

LA-UR-21-29210

Approved for public release; distribution is unlimited.

Title: Enabling rapid development of highly interactive analysis tools for
accelerator operations and experiments at DARHT through web services

Author(s): Koglin, Jason Edwin
Ronquillo, Daniel

Intended for: Report

Issued: 2021-11-05 (rev.1)

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Enabling rapid development of highly interactive analysis tools for accelerator operations and experiments at DARHT through web services

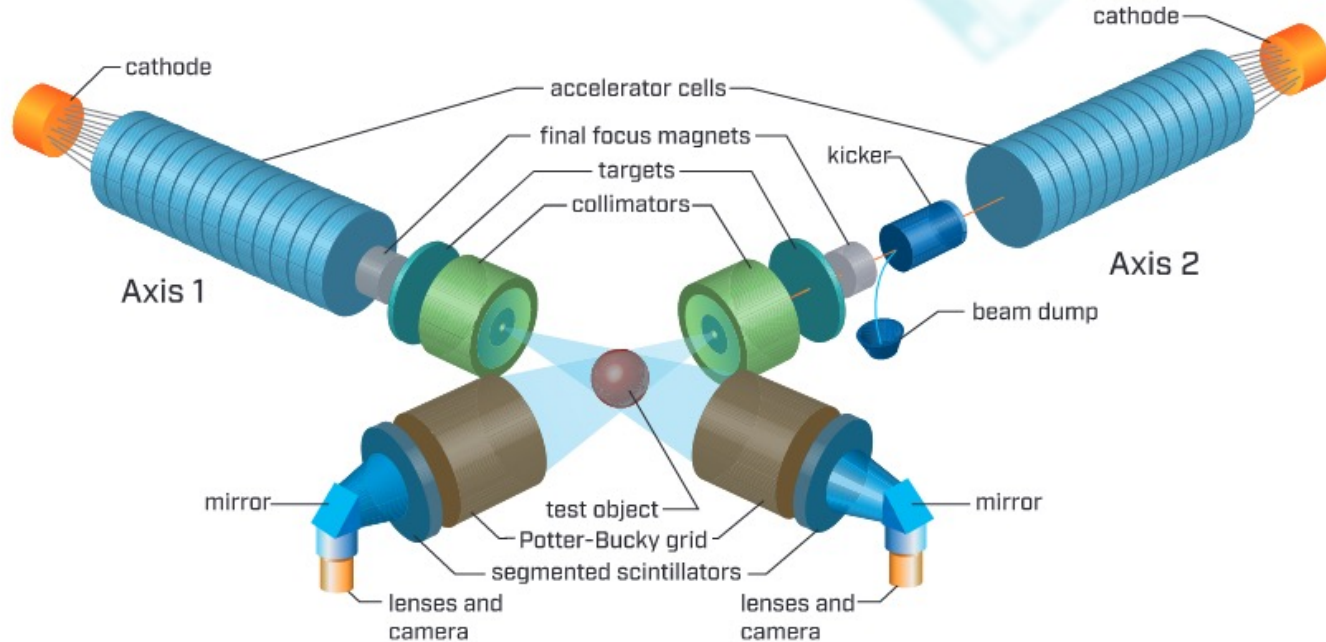
Jason Koglin (PI) and Daniel Ronquillo
J-6 DARHT Engineering, Operations, and Physics

September 14, 2021

LA-UR-21-29210

The penetrating radiography provided by DARHT is a key capability in executing a core mission of LANL

DARHT consists of two Linear Induction Accelerators (LIAs)



The Problem: DARHT has accrued substantial technical debt in analysis software over 10+ years of operations

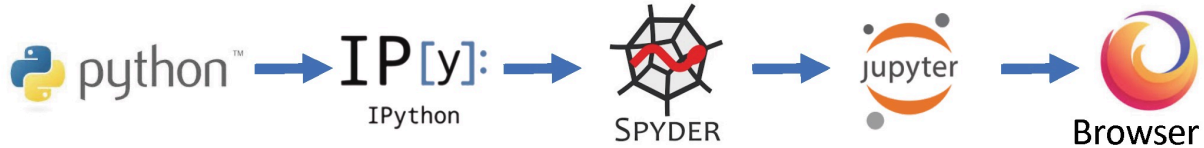
- Current DARHT accelerator analysis software is mostly
 - Shot based IDL GUI applications
 - Applications were developed to meet specific needs
 - Individual analysis code
 - The data acquisition system for the DARHT accelerators (DAAAC) provides flexible way to view and flexibly interact with individual waveform data
 - Analysis tools for new diagnostics and accelerator based experiments is typically ad-hoc and not easily shared between individuals and teams conducting development work at DARHT
 - Key staff have recently retired or are expected to in the near future
- **A new analysis framework has been developed using Python**

The Bigger Problem: How to make it easy to deploy applications and make sure they are robust

1. It is challenging to install and maintain Python packages on the air-gapped workstations at DARHT.
2. Even with relatively good instructions and well-designed Python packages, it has been difficult for most scientists and engineers to install and begin to use the new applications.
3. There is too much DARHT data to store on local drives and loading data from mapped drives is slow ($\sim 10x$ slower), particularly through VPN.
4. Local memory is easily taxed making applications slow or unresponsive. Since this depends on the workstation/laptop hardware configuration it makes it difficult to balance I/O and memory usage in app development.

→ **A new way of deploying/hosting software applications is desired**

The Solution: Rapid deployment of highly interactive applications using Python libraries and a web browser

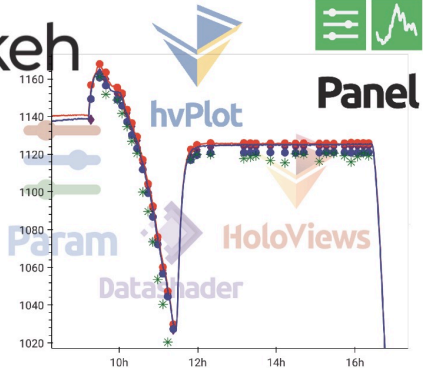


Higher Level



xarray, seaborn, pandas, scikit-image, NumPy, bokeh, hvPlot, Panel, HoloViews, DataSlider

	V	P	Tmod	Tcat	T1PkK	T2PkK
37782	59.8	80.0	2454.0	1148.7	1152.1	1148.9
37783	59.3	80.0	2403.0	1141.6	1151.3	1145.8
37784	58.7	80.0	2354.0	1134.7	1141.9	1137.2
37785	58.2	80.0	2300.0	1126.9	1134.8	1130.2
37786	57.7	80.0	2256.0	1120.5	1127.3	1123.1
37787	56.6	80.0	2120.0	1100.0	1115.5	1111.9
37788	55.5	80.0	2054.0	1089.7	1103.2	1099.4
37789	54.4	80.0	1954.0	1073.5	1091.2	1086.8
37790	53.1	80.0	1836.0	1053.6	1074.8	1072.2
37791	52.0	80.0	1761.0	1040.5	1058.5	1056.6



Graphical /Interactive

First Challenge: Find efficient way to host apps on the air-gapped 'darht-net' and keep packages up to date

- Career development for Daniel Ronquillo
 - Programmatic IT Engineer at LANL since 2018
 - Currently working on Master of Information Technology at NMSU
 - Performed work to deploy server apps
- Anaconda package/environment management
 - Found easiest way is to first use LANL Infrastructure on Demand (IoD) machine to build conda environments and test all apps
 - Copy only conda environment directories to air-gapped machine along with DARHT specific packages hosted on git.lanl.gov
 - Documentation was created to replicate efforts

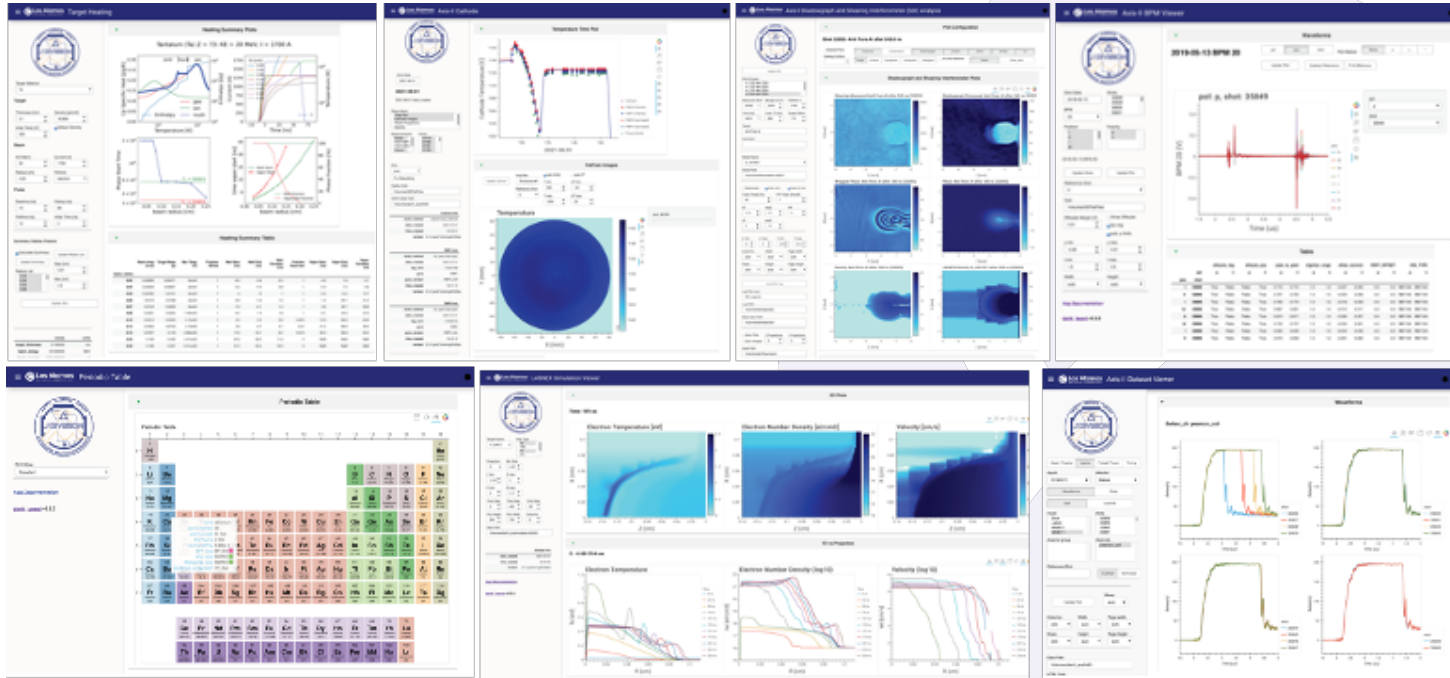
ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Work	% Complete
1		Begin Project Documentation	40 days	Wed 5/26/21	Tue 7/20/21		0 hrs	95%
2		Prerequisites	3 days	Wed 5/26/21	Fri 5/28/21		12 hrs	100%
3		Hardware Prep	6 days	Mon 5/31/21	Mon 6/7/21	2	12 hrs	100%
4		Image Server (DARHT)	2 days	Tue 6/8/21	Wed 6/9/21	3	8 hrs	100%
5		Server Hardware and Software Ready (DARHT)	4 days	Tue 6/22/21	Fri 6/25/21		32 hrs	100%
6		Contact NIE (Yellow)	0.13 days	Wed 6/9/21	Wed 6/9/21		1 hr	100%
7		Create IOD Ticket	0.13 days	Thu 6/10/21	Thu 6/10/21	6	1 hr	100%
8		IOD Yellow Net Server	1 day	Thu 6/10/21	Fri 6/11/21	7	1 hr	100%
9		Server Hardware and Software Ready (YELLOW)	4.38 days	Tue 6/22/21	Mon 6/28/21		35 hrs	100%
10		Install Anaconda 3 (DARHT)	1 day	Thu 6/10/21	Thu 6/10/21	4	1 hr	100%
11		Install Anaconda 3 (Yellow)	2 days	Fri 6/11/21	Tue 6/15/21	8	3 hrs	100%
12		ManualPKGS Directory (DARHT)	1 day	Mon 6/14/21	Mon 6/14/21		0.5 hrs	100%
13		Conda Environments Directory (DARHT)	1 day	Mon 6/14/21	Mon 6/14/21		0.5 hrs	100%
14		ManualPKGS Directory (Yellow)	1 day	Mon 6/14/21	Mon 6/14/21		0.5 hrs	100%
15		Conda Environments Directory (Yellow)	1 day	Mon 6/14/21	Mon 6/14/21		0.5 hrs	100%
16		Develop Directory Sync Profile (DARHT)	1 day	Tue 6/15/21	Tue 6/15/21	12,13	2 hrs	100%
17		Develop Directory Sync Profile (Yellow)	1 day	Tue 6/15/21	Tue 6/15/21	14,15	2 hrs	100%
18		Anaconda PKGS Sync Ready	5.63 days	Tue 6/22/21	Wed 6/29/21		45 hrs	100%
19		Install GitLab Server (DARHT)Cancelled	0.5 days	Wed 6/23/21	Wed 6/23/21		4 hrs	0%
20		Install Web Server Services (DARHT)	1 day	Thu 6/24/21	Thu 6/24/21		3 hrs	25%
21		Initial App Deployment (Yellow)	0.5 days	Wed 6/30/21	Wed 6/30/21	17	4 hrs	100%
22		Test Application Live (Yellow)	0.5 days	Wed 6/30/21	Wed 6/30/21	21	4 hrs	100%
23		Migrate Python PKGS (Yellow to DARHT)	1 day	Thu 7/1/21	Thu 7/1/21	22	3 hrs	100%

Task	Inactive Summary	External Tasks
Split	Manual Task	External Milestone
Milestone	Duration-only	Deadline
Summary	Manual Summary Rollup	Progress
Project Summary	Manual Summary	Manual Progress
Inactive Task	Start-only	
Inactive Milestone	Finish-only	

Project: ISTS-Ana.mpp
Date: Mon 8/23/21

Recommendation: use server with more RAM, all SSD and 10GB NICs

Outcome: Applications have been deployed at DARHT through web services on the air-gapped darht-net

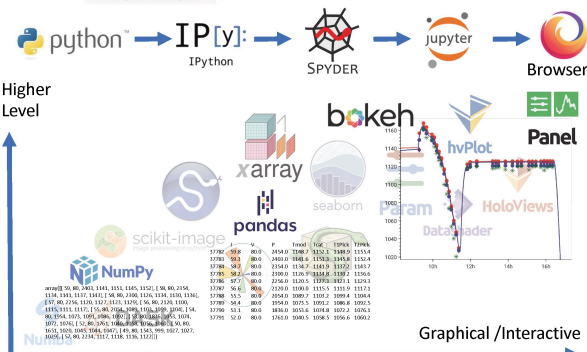
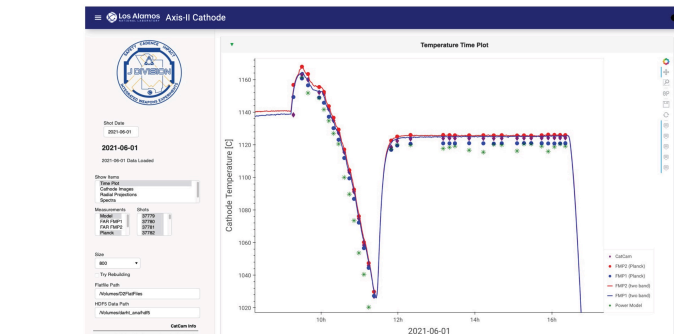


Only difference between using apps locally and on other machines is opening ports.

Second Challenge: Find consistent and secure way to deploy the same apps on the LANL Yellow network

- a) Tested using Linux machine that we currently have available
 - Apps work on Linux machine without issue and can be served up locally, but found more convenient to work with Infrastructure on Demand (IoD) Windows Server
 - Would need additional expertise to delve into opening up as a server on yellow using PIV authentication
 - b) Explored options with SEA-3 WEB Application Development Team
 - Plan to use J-6 programmatic funds to implement IoD 'front-end' server with authentication that SEA-3 will manage, which points to our IoD 'back-end' web app server that we will manage
 - c) Explored options being tested out under the Nuclear Science Data Solutions (NSDS) initiative within the LANL Weapons Research Services
 - MarkLogic is attractive way to deal with metadata (including who has access to what data) – using MarkLogic sandbox to get ready for planned FY22 deployment
- Outcome: Gained experience and developed contacts within LANL

Enabling rapid development of highly interactive analysis tools for accelerator operations and experiments at DARHT through web services



Project Description

The project goal is to explore options for hosting DARHT accelerator analysis applications on web servers, both within the air-gapped 'darht-net' at DARHT and on the LANL yellow network.

Project Outcomes

An anaconda python environment was deployed on a server to host a set of analysis apps with a start page on 'darht-net'. A path to deploying the apps on the yellow network with authentication has been identified.

PI: Jason Koglin, J-6 Engineering, Operations & Physics

Total Project Budget: \$20,000

ISTI Focus Area: Data Science and Artificial Intelligence Infrastructure; addresses data management and interactivity priorities

Highly-interactive applications for DARHT data analysis and operations are now hosted on a web server on the air-gapped 'darht-net'. The apps are built using open-source python libraries in a way that allows for rapid deployment of new analysis tools. The graphic illustrates how increasingly higher-level packages are used to build interactive plots from named data arrays and high-level parameter objects. The apps are served directly from Jupyter notebooks either locally or from a server, providing a convenient programming platform for rapid development.

END

Applications are built in a Jupyter Notebook using standard python packages

- Use param for parameters that have widget hooks

```
>>> shots = param.ListSelector()
```

- Use decorators to create callbacks that perform arbitrary actions (e.g., load data)

```
>>> @param.depends('shots', watch=True)
```

- Interactive plots with hvplot (uses bokeh)

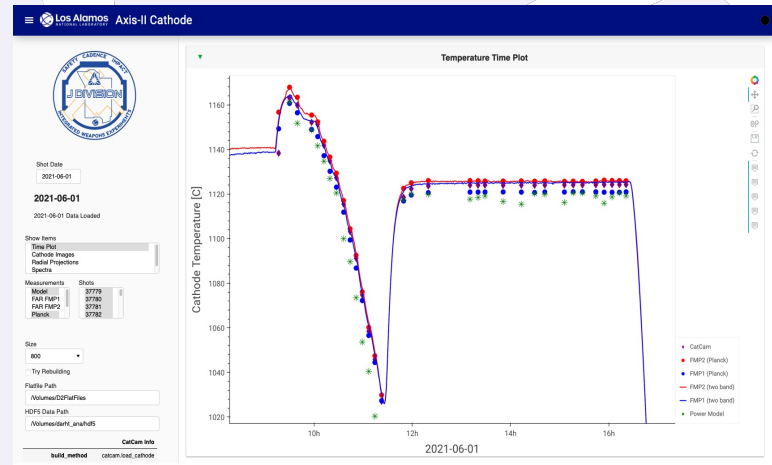
```
>>> plt = df.hvplot.scatter()
```

- Use panel to organize app using **templates** for uniform look and branding

```
>>> app = pn.template.MaterialTemplate()
```

- Build & serve from jupyter notebook

```
>>> app.show()
```



- Or serve apps from command line:

```
>>> panel serve app.ipynb --port 5009
```