Yandex

Data Placement and Data Handling in LHCb

Mikhail Hushchyn, Andrey Ustyuzhanin

Outline

1. The LHCb data management
2. Replication placement strategies
3. Cache algorithms
4. The LHCb data management improvements
5. Data grid simulation

The LHCb data management

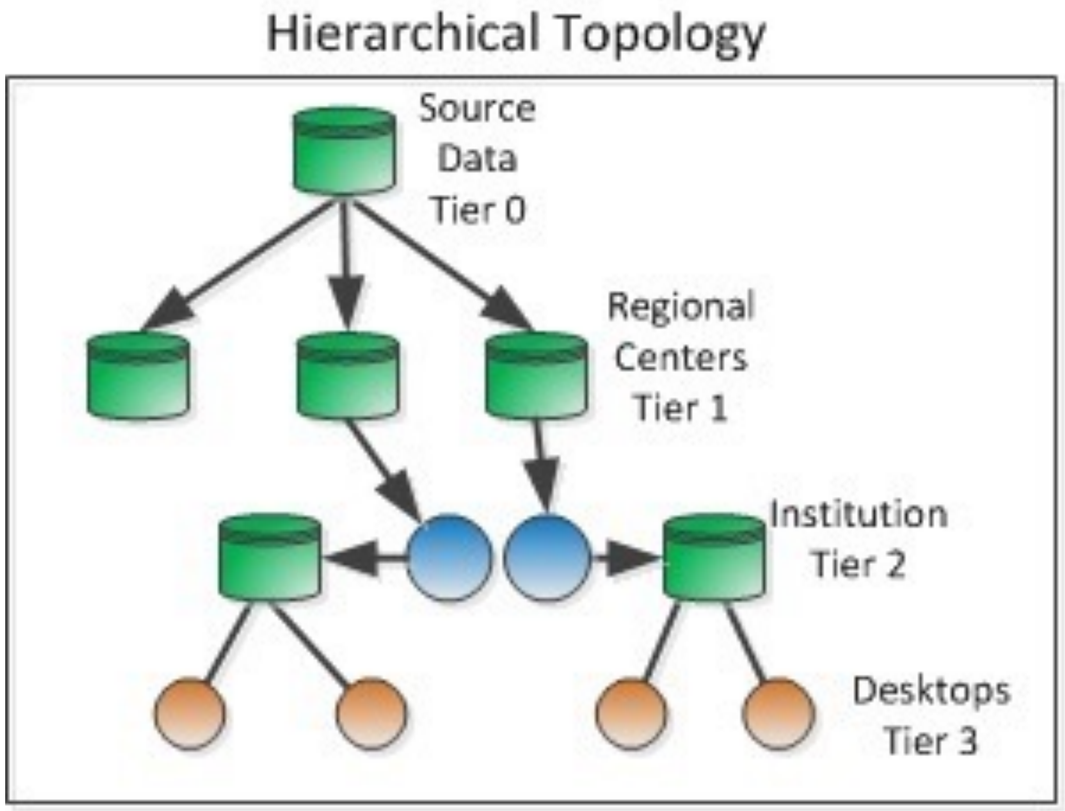
The LHCb Data Grid

Three spaces at each Tier1:

〉 LHCb-Tape: T1D0 custodial storage

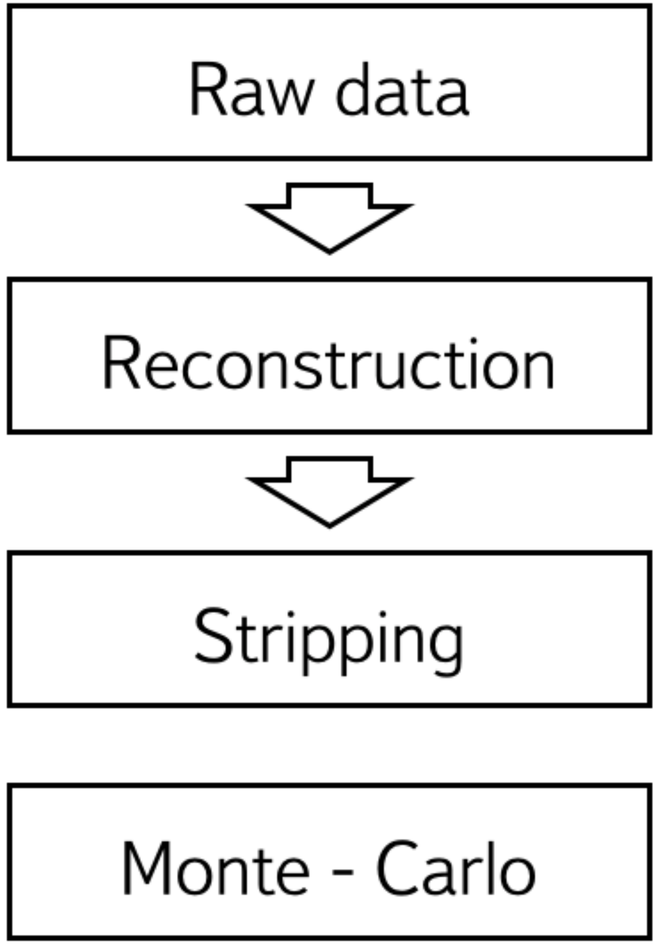
〉 LHCb-Disk: T0D1 storage for centrally managed datasets

〉 LHCb-USER: T0D1 for user datasets stored on the Grid



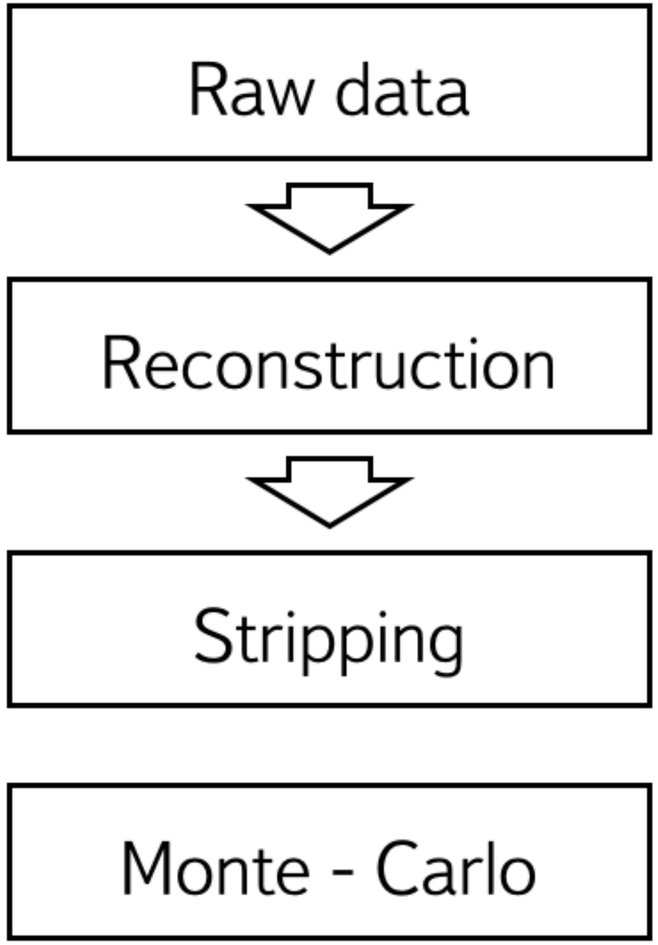
Raw data

- One replica on T0D1 at a Tier1 and one at CERN.
- One archive replicas on T1D0 custodial storage, one at CERN.



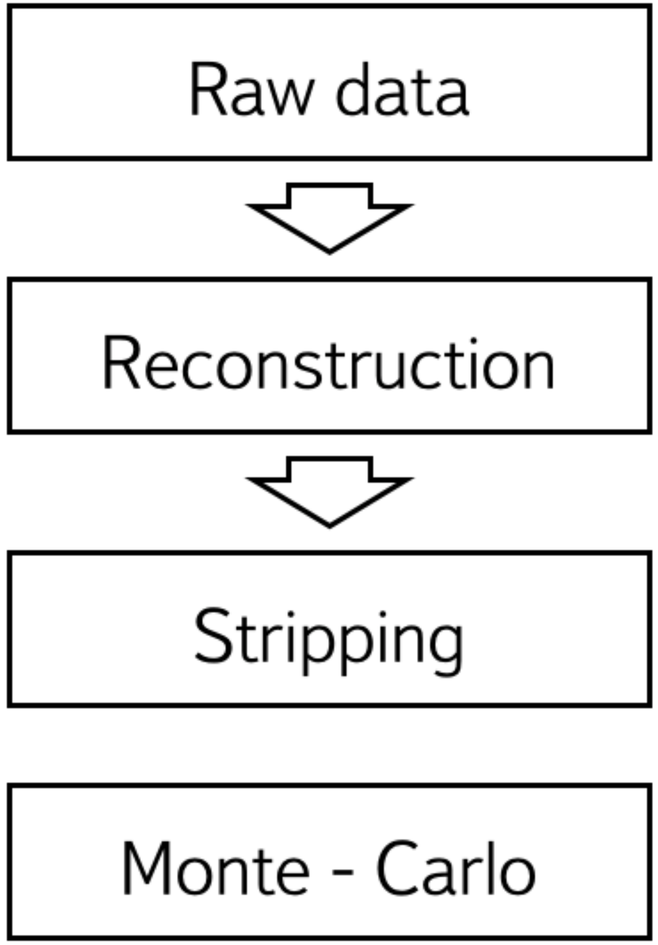
Data Reconstruction

- One replica on T0D1 at a Tier1 and one at CERN.
- One archive replicas on T1D0 custodial storage, one at CERN.



Data Stripping

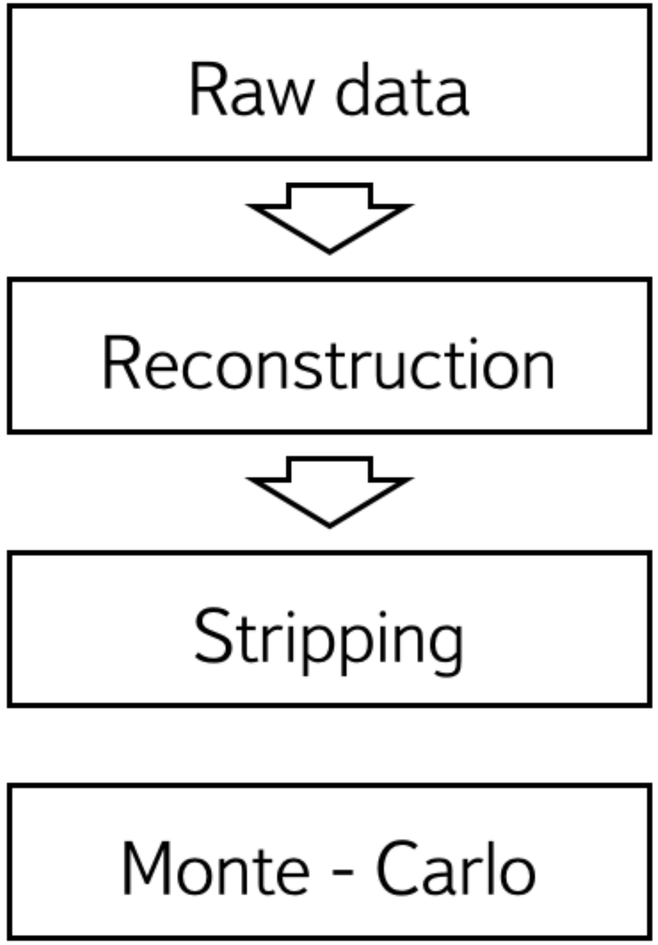
- 3 replicas on T0D1 at a Tier1 and one at CERN.
- 2 archive replicas on T1D0 custodial storage, one at CERN and one at a Tier1.



Monte-Carlo Simulation

〉 2 replicas on T0D1 storage at a Tier1 and one at CERN.

〉 One archive replica on T1D0 custodial storage at CERN or a Tier1.



Problems

1. Static replication
2. Memory pollution

Replication placement strategies

Dynamic replication

- Based on popularity of the data
- Creates replica when when the number of hits for a dataset exceeds the static replication threshold
- Decreases response time
- Helps load balance across the data grid

Adaptive replication

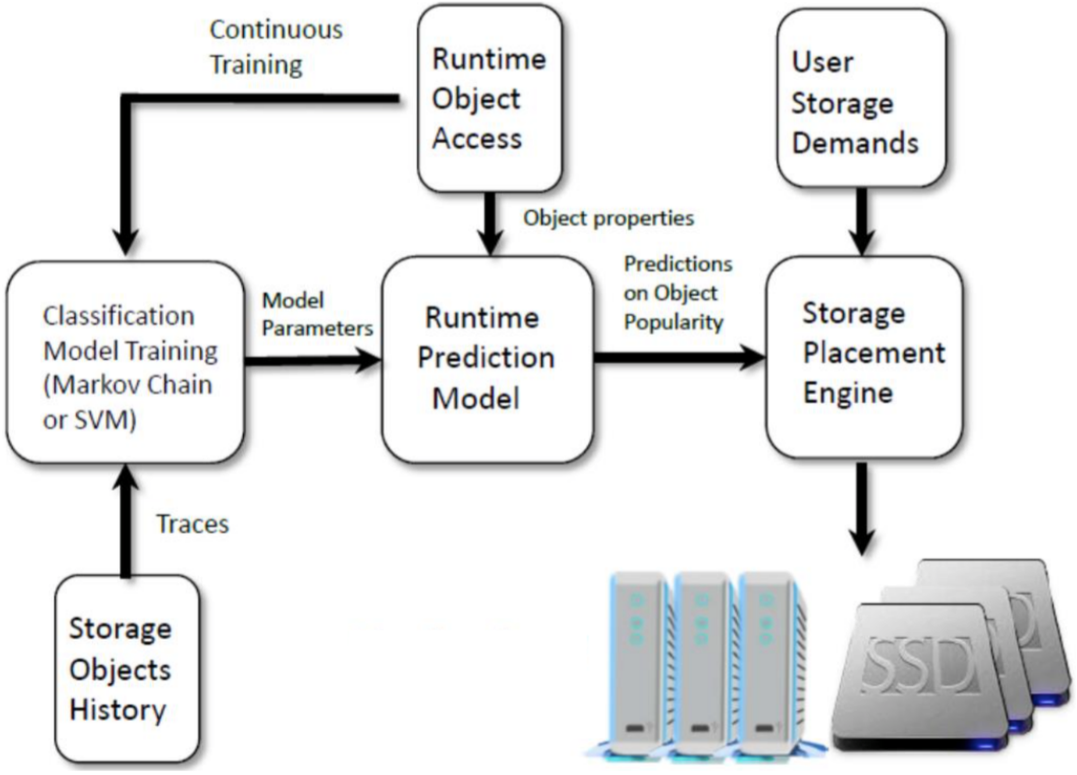
- Similar to dynamic replication
- Based on request arrival rates from clients over a period of time
- Creates replica if the number of requests on average exceeds the previous threshold and shows an upward trend

Improvements

Let’s predict future!

SSD-Optimized Workload Placement with Adaptive Learning and Classification

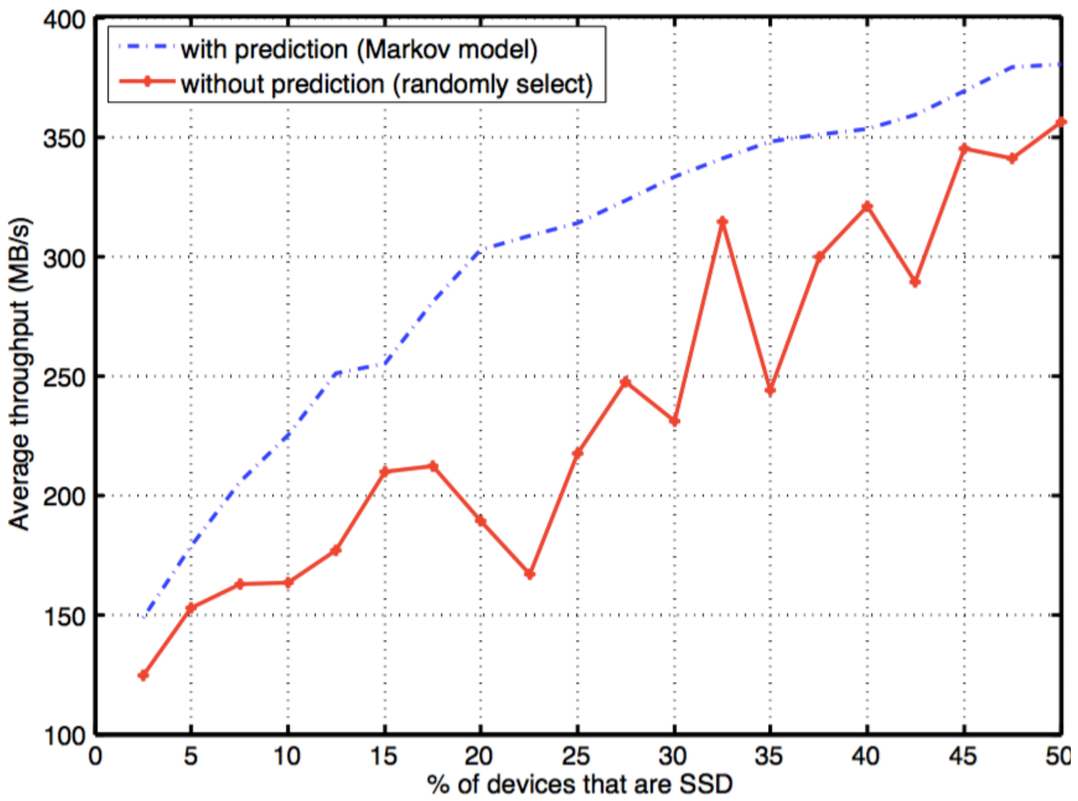
- Presented in 2014 by Lipeng Wan
- For SSD + HDD hybrid systems
- Uses machine learning algorithms to predict the popularity of data object accesses
- Takes into account user storage demands



Ref.: [7]

SSD-Optimized Workload Placement with Adaptive Learning and Classification

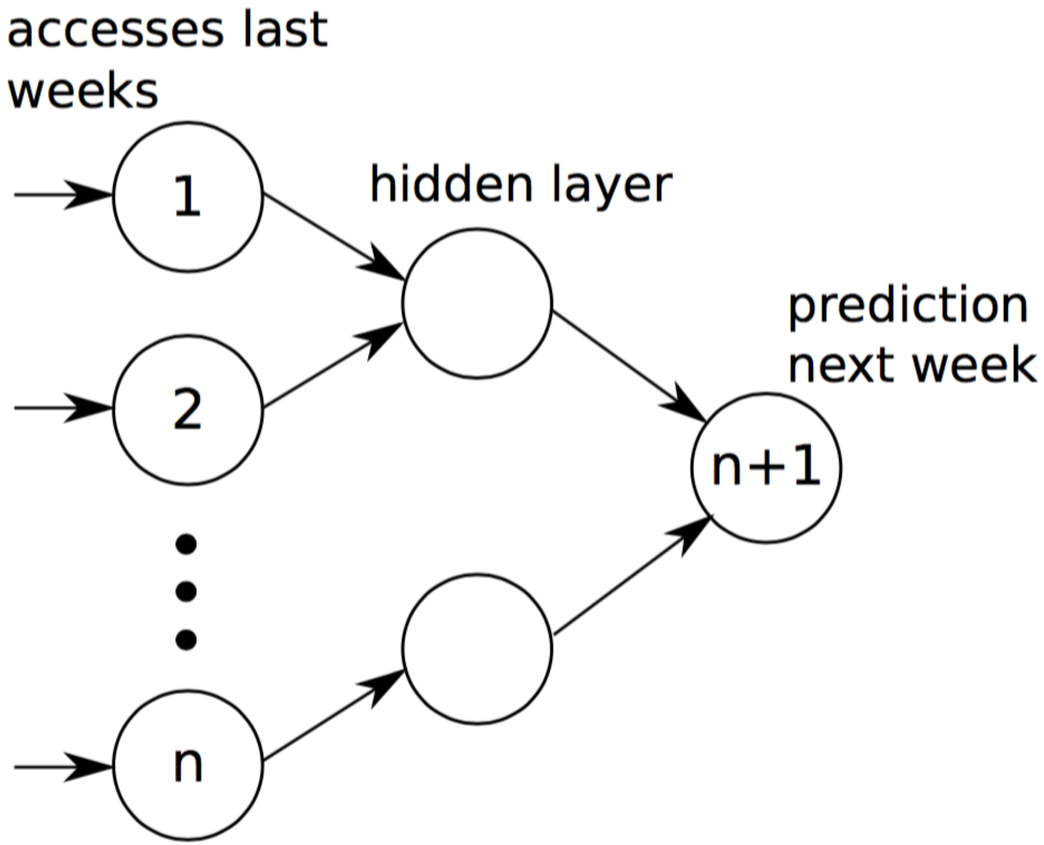
〉 Higher average read throughput than random selection



Ref.: [7]

A Popularity-Based Prediction and Data Redistribution Tool for ATLAS

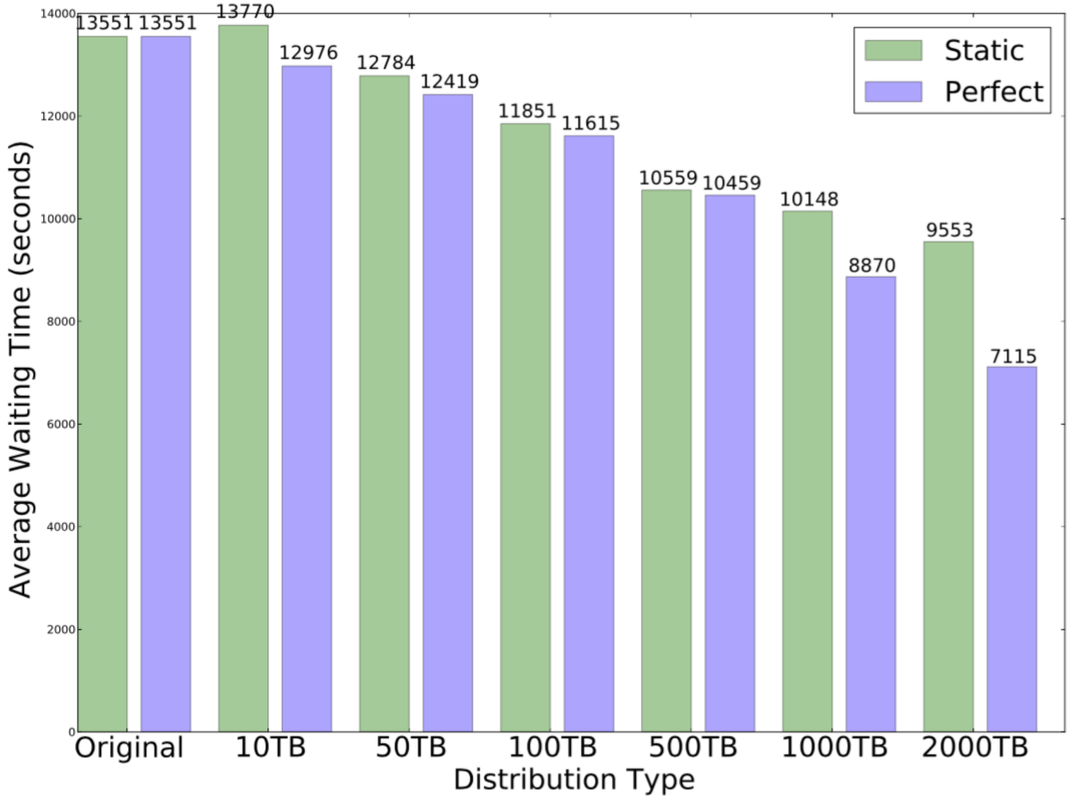
- Introduced in 2014 by Thomas Beermann
- Use Artificial Neural Networks to make forecasts of data popularity
- Removing unpopular replicas
- New replicas for popular datasets



Ref.: [8], [9]

A Popularity-Based Prediction and Data Redistribution Tool for ATLAS

- The more data is moved the bigger is the benefit
- Reduction of the waiting time from 3.75 hours to 2 hours using perfect prediction and a turnover of 2000TB

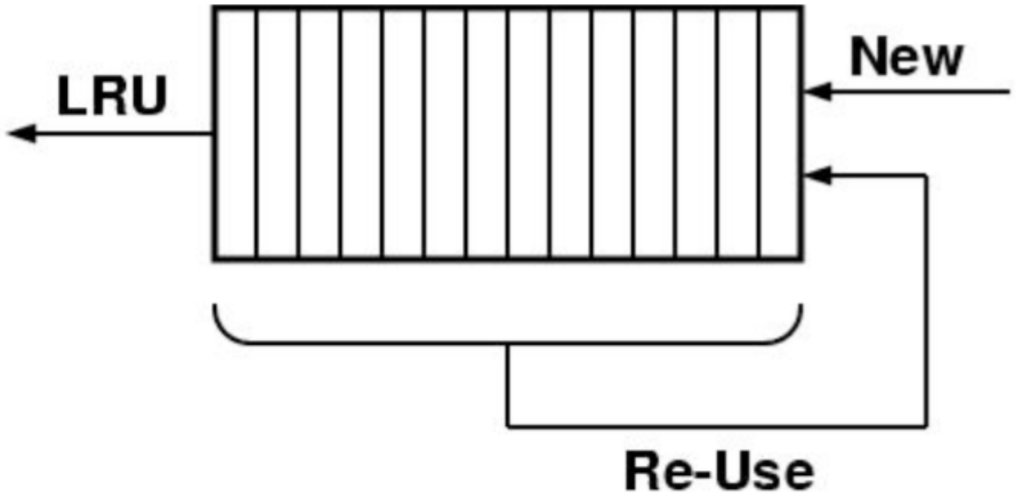


Ref.: [8], [9]

Cache algorithms

Least Recently Used (LRU)

- Described in 1965
- LRU replaces the element in the cache that was least recently used



Ref.: [1]

Other Cache Algorithms

These algorithms employ directly observable criteria:

- recency of access
- frequency of access
- size of the objects
- cost of fetching the objects from secondary memory

criteria	algorithm
–	Random, FIFO, LIFO
time	LRU, MRU, GDS, GDSF, LFUDA, LRV
freq	LFU, MFU, GDSF, LRV, LFUDA
size	SIZE, GDS, GDSF, LRV
retrieval cost	GDS, GDSF, LFUDA, LRV
ID	Hash, Bloom filter
hop-count	–
QoS priority	Stor-serv

Ref.: [3], [4]

Improvements

Let’s predict future!

Web Caching Algorithms

Use machine learning algorithms to:

- Adapt to changing conditions on the Web
- Predict the class of objects that would be re-visited
- Predict the future behavior of a client

Introduced in 1999

Adaptive Web Caching using Logistic Regression

Annie P. Foong, Yu-Hen Hu and Dennis M. Heisey[1]
Department of Electrical and Computer Engineering
University of Wisconsin
1415 Engineering Drive Madison, WI 53706, USA
Phone: +1 608.262.2445
E-mail: foong, hu@ece.wisc.edu, heisey@surgery.wisc.edu

Introduced in 2014

Performance Improvement of Least-Recently-Used Policy in Web Proxy Cache Replacement Using Supervised Machine Learning

Waleed Ali[1]*, Sarina Sulaiman[1], and Norbahiah Ahmad[2]

[1]Soft Computing Research Group, Faculty of Computing,
Universiti Teknologi Malaysia, 81310 Johor, Malaysia
email: waleedalodini@gmail.com, sarina@utm.my, norbahiah@utm.my

Introduced in 2015

Improving the Performance of a Proxy Cache Using Tree Augmented Naive Bayes Classifier

Julian Benadit.P[a,*], Sagayaraj Francis.F[a], Muruganantham.U[b]

[a]*Department of CSE, Pondicherry Engineering College, Pondicherry, 605014, INDIA*
[b]*Department of CSE, Dr.SJSPMCET, Pondicherry University, Pondicherry, 605502, INDIA*

Idea

1. Machine learning algorithms predict probability that an abject will be re-visited based on its access history
2. The objects with the small probabilities are moved closer to the cache queue bottom

Advantages

- Increase hit ratio of a cache
- Reduce cache pollution

Introduced in 2015

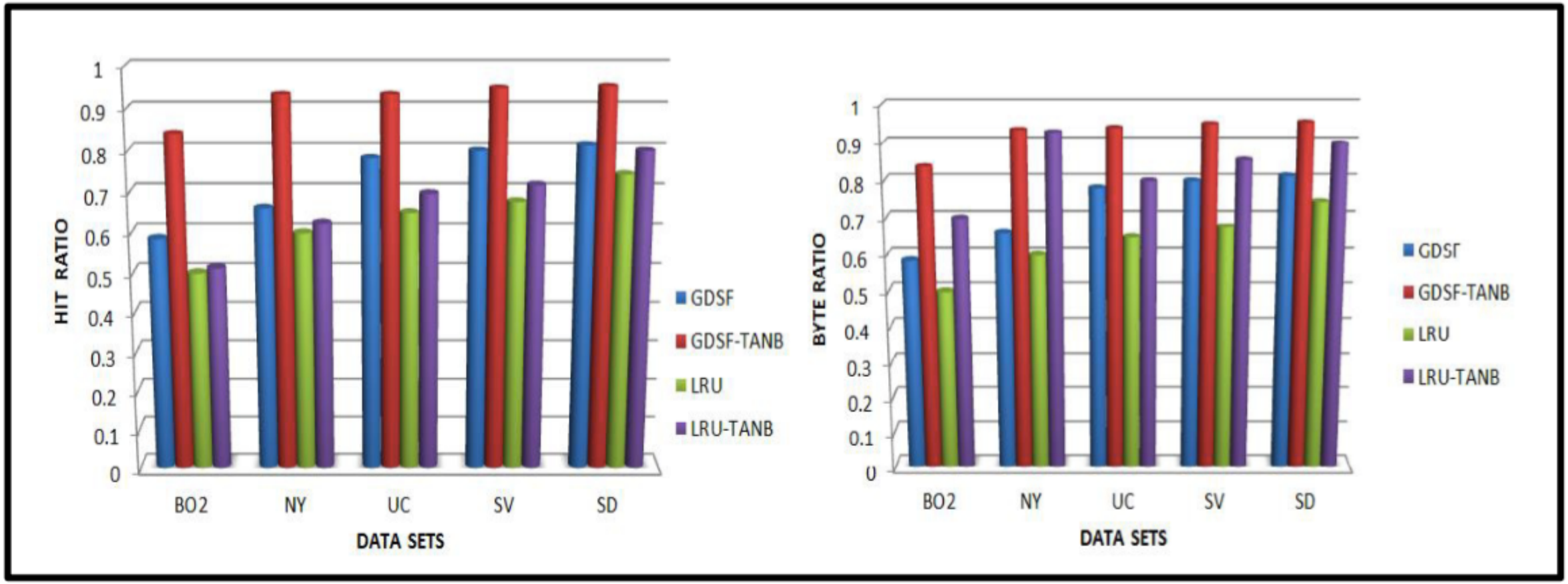
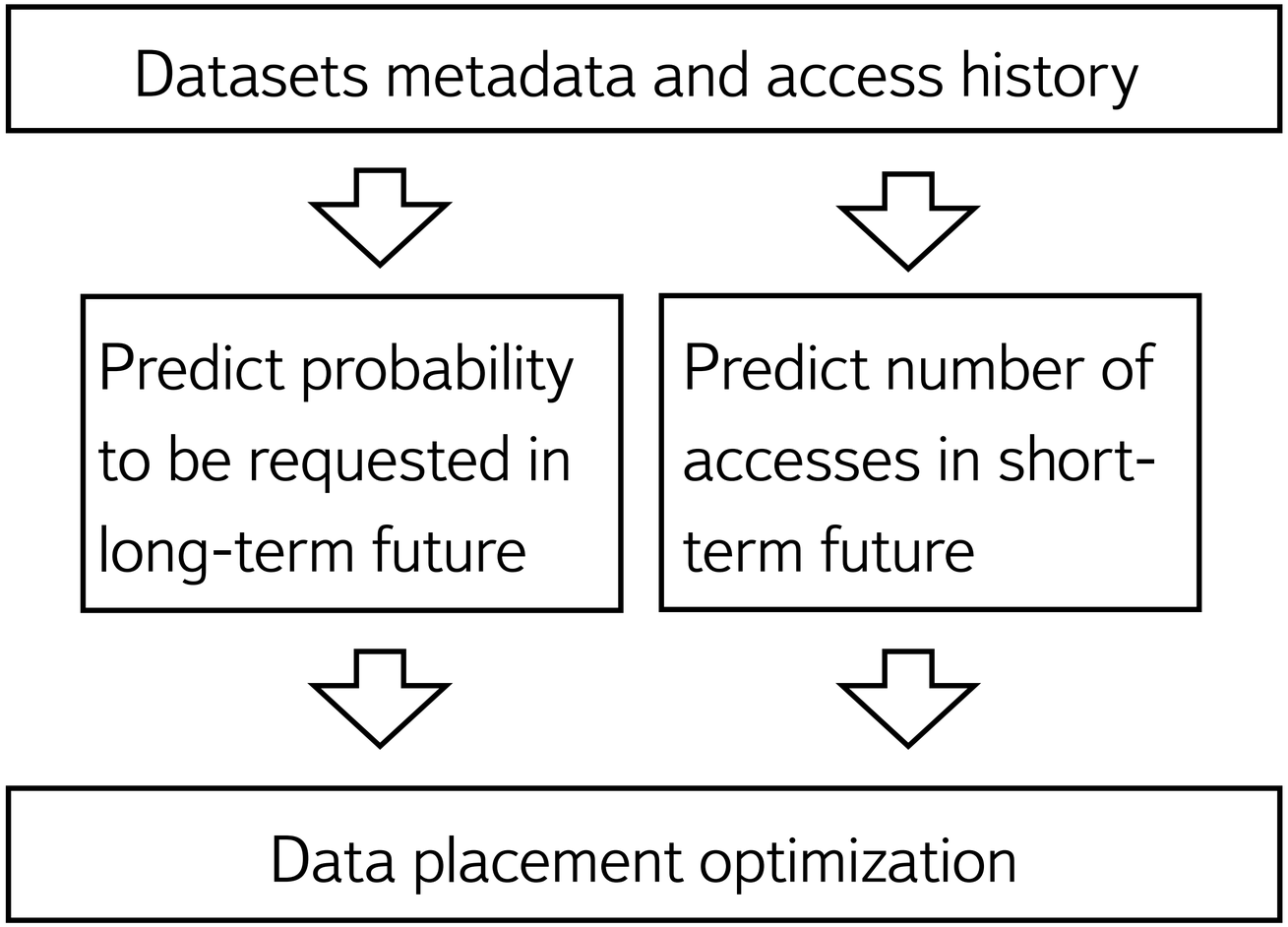


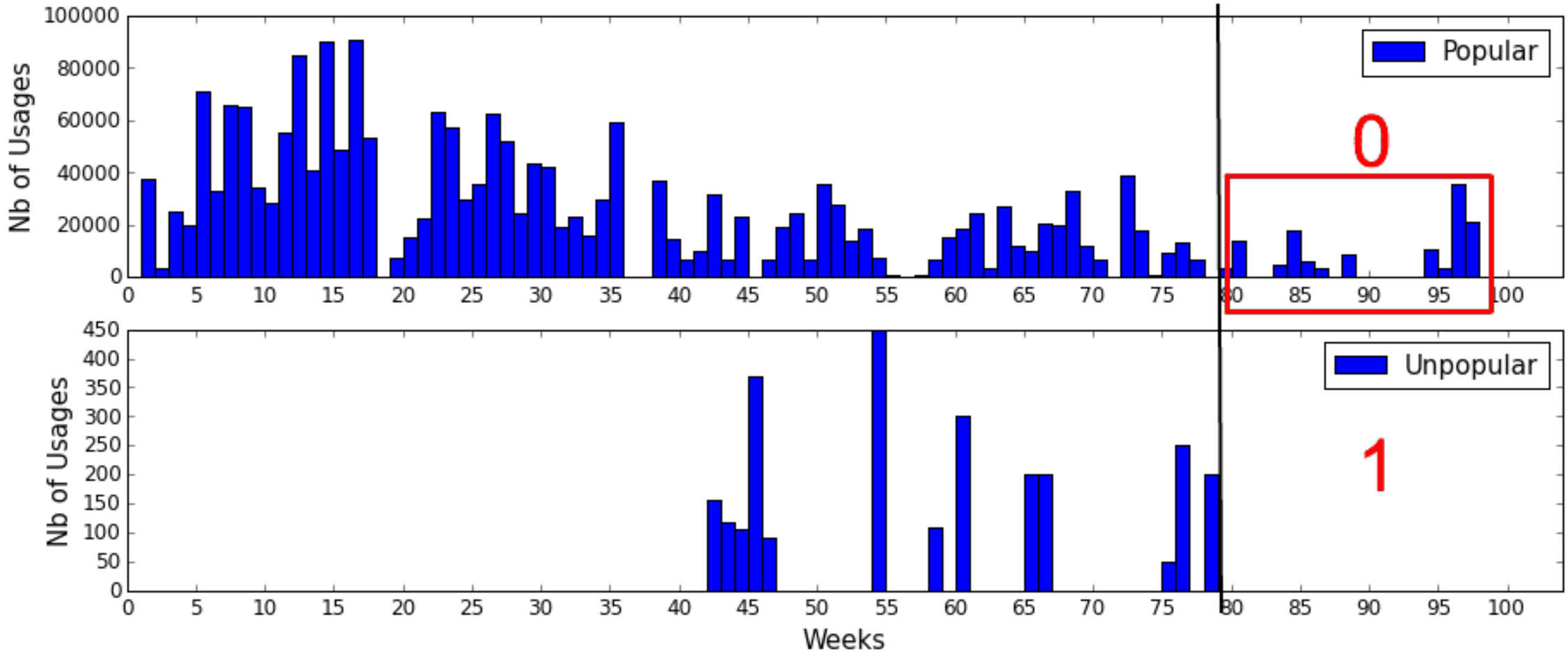
Fig. 4. Cache Hit Ratio and Byte Hit ratio for Different Data Set.

The LHCb data management improvements

Concept


Datasets metadata and access history
Predict probability to be requested in long-term future
Predict number of accesses in short-term future
Data placement optimization

The probability prediction



The probability prediction

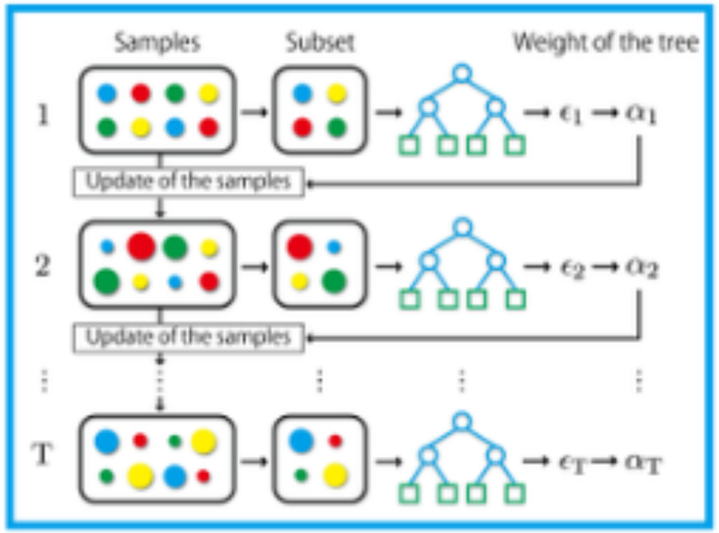
For the datasets description were used the following features:

- Recency of access
- Frequency of access
- Size
- Type
- Creation time and others

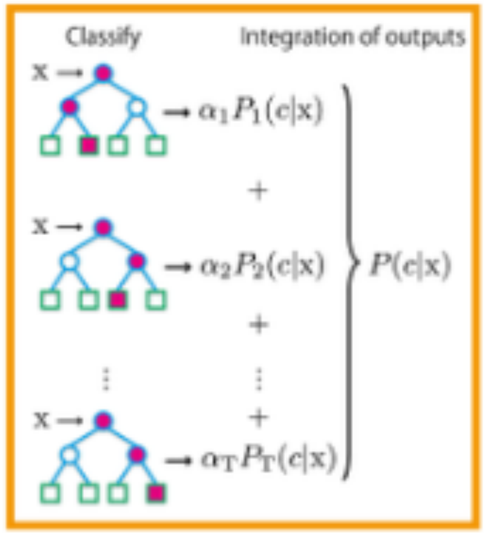
The probability prediction

Access History

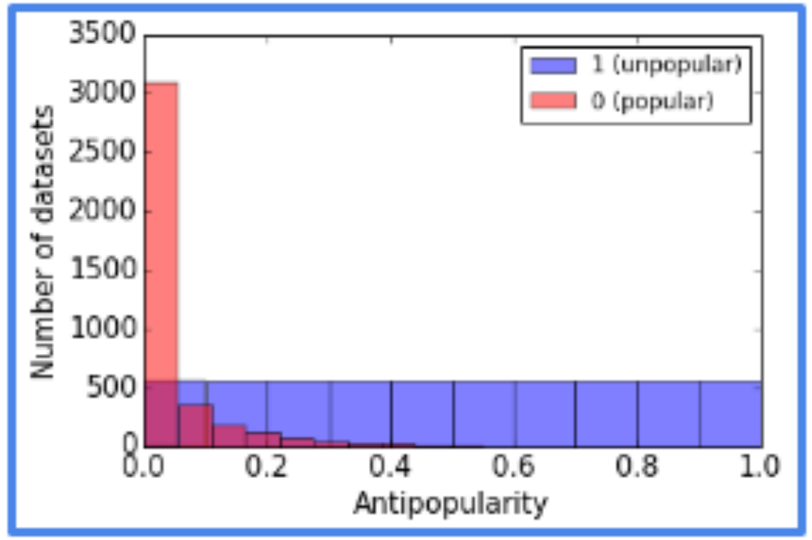
Metadata



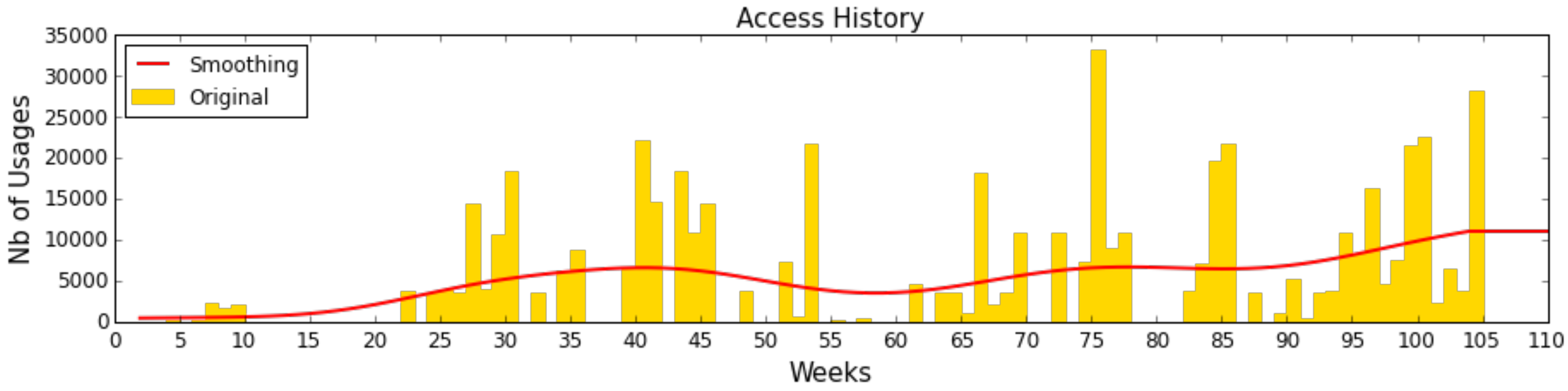
Learning Process



Classify Process



The number of accesses prediction



The data placement optimization

Following loss function was used for data sets placement optimization:

$$L = C_{disk} \sum_{i}^{n} S_i (Rp_i + \alpha \frac{I_i}{Rp_i}) \delta_i + C_{tape} \sum_{i}^{n} S_i (1 - \delta_i) + C_{miss} \sum_{i}^{n} S_i m_i, \quad (4)$$

C_{disk} - cost of storage 1Gb data on disk,
C_{tape} - cost of storage 1Gb data on tape,
C_{miss} - cost of restoring 1Gb data from tape to disk,
α - fine for low number of replicas,
S_i - size of one replica of ith data set,
Rp_i - number of replicas of ith data set,
I_i - predicted usage intensity of ith data set;
δ_i is equal 1 if i^{th} data set on disk, otherwise it is 0;
m_i is equal 1 if i^{th} data set was restored from tape to disk.

Results

Alpha	Download time	Saved space, %	Number of mistakes
0	3,35	71	8
0,001	1,03	57	8
0,005	0,72	40	8
0,01	0,68	34	8
0,05	0,63	11	8
0,1	0,62	1	8

Total number of datasets is 10 000

Base download time is 1

Further development

- Improve quality of the predictions
- Make predictions more interpretable
- Make loss function simplification

The code source:

https://github.com/yandexdataschool/DataPopularity

Data grid simulation

Data grid simulation

Different data placement strategies may work well in different situations or at different times due to:

- variations in workload
- request size
- type of processing
- relative speeds of the different memories
- load on the communication network, etc.

Data grid simulation

- Data grid simulations is needed to test different data placement strategies.
- How do the simulation?
- Any existing solutions?
- Any ideas?

Summary

Advantages of using machine learning for LHCb:

- Data popularity prediction
- Data placement with low number of mistakes (tape or disk)
- Optimal replication
- Reducing waiting time for jobs

Thank you for the attention!

Contacts

Mikhail Hushchyn

mikhail91@yandex-team.ru

Andrey Ustyuzhanin

anaderi@yandex-team.ru

References

[1] Saemundsson T. *An experimental comparison of cache algorithms*, 2013

[2] Sai Huang, Dan Feng, Qingsong Wei, Jianxi Chen, Cheng Chen *Improving Flash-based Disk Cache with Lazy Adaptive Replacement*, 2013, MSST

Avi Press, Michael Pacer, Thomas L. Griffiths, Brian Christian *Caching Algorithms and Rational Models of Memory*

[3] Robert B. Gramacy , Manfred K. Warmuth, Scott A. Brandt, Ismail Ari *Adaptive Caching by Refetching*

[4] Ismail Ari, Ahmed Amer, Robert B. Gramacy *ACME: Adaptive Caching Using Multiple Experts, 2002*

References

[5] Xin Liu, Ashraf Aboulnaga, Kenneth Salem, Xuhui Li *CLIC: CLient-Informed Caching for Storage Servers*

[6] Xin Liu *Optimizing Hierarchical Storage Management For Database System*, 2014

[7] Lipeng Wan, Zheng Lu, Qing Cao, Feiyi Wang, Sarp Oral, Bradley Settlemyer *SSD-Optimized Workload Placement with Adaptive Learning and Classification in HPC Environments*, 2014

References

[8] T Beermann1,2, P Maettig1, G Stewart2, 3, M Lassnig2, V Garonne2, M Barisits2, R Vigne2, C Serfon2, L Goossens2, A Nairz2 and A Molfetas2,4 on behalf of the ATLAS collaboration *Popularity Prediction Tool for ATLAS Distributed Data Management*, 2014

[9] Thomas Beermann, Graeme A. Stewart, Peter Maettig, on behalf of the ATLAS collaboration *A Popularity-Based Prediction and Data Redistribution Tool for ATLAS Distributed Data Management*, 2014

[10] P. Charpentier, M. Hushchyn, A. Ustyuzhanin *Disk storage management for LHCb based on Data Popularity estimator*, 2015, CHEP2015