PHYSICS APPLICATION INFRASTRUCTURE DESIGN FOR FRIB DRIVER LINAC*

G. Shen#, E. Berryman, Z. He, M. Ikegami, D. Liu, D. Maxwell, V. Vuppala,
FRIB, Michigan State University, East Lansing, 48864 USA

Abstract

FRIB (Facility for Rare Isotope Beams) is a new heavy ion accelerator facility to provide intense beams of rare isotopes, and currently under active construction at MSU (Michigan State University). Its driver linac accelerates all stable ions up to uranium, and targets to provide a CW beam with the energy of 200MeV/u and the beam power of 400 kW. A new software infrastructure for high level control and physics application is under development to support the beam commissioning and operation, which is a 3-tier structure consisting of upper level, middle layer, and low level respectively. The detailed design and its current status are described in this paper.

INTRODUCTION

FRIB [1] is a new project under cooperative agreement between US Department of Energy and MSU. It is under construction on the campus of MSU and will be a new national user facility for nuclear science. Its driver accelerator is designed to accelerate all stable ions to energies > 200 MeV/u with beam pow er on the target up to 400 kW as shown in Table 1 [2].

Table 1: FRIB Driver Accelerator Main Parameters

Parameter	**Value**	**Unit**
Primary beam ion species	H to ^{238}U	
Beam kinetic energy on target	> 200	MeV/u
Maximum beam power on target	400	kW
Beam duty factor	100	%
Beam current on target (^{238}U)	0.7	emA
Beam spot size on target (90%)	1	mm
Driver linac beam-path length	517	m
Average uncontrolled beam loss	< 1	W/m

The layout of FRIB is as shown in Fig. 1, which consists of two ECR (Electron Cyclotron Resonance) ion sources, a low energy beam transport with a pre-buncher and a chopper, a RFQ (Radio Frequency Quadrupole) linac, LS1 (Linac segment 1) to accelerate the beam up to > 15 MeV/u, LS2 and LS2 to accelerates the beam > 200 MeV/u, two folding segments to confine the footprint and facilitate beam collimation, and a beam delivery system to transport to the target. The beam is stripped to higher charge states after LS1 section.

*Work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661
#shen@frib.msu.edu

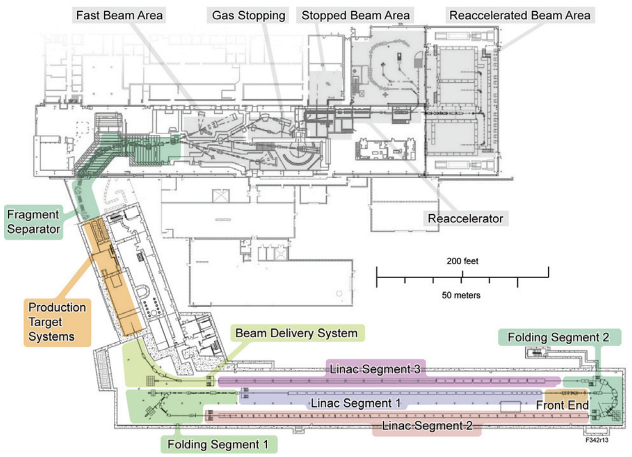


Figure 1: Layout of the FRIB driver accelerator, target and fragment separator as shown in colored area.

ARCHITECTURE

A distributed high level controls environment has been adopted for its beam commissioning and operation as shown in Fig. 2. It is a 3-tier structure consisting of upper level, middle layer, and low level respectively.

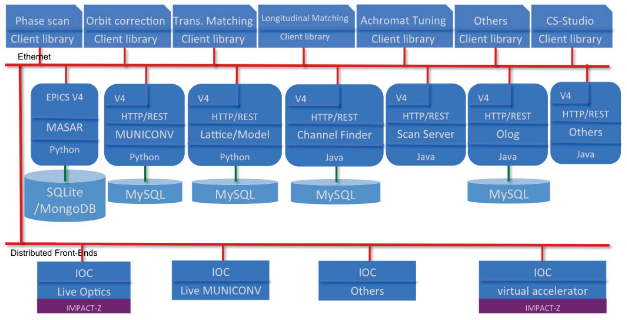

Phase scan Client library
Orbit correction Client library
Trans. Matching Client library
Longitudinal Matching Client library
Achromat Tuning Client library
Others Client library
CS-Studio Client library
Ethernet
EPICS V4 MASAR Python
V4 HTTP/REST MUNICONV Python
V4 HTTP/REST Lattice/Model Python
V4 HTTP/REST Channel Finder Java
V4 HTTP/REST Scan Server Java
V4 HTTP/REST Olog Java
V4 HTTP/REST Others Java
SQLite /MongoDB
MySQL
MySQL
MySQL
MySQL
Distributed Front-Ends
IOC Live Optics IMPACT-Z
IOC Live MUNICONV
IOC Others
IOC virtual accelerator IMPACT-Z

Figure 2: FRIB high level controls system architecture.

Since EPICS (Experimental Physics and Industry Control System) has been adopted as FRIB hardware control framework, the low level is represented as an EPICS IOC (Input Output Controller) for integration purpose. Some services are implemented in the low level layer, the middle layer consists some general services, and high level is mainly for OPI (Operator Interface).

LOW LEVEL SERVICE

At FRIB, the low level service is implemented as an EPICS IOC. One typical service is a virtual accelerator, which provides emulated accelerator behaviour such as setting and reading back a power supply setting, and reading beam orbit at each BPM position.

The architecture of FRIB virtual accelerator is as shown in Fig. 3. It uses IMPACT-Z [3] as underneath simulation engine. During first initialization, it reads accelerator from accelerator layout file and accelerator setting file, generates an IMPACT-Z input file, and initialize all hardware settings. Once the input file is ready, it executes the IMPACT-Z engine, reads simulation results, and updates the result. Every time when there is any setting change, it updates the lattice file, runs IMPACT-Z and updates the results.

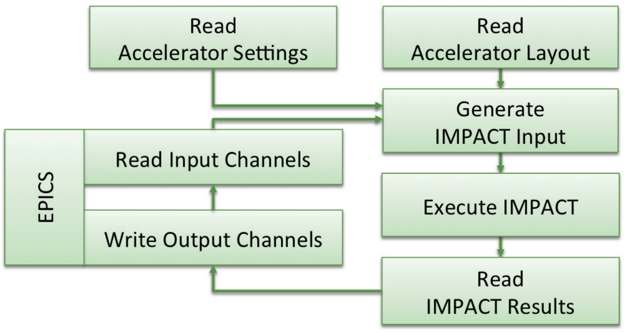


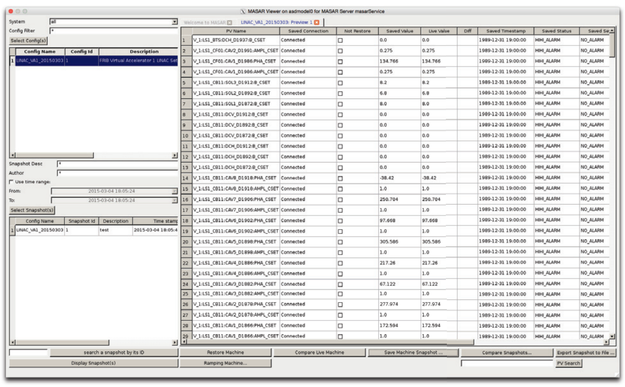
Figure 3: FRIB virtual accelerator.

MIDDLE LEVEL SERVICE

Currently, to support FRIB construction, beam commissioning and operation, a minimum set of middle layer services has been identified as below:

- eTraveler;
- Cable database;
- Configuration database (CDB);
- RBAC;
- Archiver Appliance;
- Channel Finder;
- Olog;
- Scan Server;
- MASAR;
- BEAST;
- META data store;
- Lattice/Model Web Server;
- Online model;
- Physics data server.

There are 2 different technologies adopted at FRIB to implement the middle layer services, which are EPICS V4 and web service respectively.



From: 2015-03-04 18:05:24
To: 2015-03-04 18:05:24

	Config Name	Config Id	Description
1	LINAC_VA1_20150303	1	FRIB Virtual Accelerator 1 LINAC Set

	Config Name	Snapshot Id	Description	Time stamp
1	LINAC_VA1_20150303	1	test	2015-03-04 18:05:4

	PV Name	Saved Connection	Not Restore	Saved Value	Live Value	Diff	Saved Timestamp	Saved Status	Saved Se
1	V_1:LS1_BTS:DCH_D1937:B_CSET	Connected	☐	0.0	0.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
2	V_1:LS1_CF01:CAV2_D1991:AMPL_CSET	Connected	☐	0.275	0.275		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
3	V_1:LS1_CF01:CAV1_D1986:PHA_CSET	Connected	☐	134.766	134.766		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
4	V_1:LS1_CF01:CAV1_D1986:AMPL_CSET	Connected	☐	0.275	0.275		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
5	V_1:LS1_CB11:SOL3_D1912:B_CSET	Connected	☐	8.2	8.2		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
6	V_1:LS1_CB11:SOL2_D1892:B_CSET	Connected	☐	6.8	6.8		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
7	V_1:LS1_CB11:SOL1_D1872:B_CSET	Connected	☐	8.0	8.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
8	V_1:LS1_CB11:DCV_D1912:B_CSET	Connected	☐	0.0	0.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
9	V_1:LS1_CB11:DCV_D1892:B_CSET	Connected	☐	0.0	0.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
10	V_1:LS1_CB11:DCV_D1872:B_CSET	Connected	☐	0.0	0.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
11	V_1:LS1_CB11:DCH_D1912:B_CSET	Connected	☐	0.0	0.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
12	V_1:LS1_CB11:DCH_D1892:B_CSET	Connected	☐	0.0	0.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
13	V_1:LS1_CB11:DCH_D1872:B_CSET	Connected	☐	0.0	0.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
14	V_1:LS1_CB11:CAV8_D1918:PHA_CSET	Connected	☐	-38.42	-38.42		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
15	V_1:LS1_CB11:CAV8_D1918:AMPL_CSET	Connected	☐	1.0	1.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
16	V_1:LS1_CB11:CAV7_D1906:PHA_CSET	Connected	☐	250.704	250.704		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
17	V_1:LS1_CB11:CAV7_D1906:AMPL_CSET	Connected	☐	1.0	1.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
18	V_1:LS1_CB11:CAV6_D1902:PHA_CSET	Connected	☐	97.668	97.668		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
19	V_1:LS1_CB11:CAV6_D1902:AMPL_CSET	Connected	☐	1.0	1.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
20	V_1:LS1_CB11:CAV5_D1898:PHA_CSET	Connected	☐	305.586	305.586		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
21	V_1:LS1_CB11:CAV5_D1898:AMPL_CSET	Connected	☐	1.0	1.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
22	V_1:LS1_CB11:CAV4_D1886:PHA_CSET	Connected	☐	217.26	217.26		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
23	V_1:LS1_CB11:CAV4_D1886:AMPL_CSET	Connected	☐	1.0	1.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
24	V_1:LS1_CB11:CAV3_D1882:PHA_CSET	Connected	☐	67.122	67.122		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
25	V_1:LS1_CB11:CAV3_D1882:AMPL_CSET	Connected	☐	1.0	1.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
26	V_1:LS1_CB11:CAV2_D1878:PHA_CSET	Connected	☐	277.974	277.974		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
27	V_1:LS1_CB11:CAV2_D1878:AMPL_CSET	Connected	☐	1.0	1.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
28	V_1:LS1_CB11:CAV1_D1866:PHA_CSET	Connected	☐	172.594	172.594		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM
29	V_1:LS1_CB11:CAV1_D1866:AMPL_CSET	Connected	☐	1.0	1.0		1989-12-31 19:00:00	HIHI_ALARM	NO_ALARM

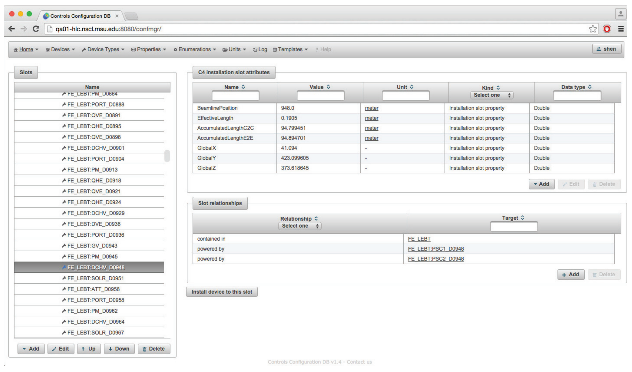
Figure 4: FRIB MASAR.

EPICS V4

The EPICS V4 has been determined as FRIB official framework for the middle layer service. The first server adopted was MASAR [4], which is a save and restore tool originally developed at NSLS II project, as shown in Fig. 4, which is one user client developed in PyQt. The development to integrate MASAR into CS-Studio is on going, and will be available shortly.

Web Service

Together with EPICS V4, web service is used to complement their weaknesses. Most databases related middle layer services listed above are developed as a RESTful web service except MASAR. One typical example is CDB as shown in Fig. 5. The CDB development is a collaboration effort with ESS (European Spallation Source) to manage machine configuration and installation information.


Controls Configuration DB
qa01-hlc.nscl.msu.edu:8080/confmgr/
Home
Devices
Device Types
Properties
Enumerations
Units
Log
Templates
Help
shen
Slots
C4 installation slot attributes
Select one
Add
Edit
Delete
Slot relationships
Up
Down
Install device to this slot
Controls Configuration DB v1.4 - Contact us

Name
FE_LEBT:PM_D0884
FE_LEBT:PORT_D0888
FE_LEBT:QVE_D0891
FE_LEBT:QHE_D0895
FE_LEBT:QVE_D0898
FE_LEBT:DCHV_D0901
FE_LEBT:PORT_D0904
FE_LEBT:PM_D0913
FE_LEBT:QHE_D0918
FE_LEBT:QVE_D0921
FE_LEBT:QHE_D0924
FE_LEBT:DCHV_D0929
FE_LEBT:DVE_D0936
FE_LEBT:PORT_D0936
FE_LEBT:GV_D0943
FE_LEBT:PM_D0945
FE_LEBT:DCHV_D0948
FE_LEBT:SOLR_D0951
FE_LEBT:ATT_D0958
FE_LEBT:PORT_D0958
FE_LEBT:PM_D0962
FE_LEBT:DCHV_D0964
FE_LEBT:SOLR_D0967

Name	Value	Unit	Kind	Data type
BeamlinePosition	948.0	meter	Installation slot property	Double
EffectiveLength	0.1905	meter	Installation slot property	Double
AccumulatedLengthC2C	94.799451	meter	Installation slot property	Double
AccumulatedLengthE2E	94.894701	meter	Installation slot property	Double
GlobalX	41.094	-	Installation slot property	Double
GlobalY	423.099605	-	Installation slot property	Double
GlobalZ	373.618645	-	Installation slot property	Double

Relationship	Target
contained in	FE_LEBT
powered by	FE_LEBT:PSC1_D0948
powered by	FE_LEBT:PSC2_D0948

Figure 5: Configuration database.

Beyond above information, CDB also stores parameters for FRIB MPS (Machine Protection System). A software-based RPS (Run Permit System) is under design to provide several key functions for safe operation of the machine. It monitors relevant inputs from hardware subsystems, which includes but not limited to cryomodule, vacuum, power supply/magnet, diagnostics, and target equipment to verify whether they are all in the desired state for the selected machine mode, beam mode, ion species with the charges, beam energy, and the beam destination, and establishes the Beam Permit signal. All parameters for RPS are saved in CDB such as allowed setting range for a device.

HIGH LEVEL CONTROL AND CONTINUOUS INTEGRATION

The OPIs for FRIB high-level controls will be developed using CS-Studio framework. Its HMI (Human Machine Interface) has been well defined and documented, and the validation will be done with the help of CS-Studio OPI validation tool.

At FRIB, an automatic software deployment procedure is adopted as shown in Fig. 6. A developer checks out source code from FRIB git [5] repository, commits and checks in any change for the review, and gets the change merged after it passes the review. Once the merge finished, a Continuous Integration process is triggered to

ISBN 978-3-95450-148-9

compile and test all changes, and a package could be created after passing all tests. Package manager could manually or automatically upload the new package, and deploy it.

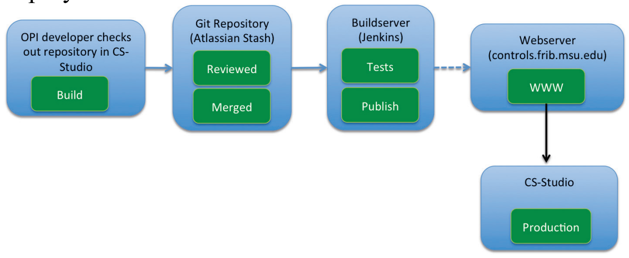

OPI developer checks out repository in CS-Studio
Build
Git Repository (Atlassian Stash)
Reviewed
Merged
Buildserver (Jenkins)
Tests
Publish
Webserver (controls.frib.msu.edu)
WWW
CS-Studio
Production

Figure 6: FRIB software development workflow.

PHYSICS APPLICATIONS

The development plan for FRIB physics application was described in [6].

Online Model

As mentioned in [6], an online model is one of the most critical components for our beam commissioning since FRIB has been designed with some specific features as below:

- Solenoid focusing lattice;
- Non-axisymmetric field components at QWRs (Quarter Wave Resonators). The non-axisymmetric nature of QWR induces dipole and quadrupole components, which has significant contribution to FRIB beam dynamics;
- Accelerate multi-charge-state beams simultaneously. FRIB is designed to accelerate up to five charge states to achieve high beam intensity. This gives a stringent beam-on-target requirements, which should be met especially for multi-charge-state beams;
- Achromat arc sections between linac segments;
- Second order achromat with sextupole magnets.

An online model named TLM (Thin Lens Model) is under active development at FRIB to meet its needs. Some details and latest status of TLM could be found in [7].

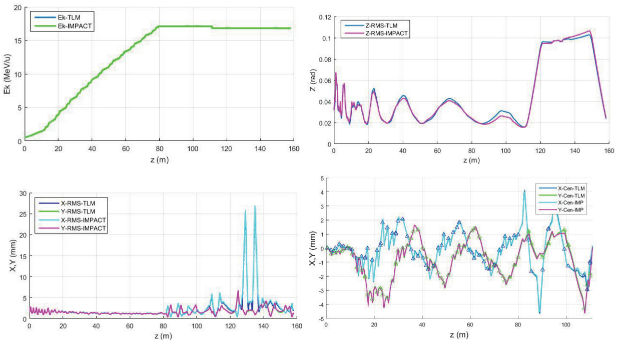


Figure 7: Benchmark of TLM against IMPACT for LS1 and FS1: (top left) Kinetic energy, (top right) RMS phase, (bottom left) transverse RMS beam size, and (bottom right) transverse beam orbit.

Fig. 7 shows a benchmark against IMPACT-Z, which compares results from the TLM and from the IMPACT-Z using the same lattice. The results show that with the lattice of FRIB LS1 and FS1 section, kinetic energy mean error is within 5e-5, longitudinal rms phase mean error is within 5%, transverse rms size mean error is within 2%, and transverse beam orbit mean error achieved within 2%.

As described above, the TLM now achieved an acceptable performance comparing with multiple particles tracking code. System integration is planned to integrate the TLM into our Python scripting environment.

Online model has been planned as a middle layer services, which is under active development. To enable physics application development from its early stage, an interface library has been developed to call it directly from physics application. Its architecture is as shown in Fig 8.

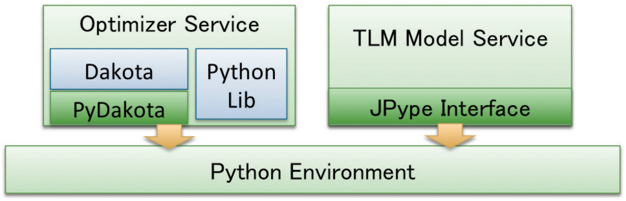


Figure 8: Architecture of TLM Model interface with Python Environment.

A prototype result for orbit correction is as shown in Fig. 9 against FRIB virtual accelerator.

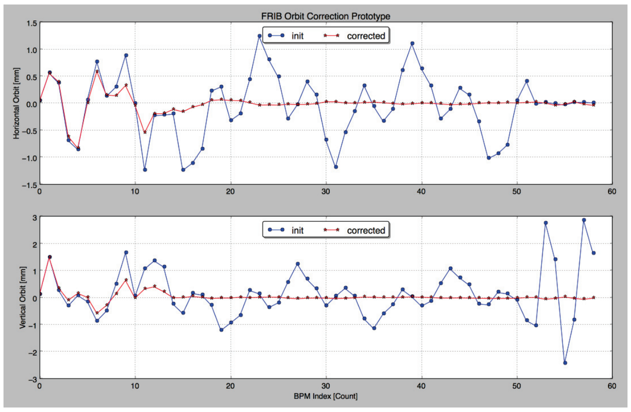


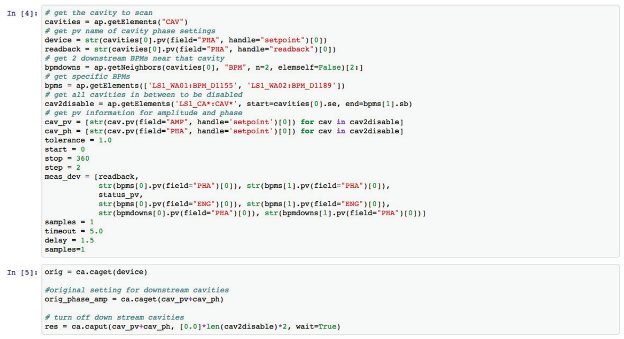
Figure 9: FRIB orbit correction prototype. The upper plot is for X direction, and lower part is for Y direction. The blue line is its original orbit, and red line is after correction.

The orbit distortion is given by artificially exciting the first steering magnet, and orbit correction is performed by using downstream correctors to flatten the downstream orbit. The results is with 5 iterations, and each iteration costs about 16 seconds with a few hundreds model calling.

System Integration

With the middle layer service, physics application development becomes much easier than ever before.

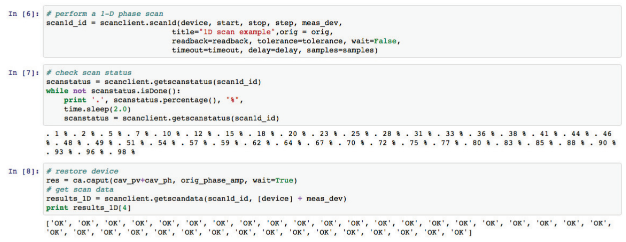
With Channel Finder service [8], there is no need to hardcode PV name any more, and the code is much clean and simple as shown in Fig. 10.



```
In [4]: # get the cavity to scan
        cavities = ap.getElements("CAV")
        # get pv name of cavity phase settings
        device = str(cavities[0].pv(field="PHA", handle="setpoint")[0])
        readback = str(cavities[0].pv(field="PHA", handle="readback")[0])
        # get 2 downstream BPMs near that cavity
        bpmdowns = ap.getNeighbors(cavities[0], "BPM", n=2, elemself=False)[2:]
        # get specific BPMs
        bpms = ap.getElements(['LS1_WA01:BPM_D1155', 'LS1_WA02:BPM_D1189'])
        # get all cavities in between to be disabled
        cav2disable = ap.getElements('LS1_CA*:CAV*', start=cavities[0].se, end=bpms[1].sb)
        # get pv information for amplitude and phase
        cav_pv = [str(cav.pv(field="AMP", handle='setpoint')[0]) for cav in cav2disable]
        cav_ph = [str(cav.pv(field="PHA", handle='setpoint')[0]) for cav in cav2disable]
        tolerance = 1.0
        start = 0
        stop = 360
        step = 2
        meas_dev = [readback,
                    str(bpms[0].pv(field="PHA")[0]), str(bpms[1].pv(field="PHA")[0]),
                    status_pv,
                    str(bpms[0].pv(field="ENG")[0]), str(bpms[1].pv(field="ENG")[0]),
                    str(bpmdowns[0].pv(field="PHA")[0]), str(bpmdowns[1].pv(field="PHA")[0])]
        samples = 1
        timeout = 5.0
        delay = 1.5
        samples=1

In [5]: orig = ca.caget(device)

        #original setting for downstream cavities
        orig_phase_amp = ca.caget(cav_pv+cav_ph)

        # turn off down stream cavities
        res = ca.caput(cav_pv+cav_ph, [0.0]*len(cav2disable)*2, wait=True)
```

Figure 10: Using Channel Finder service to search correct channel name.

Once get all needed channel names, user could ask other service to perform a desired work, for example a scan for RF cavity scan. Fig. 11 shows an example to use Scan Server to perform a 2-π scan.



```
In [6]: # perform a 1-D phase scan
        scan1d_id = scanclient.scan1d(device, start, stop, step, meas_dev,
                                      title="1D scan example",orig = orig,
                                      readback=readback, tolerance=tolerance, wait=False,
                                      timeout=timeout, delay=delay, samples=samples)

In [7]: # check scan status
        scanstatus = scanclient.getscanstatus(scan1d_id)
        while not scanstatus.isDone():
            print '.', scanstatus.percentage(), "%",
            time.sleep(2.0)
            scanstatus = scanclient.getscanstatus(scan1d_id)

        . 1 % . 2 % . 5 % . 7 % . 10 % . 12 % . 15 % . 18 % . 20 % . 23 % . 25 % . 28 % . 31 % . 33 % . 36 % . 38 % . 41 % . 44 % . 46
        % . 48 % . 49 % . 51 % . 54 % . 57 % . 59 % . 62 % . 64 % . 67 % . 70 % . 72 % . 75 % . 77 % . 80 % . 83 % . 85 % . 88 % . 90 %
        . 93 % . 96 % . 98 %

In [8]: # restore device
        res = ca.caput(cav_pv+cav_ph, orig_phase_amp, wait=True)
        # get scan data
        results_1D = scanclient.getscandata(scan1d_id, [device] + meas_dev)
        print results_1D[4]

        ['OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK',
        'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK', 'OK']
```

Figure 11: Example of 2-π scan using Scan server.

As shown in Fig. 12, the phase scan becomes one code, and the rest lines are for checking the scan progress, restoring setting to that before scanning, and manipulating the data like plotting. The scan result is as shown in Fig. 9.

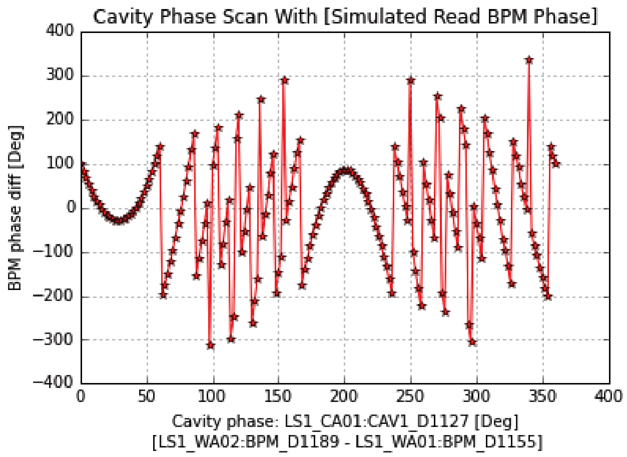


Figure 12: Phase scan result, which is for the first cavity of FRIB LS1 section. The names of related PVs are from Channel Finder Service, and the scan is done with Scan server.

CONCLUSION

Distributed service oriented architecture has been adopted for FRIB beam commissioning and operation. The infrastructure to support physics application development is under active development, and minimum set of service is ready to enable the development at its early stage. Two (2) technologies are used for our middle layer service development, which are EPICS V4 and web service respectively. We have started to develop physics applications under Python environment, and within an integrated environment.

ACKNOWLEDGMENT

The authors would thank our FRIB colleagues, especially S. Lund, E. Pozdeyev, J. Wei, Y. Yamazaki, Q. Zhao, Y. Zhang, for their helpful discussions and suggestions.

REFERENCES

[1] http://www.frib.msu.edu
[2] J. Wei et al, "FRIB Accelerator: Design and Construction Status", MOM1I02, proceedings of HIAT15, Yokohama, Japan (2015).
[3] J. Qiang et al. Impact. In Proceedings of the 1999 ACM/IEEE conference on Supercomputing, page 55. ACM, 1999.
[4] G. Shen et al., "NSLS II Middlelayer Services", MOPPC155, proceedings of ICALEPCS2013, San Francisco, US (2013).
[5] https://git-scm.com
[6] M. Ikegami, G. Shen, "Development Plan for Physics Application Software for FRIB Driver Linac", MOPWI023, proceedings of IPAC15, VA, US (2015).
[7] G. Shen et al., "Development Status of a Thin Lens Model for FRIB Online Model Service", TUDBC2, proceedings of ICAP2015, Shanghai, China (2015).
[8] https://github.com/ChannelFinder/ChannelFinder Service