Near real-time streaming analysis of big fusion data

Presenter: Ralph Kube et al.

2021-12-03

R.M. Churchill
CS Chang
K. Silber

L. Stephey
C. Snavely
R. Thomas
W. Bhimji
D. Bard
E. Dart

J. Choi
R. Wang
S. Klasky

M.J. Choi
J.S. Park
B.S Cho

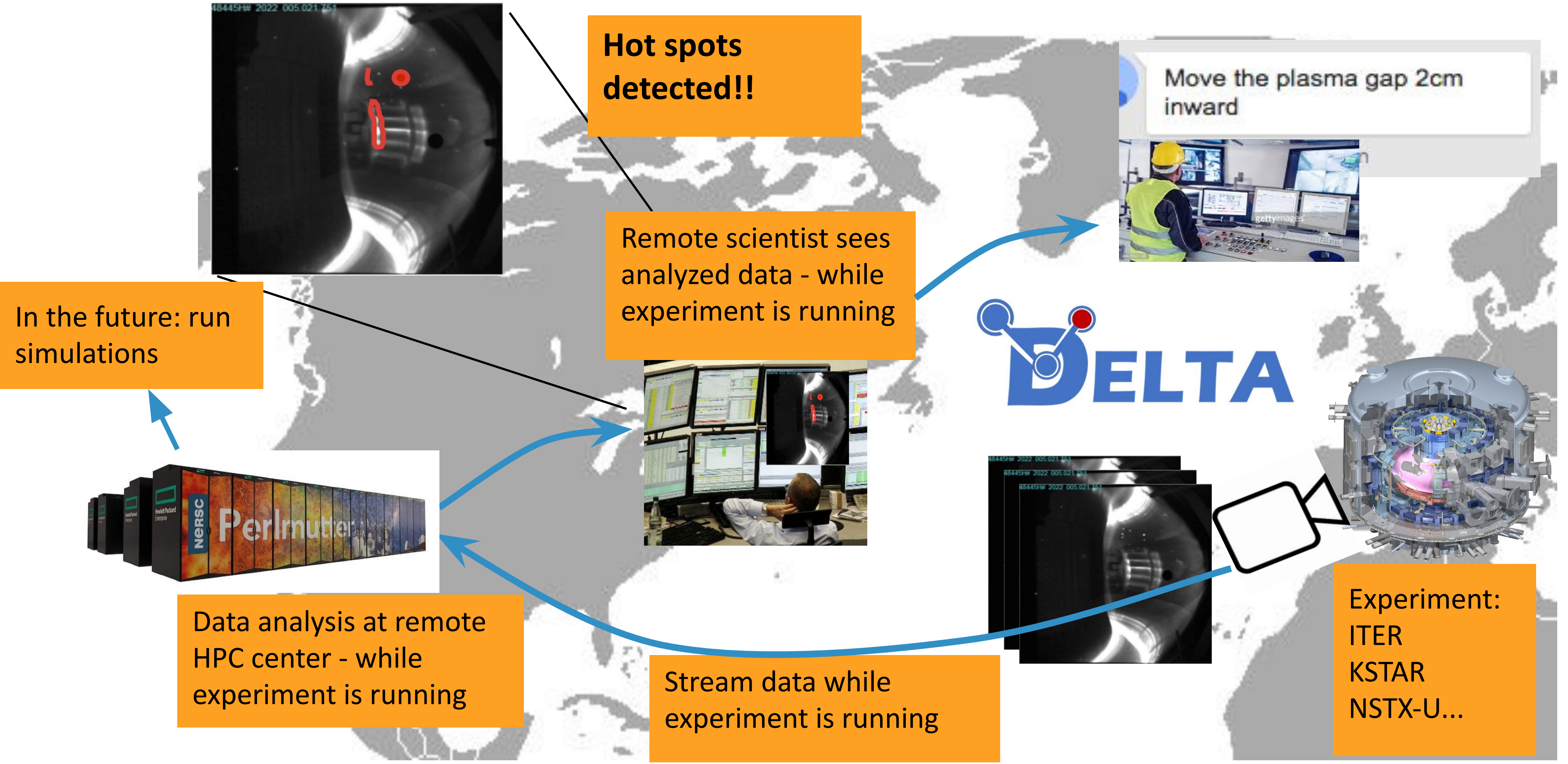
Goal: Tackle fusion data analytics at ITER scale to expand operational capabilities of fusion facilities

- **Driver: Compute environments are changing - ITER will have only minimal compute resources on-site.**
  - Increase in compute power driven by accelerators (GPU/TPU/ASIC/WSE)
  - Storage volume and speed don’t keep up with increase in compute power
- **Driver: Long-pulse devices need fast turnaround time for data analysis**
  - Measurements need to be analyzed as plasma pulses are ongoing - overlap experiments and data analysis to save time.
- **Driver: ML algorithms perform best when trained on big data sets**
  - Manual training will become infeasible. ML models will need to be trained online as new data becomes available.

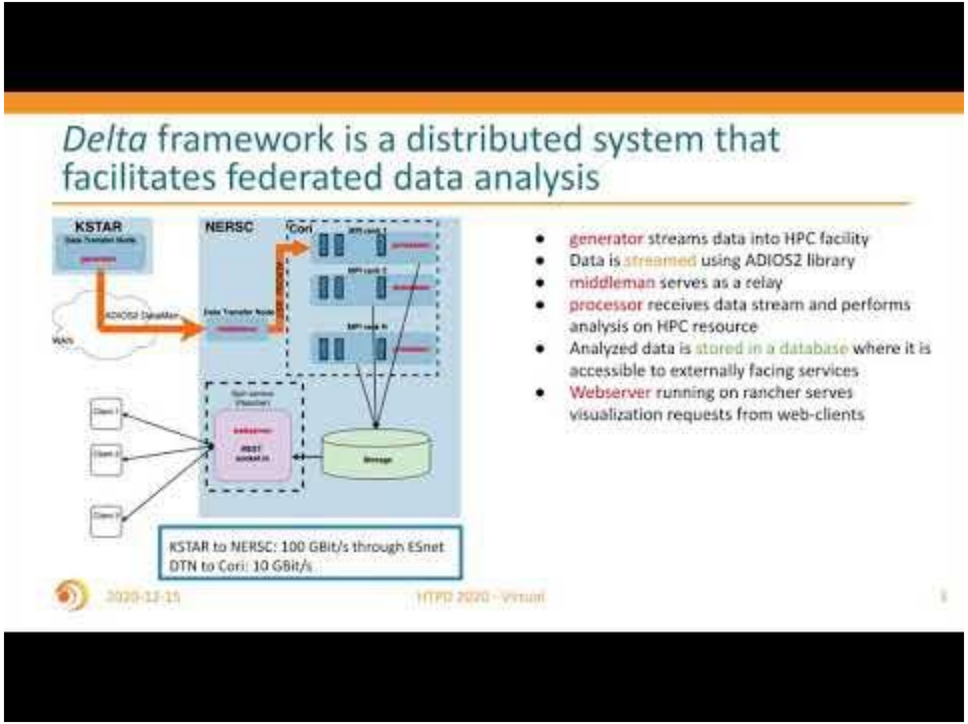
Solution: Move to near real-time streaming data analysis.

For this we are developing the DELTA framework DELTA

Connect experiments, scientists, and compute resources to optimize scientific discovery



Example: Streaming analysis of KSTAR ECEi data at NERSC in between shots



In December 2020 we used NERSC to analyze KSTAR ECEi measurements in between shots.

Streaming the measurements to NERSC, analyzing and visualizing them took about 5 minutes.

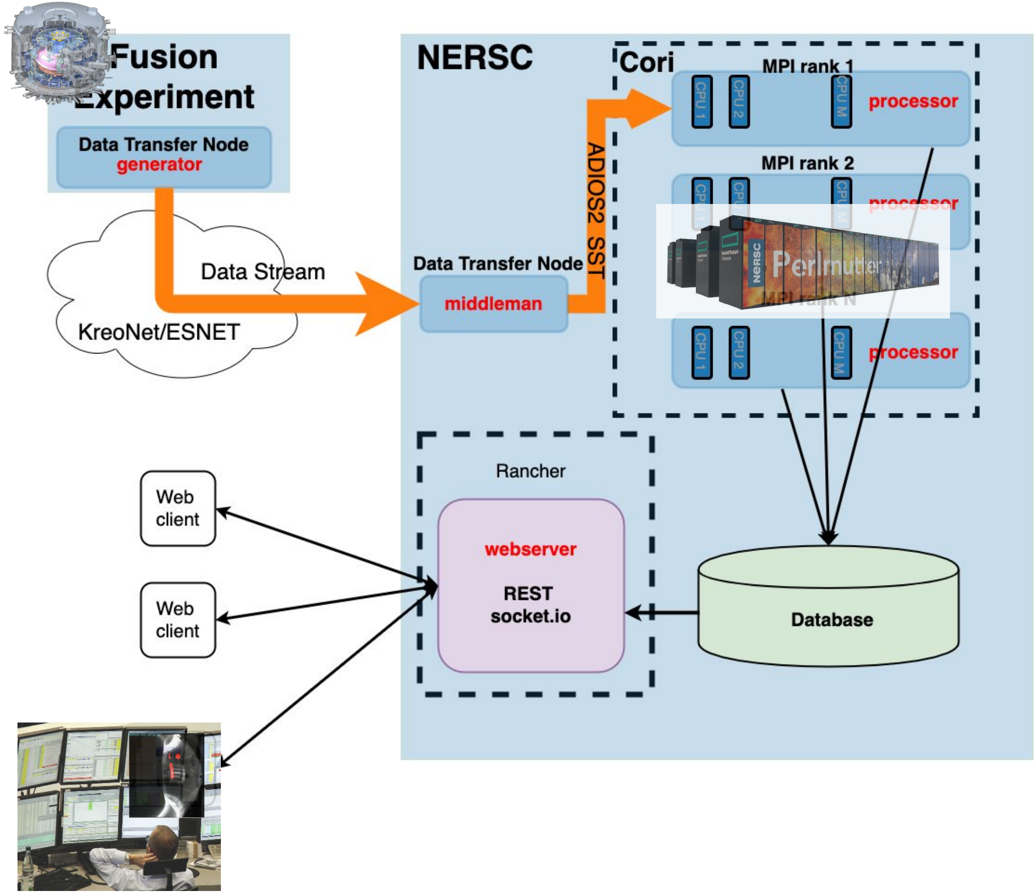
Shot cadence at KSTAR is ~10 minutes.

Screen layout:

Network bandwidth	Network bandwidth	CPU utilization
KSTAR Data generator	**NERSC** relay	**Cori**(NERSC) Data analysis

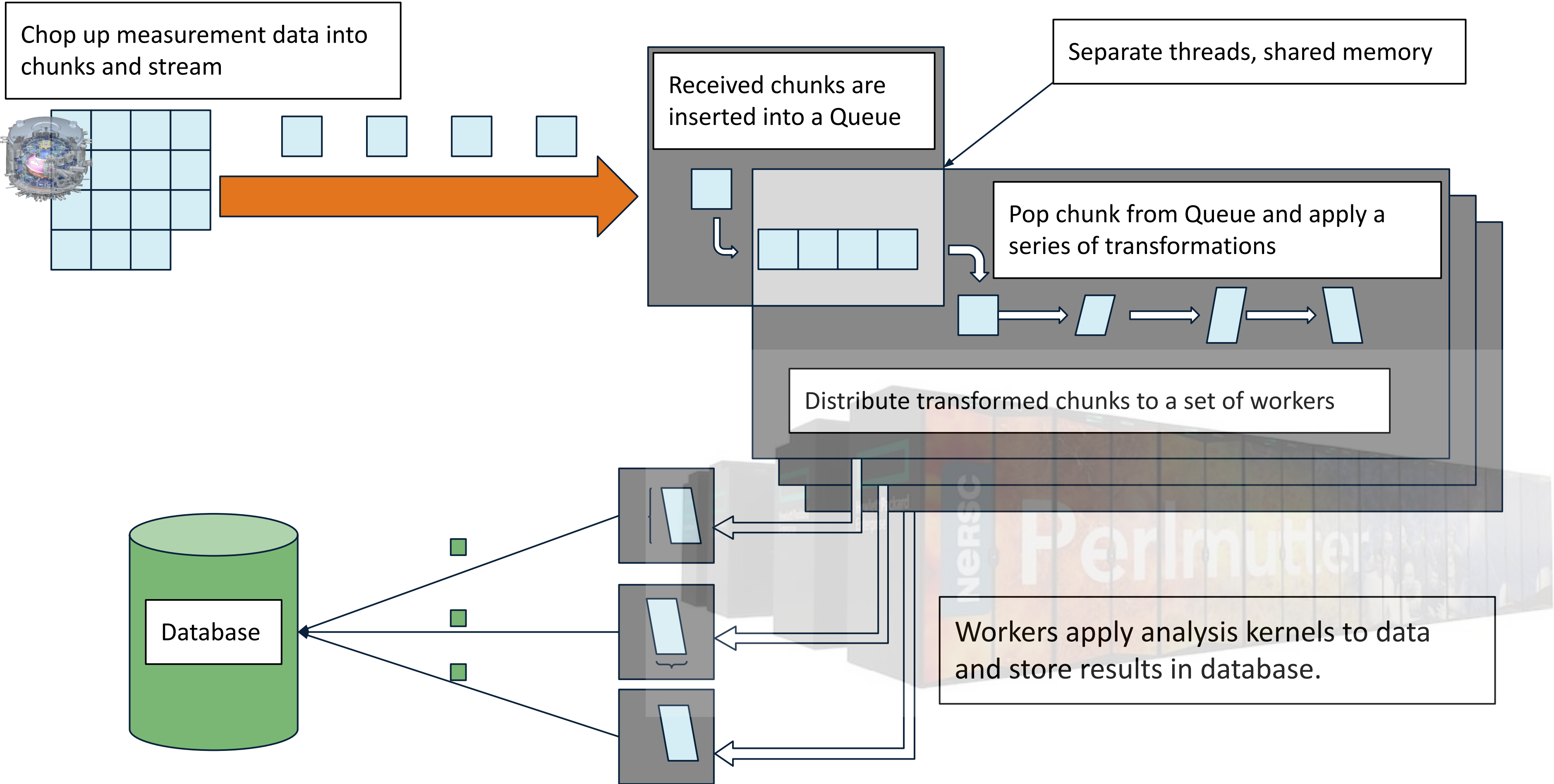
[Video link]

DELTA is a framework that enables near real-time analysis of data streams from fusion experiments



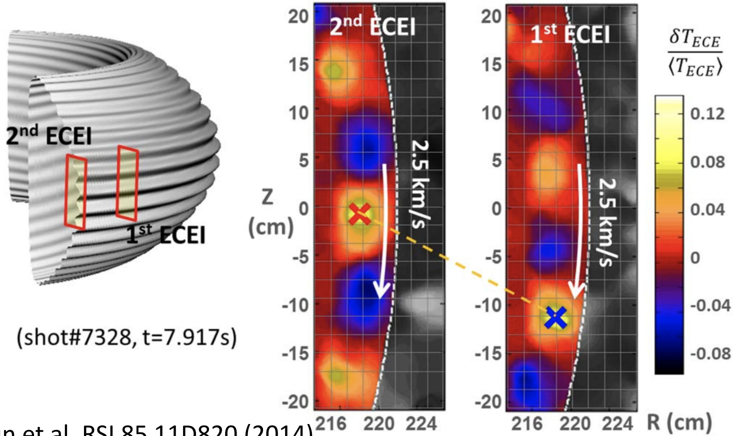
- **generator** streams data into HPC facility.
  - Input are data files. Future: stream directly from diagnostic
- **middleman** serves as a relay.
  - Data never touches the file system
- **processor** receives data stream and performs analysis on HPC resource.
  - Distributes work using a master-worker architecture
- Analyzed data is stored in a **database** where it is accessible to externally facing services.
  - Modern databases facilitate flexible real-time access by downstream applications
- **Webserver** running on Spin (container orchestration service) serves visualization requests from web-clients.
  - Make analyzed data instantaneously accessible from around the world. Allow to use modern web-based collaboration tools.

Delta processor simultaneously receives a data stream and runs analysis.



Define a compute intensive benchmark workload to test scalability of the DELTA processor

ECEI visualizes large scale 2.5d plasma structures



Yun et al. RSI 85 11D820 (2014)

- KSTAR ECE diagnostic: samples 24x8=192 channels with ~MHz sampling rate
- Diagnostics produces image time-series with about 1GB/sec

ECE benchmark workflow:

Estimate the power spectrum of each channel. Then calculate:

- Cross-power: $S_{XY}(\omega) = E[X(\omega)Y^{\dagger}(\omega)]$
- Coherence: $C_{XY}(\omega) = |S_{XY}(\omega)| / S_{XX}(\omega)^{1/2} S_{YY}(\omega)^{1/2}$
- Cross-phase: $P_{XY}(\omega) = \arctan(Im(S_{XY}(\omega)/Re(S_{XY}(\omega)))$
- Cross-correlation: $R_{XY}(t) = IFFT(S_{XY}(\omega))$

for all $\binom{192}{2} = 18336$ channel pair combinations (X,Y).

→ Each calculation above is implemented as a separate kernel.

→ For a single shot, we partition the ECEI data into 500 chunks of size 24x8x10,000.

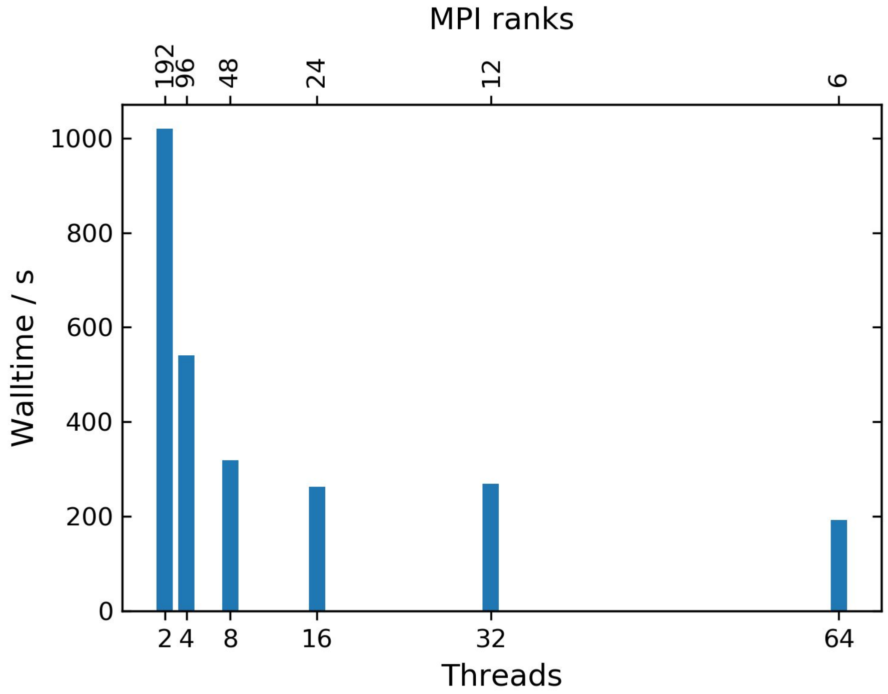
→ Code adapted from *fluctana*, M. Choi. Other codes possible.

Use cases for C, S, P, R:

Estimation of local dispersion relation (flow velocity), 2d characterization of T_e turbulence, identification of avalanche-like T_e transport events:

Choi et al. NF 57 126058 (2017); Choi et al. NF 59 086027 (2019);

The benchmark scales well over multiple CPU nodes until we hit architectural bottlenecks



Run benchmark on 6 Cori nodes (Xeon Haswell CPU with 32 cores, 128GB RAM).

- 192 MPI ranks / 2 hyperthreads: Too much communication, CPU cores are not effectively utilized.
- 6 MPI ranks / 64 hyperthreads: Shortest walltime, about 190s.

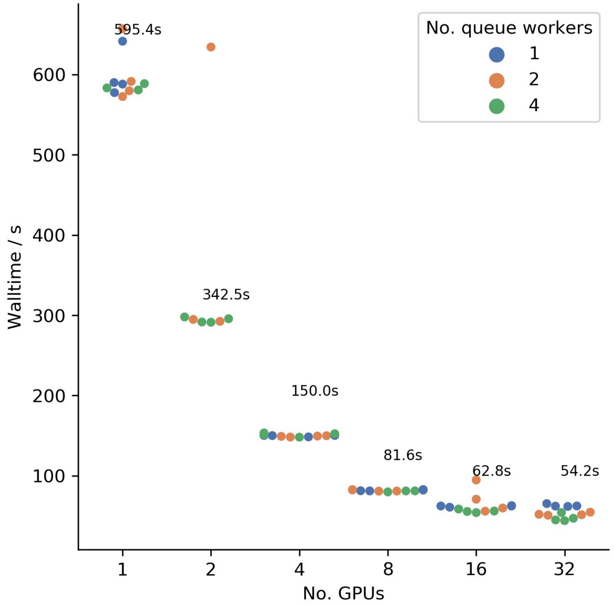
KSTAR shot cadence: approx. 10 minutes
Fastest execution time: 190 seconds.
Caveats:

- Data is read from filesystem, no streaming.
- Data analysis results are discarded

Walltime does not change much when streaming + storage is added.

Benchmark analysis scales well on multiple CoriGPU nodes until we hit architectural bottlenecks

- Kernels C,S,P perform similar computation -> Fuse them into a single kernel.
- Implement kernel using numba to enable just-in-time-compilation and GPU execution
- Strong scaling up to 16 GPU nodes.
- After that, pre-processing throughput becomes the bottleneck.



Caveats:

- Time shown is for processing 100 chunks.
- Data is read from filesystem, no streaming.
- Data analysis results are discarded
- Data chunks are pre-processed using a short-time Fourier transform and bandpass filter

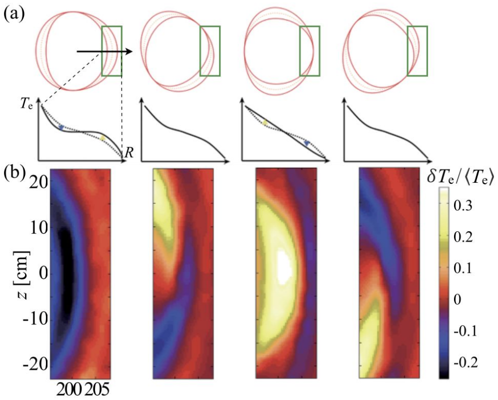
This is about 5x slower than the CPU case.

- Memory access pattern in the kernel is challenging
- Overhead due to memory transfer from/to GPU

Machine learning can be leveraged at multiple stages in the DELTA framework to aid scientific discovery

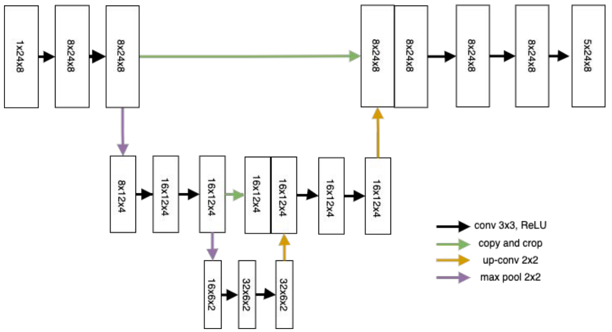
ML models can be used

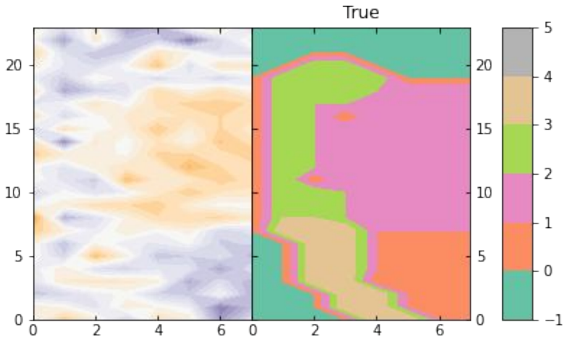
- in pre-processing to run selectively run data analysis routines based on results from a ML detector
- to store detected features in a database
- in online visualization



- Magnetic island perturb the $<T_e>$ profile as it rotates through ECEI view.
- Sampling $\delta T_e/<T_e>$, ECEI detects radial phase inversion structures caused by magnetic islands.

- U-Net is a CNN encoder-decoder architecture
- Small model with ≲10k parameters.





Input is a 24x8 array of ECEi data

Output is a 24x8 array where each pixel is assigned a value that encodes which part of the image it belongs to.

Live-demo: Use the Magnetic Island detector in the web dashboard

http://lb.test123.development.svc.spin.nersc.org/dashboard/newvue

Conclusions and future plans

Goal: Connect experiments, scientists, compute using advanced networks to optimize scientific discovery for fusion energy sciences.

For this, are developing the Delta framework which reliable facilitates near real-time streaming analysis of big fusion data using distributed HPC resources and web-based interactive visualization tools.

- **If you would like to use DELTA for your analysis we want to hear from you.**

Future work will explore

- **Operationalize streaming data analytics for fusion: Develop a persistent service to be used in daily ops.**
- Develop a power-exhaust monitoring system
  - Add support for more data sources (f.ex. IR, Mirnov Coils, bolometer)
  - Add support for more advanced workflows: simulations that use measurements as input
- Explore use of industry streaming analysis software (f.ex. ray.io and apache foundation suite) at NERSC
- Coupling to IDA systems

References

Churchill et al. 2021

Kube et al. 2021

Kube et al. 2020

Kube et al. 2020

Choi et al. 2020

Churchill et al. 2020

Choi et al. 2016

Choi et al. 2018