analysis

- Data analysis follows sequential stages:
  - spotfinding
  - indexing Bragg spots
  - model refinement
  - integrating Bragg spots

![Data Analysis Diagram](image-url)
How’s the Computation Weather Today?

- **Computational Weatherplot:**
  1. Each line shows work done by one MPI rank
  2. There is no “the cctbx.xfel workload”

Start-Up and I/O (PSANA)
Spot Detection (DIALS)
Indexing (DIALS)
Refinement (DIALS)
Integrating (DIALS)
How’s the Computation Weather Today?

- **Computational Weatherplot:**
  - Each line shows work done by one MPI rank
  - There is no “the cctbx.xfel workload”

---

**Issue:**

**MPI-Communication Bound**

---

Start-Up and I/O (PSANA)
Spot Detection (DIALS)
Indexing (DIALS)
Refinement (DIALS)
Integrating (DIALS)
How’s the Computation Weather Today?

- Computational Weatherplot:
  - Each line shows work done by one MPI rank
  - There is no “the cctbx.xfel workload”

- Issue: MPI-Communication Bound

- Issue: I/O contention

- Start-Up and I/O (PSANA)
- Spot Detection (DIALS)
- Indexing (DIALS)
- Refinement (DIALS)
- Integrating (DIALS)
How’s the Computation Weather Today?

- Computational Weatherplot:
  - Each line shows work done by one MPI rank
  - There is no “the \texttt{cctbx.xfel} workload”

Start-Up and I/O (PSANA)
Spot Detection (DIALS)
Indexing (DIALS)
Refinement (DIALS)
Integrating (DIALS)
Workflow Coordination using NERSC Spin

- **Home-grown workflow manager** `cctbx.xfel`
  - mySQL database hosted on Spin (NERSC microservice platform)
  - Each worker commits progress to DB
  - `cctbx.xfel` determines new analysis runs and “assembles” jobs (input files, job scripts, ...)
  - `cctbx.xfel` monitors slurm and DB, reporting live progress

**Diagram:**
- Datamvr
  - `cctbx.xfel` checks status of file transfers
- NX/Login Nodes
- Spin
  - slurm connector submits new jobs (multiple users can submit)
  - each rank commits status to database:
    1. spot-finding rate
    2. indexing rate
    3. crystal parameters
- Compute Nodes
  - multiple users query database
  - multiple users query database
XFEL as a Proxy for HPC Data Analysis
Why is XFEL relevant to other "Data Analysis for Science" projects?
Challenge 1: Urgent Computing Resources

- **Data Collected (run number) →**: collecting/transferring, processing, no live results

- **Wallclock time →**: Reservation with 3 nodes, each run takes 1 unit of time to collect and process ⇒ can only process 3 yellow squares at once
Challenge 1: Urgent Computing Resources

- **Green** : collecting/transferring
- **Yellow** : processing
- **Red** : no live results

Reservation with 3 nodes, each run takes 1 unit of time to collect and process ⇒ can only process 3 yellow squares at once.
Challenge 1: Urgent Computing Resources

Data Collected (run number) →

Reservation with 3 nodes, each run takes 1 unit of time to collect and process ⇒ can only process 3 yellow squares at once.

Wallclock time →
Challenge 1: Urgent Computing Resources

Reservation with 3 nodes, each run takes 1 unit of time to collect and process ⇒ can only process 3 yellow squares at once

Falling behind on live fast-feedback

Batch reprocessing

Data Collected (run number) → Wallclock time →
Challenge 1: Urgent Computing Resources

• Reservation:
  o 32 - 64 Haswell nodes for live data processing
  o Can be used for preemptible jobs in the future (avoid idle nodes)

• Realtime QOS:
  o Flexibly add up to 20 Haswell nodes for reprocessing
Challenge 2: High-Speed Data (Network and I/O)

- In data analysis workflows, file systems and network can become bottlenecks

- I/O Optimization:
  - Optimize python logger for high-frequency parallel I/O
  - Write logs to Burst Buffer

- Experience:
  - Transfers ran smoothly, can switch redirect destination
  - FS performance limited the transfer rate

- Improvements:
  - Use SSD storage at LCLS to speed up transfers
  - Improve write performance at NERSC
  - Allow users to initiate the remote transfers
  - Better monitoring and alerting
Challenge 2: High-Speed Data (Network and I/O)

- In data analysis workflows, file systems and network can become bottlenecks

- I/O Optimization:
  - Optimize python logger for high-frequency parallel I/O
  - Write logs to Burst Buffer

- Experience:
  - Transfers ran smoothly, can switch redirect destination
  - FS performance limited the transfer rate

- Improvements:
  - Use SSD storage at LCLS to speed up transfers
  - Improve write performance at NERSC
  - Allow users to initiate the remote transfers
  - Better monitoring and alerting

5-6x slowdown between “good” and “bad” days
Challenge 3: Realtime Monitoring and Workflow Coordination

- Need to identify and deal with variable performance (e.g. rank getting “stuck” on I/O)
- Weatherplots good for identifying load imbalance
- Ongoing research: How to integrate with workflow manager? How to automate?
Looking Forward: (Standardized) Facility APIs
What can an API do?

**Vision:** all NERSC interactions are callable; backend tools assist large or complex operations.

Endpoints prototyped or in prep:

- **/account**
  - data about the user’s projects, roles, groups and usage information.

- **/compute**
  - run batch jobs, query job and queue statuses on compute resources.

- **/task**
  - get info about asynchronous tasks (eg. from `/compute` or `/storage`).

- **/status**
  - query the status of NERSC component system health

- **/storage**
  - move data with Globus or between NERSC storage tiers

- **/reservations**
  - submit and manage future compute reservations (in prep)

- **/utilities**
  - traverse the filesystem, upload and download small files, and execute commands on NERSC systems
Airflow DAG's can make JID calls

Data Acquisition

ARP Trigger Service

Trigger Airflow DAG

Airflow

SDF Cluster SLURM

JID for SDF

Progress bar updates

Start run
End run
Register file

Kafka

Progress bar updates

SLAC

Data Mover

File migrated

SLAC

eLog

Register file

SLAC

Data

Mover

Kafka

Airflow DAG's can make JID calls

NERSC

SFAPI endpoint

Cori Cluster SLURM

NERSC

JID for NERSC

Progress bar updates

JID for SDF

Progress bar updates

Airflow

SLURM

SLURM

SDF Cluster

Cori Cluster

SLURM

SLURM

Progress bar updates
Conclusion
Successful Realtime Data Analysis at NERSC

• Live Feedback:
  o 10 mins from end of run to the molecular structures
    • Enable real-time feedback to beamline staff
  o No babysitting from NERSC staff needed

• XFEL Flexes the following “HPC Muscles”:
  o Urgent Computing Resources
  o High-Speed Data (Network and I/O)
  o Realtime Monitoring and Workflow Coordination

• Beamtime is scarce! Fast feedback is critical! [github.com/cctbx/cctbx_project]