RUCIO and LCLS-II Data Management

2nd RUCIO Community Workshop
Wilko Kroeger, February 27, 2019, Oslo

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

LCLS - Linac Coherent Light Source

SLAC

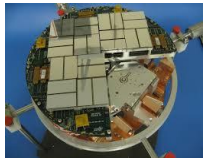
The LCLS is a free electron laser that produces ultra fast X-ray pulses.

LCLS has already had a significant impact on many areas of science, including:

- ➔ Resolving the structures of macromolecular protein complexes that were previously inaccessible
- ➔ Capturing bond formation in the elusive transition-state of a chemical reaction
- ➔ Revealing the behavior of atoms and molecules in the presence of strong fields
- ➔ Probing extreme states of matter

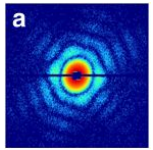
For Example: Single Particle Imaging

Megapixel detector

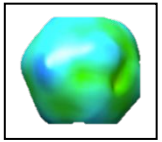


X-ray diffraction image

Intensity map from multiple pulses

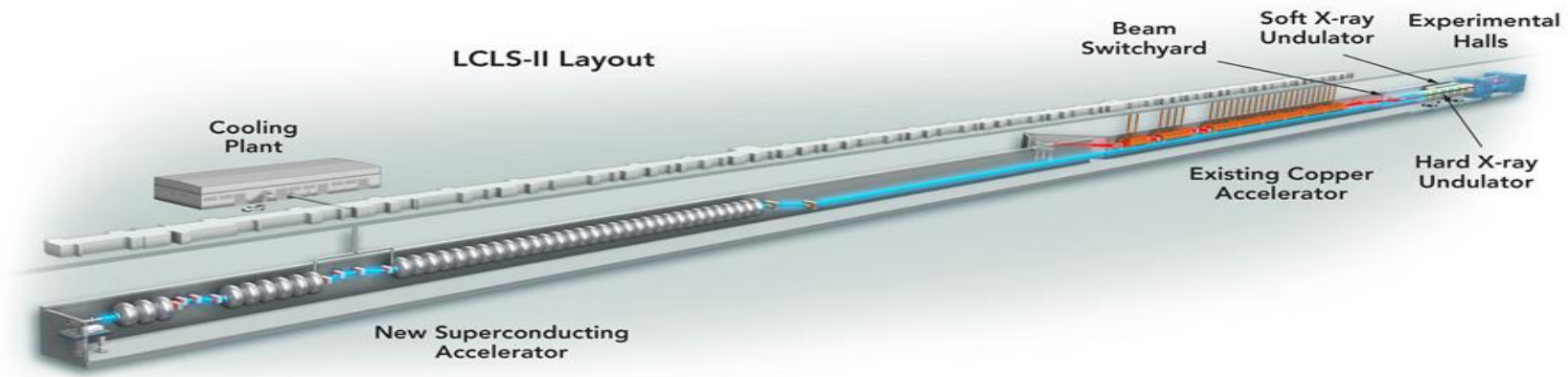


Interpretation of system structure / dynamics



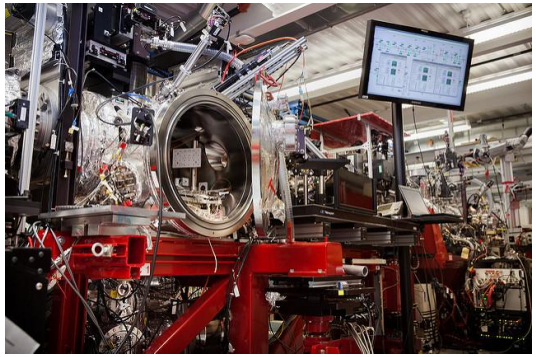
LCLS-II

- LCLS has been operating since 2009 using SLAC's warm copper linear accelerator
- LCLS has provided a wide spectrum of exiting scientific results
- For LCLS-II a new superconducting accelerator is being build and the science program is scheduled for 2021
- LCLS-II will provide a jump in capabilities allowing experiments that were not possible so far.
- The pulse rate will jump to about 1MHz from the current 120Hz of the copper linac.
- Both accelerators will be operated in parallel



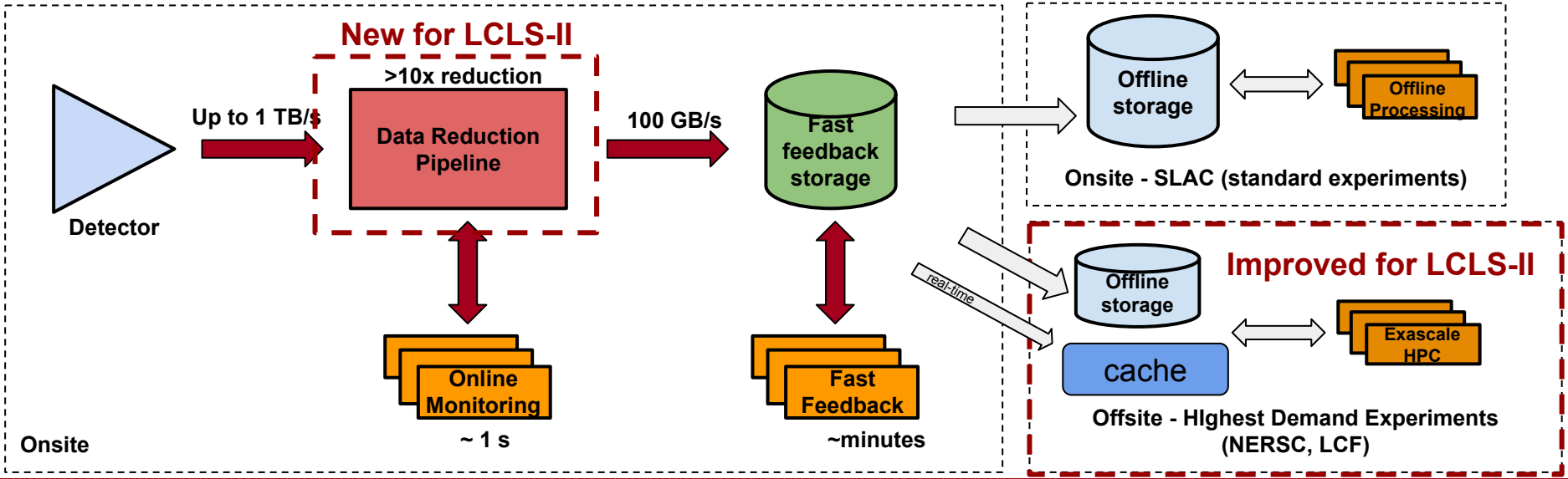
Science Instruments

- Two experimental halls, N/FEH, host the instruments for data collection.
- There are seven different instruments (three in NEH and four in FEH) each one optimised for different experimental X-ray techniques.
- Typically only one (or two if both accelerators are running) instrument is used at a time.



LCLS-II Data System

- Data needs to be reduced as we can not write everything to disk
- events are not build before written to disk. The detector contributions to an event are distributed over many files.
- LCLS: no data reduction, data rate up to 5GB/s. Events are build and all data is written to disk



Data reduction mitigates storage, networking, and processing requirements

LCLS - Data Management

SLAC

- Transfer of data between different storage resources:
  - from DAQ to different ssd/hdd storage, tape systems, storage might be local/remote
- Web tools (*DataManager*) for users to manage their Experiments and data
  - electronic logbook (Elog)
  - metadata/parameter for runs and experiments
  - FileManager (which files exist mostly, restoring runs from tape)
  - workflow tools (automatic xtc->hdf5 conversion, run quality, users job submission,...)
- Users interact with the system:
  - using the *DataManager*
  - accessing files in the file systems
- Manage disk space and file access rights
- Automatic data processing (workflows)
- Data Catalog (currently irods) and Databases for Experiment and Runs
- Experiment and Run management and metadata
- A new DM is being build: use services architecture and Kafka as a message bus

LCLS Data-Collection

SLAC

- Users submit a proposal.
- An accepted proposal becomes an unique *experiment* and it is granted beamtime, typically 3-5x 12h shifts.
- Data collection is organized into ***runs***
  - a *run* typically lasts a couple of minutes (but could be much longer or shorter)
  - after a *run* is done some configuration/setup is changed and a new run is started
  - data from a run are written to many files in parallel.
  - a run is the basic unit for data access

LCLS-II How much data to handle

SLAC

Rough estimates of how many experiments/runs/files we expect

	total
experiments	~10K's
runs	millions
number of files	10's of millions (maybe 100's)

LCLS Data Files

SLAC

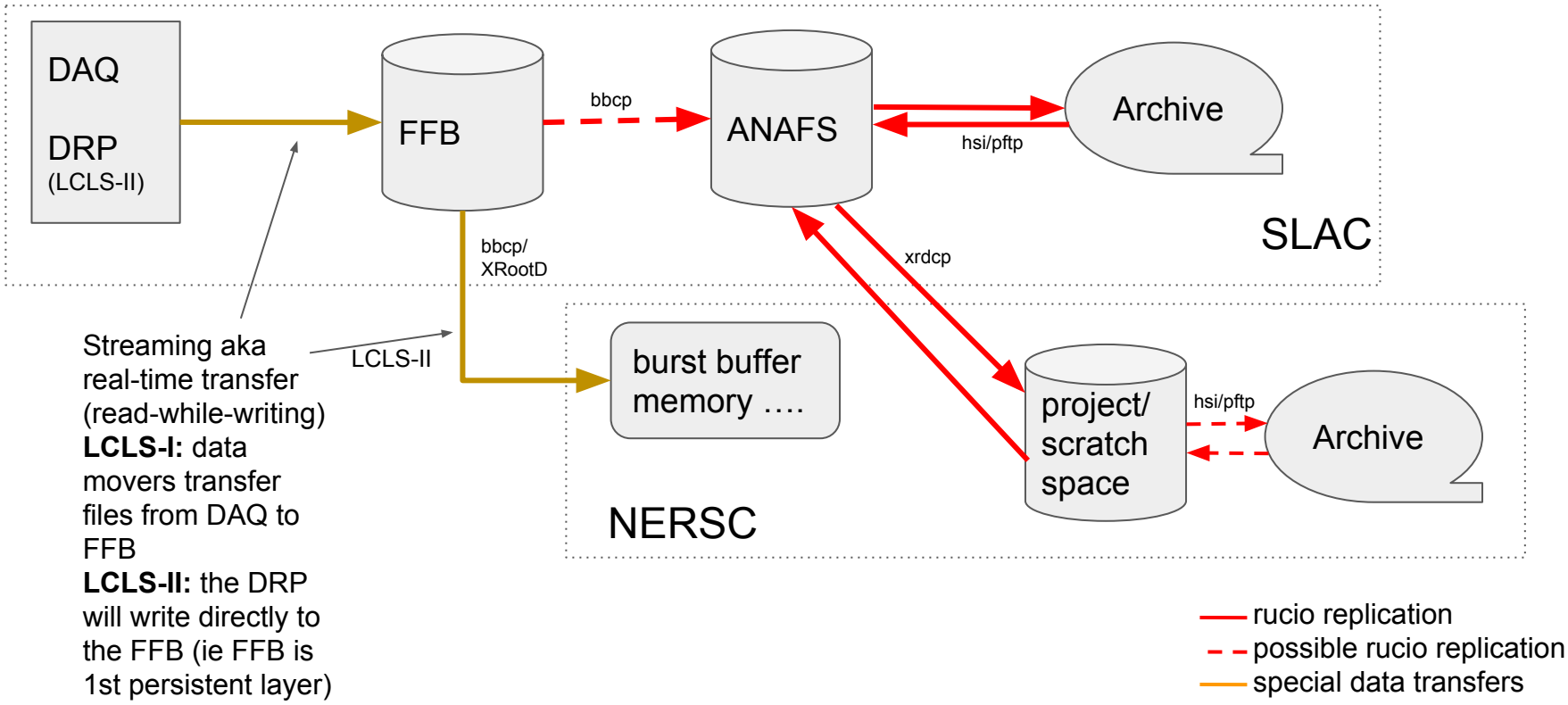
- The Data Management handles three different data types:
  - raw data in xtc format (grouped into runs)
    - most files are in this group
  - user daq data (not associated with a run)
    - only collected by a few experiments, files size typically small O(MB)
  - Secondary processed data (could be run associated)
    - example: process and converting raw data to hdf5 format.
- Users access files/runs using a parallel file system (Lustre), files are found by using a predefined path e.g.: /reg/g/psdm/<instr>/<exper>/xtc.
  A user is not directly accessing the file-catalog.
- User created files (during processing/analysis) are not managed by us we just provide disk space.

LCLS-II File Catalog

SLAC

- Catalog of files and metadata
  - raw data files, secondary products files and special daq files.
  - metadata checksum, size, run number
- Files Replications
  - automatic replicas to remote sites, archive storage and ….
  - user requested replication: restore run from tape, copy to scratch space at remote site
  - *typically all files for a run need to be replicated*
- Lifetime management
  - apply retention policy for files (see later)
- Permissions
  - only users of an experiment are allowed to access the experiments files
  - all files are owned by an production account
  - users do not have access to the file catalog directly but only through a web application that handles the permissions.
  - on disk access is managed by unix-groups and file acls.

LCLS - Data Transfers/Replication



LCLS-I: data movers transfer files from DAQ to FFB

LCLS-II: the DRP will write directly to the FFB (ie FFB is 1st persistent layer)

LCLS - Data Transfers/Replication

SLAC

- The raw data are written to disk and registered into the file catalog
- The following replications need to take place
  - all files are replicated to an archive resource (HPSS) at SLAC and possibly NERSC
  - experiments request to have all files replicated to a remote site for online processing
  - user requested restore of individual runs from archive (local or remote)
  - user requested replication of individual runs to a remote site
- Different tools are used for the file transfers
  - hsi/pftp for HPSS
  - XRootD and/or bbcp for SLAC to NERSC transfers
  - maybe other tools?
  - real time transfers: bbcp allows transferring a file while it is being written to, this will be used to do real time processing at NERSC
- File aggregation
  - some directories contain thousands of small files. Archive to tape is inefficient and we need to bundle them (currently using htar).

File/run lifetime management

SLAC

The current data retention policy that we apply is:

- All newly created raw data stay on disk for at least 6 month
- Restored runs stay on disk for at least one month
- Files are purged only if disk space is needed
- The least recently accessed runs are purged first
- In special cases we ensure that experiment data stay on disk for an extended time

Testing RUCIO

SLAC

- Using RUCIO demo container and FTS container.
  Just started installing FTS and didn't explore replication yet

- Example Usage:
  1) Create a RSE (PSDM_DISK) with deterministic='False' and prefix '/reg/d/psdm'
  2) An experiment becomes a *scope*. Files are registered to RUCIO :
  3) add_replicas(PSDM_DISK, did)

     did = { 'adler32': '1aa535ac', 'scope': '**rtel12318**', 'bytes':1231, 'name':'**xtc.e_rtel12318_r0003_s0002.xtc**',
     'pfn': '**file:/reg/d/psdm/rte/rtel12318/xtc/e_rtel12318_r0003_s0002.xtc**'}
  4) file is attached to xtc dataset (**rtel12318:xtc**)
  5) run_number metadata is set (3 in this example)
  6) Query for files of a run:
     % **rucio list-dids --filter "type=FILE,run_number=3" rtel12318:xtc.***

Open Questions

SLAC

- **Different scope/dataset/file layout**
  - Make one dataset for each run
    - scope:xtc is container contains scope:xtc.run*N* datasets and files are attached to the run*N* dataset.
- **Using run_number metadata to find all files for a run**
  - Is it possible to query for a run_number range e.g. 5-16
  - and return metadata as part of the query result
    <run_number> <file-name> <size> <pfn> ……..
- **File aggregation**
  - Some data files are small and many 1000s of files are created. So far these files are archived to tape using tar (htar for HPSS). Handling this required some code and is not well integrated with current file-manager.
  - Does RUCIO provides any help?
- **Replication**
  - need to test FTS
  - how to integrate HPSS / calling custom transfer tools
  - Is is possible to write one's own FTS

Comments

SLAC

- Using RUCIO
  - Using the demo container easy to use (after a learning a little bit of docker)
  - Would like to see more examples
    - took a little bit of time to figure out how to form the arguments for the python api (e.g.: dicts as arguments and the required keys)
    - how to specify urls for RSEs
    - some examples are wrong (list-dids shows files)
  - Would like more in depth docs about how the tools work together (RUCIO, FTS and other daemons)
- metadata is LHC specific
  - metadata: panda_id, phys_group, stream_name, lumiblocknr

What if LCLS (or photon sources) like to have different names?

Next Steps and Summary

SLAC

- Testing replication
- Setup a full demonstration including
  - file registration, replication
  - purge replicas from disk
  - interaction between LCLS data management system and RUCIO
  - Try to scale to LCLS expected experiments/files/runs (What DB to use?)
- Thanks to Andy, Carlos and Wei for discussions and help testing.
- How to ask questions: forum, email lists…?
- Need to decide by soon as we need a production system by the end of the year (LCLS starts in March 2020).