ACAT 2019

Automated and Intelligent Data Migration Strategy in High Energy Physical Storage Systems

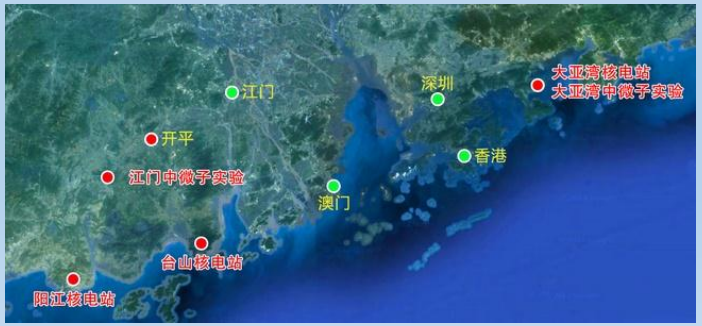
Zhenjing CHENG

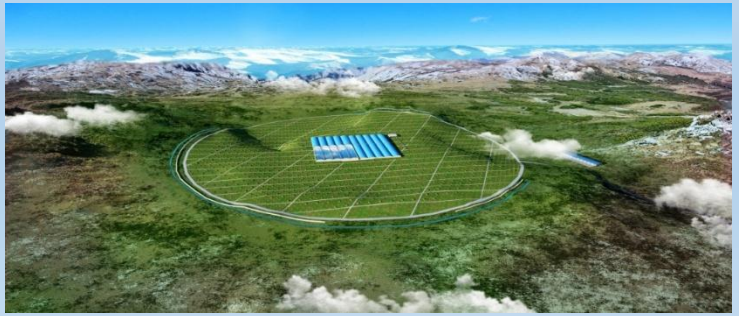
(IHEP Computing Center)

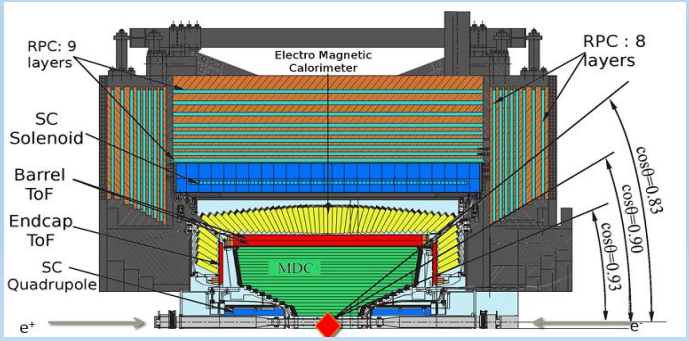
Motivation

- High Energy Physics Computing ⟶ data intensive
  - Experiments like JUNO, LHAASO and BESIII store and produce near 100 PB data(***increasing***)
  - need better data access performance(or I/O bandwith)

- Future Storage ⟶ Huge and distributed clustered storage
  - Hundreds of servers and tens of thousands clients
  - SATA HDDs can't provide **higher IOPS**! ⟶import flash disks
  - Limited fundings ⟶ all-flash storage is too expensive!
  - **Build hierarchical and tiered storage(tapes, disks, SSD)**







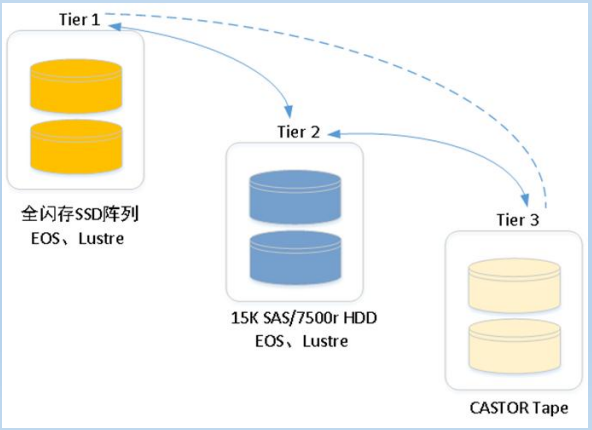
Motivation

- Tiered storage need data migration strategy
  - less active data be moved to lower cost storage devices regularly

Jiang S, Davis K, Zhang X. Coordinated multilevel buffer cache management with consistent access locality quantification[J]. IEEE Transactions on Computers, 2007, 56(1): 95-108.

- Local site storage: Data access requests are not completely random
  - Data access locality ⟶ **a small set of data keep active** for a certain period of time
    - e.g. certain physics channel events datasets
  - multiple users might analyze the same datasets within a specific period of time

- Can we predict future file access?
  - **identify** hot/warm/cold data or different data use cases based on file access
  - **optimize data migration strategy based on file heat changes**



Related studies

- Huge gap of data access times between memory and disk
  - **To alleviate the problem**: caching and prefetching
  - prefeching: bring data in memory before they are needed
  - similarily, bring hot file in **SSD tiers** to improve I/O performance and move cold file out
  - Make as many correct predictions as possible and as few false predictions as feasible

- Widely applicable file access predictors
  - Stable Successor:
  - Recent Popularity:
  - **Disadvantages:**
  - Short-term prediction
  - stand-alone prediction, not suitable for mass parallel storage system like EOS
  - rely on file access order heavily

Challenges:

- EOS cluster operators don’t understand users’ data meanings
  - We only know users’ file history access statistics - by analysising eos fst logs

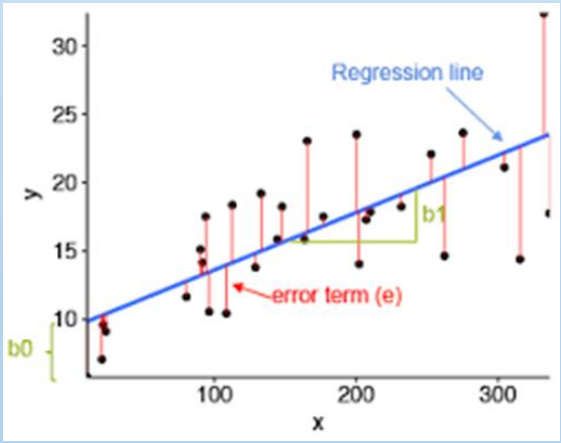
- Prediction model
  - should be suitable for massive files and data parallel access
  - shouldn’t rely too much on file access order

- ***regression analysis***

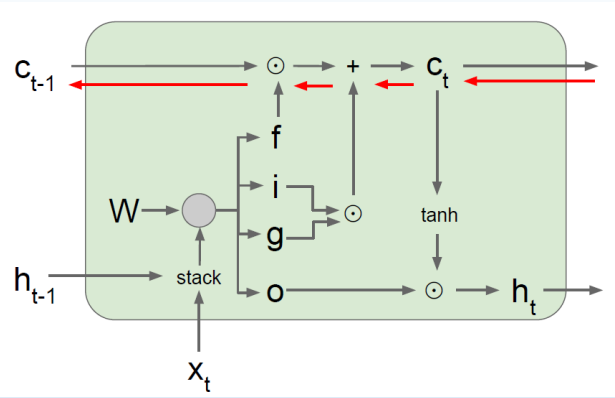
  - load predictions for continuous time :

  — High-energy physics storage : billions of files

  — impossible to build a regression model for each file

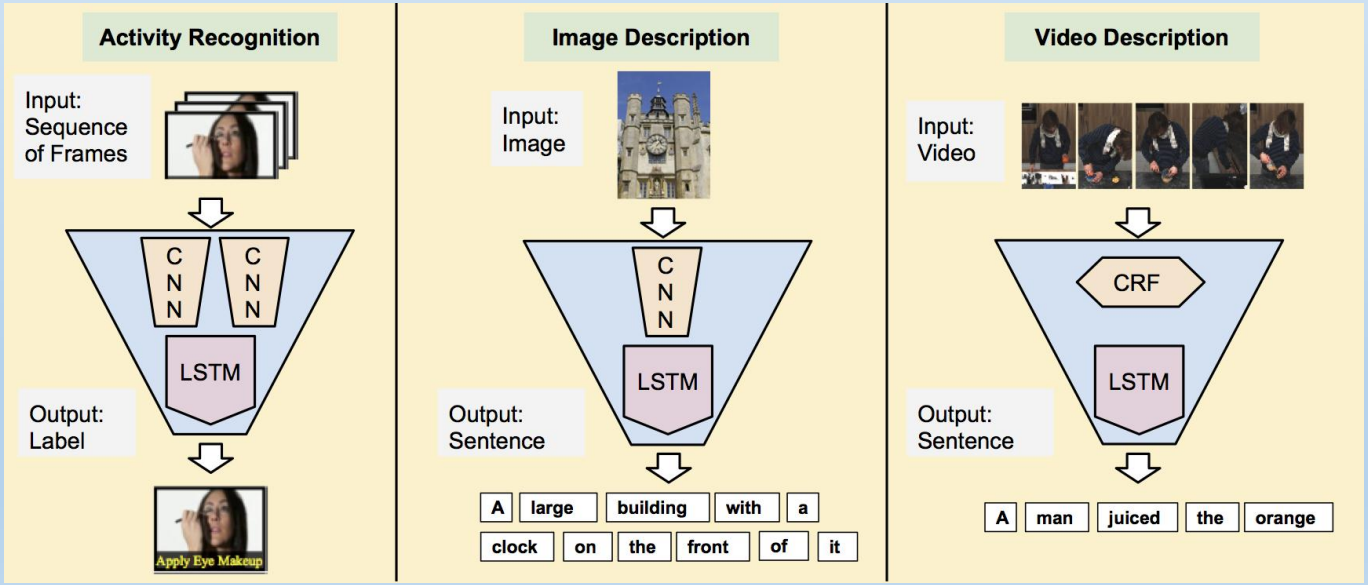


Prediction Model



- Deep Learning Algorithm : LSTM(Long short-term memory)
  - improved RNN capable of learning the long-term dependencies
  - recognize patterns in sequences of data, like text, handwriting, or numerical times series data from sensors, stock markets. E.g.

- *Application*



Prediction Model

- *Define data heat*

  (1) "*hot data*",

  - substantial reuse of small amount of high-energy datasets by users for a long time.
  - migrate to faster storage like SSD and SAS

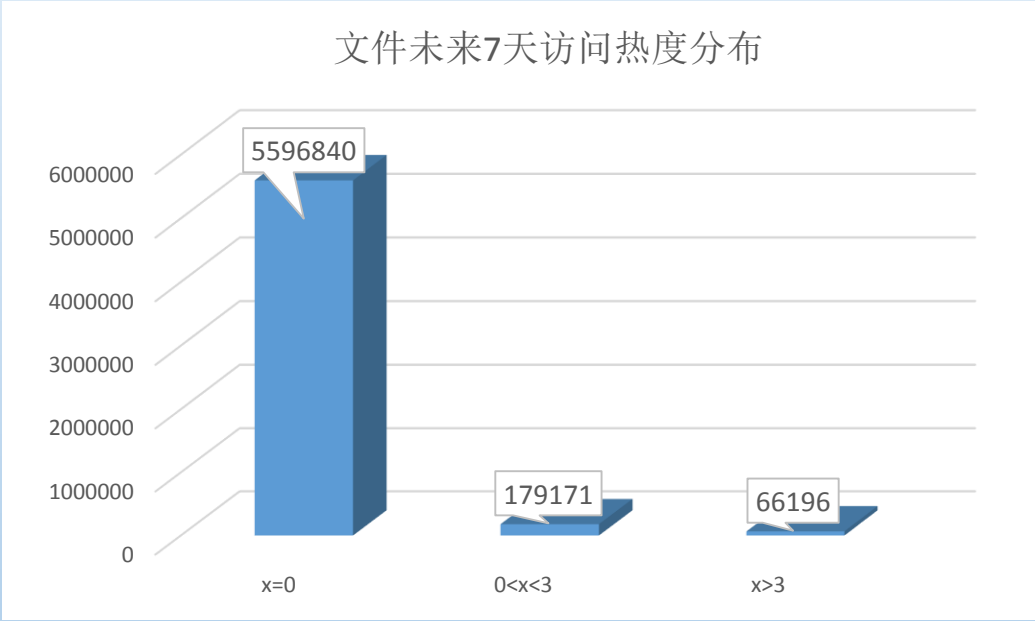
  (2)" *cold data*",

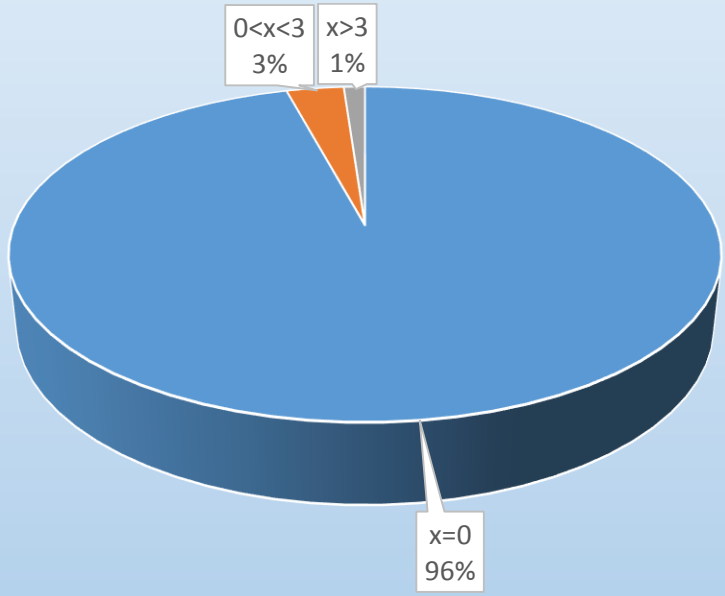
  - mass high-energy datasets used by a single user for limited processing times.
  - migrate to lower but high-capacity storage like HDD

- *So we divided files into* different categories ***<u>for different data heat</u>***

  — based on number of file access

- *Distribution of file access number within 7 days*





Model Input(*File Access Feature*)

- *EOS: FST logs keep data access records in file units, as follows*
- Provide file history read/write ratio, Re-read, Re-write, Random read, Random write and so on

log=a048f57a-6034-11e8-8f98-288023415e08&path=/#curl#/eos/user/b/biby/yinlq/rootdata/QGSJET-FLUKA
*/Helium/1.e14_1.e15/wcda003363.root&ruid=10408&rgid=1000&td=*CioA-gA.1639102:551@vm088029&host=eo*
s07.ihep.ac.cn&lid=1048578&fid=123971808&fsid=25&ots=1527263977&otms=887&cts=1527263998&ctms=734&
rb=0&rb_min=0&rb_max=0&rb_sigma=0.00&wb=8830528&wb_min=63&wb_max=32768&wb_sigma=2225.83&sfwdb=881
4629&sbwdb=8814592&sxlfwdb=8781824&sxlbwdb=8814592&nrc=0&nwc=271&nfwds=3&nbwds=1&nxlfwds=1&nxlbwd
s=1&rt=0.00&wt=24.91&osize=0&csize=8830565&sec.prot=unix&sec.name=root&sec.host=vm088029.ihep.ac.
cn&sec.vorg=&sec.grps=root&sec.role=&sec.info=&sec.app=fuse

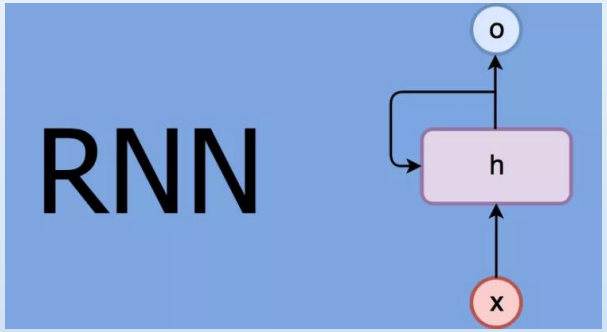
- *Make file access vectors*

  <timestamp, filename, filesize, read/write ratio, read/write bytes sequence/random read >

Model Input(*File Access Feature*)

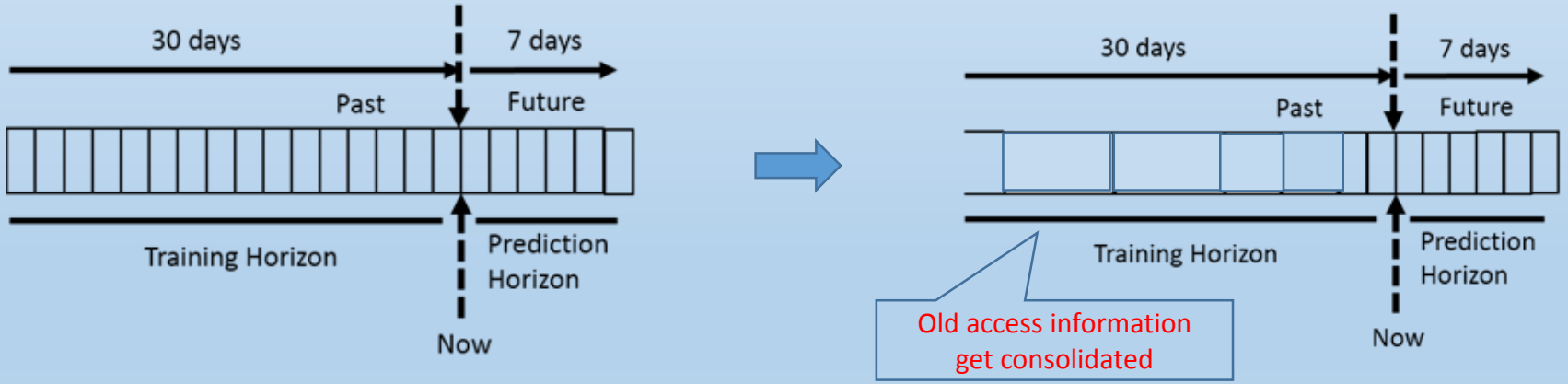
- *Compact multiple vectors into a sequence of time series by hour*

very suit for RNN!



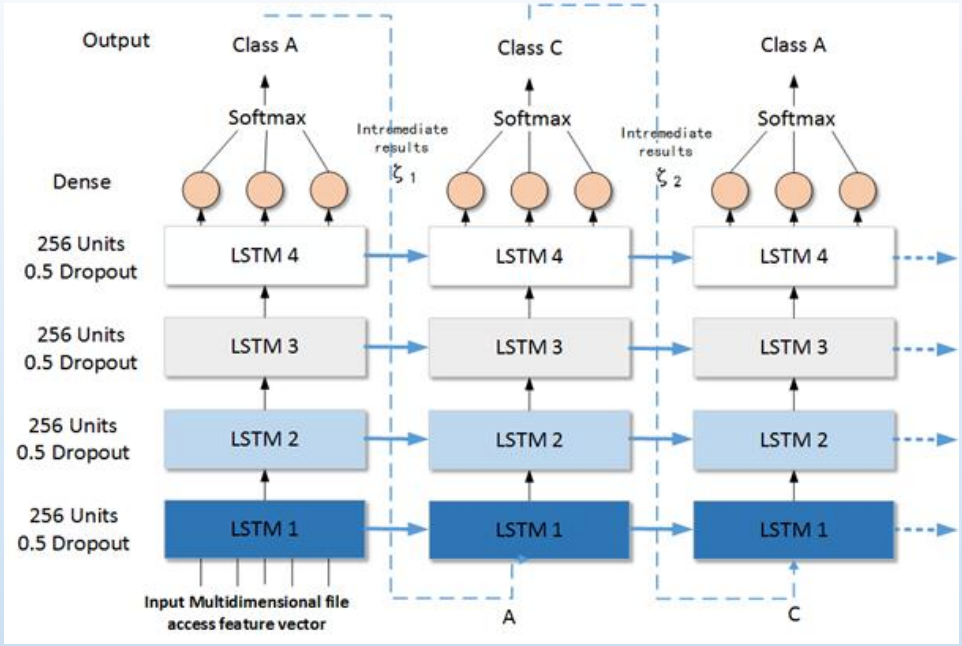
T: timestamp **F_1:**filename **F_2:** file size **R_1:**file read/write ratio **S:** sequential/random ratio **R_2:** file read/write bytes

- *Use access features in the past to predict future file heat*
  - — *dynamical* training time window
  - — the same model complexity, but more historical information used!



System design

- *Set a goal：predict **file heat** in next 7 days*



- *LSTM model :*
  - — *4-layer* fully connected RNN with *64* LSTM cells per layer
  - — learning rate decades *(0.001, 0.0001)*
  - — *256* samples per batch, training at the same time

- *Data set*
  - — source: EOS for LHAASO cooperation group user
  - — 5,842,207 files access records (2018.4.1-2018.5.1)
  - — divided into three groups, *training data set(80%), verification data set(10%), test data set(10%)*

Results

- Accuracy
  - Hot file prediction accuracy: 87.52%
  - Cold file prediction accuracy: 92.89%
  - Overall classification accuracy : 91.78%

- Other metrics
  - TPR/Recall: 0.9532 FPR: 0.1028

Conclusion and next step

- *Hierarchical storage* is the trend for IHEP storage. Deep learning helps make file heat prediction.
- Now binary classification, multiple decisions in future for *multiple storage layers*

- Didn’t consider impact brought by data migration to the storage performance
- Introduce the concept of migration cost, consider impact on storage performance
- Adaptive and Automated file migration strategy, more adaptive to storage load changes

Thanks