EPICS in Context of Machine Learning, Beam Tuning and Offline Scientific Computing

Greg White,
19 April, 2023, SLAC
For EPICS Collaboration Meeting, Fermilab Spring 2023.

Greg White, for Auralee Edelen, Alberto Lutman, Chris Mayes, Bob Dalesio, Kay Kasemir, Jacqueline Garrahan, Michael Davidsaver, Matej Sekoranja, Murali Shankar, EPICS 7 Development Team, and many others.

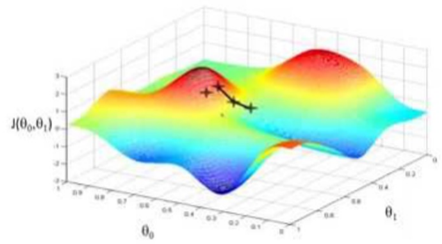
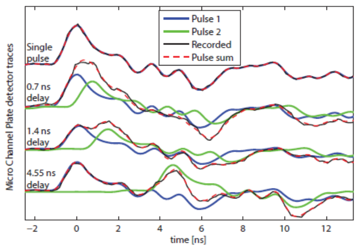
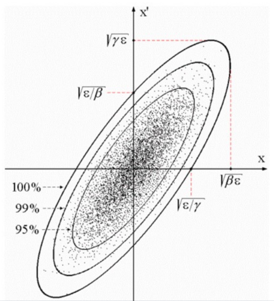
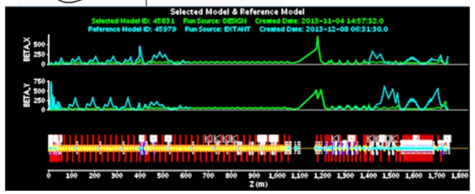
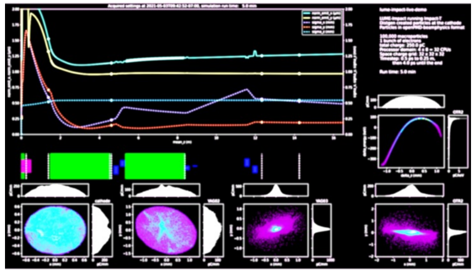
Talk Outline: EPICS is changing to include AI/ML and HPC

- Scope: Controls → += Physics
- EPICS Version: 3 → 7
- Network: Controls → += HPC

Physics ML / Modelling EPICS Drivers

- Beam Optimization no longer only on ad-hoc "correlation plots" and Singular Value Decomposition. Rather continuous multi-parametric and NN online, and big-data offline, informing online in real time
- Models no longer linear only. Space charge, RF kicks, magnet errors, dynamic initial conditions, will have to be included. That implies large HPC and ML, reading and writing to online PVs!
- Tuning on Pulse identified data. Pulse Synchronous Archiving All pulses, (soft) Real time pulse synch accelerator-detector optimization, Continuous image acq/store, Pulse based Modeling
- EPICS implications. Move and support physics analysis on fast networks off production; Control system moves into HPC -> Cyber security extends from HPC to Control Sys.

Emerging EPICS Ecosystem Roles in AI/ML & Physics Applications

Device Abstraction, Data Measurement & Transport	Online non-linear optimization	Collection & Event Building / Big Data	Online Linear Modeling	Offline big modeling and ML
EPICS 7 Structured data. EPICS Security	 Ad-hoc Bunch sync ML regression analysis / ML. Non-linear Model. Gaussian Process. Gradient decent	Beamline - Accelerator Synchronous tuning Bunch resolved mass data archived w/ meta data Sync Camera data  	Online linear model of whole machine, from pulse sync data   	Online Multi-particle simulation (Impact-T). ML real time optimization. 
PV carries many signals/records. E.g. NTNDArray image	PVs carry physics measurements & tuning results	1 PV carries whole synchronous orbit data etc.	1 PV carries whole lattice, all Twiss etc.	PVs from offline HPC carry tuning for immediate use online.

EPICS Version 7 in a Nutshell

1. New Fast Protocol, “pvAccess” replaces CA
2. Structured data
3. Python and MATLAB APIs (and C++ and Java)
4. Data Services (Process variables with arguments)
5. Standardized Scientific Types
6. Dynamic typing
7. Introspection interface
8. New smart process database

```
$ pvget XCOR:LI24:900:TWISS
structure
    double energy 5.00512
    double psix 37.7625
    double alphax 13.6562
    double betax -2.78671
    double etax -0.00698294
    double etaxp 0.00107115
    double psiy 31.9488
    double alphay 116.762
    double betay 5.2592
    double etay 0
    double etayp 0
```

Figure: pvAccess getting PV of a structure of optics parameters. In this case a standard “Normative Type” type was not used, so the raw structure is displayed

Faster. Structured data. Metadata. Easier.

EPICS 7 Image Type (NTNDArray) Example

Helping ML answer e.g. – What does the cathode look like when the photon flux intensity is high

```
16:57|bmartins@bmartins-pc ~$ pvget -r 'field()' Eiger1M:pva1:Image
Eiger1M:pva1:Image
epics:nt/NTNDArray:1.0
    union value
        ushort[]  [1,0,1,0,0,0,0,0,0,0,0,0,0,[illegible]0,0,0,0,0,2,1,0,0,0,1,0
,0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,[illegible]1,1,0,0
,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,1,0,0,0,0,0,0,[illegible]
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1,1,0,0,0,0,0,0,[illegible]
,0,0,0,0,0,0,0,1,1,0,0,0,1,1,0,0[illegible]
A lot of  data snipped
[illegible],0,0,0,3,1,1,1,2,2,0,2,0,1,0,1,0,1,1,3,1,0,0,0,1,
2,2,1,0,0,0,0,2,1,0,0,0,1,2,2,1,0,3,0,0,0,2,0,3,1,0,2,0,0,2,0,1,0,1,0,0,1,0,2,0,
0,0,0,0,1,1,2,0,0,0,0,3,1,0,0,0,1,0,1,1,1,1,0,0,3,0,1,1,0,0,1,0,1,0,0,0,2,1,0,0,
1,2,0,1,2,2,1,0,3,0,1,0,0,1,1,0,0,1,0,0,0,0,0,2,0,0,0,0,1,0,0,1,0,0,0,0,1,0,0,1,
1,0,0,0,1,0,0,0,0,1,0,0,2,1,0,0,0,0,1,1,0,1,1,2,1,1,1,0,0,1]
    codec_t codec
        string name
        any parameters
            (none)
    long compressedSize 2193900
    long uncompressedSize 2193900
    dimension_t[] dimension
        dimension_t
            int size 1030
            int offset 0
            int fullSize 1030
            int binning 1
            boolean reverse false
        dimension_t
            int size 1065
            int offset 0
            int fullSize 1065
            int binning 1
            boolean reverse false
    int uniqueId 120682
    time_t dataTimeStamp 1995-10-08T16:57:40.297 0
    epics:nt/NTAttribute:1.0[] attribute
        epics:nt/NTAttribute:1.0
            string name ColorMode
            any value
                int  0
            string descriptor Color mode
            int sourceType 0
            string source Driver
    string descriptor
    alarm_t alarm NO_ALARM NO_STATUS <no message>
    time_t timeStamp 1995-10-08T16:57:40.301 0
    display_t display
        double limitLow 0
        double limitHigh 0
        string description
        string format
        string units
```

Command line request for a given image PV by name.

Data Type Identifier

Raw image data.

Widget learns how to display image from metadata

The image meta data; giving parameters to interpret data in the *value* field, and other information.

Figure: A screenshot of the output of the EPICS 7 "pvget" command, for an image.

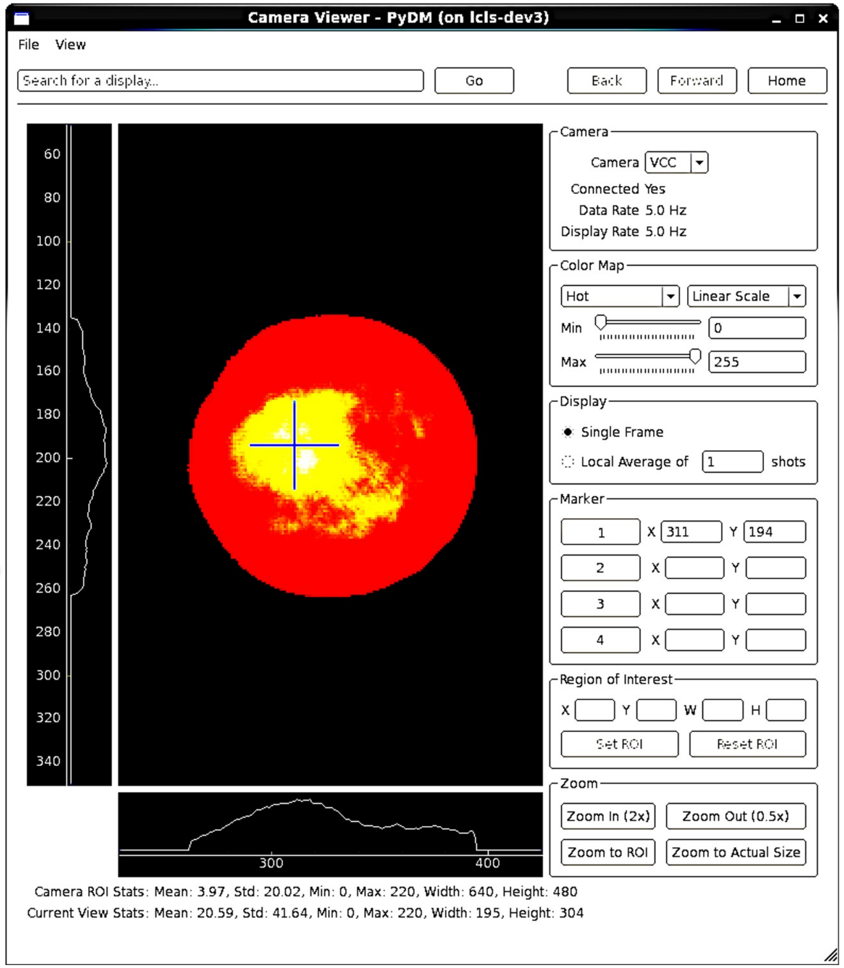


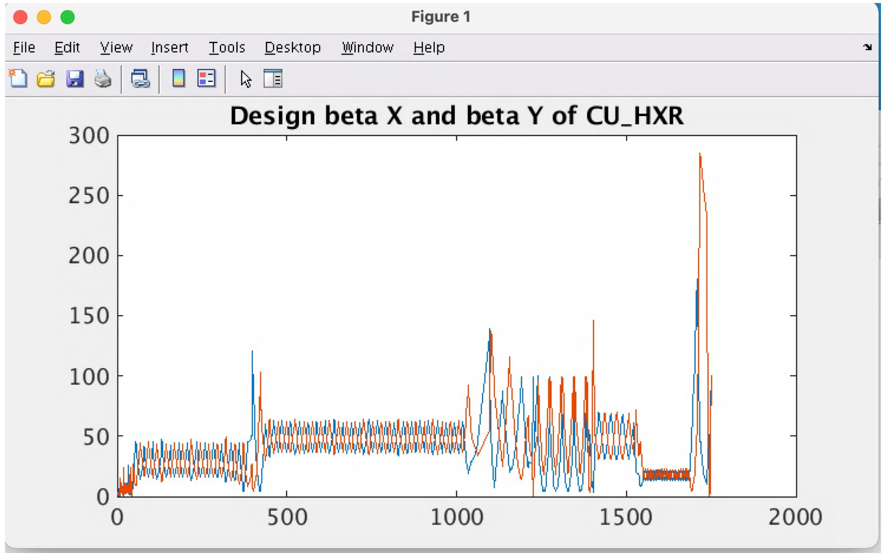
Figure: A screenshot of PyDM displaying a beam profile image.

B. Martins, BNL, M. Gibbs SLAC

Model Data through EPICS 7 Control System

Twiss Parameters

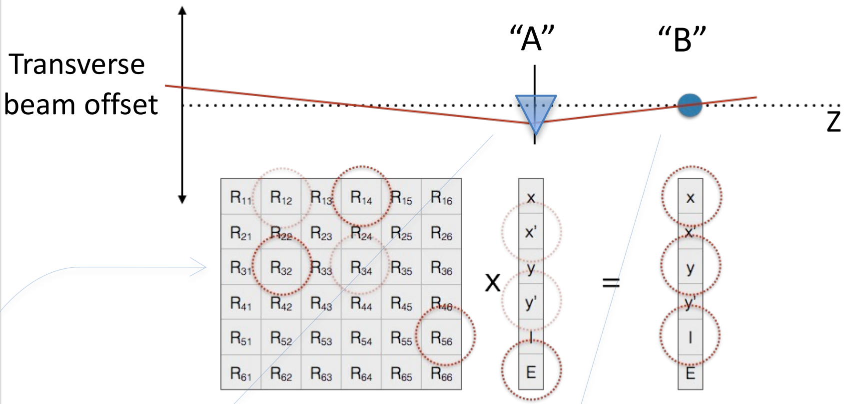
```
>> twiss_d=eget('BMAD:SYS0:1:CU_HXR:DESIGN:TWISS',{'provider','pva'});
>> plot(twiss_d.s, twiss_d.beta_x);
>> hold on
>> plot(twiss_d.s, twiss_d.beta_y);
>> title('Design beta X and beta Y of CU\_HXR');
```




```
>> twiss_l = eget('BMAD:SYS0:1:CU_HXR:LIVE:TWISS',{'provider','pva'});
>> twiss_l(find(strcmp(twiss_l.device_name,'BPMS:LI21:233')), …
[2,6,7,11,12])
ans =
        1x5 table

device_name      alpha_x beta_x alpha_y beta_y
_______________  _______ ______ _______ ______

"BPMS:LI21:233"  1.9887  7.9179 -0.20858 4.8231
```

Structured data, presented as table by EPICS

Transfer Matrices, example shows A to B



Process Variable value, computed subject to user's arguments

```
$ eget -s XCOR:IN20:491:RMAT -a b BPMS:IN20:525
```

0.669591	0.694604	0	0	-3.08532e-19	2.41325e-19
-0.570851	0.901275	0	0	-1.23627e-19	1.45491e-19
0	0	1.33379	0.966896	0	0
0	0	0.358415	1.00957	0	0
-2.29302e-24	8.92892e-20	0	0	1	1.20724e-05
1.00974e-28	0	0	0	0	1

Figures: EPICS 7 modelling service giving orbit response matrices and Twiss parameters for given devices - the basis of 90% of emittance minimization applications like feedback, steering, bumps, etc.

Accelerator Infrastructure data through EPICS

Data Services now trivial to implement in EPICS. In Python, Java, C++, even MATLAB

Directory Services

```
# The names of PVs, by device name pattern:
$ eget -s ds -a name=XCOR:LI21:135:%
       name
       XCOR:LI21:135:ABORT
       XCOR:LI21:135:ACCESS
       XCOR:LI21:135:ALLFUNCGO
       XCOR:LI21:135:BACT
       XCOR:LI21:135:BACTFO
… (many rows snipped)
```

```
# Regular expression (restrict to sectors LI25-LI29)
$ eget -s ds -a regex='XCOR:LI2[5-9]:.*:BDES'
…
# Device names of the instruments in the laser heater region
$ eget -s ds -a etype INST -a tag LSRHTR -a show dname
…
# A recent search for invalid data in corrector PVs
$ eget -tTs ds -a name %COR:LTU%:%:%DES | \
eget -p ca -f - | grep nan
XCOR:LTU1:558:BDES nan
XCOR:LTU1:558:IDES nan
…
```

Oracle Database accessed through Control System:

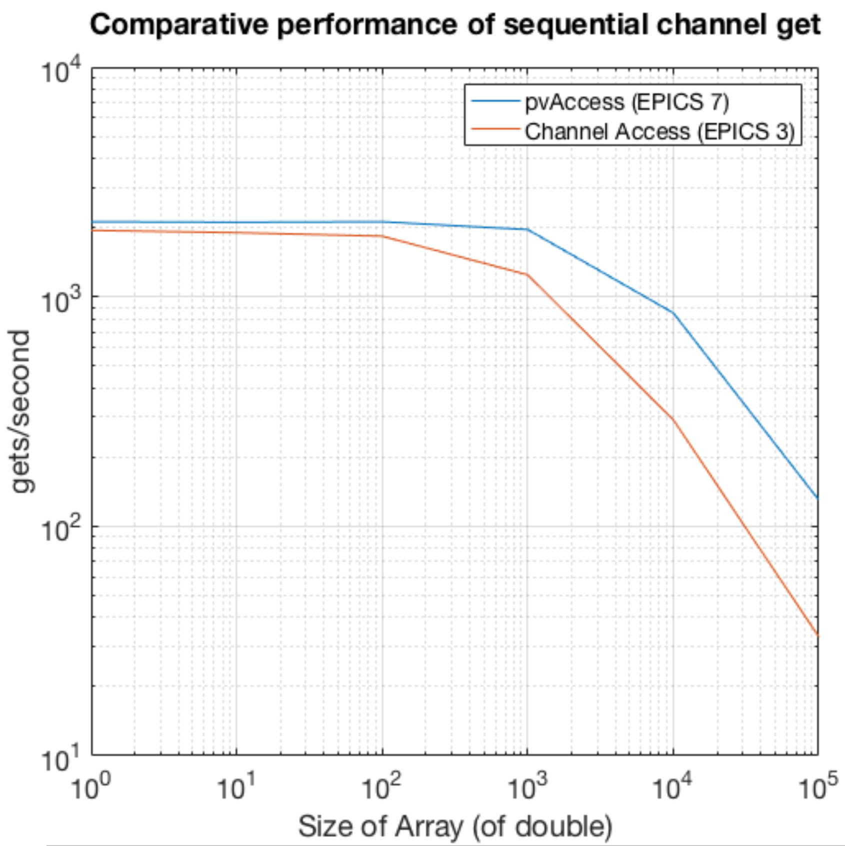
```
[physics@lcls-srv01 ~/greg]$ pvcall INFR:SYS0:1:SC_HXR | more

 ELEMENT              DEVICE KEYWORD  AREA SECTOR MODEL   SUML_M LINACZ_M S_DISPLAY                       OPSTAT OBS
 SOL1BKB        SOLN:GUNB:100    SOLE  GUNB    S00   MAD -0.071705 -10.1164  -10.1164         "Commissioned, Online"   N
CATHODEB        CATH:GUNB:100    INST  GUNB    S00   MAD         0 -10.0447  -10.0447                          " "   N
   CQ01B      QUAD:GUNB:212:1    QUAD  GUNB    S00   MAD   0.24653 -9.79814  -9.79814         "Commissioned, Online"   N
   SQ01B      QUAD:GUNB:212:2    QUAD  GUNB    S00   MAD   0.24653 -9.79814  -9.79814         "Commissioned, Online"   N
   SOL1B        SOLN:GUNB:212    SOLE  GUNB    S00   MAD   0.24653 -9.79814  -9.79814         "Commissioned, Online"   N
   VV01B "- NO EPICS NAME -"     INST  GUNB    S00   MAD   0.38748 -9.65719  -9.65719                          " "   N
   XC01B        XCOR:GUNB:293    XCOR  GUNB    S00   MAD    0.4845 -9.56017  -9.56017         "Commissioned, Online"   N
   YC01B        YCOR:GUNB:293    YCOR  GUNB    S00   MAD    0.4845 -9.56017  -9.56017         "Commissioned, Online"   N
```

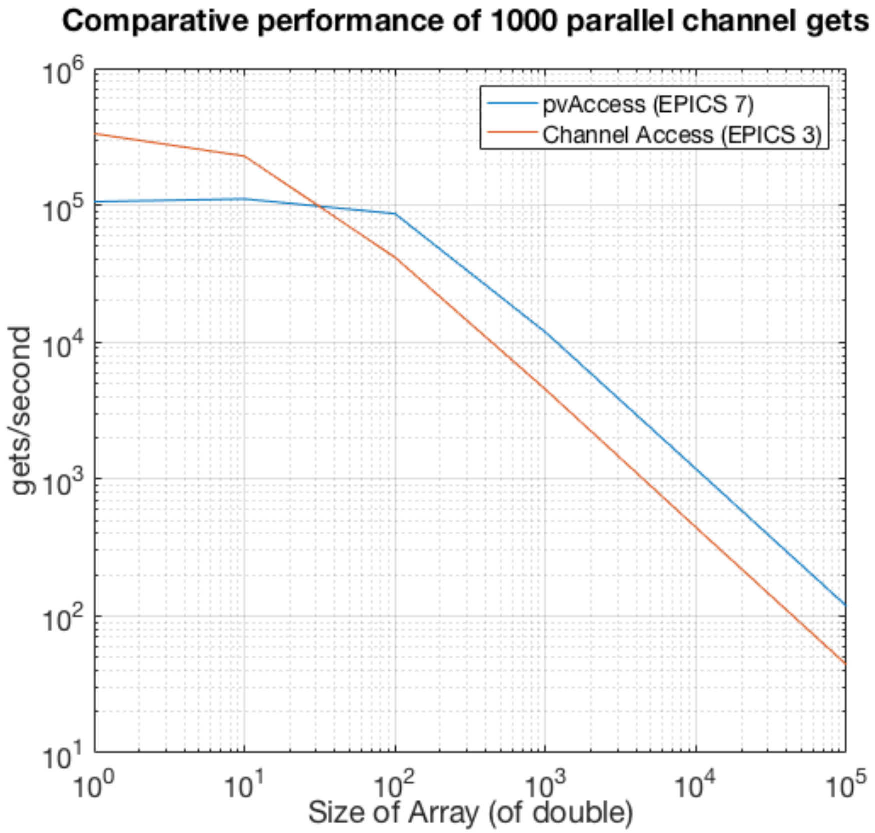
Figure: Access to Oracle gives device infrastructure, magnet calibrations, drawing names, etc.
Will be used in LCLS-II for such things as cryogenic plant system hierarchy etc.

Greg White, Murali Shankar SLAC

EPICS 7 network performance (pvAccess)

Comparison of new data protocol included in EPICS 7 compared to EPICS 3





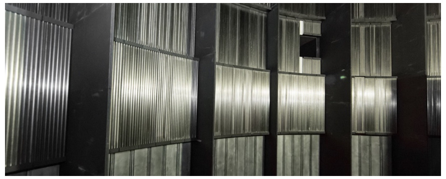
pvAccess is significantly faster than EPICS 3 at big data. CA is faster at many small ops in parallel.

Use Cases Indication: Use pvAccess for machine beam synchronous accelerator data, experimental data, camera data, structured data or data & metadata. Maybe use CA for very full synoptic displays and compatibility with legacy display mgr.

Matej Sekoranja, Cosylab

SNS uses EPICS v7 for high throughput event readout, of structured PV data.

Neutron Detector



SNS SEQUOIA detector ray consisting of >800 ^{3}He tubes covering a solid angle of 0.8 steradian

nED

- pvAccess Server
- pvDatabase
- Classic EPICS IOC

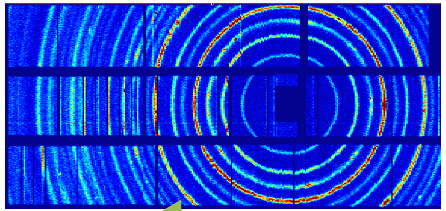
nED provides the driver interface to the detector electronics and streams experiment data using pvAccess

pvAccess streams event data at 80 Mbytes / sec

```
CHANNEL   : BL14B:Det:Neutrons
STATE     : CONNECTED
ADDRESS   : 10.111.29.150:57267
structure
    time_t timeStamp
        long secondsPastEpoch
        int nanoseconds
        int userTag
    epics:nt/NTScalar:1.0 proton_charge
        double value
    epics:nt/NTScalarArray:1.0 time_of_flight
        uint[] value
    epics:nt/NTScalarArray:1.0 pixel
        uint[] value
    epics:nt/NTScalarArray:1.0 sample_a1
        uint[] value
    epics:nt/NTScalarArray:1.0 sample_a2
        uint[] value
    epics:nt/NTScalarArray:1.0 sample_a8
        uint[] value
    epics:nt/NTScalarArray:1.0 sample_a19
        uint[] value
    epics:nt/NTScalarArray:1.0 sample_a48
        uint[] value
    epics:nt/NTScalarArray:1.0 sample_b1
        uint[] value
    epics:nt/NTScalarArray:1.0 sample_b8
        uint[] value
    epics:nt/NTScalarArray:1.0 sample_b12
        uint[] value
    epics:nt/NTScalarArray:1.0 position_index
        uint[] value
    epics:nt/NTScalarArray:1.0 position_x
        uint[] value
    epics:nt/NTScalarArray:1.0 position_y
        uint[] value
    epics:nt/NTScalarArray:1.0 photo_sum_x
        uint[] value
    epics:nt/NTScalarArray:1.0 photo_sum_y
        uint[] value
```

The EPICS 7 data structure includes an array of pixels and a corresponding array of times of flight for neutron event. Additional fields record accelerator metadata e.g. pulse information and detector diagnostic information.

CS-Studio



Diffraction rings of a powder sample at SEQUOIA

ADnED

- CA Server
- areaDetector data processing
- pvAccess Client

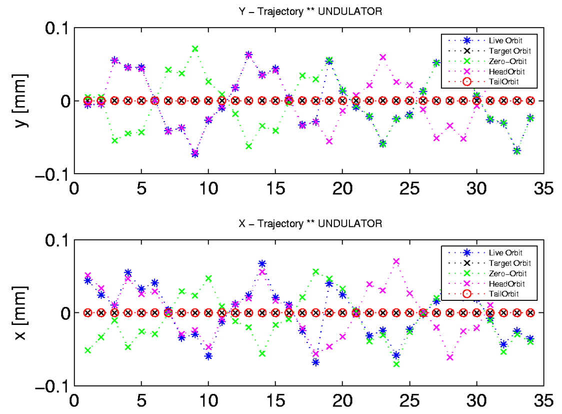
ADnED, a pvAccess client, generates online histograms and counting statistics from the nED data stream and serves them using the CA protocol to clients including CS-Studio

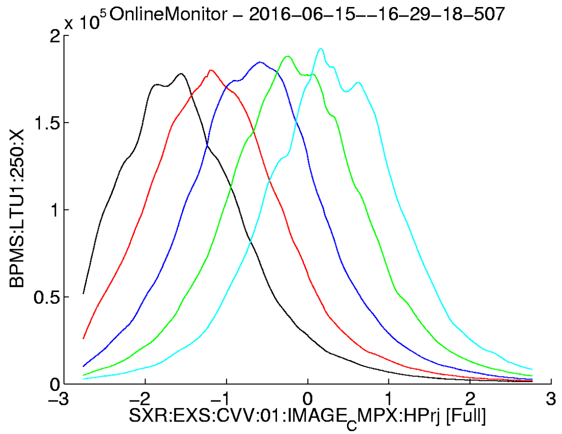
SNS/HFIR Outcomes:

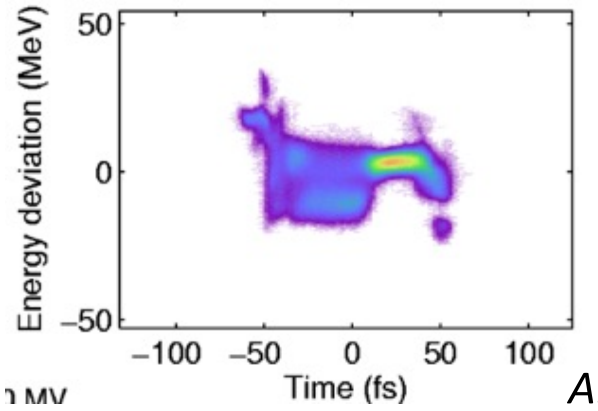
1. v7 in use at 21 Beamlines
2. 10M events/s
3. Excellent reliability.

M. Pearson, K. Kasemir, S. Hartman, ORNL

EPICS 7 – Beam Synchronous Data – and Event building







A. Lutman

Problem: assemble pulse synchronous data & machine meta data. Complex, incomplete.

⇒ In Accelerator – Beam Synchronous Acquisition system (BSA).

⇒ In Experimental support controls, Beam Line Data System (BLD).

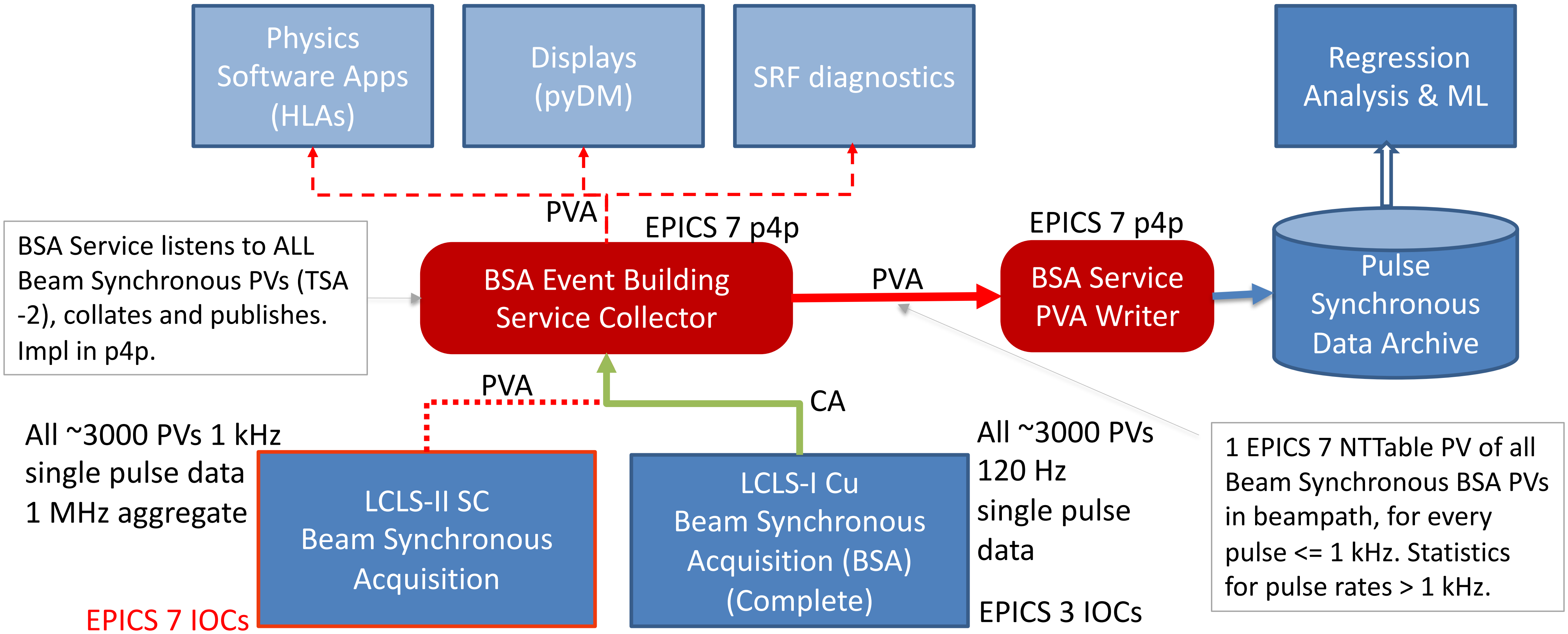
- Requirement: Make it easy. Support Regression Analysis and ML (BSA Service), and event build for FEL tuning

Objective: Fast transfer of pulse identified data as atomic structures of the measured data and their “meta-data” for analysis, experiment tuning, and ML.

Alberto Lutman, SLAC

“All the data, all the time” at SLAC’s LCLS

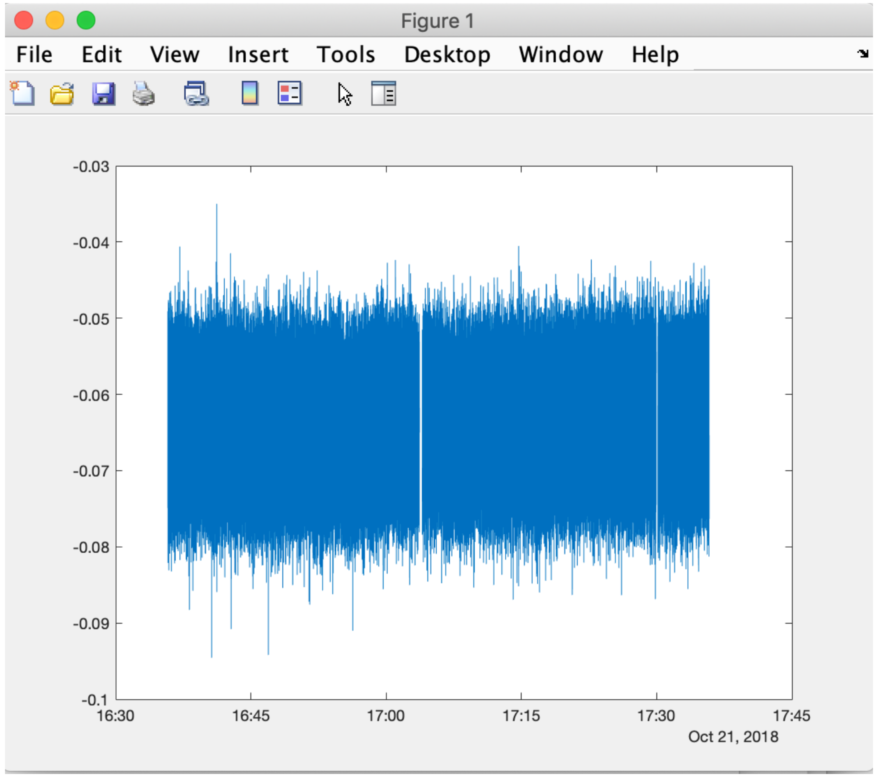
Figure: Accelerator Event Building Service collects all bunch-by-bunch data, lines up by bunch ID, tags with accelerator meta data, stream to clients, and archives for Machine Learning and diagnostics. ~2.5 GB/hr to HDF5 continuously



https://www.slac.stanford.edu/grp/ad/docs/model/matlab/bsd.html, M. Davidsaver, G. White

All past beam synchronous data, available to Matlab

```
>> % Load data of all beam synchronous PVs, for hour following Oct 22 2018, 00:35.49
>> load('~/Development/bsd/CU_HXR_20181022_003549.h5','-mat’);
>> % Make a single vector of real seconds POSIX time, and convert to local time
>> ts=double(secondsPastEpoch)+double(nanoseconds)*1e-9;
>> t=datetime(ts,'ConvertFrom','posixtime','TimeZone','America/Los_Angeles’);

>> % E.g. Plot every pulse at one BPM PV over the hour
>> plot(t,BPMS_IN20_525_X);
```

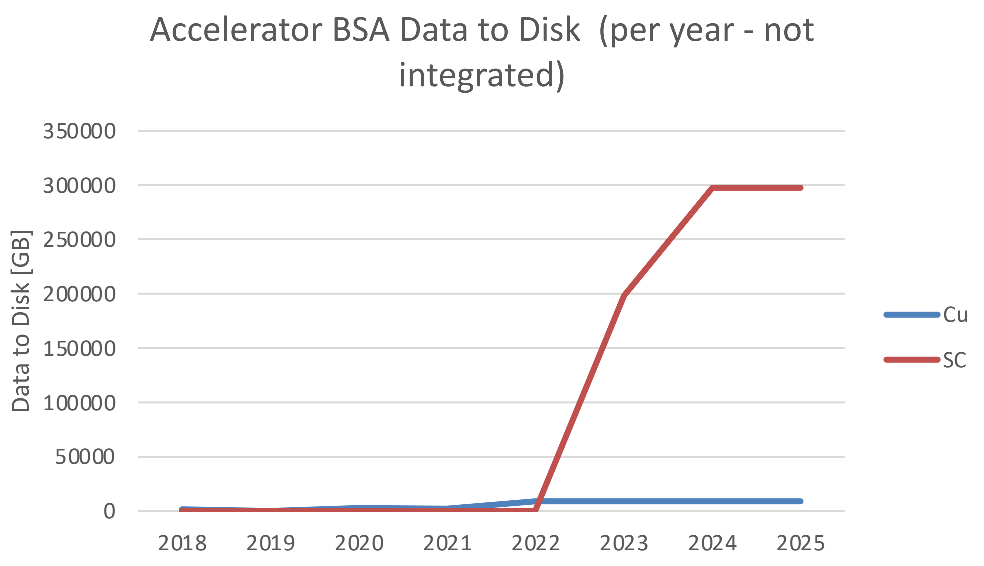


All the data of every beam synchronous device, at every pulse, continuously 24x7, for post-facto ML and other numerical analysis.

https://www.slac.stanford.edu/grp/ad/docs/model/matlab/bsd.html, *M. Davidsaver, G. White*

Data will drive analytics. Both data generated and storage requirement are set to grow enormously

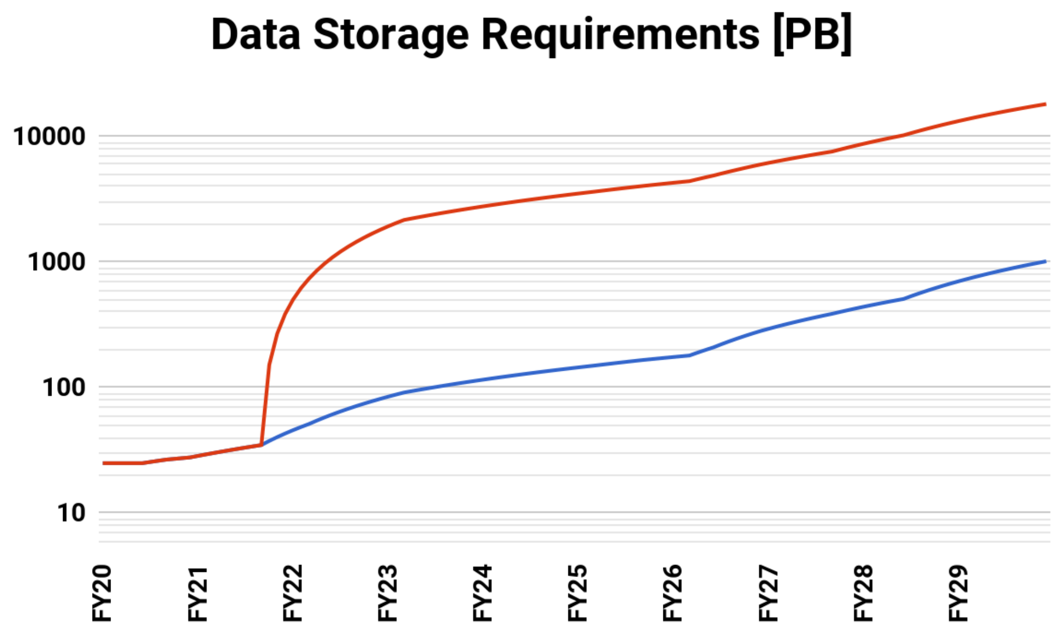
LCLS Accelerator Data



G. White, AD

Optimistic: Assumes continuous 24/7 data recording and file format does include data summary tuple for characterizing data >1 kHz
Data roll-up DOES NOT Include LCLS-HE assumed coming after 2026
Data roll-up DOES NOT include Camera image data

LCLS Experiment Data

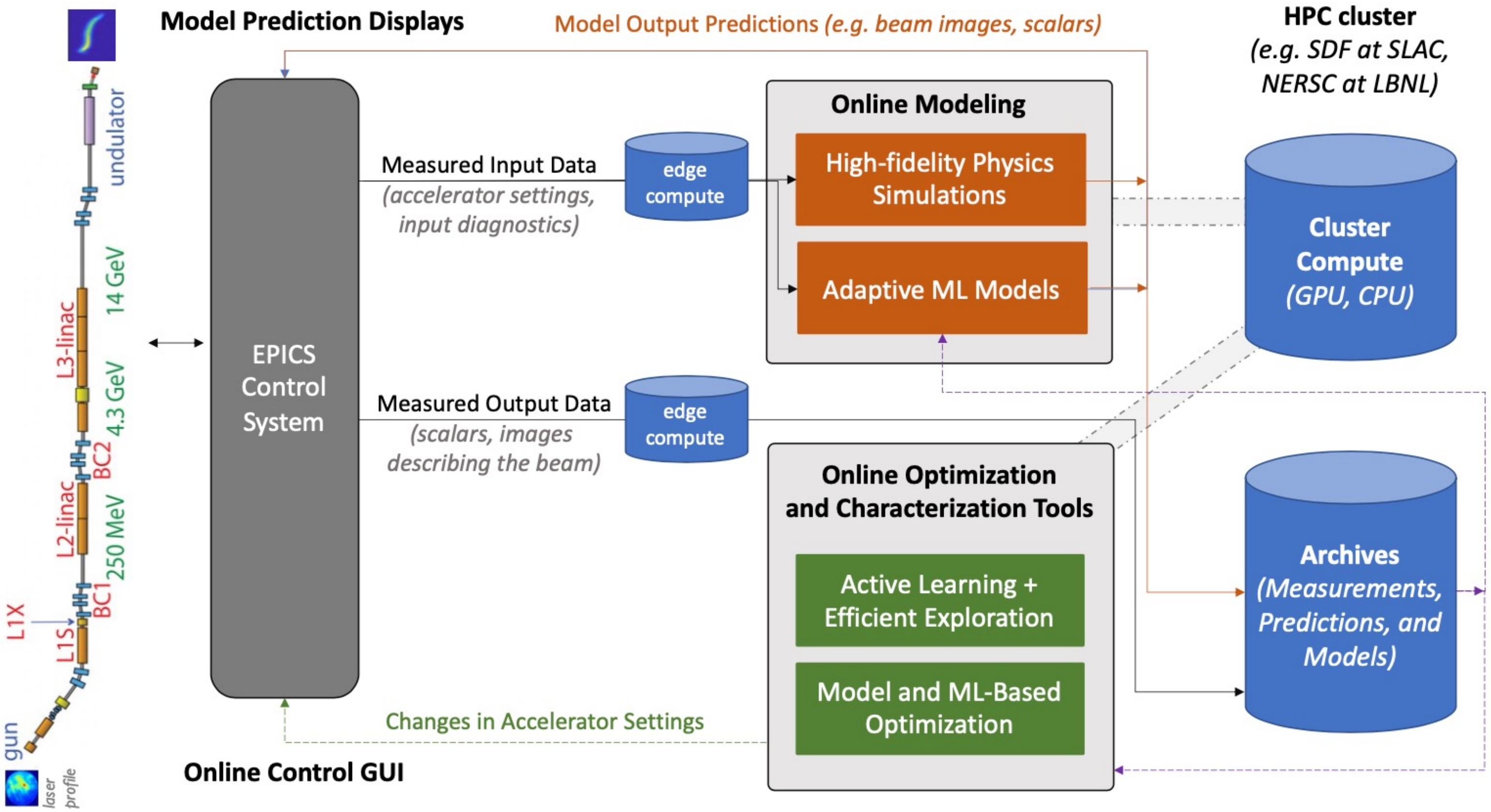


Jana Thayer, LCLS

Accelerator data storage size will be large for us, but small compared to LCLS experiment (300 TB << 1000 PB)

Machine Learning and Multi-particle simulation computing pipeline in development at SLAC

Figure: Computer architecture of ML and physics simulation for modelling and tuning the online accelerator in real time.



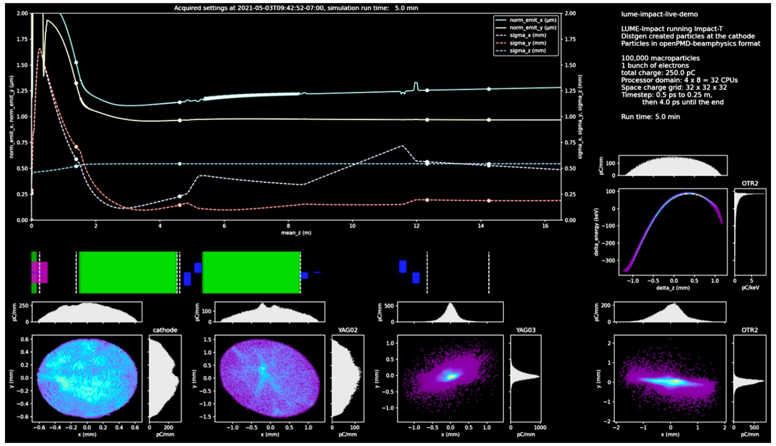
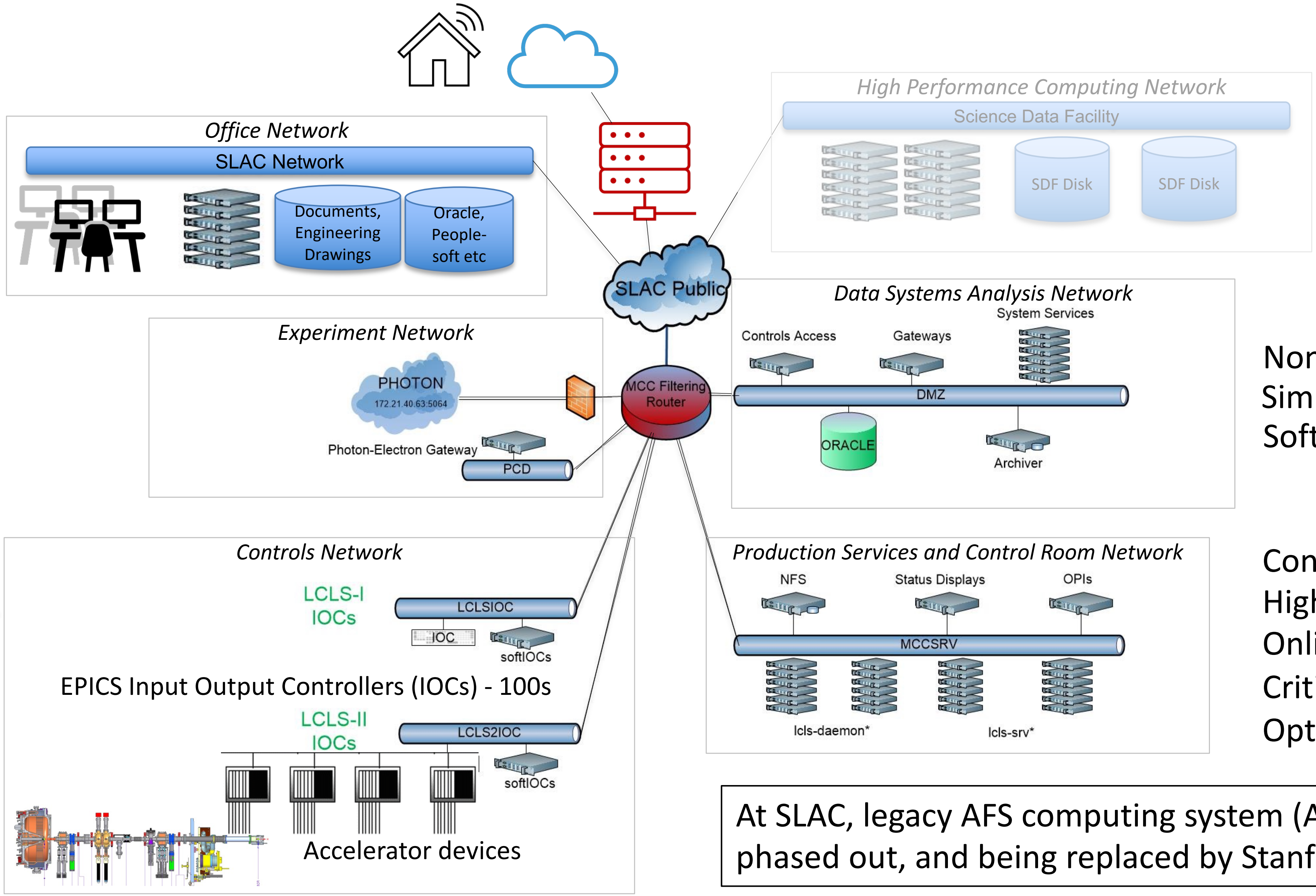


Figure: Screenshot of dashboard of LCLS LUME start-to-end multiparticle simulation. FY23-34 productize for active tuning, identify loss sources, plan transport strategies.

A. Edelen, C. Mayes, J. Garrahan, SLAC

Accelerator Computing Architecture Early 2023



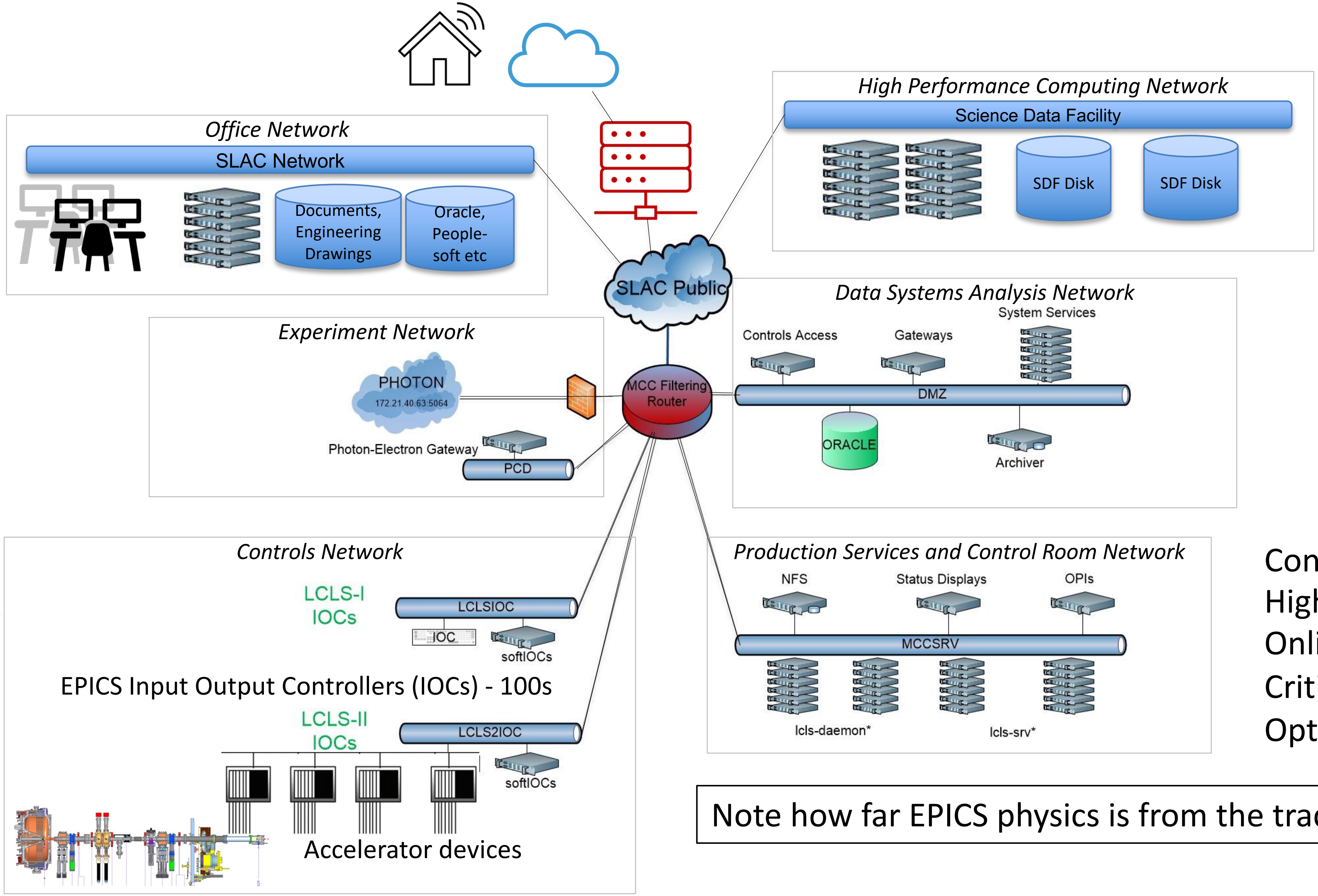
Big Data stores
Experiment Data Analysis
Accelerator Physics Analysis
Multi-particle Tracking (s-2-e)
Machine Learning Training

Non-critical Data Services
Simulation
Software development

Control room displays
High Level Apps execution
Online modeling
Critical Data Services
Optimization, ML and slow feedback

At SLAC, legacy AFS computing system (Accelerator Data Systems) being phased out, and being replaced by Stanford SLAC Science Data Facility (S3DF).

Accelerator Computing Architecture Late 2023



Big Data stores
Experiment Data Analysis
Accelerator Physics Analysis
Multi-particle Tracking (s-2-e)
Machine Learning Training
Non-critical Data Services
Simulation
Software development

Control room displays
High Level Apps execution
Online modeling
Critical Data Services
Optimization, ML and slow feedback

Note how far EPICS physics is from the traditionally sequestered control system

Summary: EPICS is changing to include AI/ML and HPC

- Scope: Controls → += Physics

  SCADA, optimized control, synoptic display → AI/ML, linear modelling, multi-particle modeling, physics applications and daemons

- EPICS Version: 3 → 7

  Scalars & waveforms, get/set/monitor → Structured data, fast protocol for complex data, RPC services, (relatively) big data & datastores

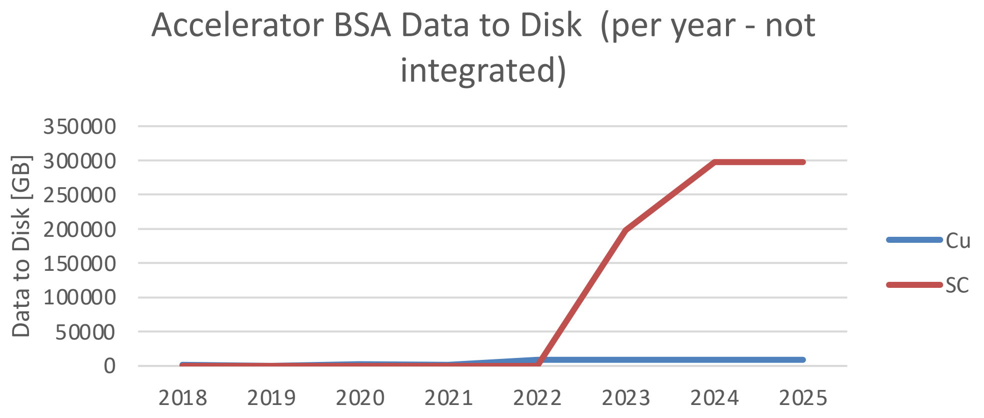
- Network: Controls → += HPC

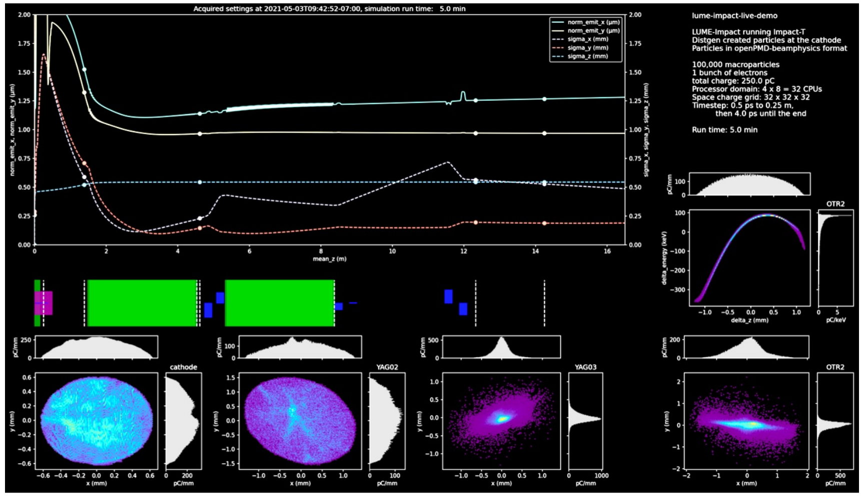
  ML needs HPC, especially training. Multiparticle needs HPC. HPC tends to be offline, away from controls. Results PV write → Authentication and Authorization.

EPICS 2023 ML intro talk ends.

New Data Sizes, ML and Online Multi-particle Model, their Implications and Requirements on Architecture

<u>Premise: Future accelerator will be optimized by Data Systems, ML and HPC Model Systems</u>





Implications

- Data Science driven accelerator tuning.
  BSA Data store, ML on SDF. ML on prod -> network, Physics Analysis Env & documentation
- Complicates requirement to produce quick "feedback" to experiment
- Many controls and physics processes, not just IOCs --> PVA gateways, EPICS 7 IOCs, production HLAs online
- Production at >1 network. Dev and SDF now incl.
- Development in DMZ and SDF networks, not prod
- New maths and algorithms to be prototyped -> Matlab from home, PVA matlab, python env
- Data access across networks (prod, dev/AFS, SDF)
- Large amount of data to be stored, managed, understood
- Fast and easy to use.

Consequent Requirements

1. Production architecture that includes S3DF.
2. Simple, fast, automated Workflows and Data Flows