



DIRAC Caching demonstrator

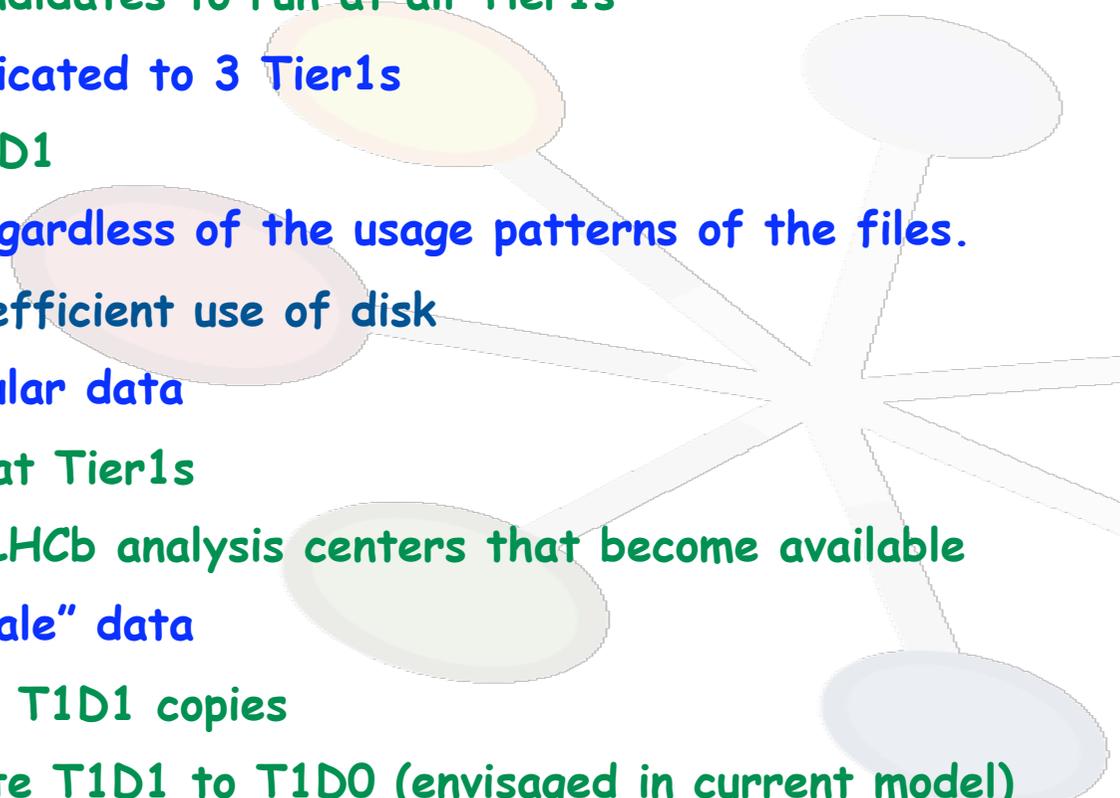
*Ph. Charpentier
for
A.C. Smith (LHCb-DIRAC)*





Current model and goals

- Current model
 - DSTs from real data replicated to 7 Tier1s
 - ☆ 2 T1D1 and 5 TOD1
 - ☆ Analysis jobs candidates to run at all Tier1s
 - DSTs from MC replicated to 3 Tier1s
 - ☆ 2 T1D1 and 1TOD1
 - This is the same regardless of the usage patterns of the files.
- Goal is to make more efficient use of disk
 - More copies of popular data
 - ☆ Up to 7 copies at Tier1s
 - ☆ ...plus any new LHCb analysis centers that become available
 - Fewer copies of "stale" data
 - ☆ Reduce to only 2 T1D1 copies
 - ☆ Eventually migrate T1D1 to T1D0 (envisaged in current model)





Caching and job scheduling

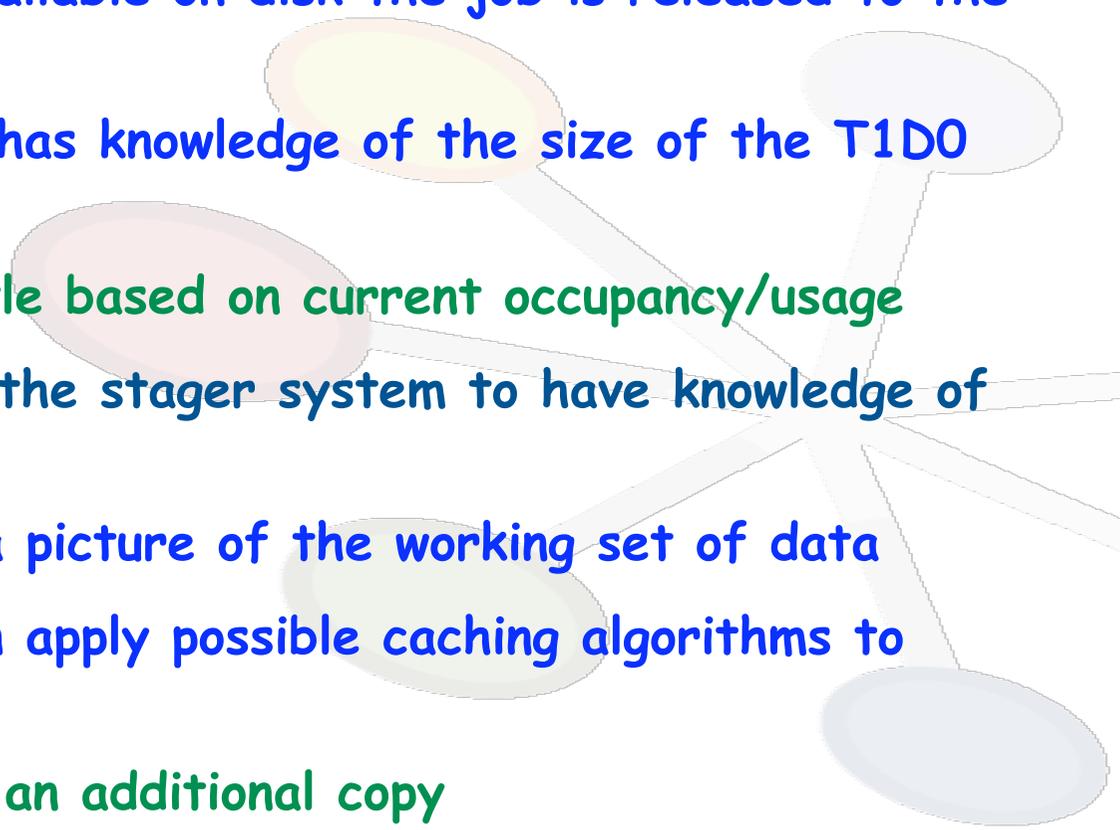
DIRAC CACHING DEMO

- Primary assumption:
 - *The usage of files today gives useful information about the usage of files tomorrow*
- The success of this demonstrator relies on this
 - If you don't hear any more from us you know that this is a fallacy
- We do not want to couple job scheduling to data movement
 - Difficult to perform scheduling with so many variables involved
 - ★ Success in this field is tough (ask WMS guys)
 - Data will not be replicated in anticipation of particular jobs
 - ★ Do not want to waste wall clock waiting for data to be available



Proposed implementation

- DIRAC staging system that manages the T1D0 files for jobs
 - WMS optimisers request stager to stage T1D0 files for jobs
 - Once the data available on disk the job is released to the task queue
 - Stager currently has knowledge of the size of the T1D0 caches
 - ☆ Tries to throttle based on current occupancy/usage
- We want to extend the stager system to have knowledge of T*D* files
 - This will give us a picture of the working set of data
 - From here we can apply possible caching algorithms to determine
 - ☆ When to make an additional copy
 - ☆ When to reduce the number of copies





Metrics

- How to measure success?
 - Our goal is to make popular files more available
- Metrics must include a
 - measure of availability
 - ☆ easy: the number of replicas
 - measure of popularity
 - ☆ not so easy: the number of jobs in a given time period
 - * The time period being the hard thing to decide
- But plain cache hit rate
 - Doesn't make sense since all files will be on disk (cache hit 100%)
- Comparing possible algorithms will be easy
 - With any set of metrics (straight comparison)
- With such a change in model the problem is to compare to what we have now

