Cost model study on cache size / network trade off using CMS data

Andrea Sciabà

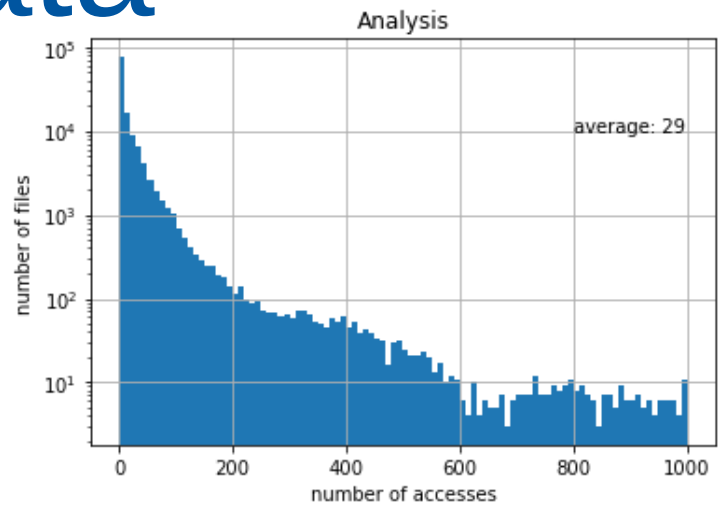
LHCOPN/LHCONE workshop, 14/1/2020

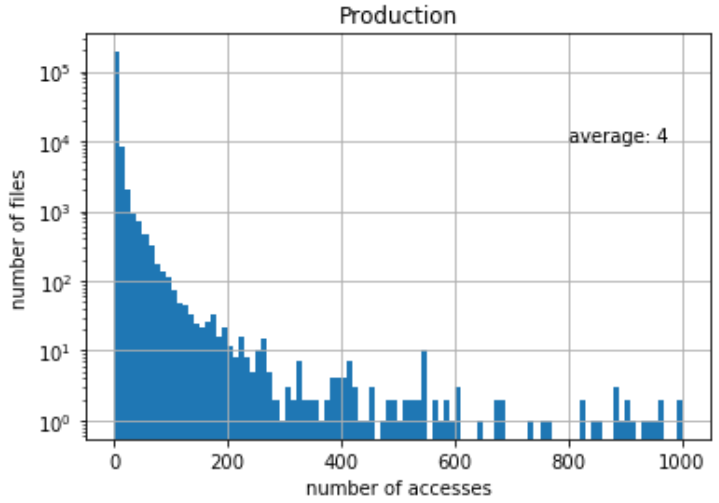
Introduction

- Data access at HL-LHC will need to be extremely optimized to fit within constrained storage (and network) resources
- Assumed to have few (big) centres hosting the data and several (small) centres analysing it, possibly after having cached it locally
- Need to understand what is the “optimal” cache size/retention time for a site
  - Optimal in terms of cost
  - Latency hiding benefits not yet considered, though may have an effect on cost if CPU efficiency is significantly improved
  - Same for saving in operational costs…
- Benefits can be reaped from now, no need to wait for Run4 of course

Using CMS popularity data

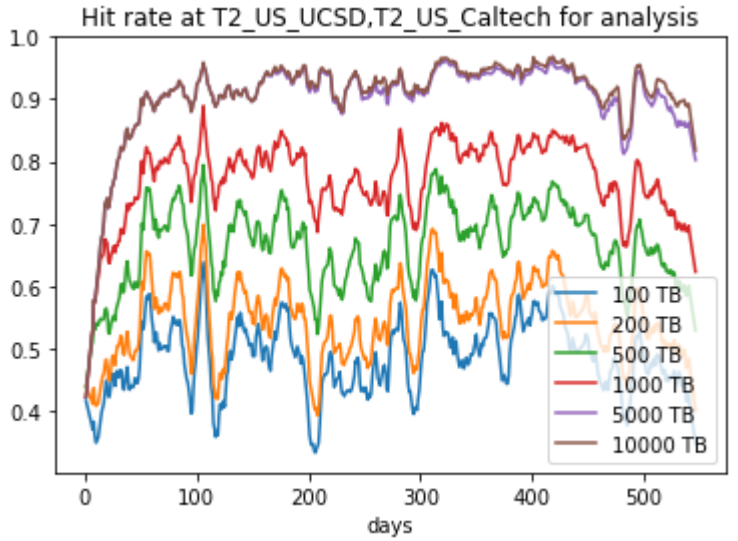
- CMS collects detailed information about files accessed by CMSSW
  - File name, size, location, time, site where the job ran, number of bytes, user, etc.
- Easy and fast to extract data access patterns using a Spark/HDFS cluster at CERN
- Example: number of accesses per file for production jobs is much smaller than for analysis jobs

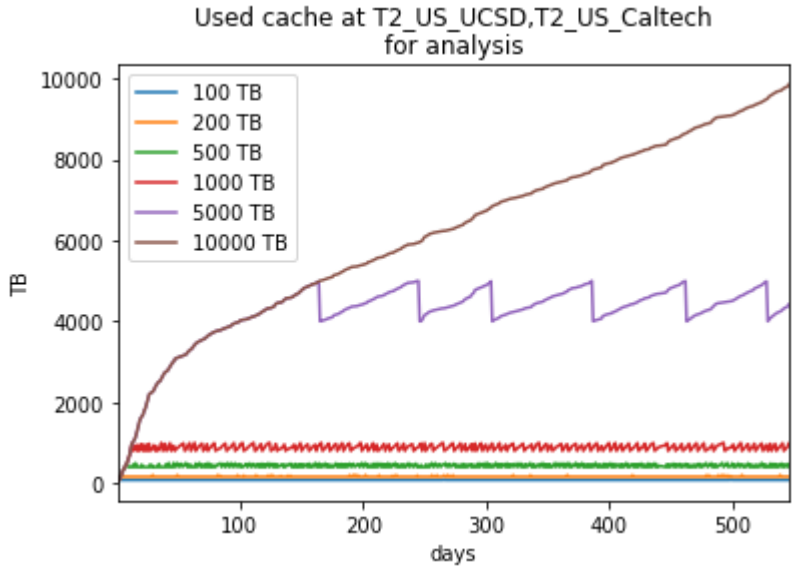




Simulating a site cache

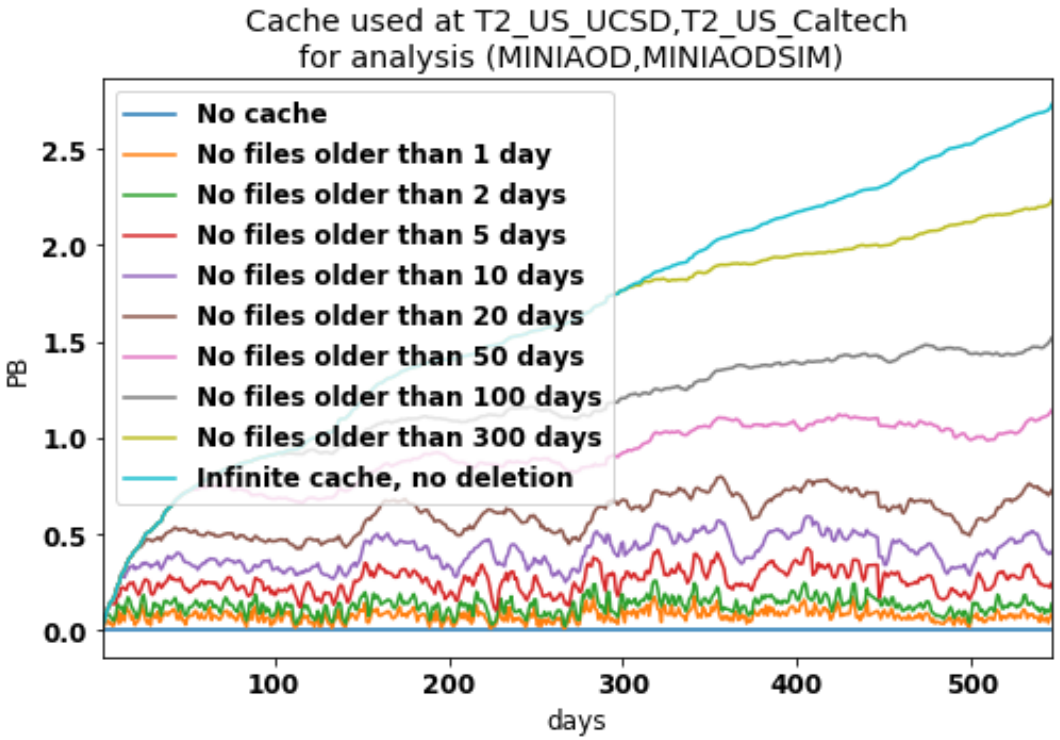
- Only very few sites have a local cache, but we can use real file access information to simulate the impact of a cache, if it existed
  - Example: first time a file is read is a cache miss, subsequent accesses are hits
- Several sites analyzed, but here showing results for just for the CMS Tier-2 SoCal site (UCSD + Caltech)
  - Big T2, it has even an actual production Xcache
- Looking at MINIAOD(SIM) just because it is the most popular format for analysis (similar patterns for AOD(SIM))





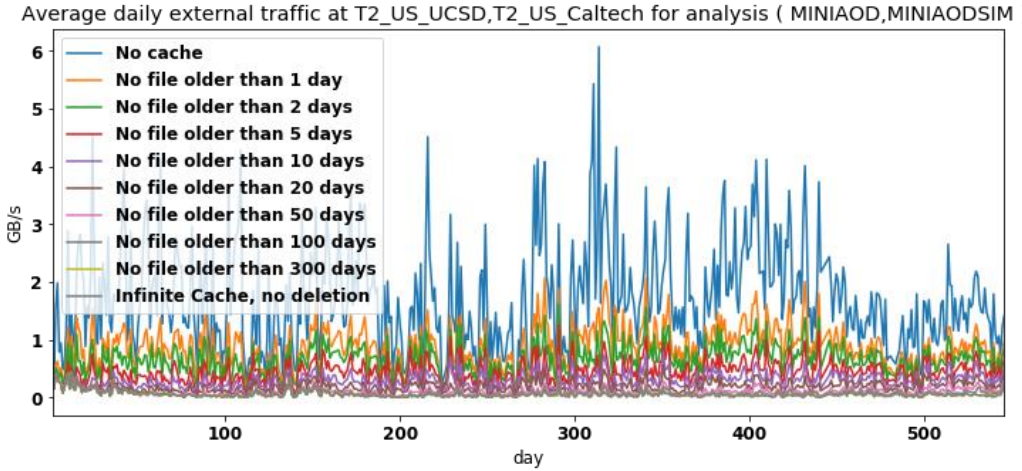
Optimal cache size

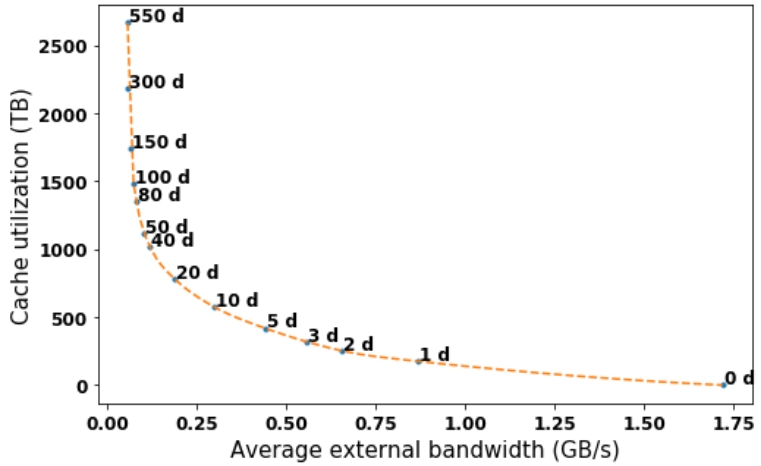
- From a cost perspective:
  - Too large cache → expensive storage
  - Too small cache → too much WAN traffic → need to buy more network bandwidth
- Different cache management strategies:
  - High/low watermarks to free up space, e.g. according to LRU criteria, or
  - Remove files not accessed since more than N days, or
  - More sophisticated strategies (might even use ML…)
  - The first two are mostly equivalent, as a given maximum file age leads to a more or less constant cache occupancy



WAN traffic vs. used cache

- WAN traffic is generated for files not cached
- Cache utilization is generated by files cached…
- There must be an optimal value for the maximum file age that minimizes cost
  - For the given site, type of job, time period and data format
- A cost function can be defined





Cost function

- Total cost = network cost + storage cost
- Storage cost = max(cache occupancy) × cost / unit of disk storage
  - Relatively straightforward, caches have low QoS, so cheap HDDs in JBOD configuration are sufficient
- Network cost = avg(external traffic / time) × cost / unit of bandwidth
  - Much more difficult to estimate, as it is not proportional to usage

Cost estimates

- Disk
  - Cost estimated in the WLCG/HSF cost model working group
    - 1 HDD: 8 TB, 400 EUR, 4 years lifetime
    - Disk server cost / TB ~ twice disk cost
    - ⇒ cache cost ~ 25 EUR/TB/year
  - Baseline HDD scenario: 25 EUR/TB/year
  - Pessimistic HDD scenario: 50 EUR/TB/year
  - SSD scenario: 100 EUR/TB/year
- Network
  - NREN #1: 3.5 Tbps for 20 MEUR/year → 1.4 EUR/TB
  - Provider #2: 20 Gbps for 4000 EUR/month → 0.6 EUR/TB
  - My internet provider: 100 Mbps for 40 EUR/month → 1.3 EUR/TB
  - Baseline: 1 EUR/TB
  - Pessimistic: 10 EUR/TB
- These estimates can be very different at different sites, so take them just as arbitrary but meaningful references

Cost optimization results

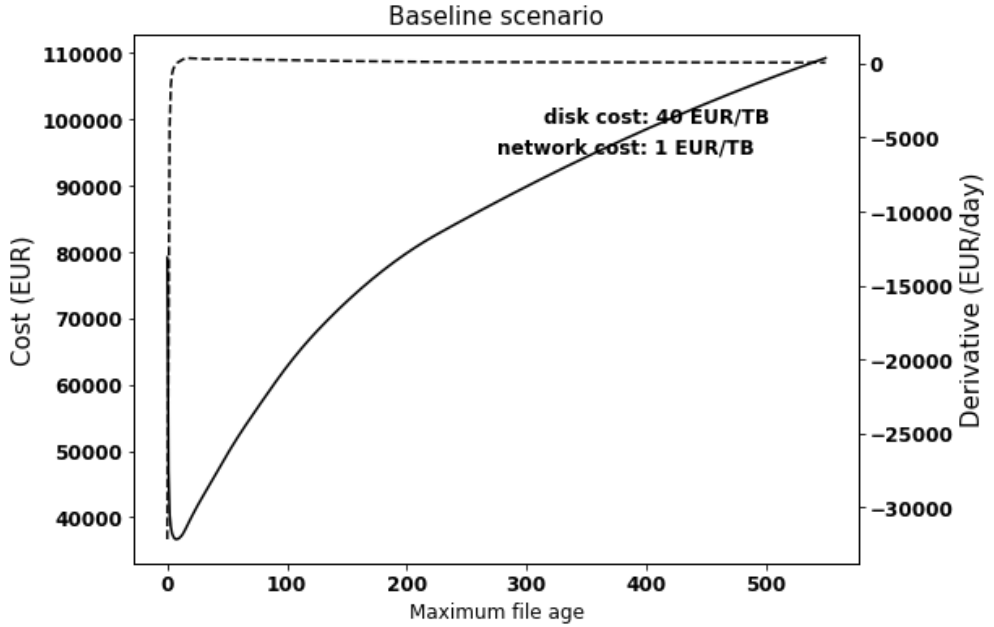
		Disk cost (EUR/TB)		
		40	100	150
Network cost (EUR/TB)	1	8	2	1
	10	70	35	25

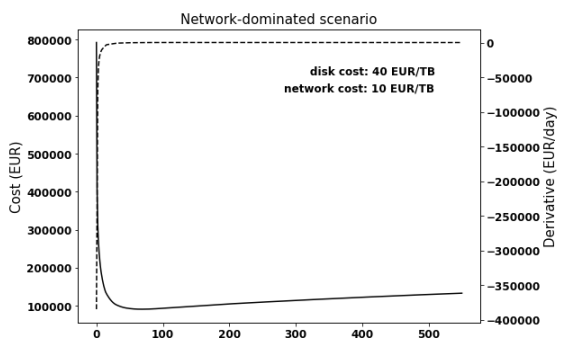
Optimal file age (days)

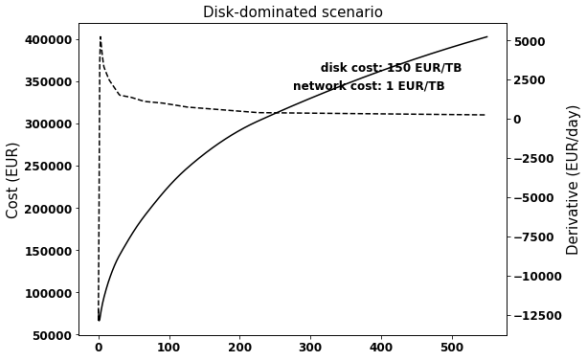
		Disk cost (EUR/TB)		
		40	100	150
Network cost (EUR/TB)	1	520	250	175
	10	1260	980	870

Optimal cache size (TB)

These numbers are for 1.5 years







Things to do

- Estimate cost of CPU inefficiency due to high network latency
  - bandwidth saturation at high data rates
  - very sparse reads, making latency hiding more difficult
  - high miss rates, for sites with very small caches (or none), reading across long distances
- Use a more realistic network cost estimate
  - It is not a linear function of the used bandwidth!
  - For some sites, network is free (but somebody else is paying nonetheless)
- Consider all data and job types
- Compare with actual site cache costs at sites with a cache… e.g. SoCal
- Perform the analysis for other experiments
  - Ongoing for ATLAS

Conclusions

- A simple exercise to show how to choose the best cache size
- The optimal point critically depends
  - on the access patterns and the scale of the dominant workloads
  - on the cost scenarios at the site
- Using real access data, so results become invalid if data access patterns change significantly
- Need to make this kind of estimation easy and available for all sites