Control Infrastructure, Data Analysis and Simulation Feedback at the NDCX-II Accelerator Facility

BERKELEY LAB

T. Schenkel P. A. Seidl A. Persaud Q. Ji W. L. Waldron M. Stettler M. Regis V. Vytla

J. J. Barnard A. Friedman D. P. Grote

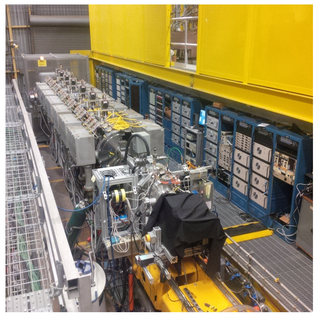
U.S. DEPARTMENT OF ENERGY

LLNL

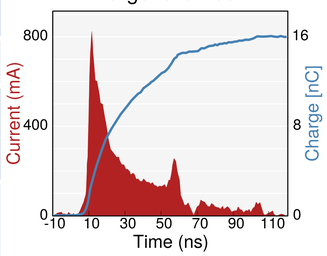
CASIS Workshop, Livermore, May 18th, 2016

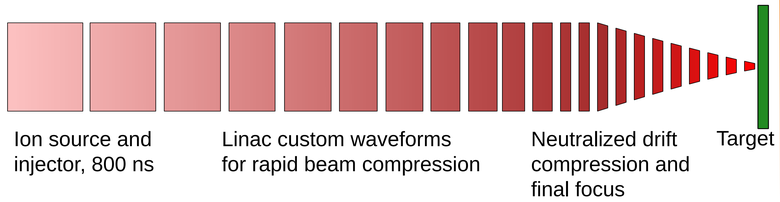
The NDCX-II Facility

	NDCX-II (goals)
Ions/pulse (total)	10^{10} (3×10^{11})
Pulse length	2 ns (~1 ns)
Typical spot size	1 mm^2
Ion species	He, (H, d, …)
E_{kin}	0.12 - 1.2 MeV
Energy spread	~10 %
Repetition rate	2/min
Target temperature	< 0.1 eV (~1 eV)
Radiation environment at the target	Benign, no shielding required
Heated volume	~1 mm^2 × 5 µm = 5×10^6 $µm^3$

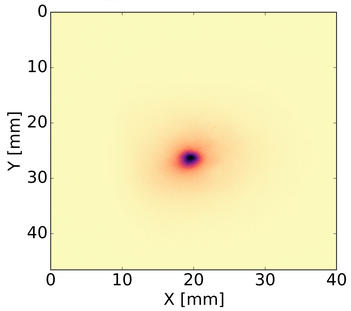


Target chamber

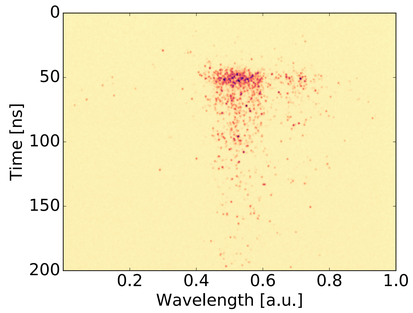




CCD camera



Streak camera



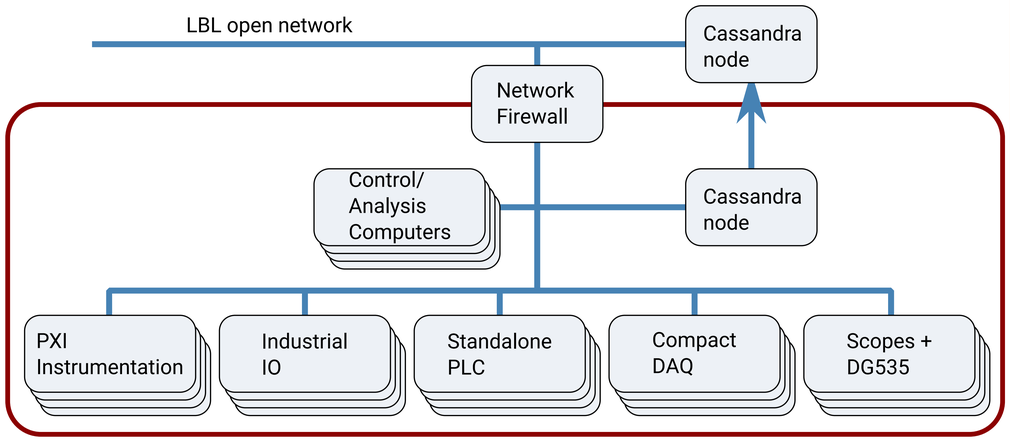
Research Topics:

- Defect dynamics in materials
- Intense beams and beam-plasma physics
- Materials studies for fusion reactors
- Ion heating of matter (Warm-Dense Matter)
- Extreme chemistry

Controls and Database

Pulsed	Non-pulsed
Triggers(NI-PIX 6652/ GFT 9404)	PLC logic (Relays, AI, AO)
Scopes (NI-PXI 5105/5133)	NI-DAQs
Scopes	NI-FieldPoints
DG535	
Cameras	

Approximately 160 scope channels, 80 delay pulse generator channels, 50 power supplies with around 450 analog and digital IO ports.



Cassandra Database:

- Highly scalable; very easy to add nodes to a running system
- Designed to handle large amounts of data
- Optimized for fast writes and sequential reads
- CQL query language similar to SQL-DB
- Python driver available, created a LabVIEW driver

cassandra

Message broker ZMQ:

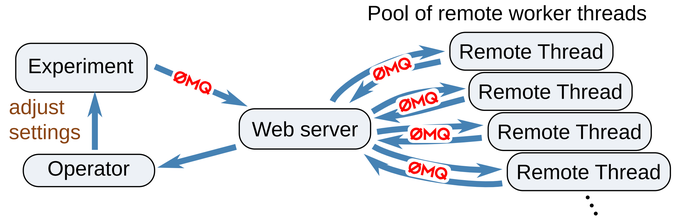
ØMQ

- Available for many languages (including LabVIEW and Python)
- Fast and stable (originated in finance industry)
- Used for remote control of LabVIEW

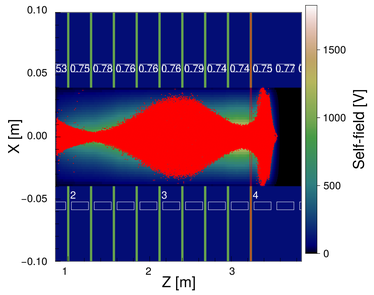
```
class Shot(Model):
    lvtimestamp = columns.BigInt(primary_key=True)
    devicename = columns.Text(primary_key=True)
    data = columns.Text()
    settinghash = columns.Text()
    version = columns.Integer()

class Setting(Model):
    hash = columns.Text(primary_key=True)
    data = columns.Text()
    version = columns.Integer()

class Data(Model):
    devicename = columns.Text(partition_key=True)
    date = columns.Integer(partition_key=True)
    eventtime = columns.DateTime(primary_key=True)
    value = columns.Float(primary_key=True)

class Comments(Model):
    month = columns.Integer(primary_key=True)
    daytime = columns.Integer(primary_key=True)
    species = columns.Text(primary_key=True)
    comment = columns.Text()
    sample = columns.Text()
    shots = columns.Set(colums.BigInt)
```

Simulation Feedback

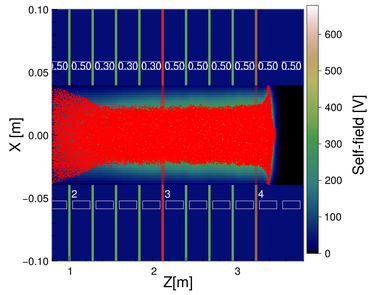
- Simulate complete beamline using Warp
- Use voltage waveforms, timing, etc. from experiment
- Simulate each shot live using 2 desktop computers and a small ZMQ python script, upload results to web page



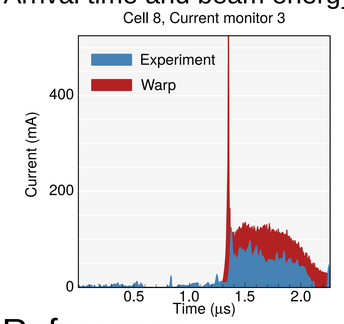
Warp simulation before tuning

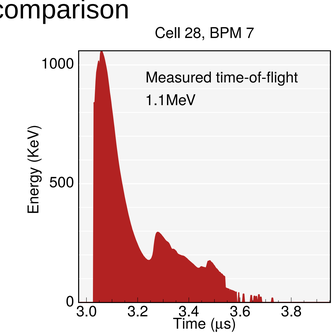


after tuning



Arrival time and beam energy comparison





References:

- P.A. Seidl et al., NIM A 800 98 (2015)
- controls: arxiv.org/abs/1506.04212
- Phys. Proc. 66, 604 (2015)
- warp: http://warp.lbl.gov
- https://bitbucket.org/berkeleylab/atap-labview-cassandra-driver

This work was supported by the Office of Science of the US Department of Energy under contracts DE-AC02–05CH11231 (LBNL), DE-AC52-07NA27344 (LLNL) LLNL-POST-692079