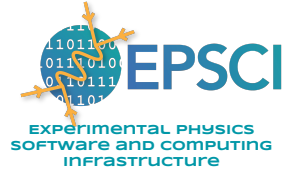




EIC Project Detector-1
"We need a real name"



JANA2 Summary

See full presentation slides from last week here:

<https://docs.google.com/presentation/d/1LI1dAqog58A6dFC0YLyhJHzlC43OVctey00swpP8E3Q/edit?usp=sharing>

July 6, 2022

David Lawrence JLab



Requirements:

- The reconstruction framework must be able to run on both simulated events and real data.¹ Even if there may be algorithms that use truth information (or even require truth information, initially), the reconstruction framework itself should allow for running without truth information.
- The reconstruction framework must be able to take advantage of heterogeneous computing resources (multiple cores, GPUs, etc).²
- The reconstruction framework must encourage modular development, using defined interface layers.³
- Algorithms must be implemented using the selected model, and ensure that data (event data, geometry description, and algorithm parameters) are kept separate from the algorithm itself.
- Algorithms must be able to run independently from any scheduling strategies; an algorithm should not need to know how it is orchestrated, whether it is

¹ "We will have a common software framework for the processing of the software, including the processing of streamed data and its time-ordered reconstruction." Statement of Principles, 2a.

² "We will enable distributed workloads across the computing resources of the worldwide EIC community, leveraging not only HTC but also HPC systems." Statement of Principles, 3a.

³ "We aim for a modular development paradigm for algorithms and tools without the need for users to interface with the entire software environment." Statement of Principles, 4c.

running in parallel, in single or multithreaded mode, concurrent or not, in online or offline analysis mode.

- The reconstruction framework must be open source, accessible to the entire community, and managed by a sustainable core team.⁴
- The reconstruction framework must be able to pass (and add) metadata and so-called slow control information to the output it produces, so input files are not needed and output files can stand on their own.
- The reconstruction framework must be able to run in streaming readout mode, that is:
 - with access to only parts of an event (single detector, single sector),
 - with events (or parts of events) appearing out of sequence,
 - individual algorithms must not rely on an algorithm-specific internal state to be able to make sense of disconnected parts of events.

Additional assessment criteria

- Amount of 'boilerplate' code that must be written by algorithm developers.
- Ability by the framework to avoid common safety errors through interface enforcement mechanisms (e.g. code signing).
- Ability for shared algorithm development between the two EIC detector collaborations (and/or outside of the EIC).
- Use of modern and sustainable coding practices, including in the code written by algorithm developers and other contributors.
- Demonstration of performance in production environments.

**JANA2 Meets all requirements
and Additional assessment criteria**

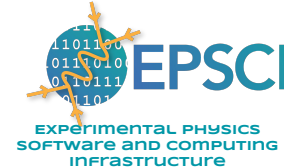
*(details on how it meets each of them given
in slides shown at June 29th meeting)*



EIC Project Detector-1
"We need a real name"

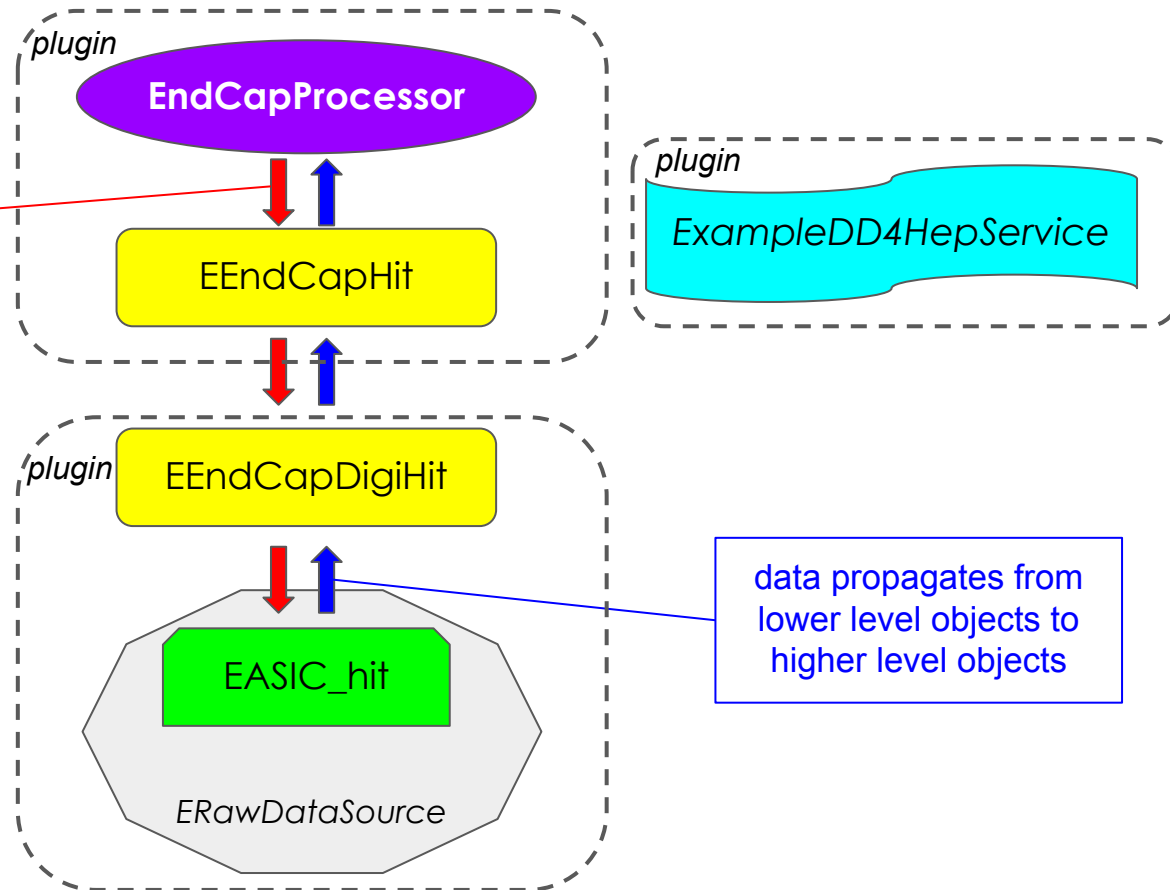
Example with Geometry Service

https://github.com/faustus123/EIC_JANA_Example



requests start with higher level objects and propagate to lower level objects

Wouter suggested example:
"...[take] a collection of hits and selecting those hits that are on a particular endcap tracking detector and have a position outside a minimum radial range."

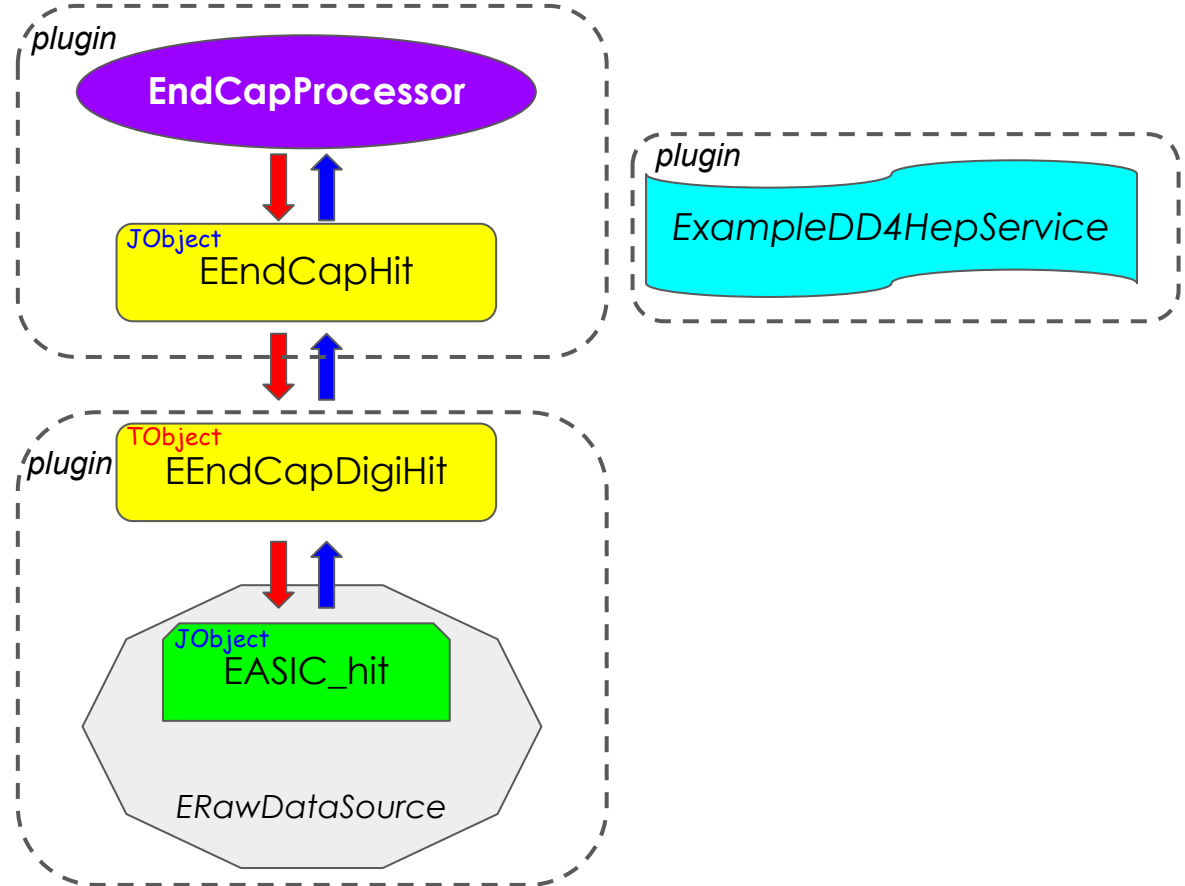


data propagates from lower level objects to higher level objects



Example with Mixed TObject and JObject

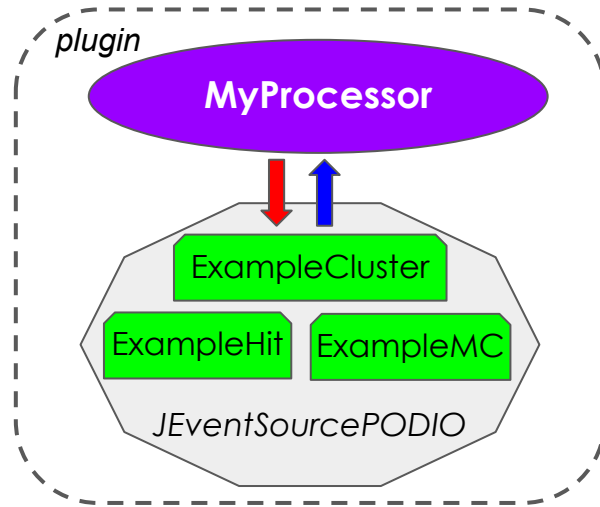
https://github.com/faustus123/EIC_JANA_Example/tree/TObject_example





Example with PODIO Event Source

https://github.com/faustus123/EIC_JANA_Example/tree/PODIO_example



ALL EXAMPLES

- *detailed build instructions accompany each example*
- *extensive comments in code*

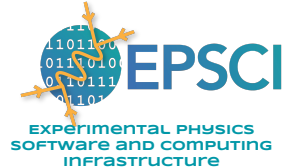
Example implements a JANA event source that can read from the *example.root* file produced by PODIO *tests/write* program.

n.b. **objects are still owned by PODIO EventStore**



EIC Project Detector-1
"We need a real name"

Python support in JANA2



As pure python script

`python3 jana.py`

```
4 # This example JANA python script
5 import time
6 import jana
7
8 print('Hello from jana.py!!!')
9
10 # Turn off JANA's standard ticker so we can print our own updates
11 jana.SetTicker(False)
12
13 # Wait for 4 seconds before allowing processing to start
14 for i in range(1,5):
15     time.sleep(1)
16     print(" waiting ... %d" % (4-i))
17
18 # Start event processing
19 jana.Start()
20
21 # Wait for 5 seconds while processing events
22 for i in range(1,6):
23     time.sleep(1)
24     print(" running ... %d (Nevents: %d)" % (i, jana.GetNeventsProcessed()))
25
26 # Tell program to quit gracefully
27 jana.Quit()
```

As embedded interpreter

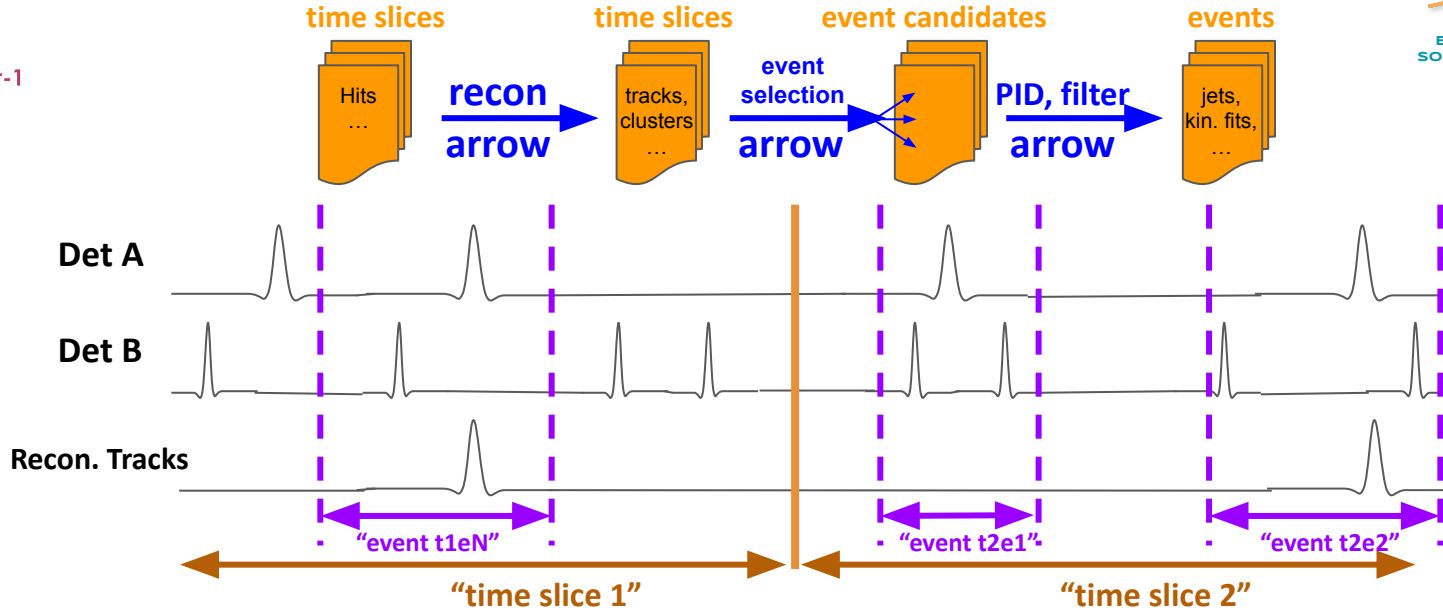
`jana -PPLUGINS=janapy -PJANA_PYTHON_FILE=myfile.py`

```
4 # This is a simple example JANA python script. It shows how to add plugins
5 # and set configuration parameters. Event processing will start once this
6 # script exits.
7 import jana
8
9 jana.AddPlugin('JTest')
10 jana.SetParameterValue('jana:nevents', 200)
11
12 jana.Run()
```



EIC Project Detector-1
"We need a real name"

Streaming Data



- **JANA2** has streaming readout features tested under multiple detector setups and in beam conditions*
- **EPSCI** has multiple experts working on streaming DAQ systems in same group as **JANA2** developers
- **EPSCI** works closely with the *JLab Fast Electronics Group* (Chris Cuevas) and partners routinely in performance testing in the DAQ Lab.

*Publications relevant to Streaming:

<https://arxiv.org/abs/2202.03085> Streaming readout for next generation electron scattering experiments

<https://doi.org/10.1051/epjconf/202125104011> Streaming Readout of the CLAS12 Forward Tagger Using TriDAS and JANA2

<https://doi.org/10.1051/epjconf/202024501022> JANA2 Framework for Event Based and Triggerless Data Processing

<https://doi.org/10.2172/1735849> Evaluation & Development of Algorithms & Techniques for Streaming Detector Readout

EPSCI Members

group formed Feb. 2020



David Lawrence PhD (physics) - Group Lead

Expertise: Physics, C++, software framework, online systems



Nathan Brei BS (aerospace engineering) MS (CS)

Expertise: Programming languages, parallel processing



Thomas Britton PhD (physics)

Expertise: Physics, software, OSG, AI DQM



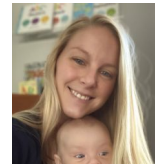
Michael Goodrich MS (physics) PhD (Computational Modeling/Simulation)

Expertise: Computational Modeling, Data Science, physics, real-time, Data Acquisition



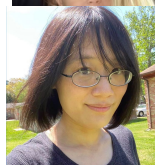
Vardan Gyurjyan PhD (physics) MS (CS)

Expertise: Data Acquisition, Java, C++, software frameworks



Torri Jeske PhD (physics)

Expertise: Experimental Nuclear Physics, Data Analysis, Detector Calibration



Xinxin "Cissie" Mei PhD (CS)

Expertise: Computer Science, GPU performance



***Diana McSpadden BS (math) MSDS in progress**

Expertise: Data Science, AI/ML



***Kishan Rajput MS (CS)**

Expertise: Data Science, AI/ML



Carl Timmer PhD (physics)

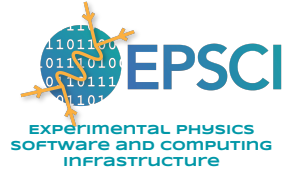
Expertise: Data Acquisition, Java, file format, I/O

* Member of DS Dept.



EIC Project Detector-1
"We need a real name"

Institutional Support

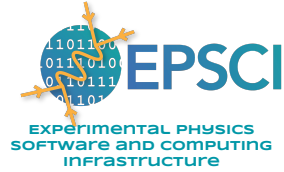


- JLab will support **JANA2** throughout the lifetime of the EIC project.
 - This is institutional and independent of any one person(s)
 - JLab is a full partner in the EIC and as such, considers it a top-tier priority
- JLab will commit ~1 FTE to feature development, support and implementation of **JANA2** in the EIC software stack
 - Significantly more initially to meet Oct. deadline
- **EPSCI** includes personnel dedicated to Streaming DAQ development *AND* the **JANA2** core development team

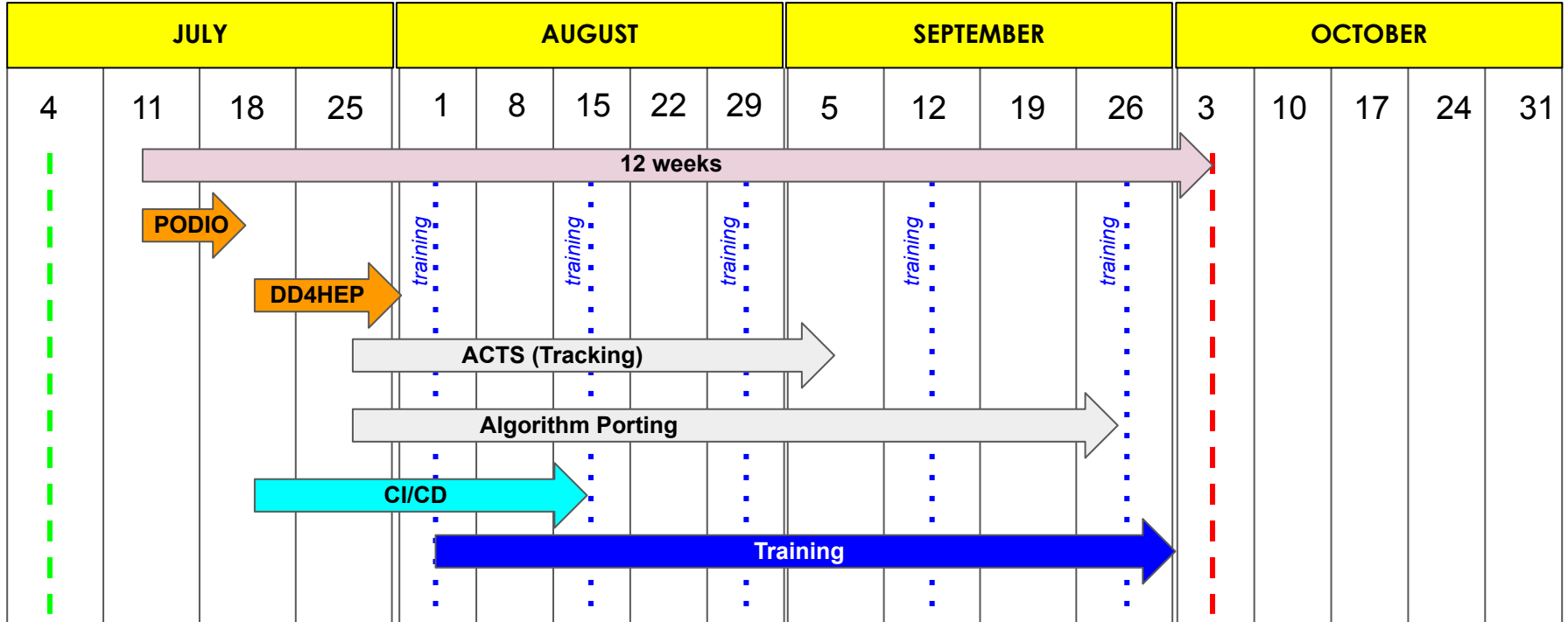


EIC Project Detector-1
"We need a real name"

Reconstruction Software Schedule for JANA2



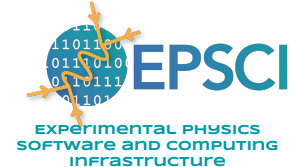
we are not starting Green Field !





EIC Project Detector-1
"We need a real name"

Summary



- JANA is a multithreaded framework project with nearly 2 decades of experience behind it
 - Multicore architecture standard at Jefferson Lab
- JANA2 is a rewrite based on modern coding and CS practices
 - Multithreading model improved using lessons learned
 - Developed for modern NP experiments with streaming readout and heterogeneous computing
- JANA2 is well-suited for a large-scale experiment in the era of streaming readout and AI
 - Meets all EIC requirements
- Jefferson Lab is a full partner in leading the experimental program for the EIC. We are ready to commit 1 FTE to support and further develop JANA2 for the EIC:
 - EPSCI group will ensure in the next months that JANA2 will be ready for the EIC collaboration.
 - We will work with the community to make JANA2 a community-wide project.

Documentation: <https://jeffersonlab.github.io/JANA2/>

Example projects: https://github.com/faustus123/EIC_JANA_Example

https://github.com/faustus123/EIC_JANA_Example/tree/PODIO_example

https://github.com/faustus123/EIC_JANA_Example/tree/TObject_example

Publications:

<https://arxiv.org/abs/2202.03085> *Streaming readout for next generation electron scattering experiments*

<https://doi.org/10.1051/epjconf/202125104011> *Streaming Readout of the CLAS12 Forward Tagger Using TriDAS and JANA2*

<https://doi.org/10.1051/epjconf/202024501022> *JANA2 Framework for Event Based and Triggerless Data Processing*

<https://doi.org/10.1051/epjconf/202024507037> *Offsite Data Processing for the GlueX Experiment*

<https://iopscience.iop.org/article/10.1088/1742-6596/119/4/042018> *Multi-threaded event reconstruction with JANA*

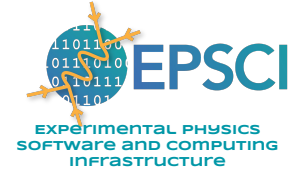
<https://pos.sissa.it/070/062> *Multi-threaded event processing with JANA*

<https://iopscience.iop.org/article/10.1088/1742-6596/219/4/042011> *The JANA calibrations and conditions database API*

<https://iopscience.iop.org/article/10.1088/1742-6596/1525/1/012032> *JANA2: Multithreaded Event Reconstruction*



EIC Project Detector-1
"We need a real name"



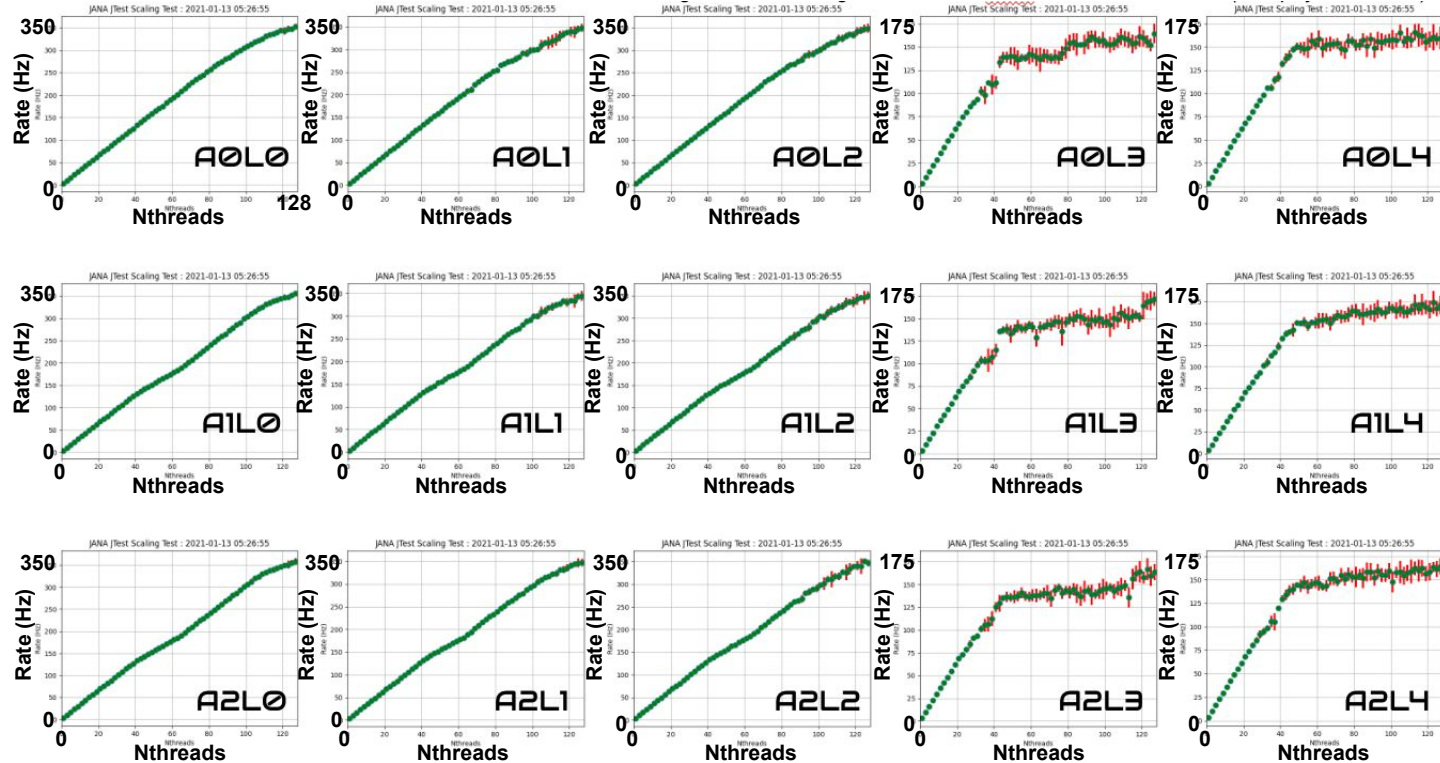
Backups



Multiple Affinity and Locality strategies

EIC Project Detector-1
"We need a real name"

OS, chip type, memory architecture, and nature of job all can affect which model yields optimal performance



```
enum class
AffinityStrategy {
    None,
    MemoryBound,
    ComputeBound };
```

```
enum class
LocalityStrategy {
    Global,
    SocketLocal,
    NumaDomainLocal,
    CoreLocal,
    CpuLocal };
```

Configurable at run time via Config. Parameters

JANA2 Scaling test: PSC Bridges-2 RM Two AMD EPYC 7742 CPUS (128 physical cores)