Toward Ten-Minute Turnaround in CMS Data Analysis: The View from Notre Dame

John Lawrence
On Behalf of CMS Collaboration and
Notre Dame Cooperative Computing Lab
May 8th, 2023

UNIVERSITY OF NOTRE DAME

Analysis Grand Challenge for HL-LHC

- The HL-LHC data volumes pose a serious challenge to current analysis approaches
- IRIS-HEP has initiated Analysis Grand Challenge (AGC) to develop analysis framework that provides necessary:
  - Throughput
  - Flexibility
  - Ease of use
  - Low latency

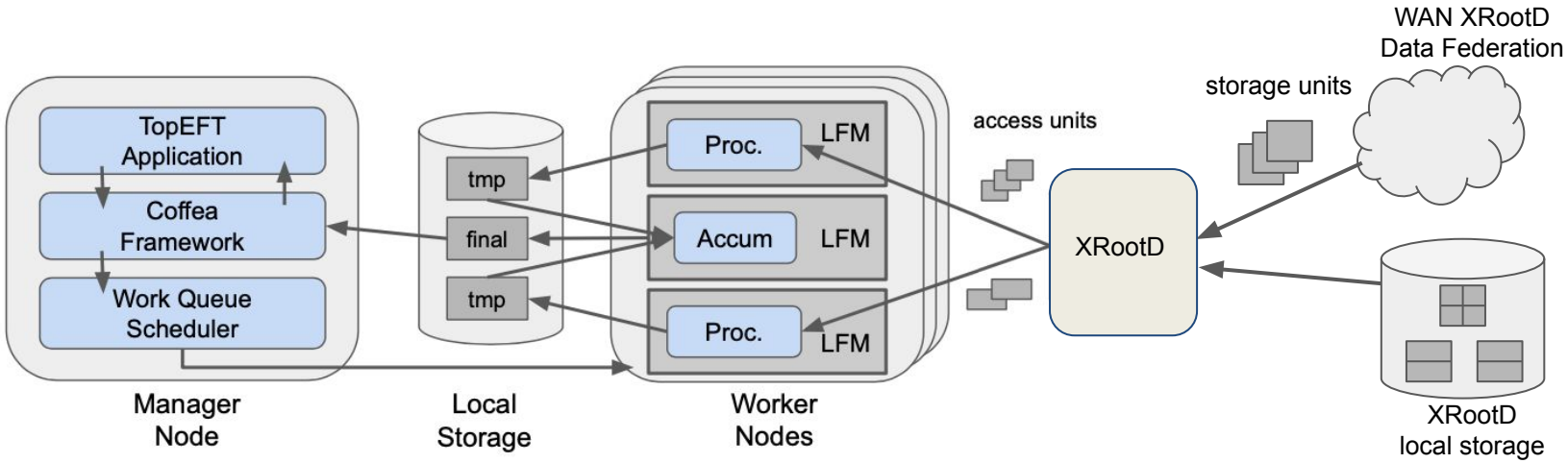
Analysis: TopEFT Framework

- Use TopEFT analysis to test current framework
  - Full Run 2 analysis (~150/fb, HL-LHC~3000/fb)
- Designed to analyze CMS data in order to search for new physics using the framework of Effective Field Theory (EFT) CMS-PAS-22-006
- Built on Coffea framework with columnar approach relying on scientific python ecosystem

TopEFT overview

- The TopEFT workflow:
  - Inputs are flat n-tuple (CMS NanoAOD) formatted proton-proton collision data from CMS (~2TB)
  - Processing step consists of calculating relevant properties of the events and filling histograms
  - Accumulation function merges together the histograms to produce the final output
- Memory considerations of the histograms produced and accumulated with TopEFT:
  - TopEFT histograms are heavier than conventional histograms
  - Each bin carries an array of 378 numbers for its EFT framework
  - The accumulation step can cause large memory requirements

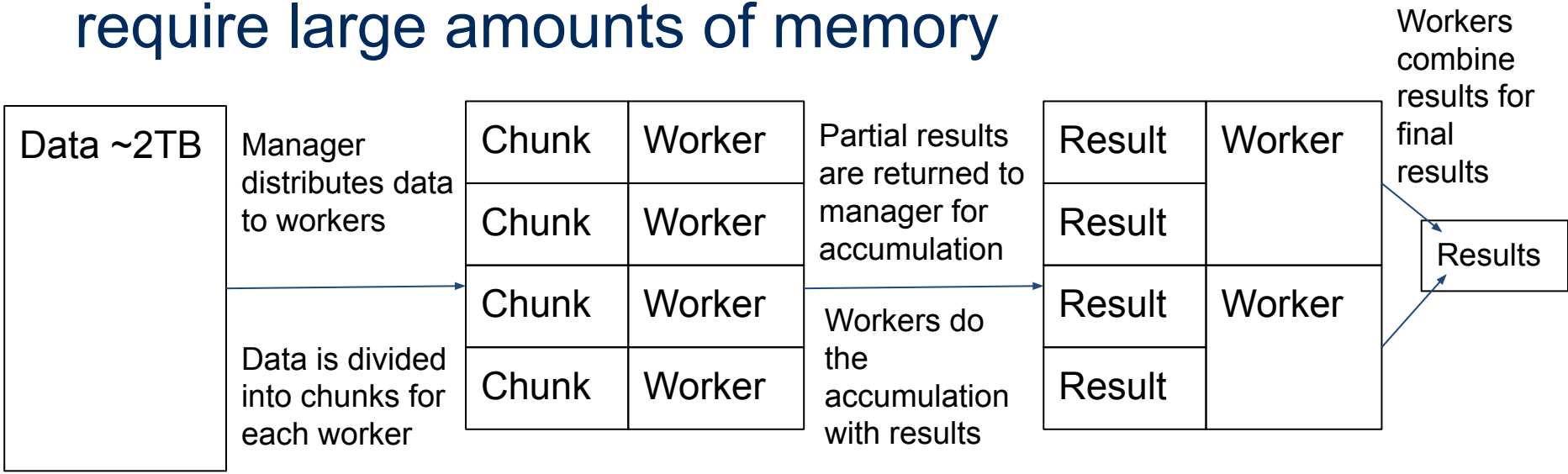
Scaling out TopEFT with Work Queue



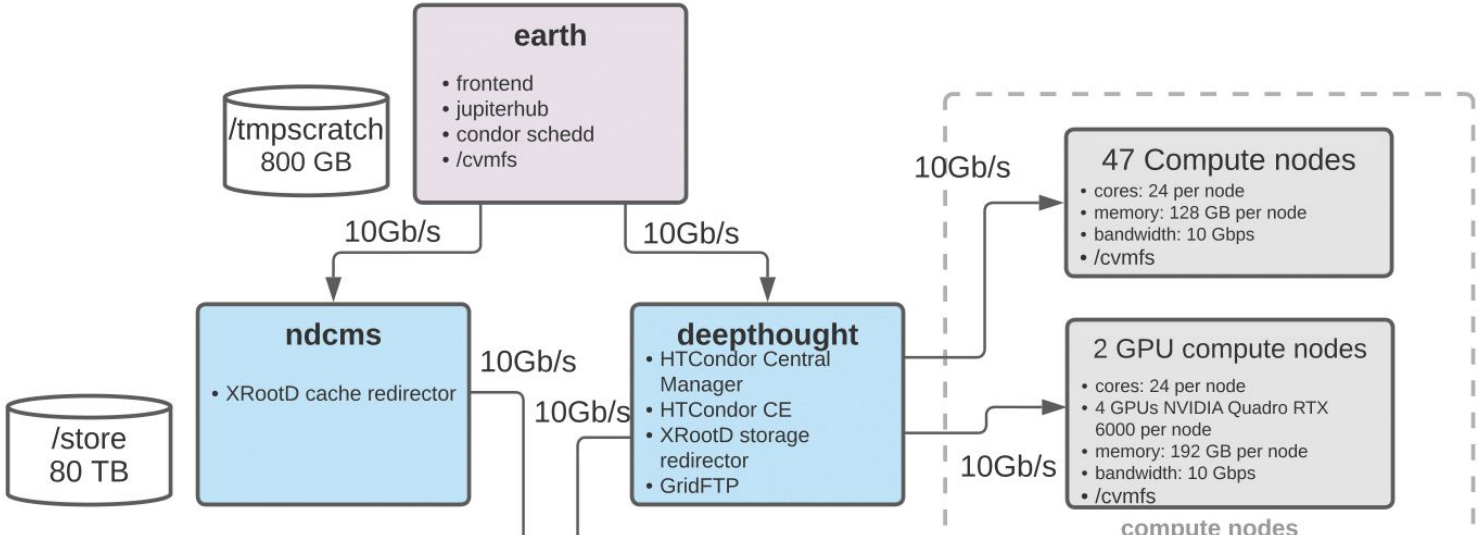
- Work Queue is a system for creating and managing scalable manager-worker style programs developed by ND CCL team
- To efficiently utilize distributed resources, TopEFT employs the Work Queue executor
- The Work Queue manager accepts task definitions from Coffea (for processing and accumulation tasks)
- Schedules the tasks to remote workers
- Sends along the relevant python environment with the task

Accumulation

The final step of merging all the histograms can require large amounts of memory



ND Tier-3


earth
• frontend
• jupiterhub
• condor schedd
• /cvmfs
/tmpscratch
800 GB
10Gb/s
10Gb/s
ndcms
• XRootD cache redirector
/store
80 TB
10Gb/s
10Gb/s
deepthought
• HTCondor Central Manager
• HTCondor CE
• XRootD storage redirector
• GridFTP
10Gb/s
47 Compute nodes
• cores: 24 per node
• memory: 128 GB per node
• bandwidth: 10 Gbps
• /cvmfs
10Gb/s
2 GPU compute nodes
• cores: 24 per node
• 4 GPUs NVIDIA Quadro RTX 6000 per node
• memory: 192 GB per node
• bandwidth: 10 Gbps
• /cvmfs
compute nodes

TopEFT performance today at ND Tier-3

cpu needs

runtime:	100min
cores:	up to 1000
mem total:	18.5 TB
disk total:	7.2 TB

IO needs

total root data:	1.7 TB
data actually used:	0.75 TB
IO temp files:	**0.25 TB**
origin:	xrootd local

processing tasks

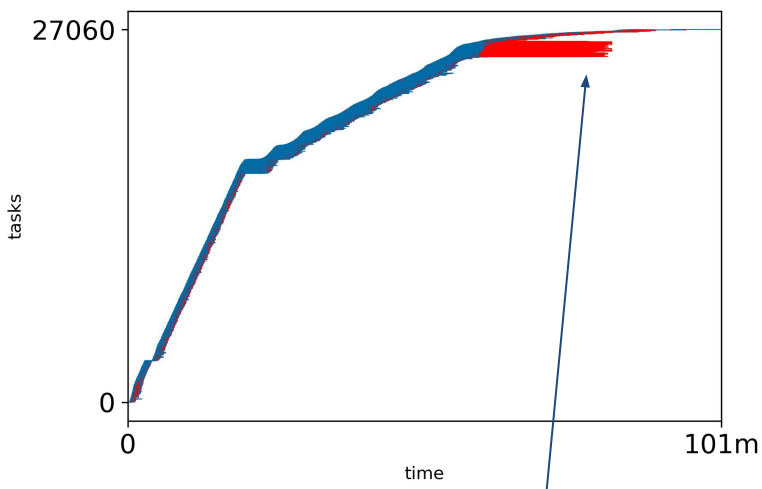
total:	23K
avg time:	110s
slowest:	318s
largest mem	4 GB
largest disk	0.5 GB

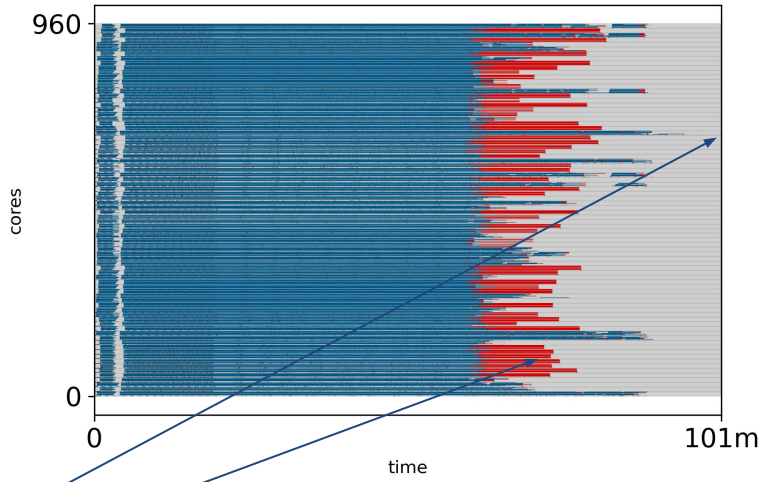
accumulation tasks

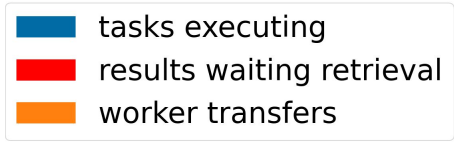
total:	1.2K
avg time:	6s
slowest:	**141s**
largest mem:	**12 GB**
largest disk:	**20 GB**

Current Bottlenecks Visualized

Accumulation
Data Returned
TopEFT
+ Work Queue







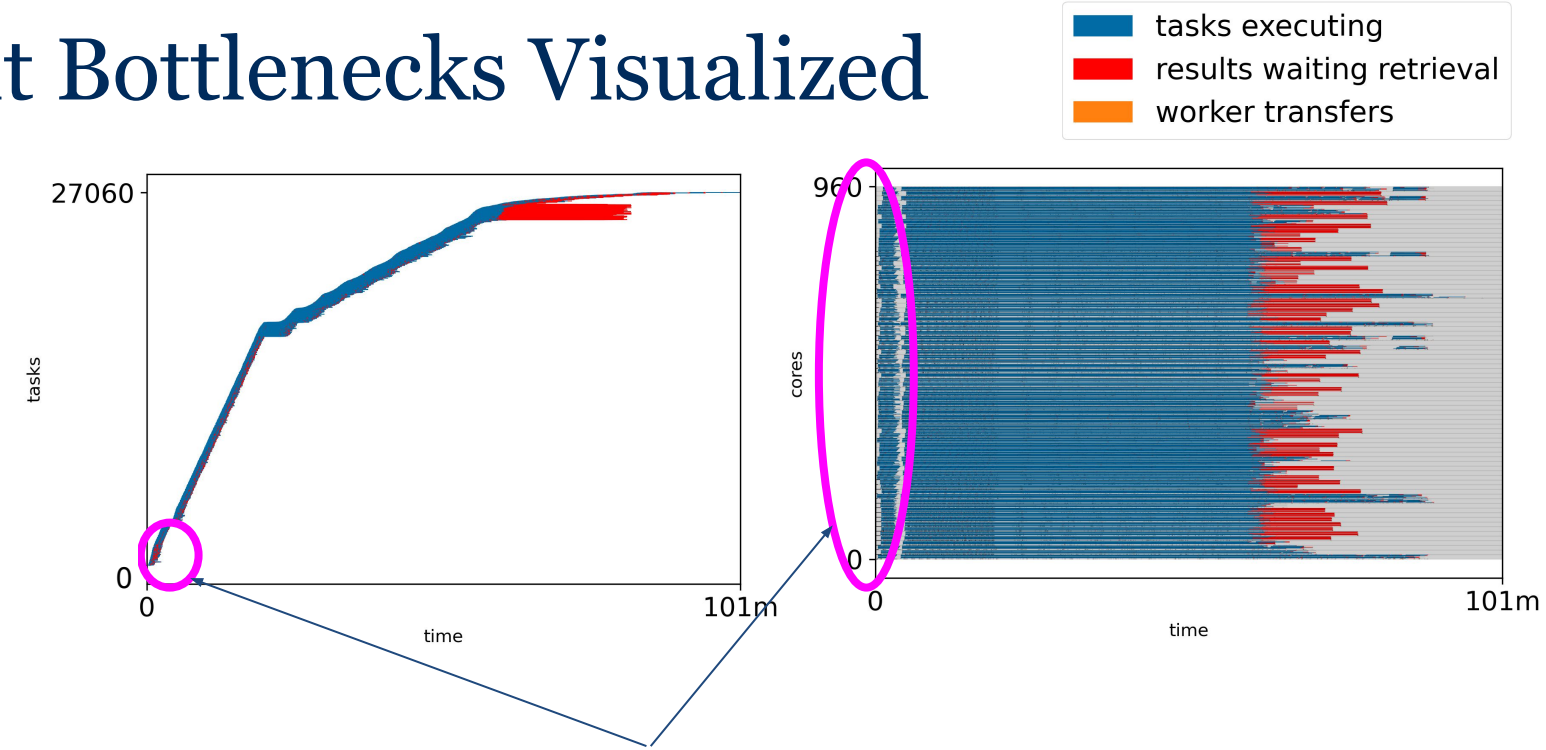
Long worker down time

Long accumulation tail

Whole Analysis: 101 minutes
Run over all Run 2 data and Monte Carlo
~1 billion events

Current Bottlenecks Visualized

Accumulation Data Returned
TopEFT
+ Work Queue



Preprocessing (~2 minutes)
XRootD requests and asset management

Current Performance Bottlenecks

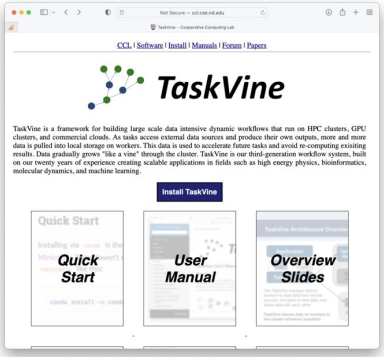
In order of impact:

1. All partial results are returned to the manager, and sent back to workers for accumulation
2. XRootD servers on top of spinning disk, which greatly limits bandwidth
3. Extra data read by the XRootD protocol that is not part of the read requests
4. Accumulation tasks may need tens of GB of memory, which reduces parallelism
5. Manager does not efficiently had out tasks to workers or obtain workers

Changes Needed to Get There

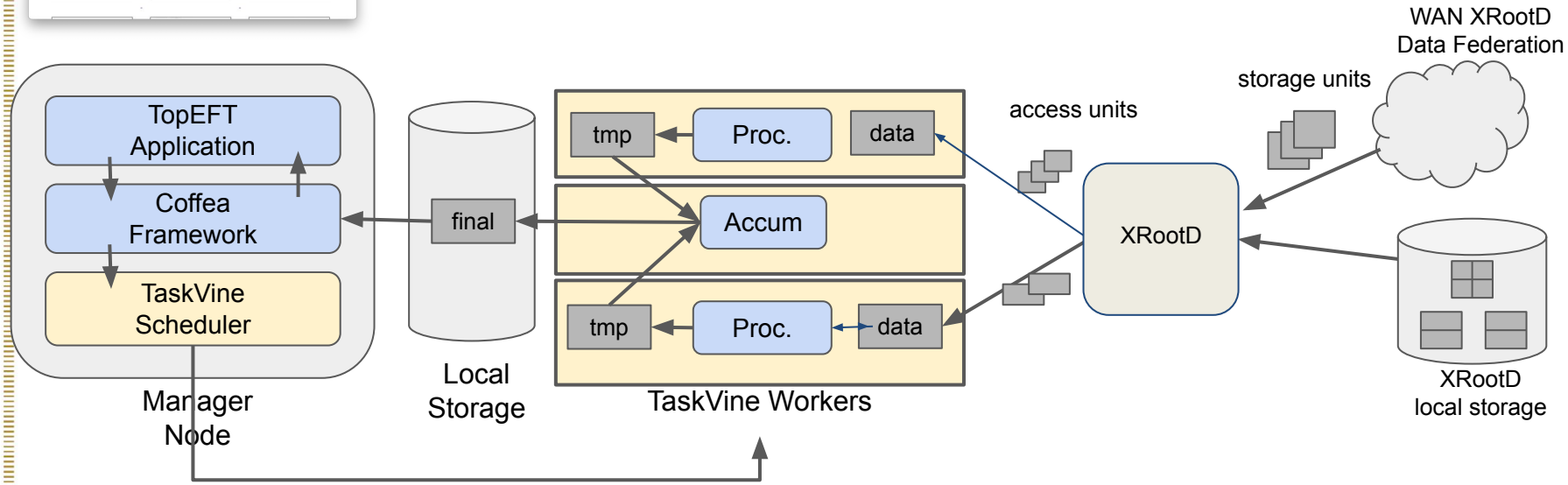
- **Data Storage System:** Every task in the system reads out a different selection of data. Need a data storage system that provides low latency (from open to first read) and high throughput (many clients reading separate data at once.)
  - **New Approach**: Migrating away from HDFS on spinning disk cluster to Ceph on experimental NVMe cluster.
- **Managing Assets for Startup:** A significant amount of turnaround time is lost to startup: allocating nodes, transferring software environments, establishing connections.
  - **New Approach**: Retain as much as possible on each cluster node, and design systems to exploit assets already present.
- **Managing Data Reduction**: TopEFT in particular produces large quantities of intermediate data: transferring it back to a central point results in exponential growth of network traffic:
  - **New Approach:** Leave data where it is created in the cluster, and dispatch accumulation tasks to consume it in place. (Requires closer attention to failure and recovery.)

Next: TaskVine Workflow Scheduler



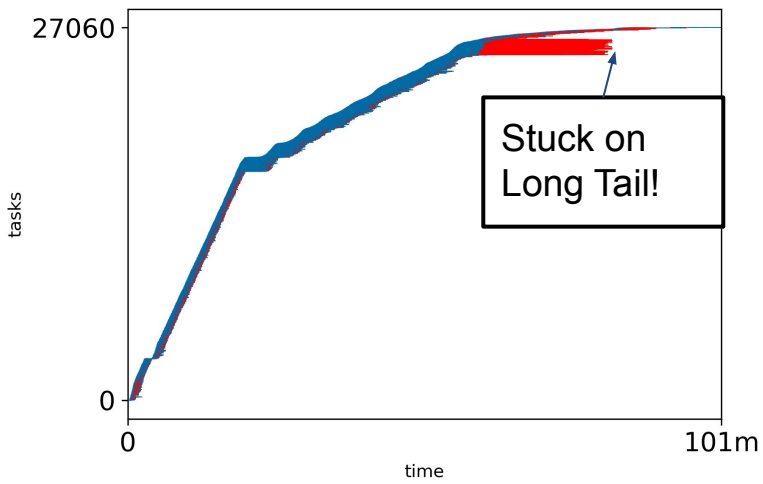
TaskVine is our next generation of workflow scheduler that improves upon Work Queue. Key idea: **data stays in the cluster** where it is accessed or created, so that tasks can simply use data in place, rather than moving it around. Our prototype of TopEFT running on TaskVine eliminates the "long tail" of accumulation tasks by keeping the intermediate data in place.

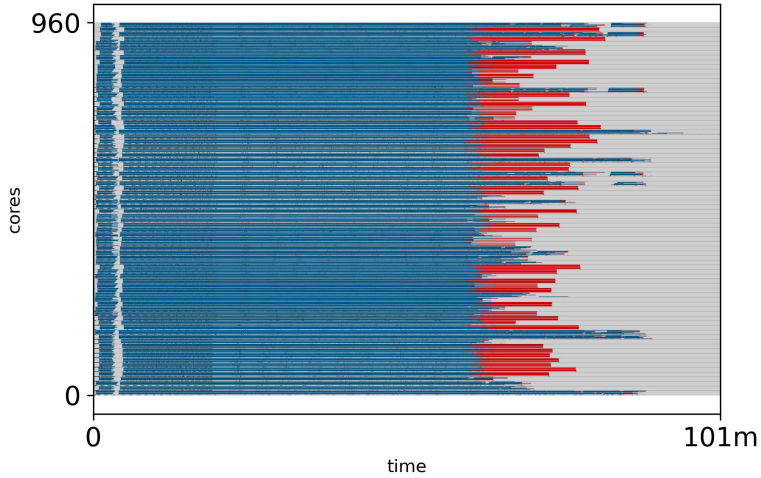
http://ccl.cse.nd.edu/software/taskvine



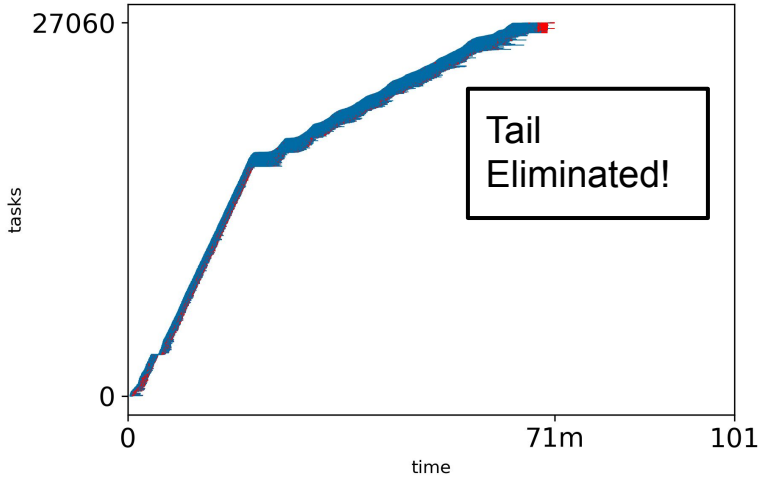
UNIVERSITY OF NOTRE DAME

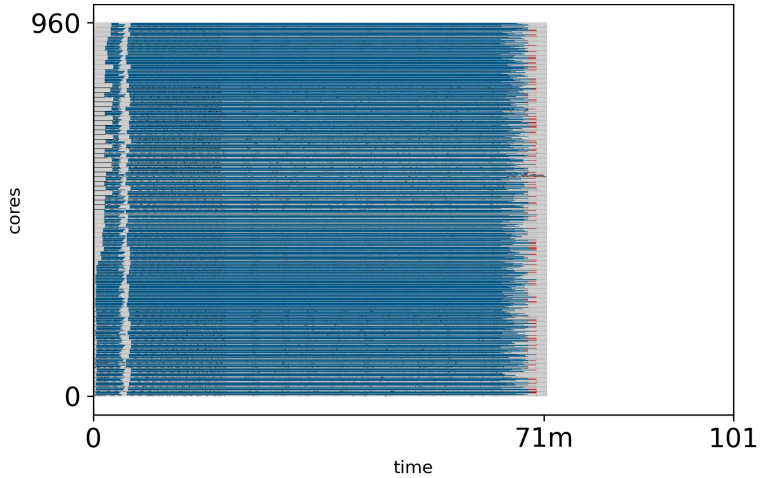
Old:
Accumulation
Data Returned
TopEFT
+ Work Queue





New:
In-Cluster
Accumulation
TopEFT
+ TaskVine





Conclusion

- TopEFT analysis using the Work Queue system as an example for analyses highlighting problems and potential solutions
- Implementing TaskVine to improve throughput by keeping partially finished results at worker nodes
- Improved a major bottleneck and cut runtime down by 20%

Thank You!

Notre Dame CMS group and CCL team

TaskVine

TaskVine