The popularity framework has been developed to provide information on the popularity of datasets in ATLAS and to facilitate user monitoring.

A record of every data access event occurring on the grid creates a corresponding tracer entry, submitted to a central front end by a tracer agent. The event information that should be captured by each tracer entry includes the source site, destination site, time stamp of when the operation began, time stamp when the operation ended, the event’s error state, and the application and user initiating the event.

These traces are collected by an Apache front end and inserted into an Oracle database. The structured storage should employ simplistic data structures to store the tracer events, particularly if the data grid has a very high rate of data access events. By simplistic, we mean that the structure should not employ any B-tree or bitmap indexes. The traces should be partitioned in daily increments on storage by the event’s starting time stamp.

One of the limitations of using one level daily aggregation is that the generated summaries have insufficient granularity for queries that looked for short term data access patterns. For example, if one wants to query for the most active users in the last two hours it is not possible to do this by querying the daily summaries, and we instead have to query the tracer collection directly. For this reason we introduce a two step aggregation process, where we generate hourly summaries then long term summaries (for example daily summaries) from the short term summaries.

The popularity framework was first introduced to provide data access information with daily granularity. This system reduces the number of traces by aggregating tracer entries into summaries using the following parameters:

1. Dataset Name
2. Day the data access occurred
3. Other tracer parameters including user, source site, and destination site

This aggregation occurs once a day and is stored into structured storage with corresponding indexes and uniqueness constraint insuring no duplicate entries for a given set of tracer parameters exist for the same period.

Since the tracer has been in operation the number of tracer entries collected by the system has linearly increased from 20 million traces a month to over 140 million traces a month.

The tracer system is receiving approximately a monthly average of 50 to 60 trace events per second. Furthermore, spikes in data access may mean that this rate can increase to more than 1200 trace events per second.

As the tracer system employs a simple data structure, with no B-Tree and bitmapped indexes, it has been able to cope with this high levels of tracer insertions.