October 19 - 25, 2024
CHEP 2024

ML-based Adaptive Prefetching and Data Placement for US HEP Systems

Authors: *Sai Lamba Karanam*, Sarat Barla*,*
*Prof. Byrav Ramamurthy** and *Dr. Derek Weitzel~**
**School of Computing ~Holland Computing Center*
University of Nebraska-Lincoln (UNL)

Presented by ***Prof. Byrav Ramamurthy***
ramamurthy@unl.edu

Presented at **CHEP 2024**
https://indico.cern.ch/event/1338689/
October 23, 2024

Project funded by the US Department of Energy (DoE) AI for HEP grant (Award # DE-SC0024648)

U.S. DEPARTMENT OF ENERGY

cse.unl.edu/~netgroup

University of Nebraska
HOLLAND COMPUTING CENTER

IN OUR GRIT, OUR GLORY

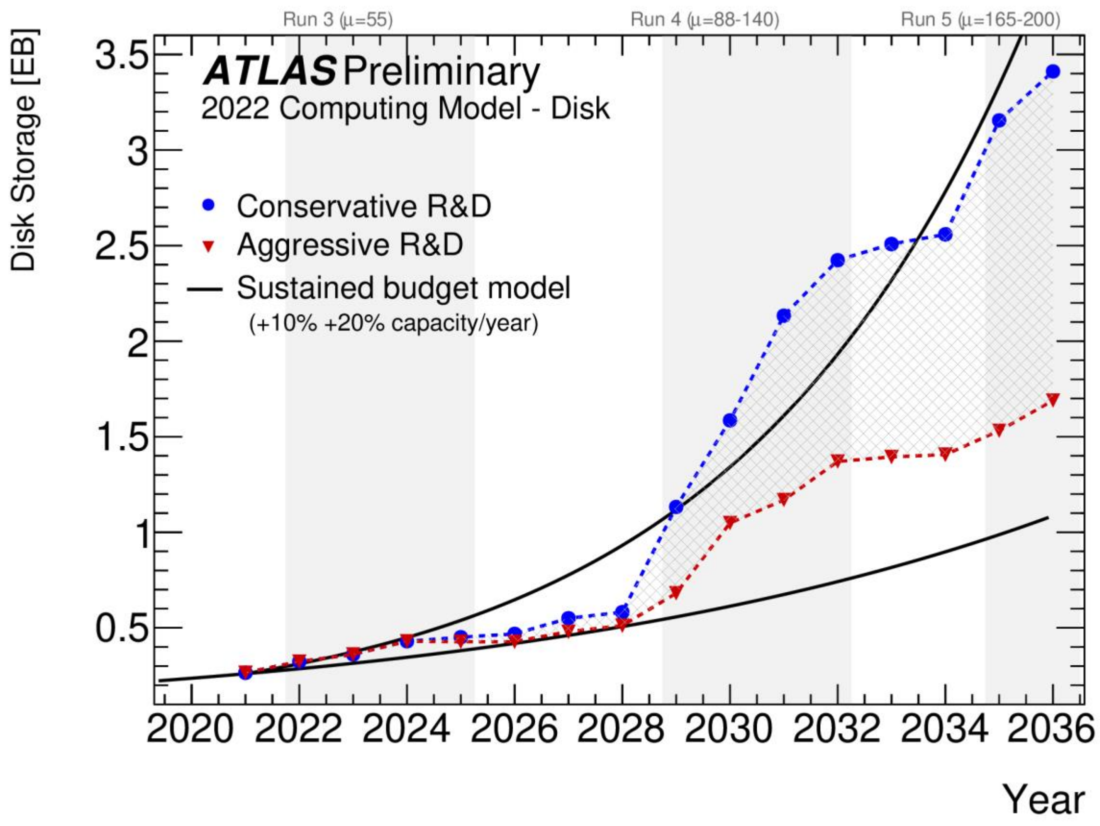
Caching in HEP

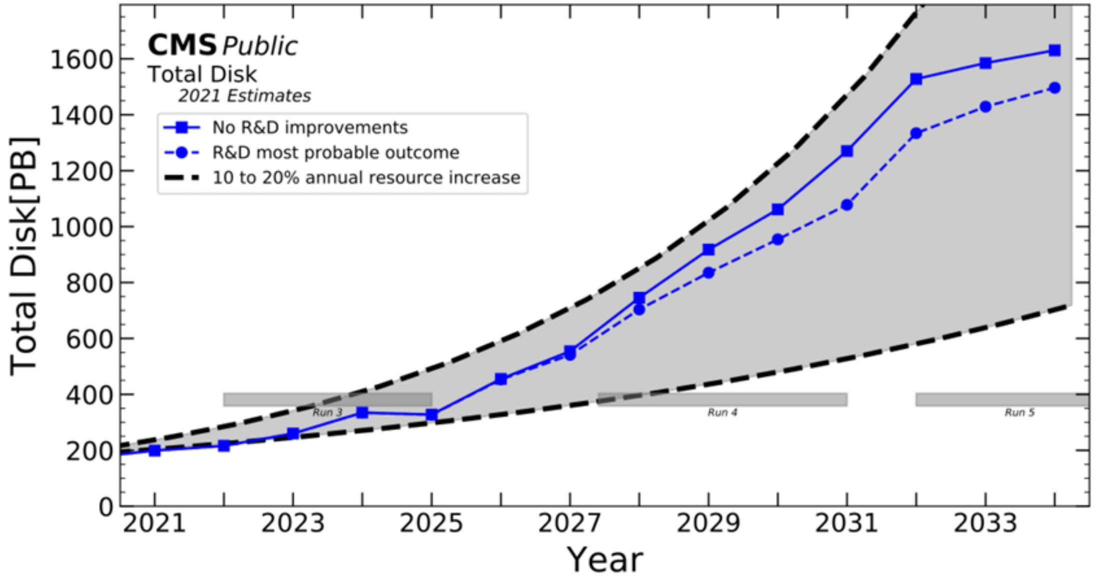
- Benefits from caching in High Energy Physics (HEP) infrastructure are well established [1]
- Current caching strategies are *not adaptive* i.e., do not adapt to changing access patterns
- Not all sites have dedicated cache servers due to administrative control and budget

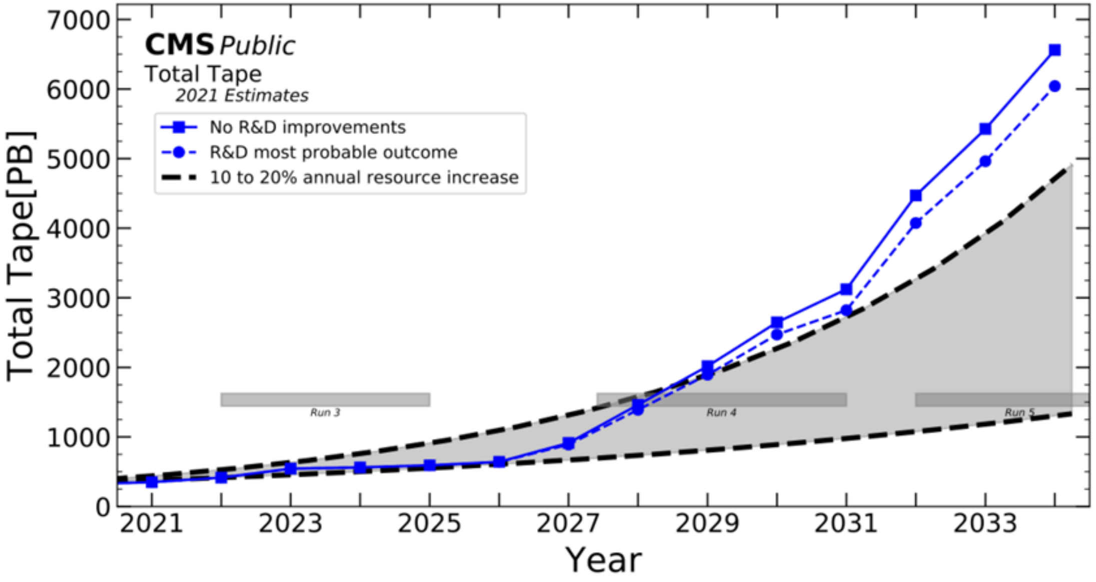
Newer Developments in HEP

- Newer experiments like High Luminosity-LHC (HL-LHC) and Deep Underground Neutrino Experiments (DUNE) will produce >10x more data and equally large computational loads
- Newer Data Acquisition Systems (DAQs) are moving towards offloading storage to remote sites [2, 3]

Projected evolution of Disk Usage for ATLAS and Disk and Tape Needs for CMS into HL-LHC







[ATLAS Software and Computing HL-LHC Roadmap](#)
[ATLAS Software and Computing HL-LHC Roadmap - v 2.1 (cern.ch)](#) (PDF)

[CMS HL-LHC Roadmap](#) (PDF)

Some Observations

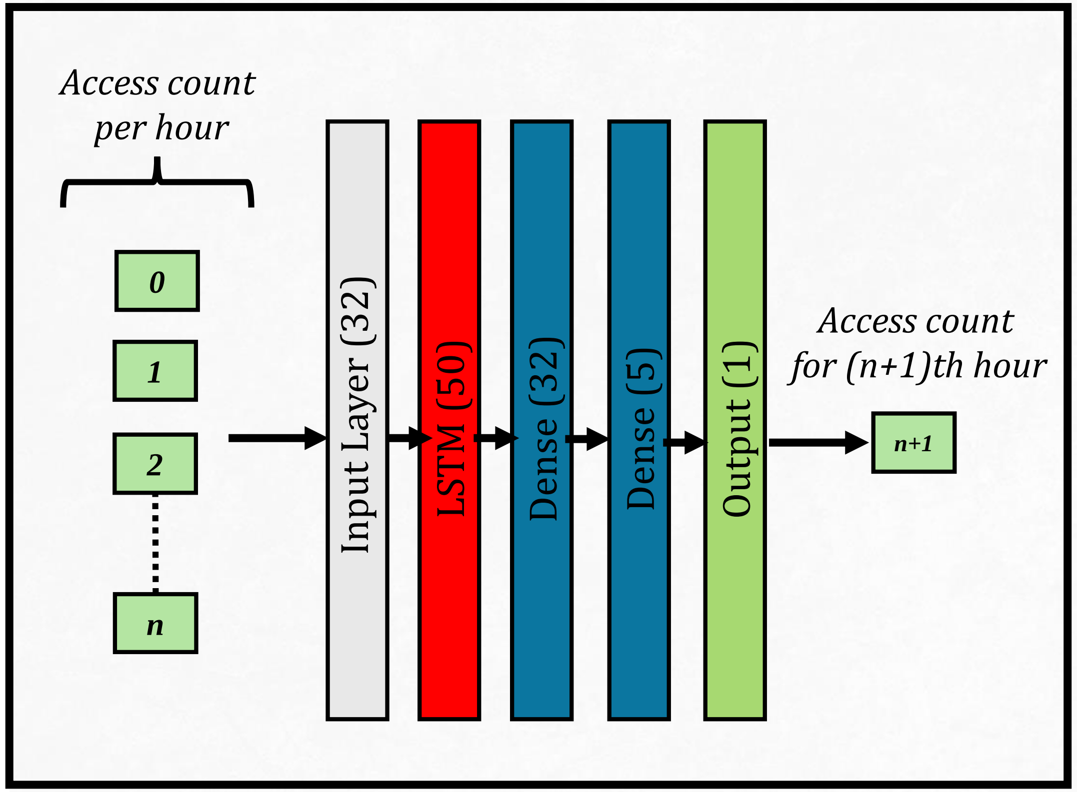
- CPU efficiency and latency approx. same between local reads at non-custodial facilities vs. cache reads from custodial sites [4]
  - *Reading from remote non-custodial sites often adds latency* → showed worse performance [4]
- We do not have enough capacity to fulfill data locality!
  - Current cache deployments are optimized for speed, but ***not for storage***
  - *e.g., 20TB cache vs 100TB total at the origin for a CMS NANOAOD regional Data Lake* [US CMS Data Lake Proposal 2020]

Some Current Developments

- Reduced data format storage size: RECO → AOD → MiniAOD → NANOAOD
  - Currently for CMS, as of 2022 (end of Run 2), 30% of analyses adopt NANOAOD; expected to increase up to 50% by the end of Run 3 [CMS HL-LHC Roadmap]
- Current ATLAS strategy is to unpin all data once a copy is created on tape [ATLAS HL-LHC Roadmap]
  - *Intuition is that it only retains popular data in disk and unpopular data are retained/stored in tape (at custodial site)*
- DAQs are moving towards streaming readout systems from the Triggered Data Acquisition systems (TDAQs) [Streaming Readout for Next Gen Scattering Experiment]
  - Supports near real-time processing of raw data by offloading to compute & analyses facilities
  - Increased data movement places burden on the limited storage
- Prefetching solutions exist but they are ***not adaptive*** [XCache Developments and Plans 2023]
  - Reading the missing blocks **in order** → cannot keep up with faster compute/read rates

- Predicting cache usage using Neural Networks (NN) such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs)
  - Great for time-series forecasting
- Prediction of cache usage ***per day*** was done previously [CHEP 2022]
  - Here, we investigate predicting cache usage ***per hour*** using LSTM
  - Hourly cache usage prediction is useful for:
    - Prefetching data into analyses locations
    - Reducing the data transfer redundancy by retaining data that are likely to be accessed in the near future

LSTM Model Architecture



Model Hyperparameters

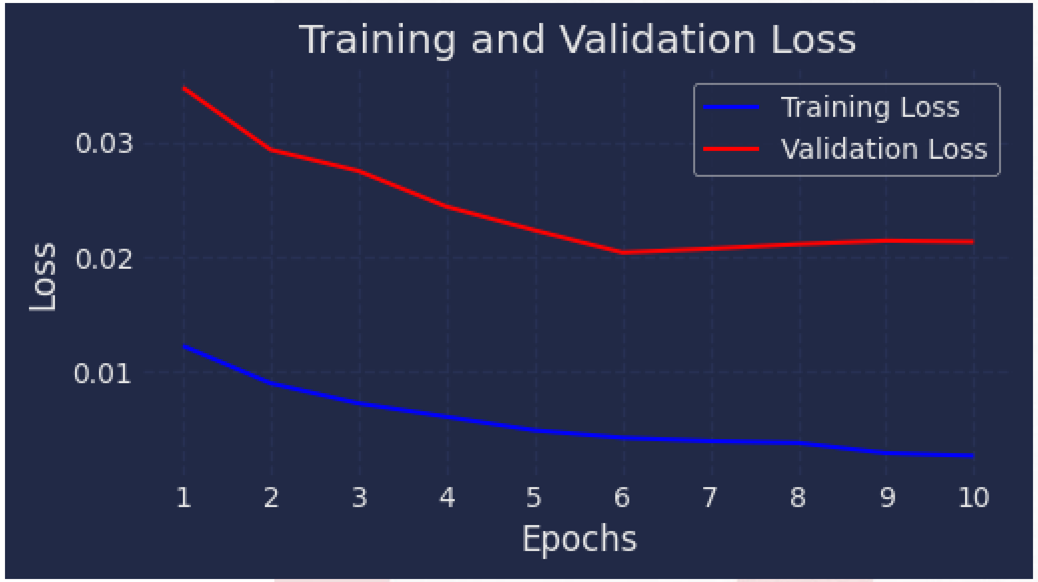
Name	Value
Optimizer	Adam
Learning Rate	0.01
Batch size	32
Train-Val-Test ratio	4:1:1
Activation	ReLU
Recurrent Activation	Sigmoid
Loss	Mean Square Error (MSE)

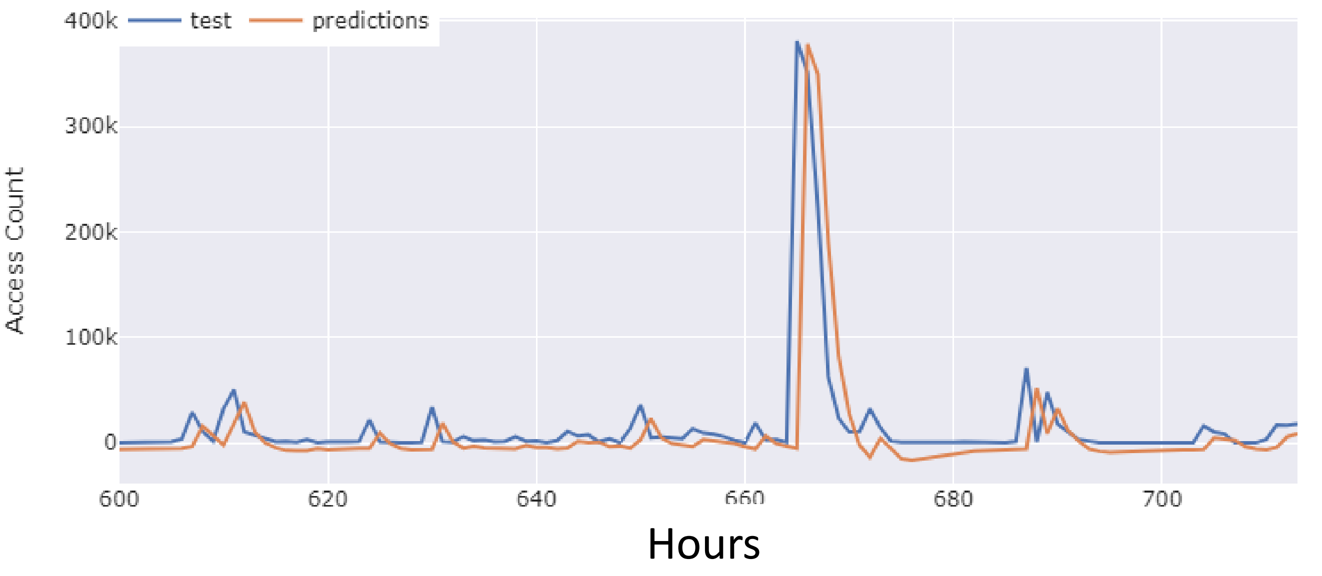
What we found with hyperparameters:

- Choice of optimizer and other batch sizes like 8, 16, 64 only affected the model convergence but not its predictions
- Multi-step prediction i.e., predicting access counts of more than one step (i.e., hour) into future was explored.
  - The performance of multi-step prediction is worse when there are unexpected peaks in accesses. Example: when there is a highly popular data for short periods of time

- Cache usage prediction: Access counts
- A simple Long Short-Term Memory (LSTM) model is able to predict the access frequency at each cache server

*xrd-cache-**xx***



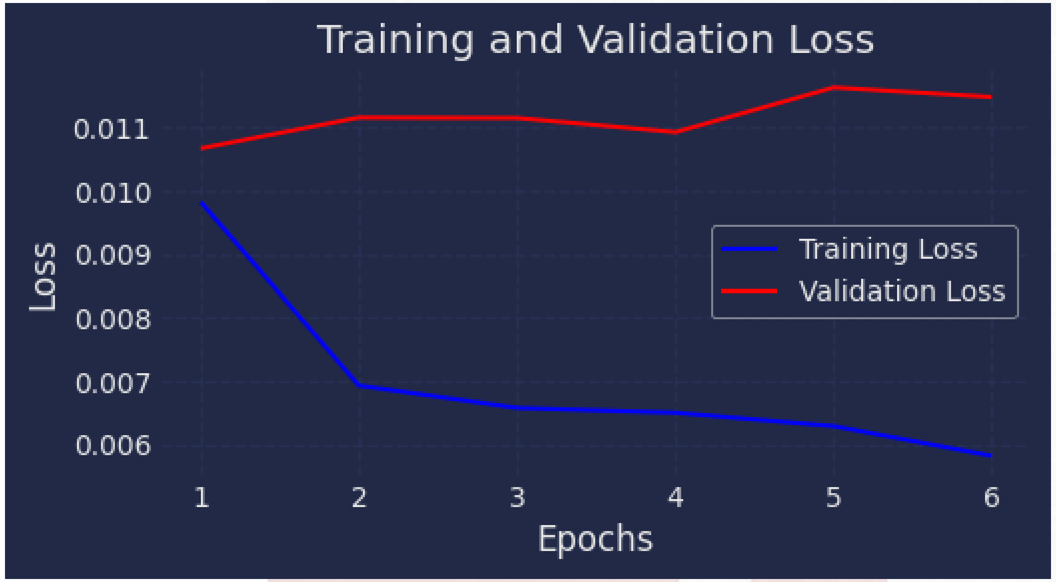


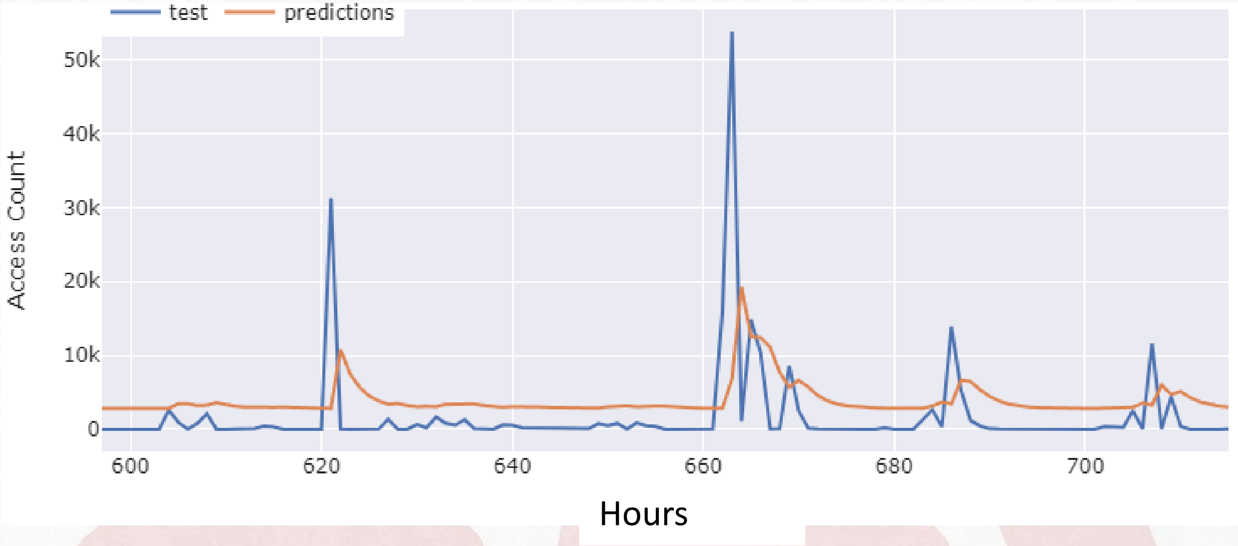
Relative error (MAE): 23.88 %

PRELIMINARY ANALYSIS (CMS CACHE): SOCAL REPO (2)

- Overall-MAE is higher because the model is worse at predicting sudden popular data

xrd-cache-yy





Relative Error (MAE): 150.13% **overall**

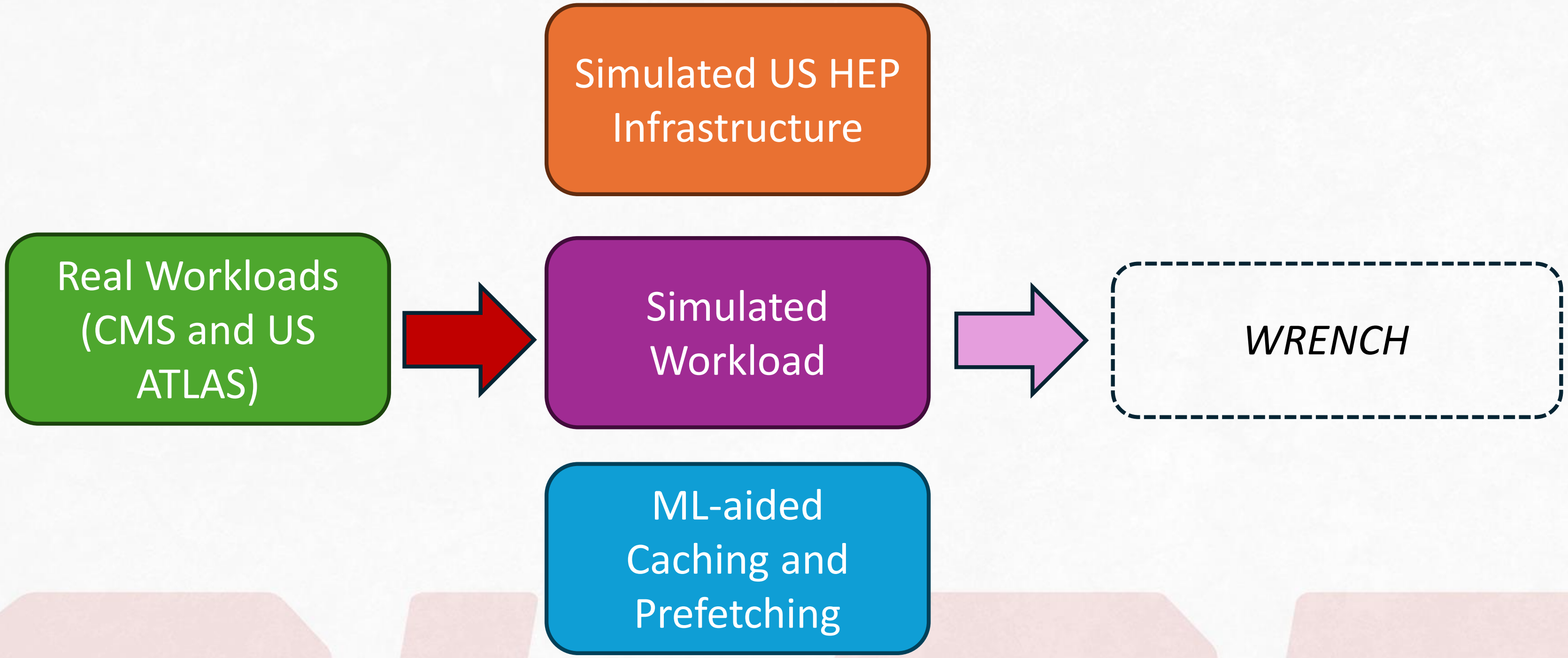
- Data set access is highly skewed, and prior studies support this finding as well[1]
  - We also validated this with our preliminary analysis
- Choice of remote source site for a file impacts the transfer latency
  - With multiple choices of sources (cache and non-cache), this becomes a combinatorial problem
- Prefetching is an attractive solution in parallel with caching
  - So is retaining files in the cache that are likely to be accessed in the ***near future***
- ***Above strategies require:***
  - *Accurate and realistic prediction*
  - *Testing on real workloads and on real infrastructure (combination of candidate storage, compute and network values)*

1. Bellavita, Julian, Caitlin Sim, Kesheng Wu, Alex Sim, Shinjae Yoo, Hiro Ito, Vincent Garonne, and Eric Lancon. "Understanding Data Access Patterns for dCache System." In *EPJ Web of Conferences*, vol. 295, p. 01053. EDP Sciences, 2024.

Goals

- Design ML-aided caching and prefetching strategies for US HEP environments
- Difficult to test developed strategies empirically on real infrastructure
- Testing of developed cache prediction strategies on real testbeds is an issue
  - Access to testbeds
  - Impact on existing workflows
- **Plan:**
  - **Current**
    - Understand cache access patterns in the US HEP ecosystem, including DUNE in the future
  - **Future**
    - Design simulator to reflect the real-world US HEP ecosystem
    - Exemplary topology w/ storage, compute and network capabilities for USCMS and US ATLAS Collaborations
    - Workload simulation from real world CMS and US ATLAS cache usage logs
    - Conduct testing of designed ML-aided caching and prefetching strategies

DEVELOPMENT PLAN



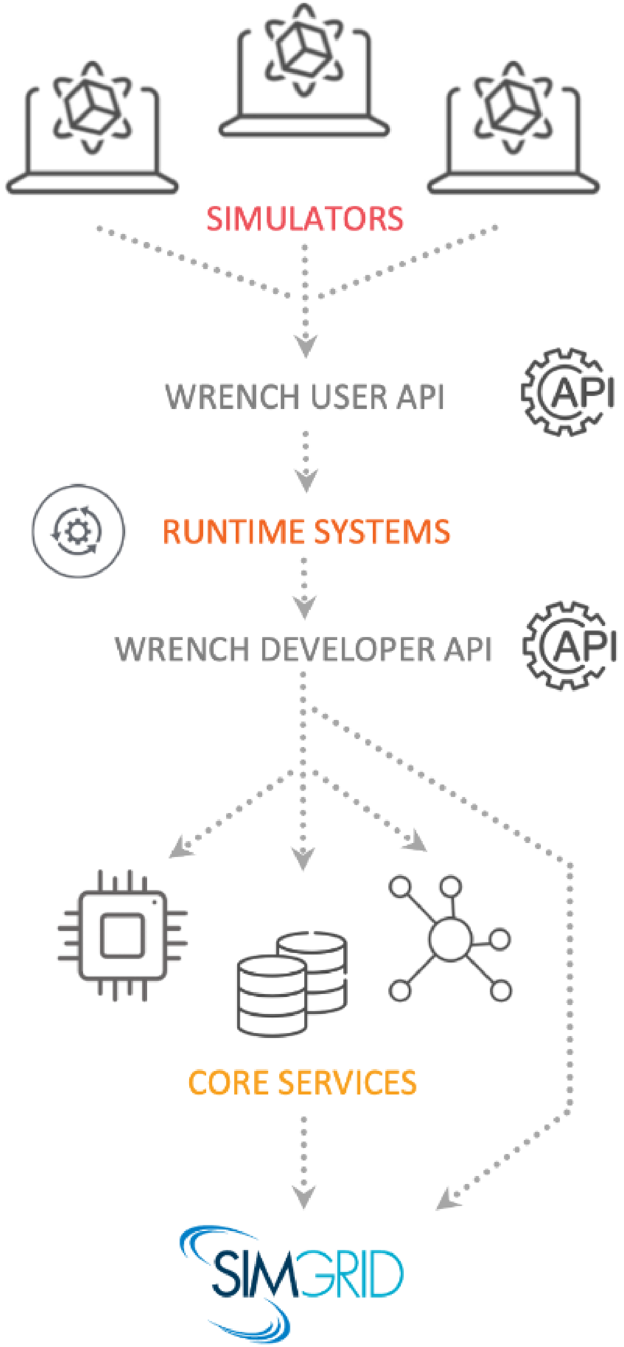
WRENCH SIMULATOR (1)

1. Workload Execution

2. Workload Management Systems (WMS)
   - HTCondor

3. Storage (XRootD), caching/data locality*
   Computation (Batch and Streaming*)
   Networking

4. Low-level abstractions (C++)



*specified support was added in the DCSim extension ----→" Modeling Distributed Computing Infrastructures for HEP Applications"

Why we need a simulator in the first place?

- Simulations allow
  - Comparison between candidate compute & storage w/o testbeds
  - Workflow Management Systems (WMS) w/o empirical experiments
  - Predicting the performance of deployed candidate ML/AI-based caching and data placement strategies w/o impacting existing workflows
  - Comparison against complex data and cache access patterns
  - Reproducibility!
- ***Validation***: when simulation executions agree with ground truth
  - Split the historical data into train and test for comparison

- Simulated workload is obtained from USCMS cache servers (MINIAOD and NANOAOD) and US ATLAS (XAOD)
  - Extracted information from workloads: *cache server name, client name, experiment identifier, read/write time (total, average, number of read operations etc.,), number of copies per file*
- Logs from MONIT dated January though March 2024
  - Extracted information: *total data processed by volume, amount of read/write data, memory consumed, total job time, #CPU Cores*
- Create a classes of jobs by type: *analysis, simulations, processing, merge jobs, admin*
- Probability distribution of jobs per class → implemented as events in the simulator with measured properties

- One Cache Server and multiple Analyses sites approx. representing the real-world US HEP infrastructure
  - The values are representative (not necessarily accurate) and will change in the future as capabilities improve

Property	***Cache Server***	***Analysis Site***
Compute	*50,000 CPU cores*	*15,000 CPU Cores*
Storage Space	*Variable*	*Variable*
Storage BW	*80 Gbit/sec*	*NA*
Cache Space	*NA*	*15% of storage*
Cache BW	*NA*	*80 Gbit/sec*
LAN BW	*2 x 100 Gbit/sec*	*40 Gbit/sec*
WAN BW	*>200 Gbit/sec (LHCONE/ESNet)*	*>200 Gbit/sec (LHCONE/ESNet)*

- Integrate the ML-aided caching into the simulator
- Extend the current model to support a file-level and hourly access prediction
  - Integrate additional parameters from the infrastructure and the workflow as input features
- Specifically, to predict future accesses using LSTM/NNs for:
  - Hot pinning popular data/files in the cache
  - ML-aided Least Recently Used (LRU) cache eviction scheme
  - Prefetch future files from remote caches

CONCLUSION

- ML-aided caching and prefetching for US HEP ecosystem, specifically USCMS and US ATLAS

Current work:

- Implementing Prefetching aided by LSTM-based cache usage prediction
  - ***Hourly***
  - ***Per file***
- Data collection and analysis
- WRENCH implementation (ongoing)

Future line of work:

- Incremental updates for better accuracy, support for additional infrastructure elements and workloads
- Alternative ML architectures beyond LSTMs and GRUs and cache policies
- Long term goal is to deploy these solutions in the operational frameworks for HEP ecosystems, starting with OSG

- GitHub - HEPCompSim/DCSim at v0.3
- WRENCH (wrench-project.org)
- GitHub - wrench-project/wrench: WRENCH: Cyberinfrastructure Simulation Workbench

1. Fajardo, Edgar, Derek Weitzel, Mats Rynge, Marian Zvada, John Hicks, Mat Selmeci, Brian Lin et al. "Creating a content delivery network for general science on the internet backbone using XCaches." In *EPJ Web of Conferences*, vol. 245, p. 04041. EDP Sciences, 2020.
2. Lawrence D. Streaming Readout and Remote Compute. Thomas Jefferson National Accelerator Facility (TJNAF), Newport News, VA (United States); 2023 May 1.
3. Suiu, Alice-Florenţa. "EPN2EOS Data Transfer System." PhD diss., University POLITEHNICA of Bucharest, 2023.
4. Horzela, Maximilian, Henri Casanova, Manuel Giffels, Artur Gottmann, Robin Hofsaess, Günter Quast, Simone Rossi Tisbeni, Achim Streit, and Frédéric Suter. "Modeling Distributed Computing Infrastructures for HEP Applications." In *EPJ Web of Conferences*, vol. 295, p. 04032. EDP Sciences, 2024.

Thanks!

Questions?

ramamurthy@unl.edu

Project funded by Department of Energy (DoE) AI for HEP program grant with award number DE-SC0024648

Any opinions presented in this talk reflect only the opinions of the authors and not of DoE

U.S. DEPARTMENT OF ENERGY