Available on the CMS information server **CMS IN 2014/XXX**

DRAFT
CMS Internal Note

Popularity Metrics of Dynamically-Managed Datasets

S. Narayanan, M. Goncharov, and C. Paus
Massachusetts Institute of Technology

Abstract

CMS data is ordered in datasets, which have some common properties and are usually analyzed together. As data taking progresses and Monte Carlo tools improve, datasets are often replaced and site administrators have to identify and delete outdated datasets. Dynamic Data Management is a novel method to automatically manage the distribution and deletion of datasets. In this note, we describe a metric of DDM performance, based on the number of user requests per dataset replica.

This box is only visible in draft mode. Please make sure the values below make sense.

PDFAuthor: S. Narayanan, M. Goncharov, C. Paus
PDFTitle: Popularity Metrics of Dynamically-Managed Datasets
PDFSubject: CMS
PDFKeywords: CMS, Dynamic Data Management, popularity

Please also verify that the abstract does not use any user defined symbols

Contents

1 Introduction 1
2 Dataset attributes 1
2.1 Computing average N_{replicas} 1
2.2 Computing N_{accesses}, size, and N_{files} 2
3 Filling the plot 2
3.1 Dataset selection 2
3.2 Binning 2

1 Introduction

Dynamic Data Management currently manages a pool of approximately 20 PB across several Tier 2 and Tier 1 sites. If a dataset is very popular, it will be replicated at several sites. On the other hand, we delete replicas of unpopular datasets, as long as it is not the last copy. If a dataset is declared deprecated, all copies will be deleted. A good measure of the performance of this algorithm is the number of accesses per replica. If, for a given dataset, this number is very large, then the dataset is not sufficiently replicated. On the other hand, if it is very small for many datasets, then we are maintaining too many copies of unused datasets. To make popularity plots as shown in Figure 1, four attributes are computed for each dataset: number of accesses, size on disk, number of files, and average number of replicas.

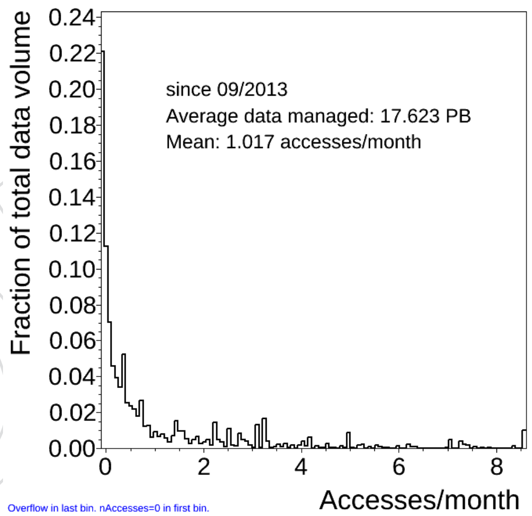

Figure 1: Dataset usage plot for the time interval $[09/2013, \text{present}]$. Only datasets in AnalysisOps are included.

2 Dataset attributes

2.1 Computing average N_{replicas}

Phedex directly provides us with the current locations of all datasets. However, this information is not directly available for the past. Thus, Phedex transfer and deletion histories are used to infer the presence of a dataset on a site. The histories are 'sanitized' to remove self-inconsistent entries such as the transfer of a dataset to a site on which it already exists (it is assumed that each site can only contain one copy of a dataset). If there is no Phedex history for a given dataset on a given site, but we know that the dataset is currently on that site, then it is assumed to have existed since its creation time (which is determined using DAS).

Having collected this information, $\langle N_{\text{replicas}} \rangle$ can be computed for a given time interval $[t_0, t_1]$. Then, summing over the sites:

$$\langle N_{\text{replicas}} \rangle = \sum_{S \in \text{sites}} \frac{\text{time on } S \text{ during } [t_0, t_1]}{t_1 - t_0} \tag{1}$$

This gives the average number of replicas of a dataset in a specific time interval.

2.2 Computing N_{accesses}, size, and N_{files}

The remaining variables are relatively easily calculated. The number of files and size of a dataset are gotten from DAS. It should be noted that we assume each replica is a "full" replica. An incomplete replica may be missing files and consequently a smaller size. We compute N_{accesses} using the caches maintained by Detox. Detox is the Dynamic Data Management tool which deals with the deletion of deprecated datasets and under-utilized replicas. In order to make the latter decision, Detox keeps a local record of the number of accesses made to each replica of each dataset, derived from the popularity API. N_{accesses} for a dataset is defined as the total number of accesses over all replicas of that dataset.

3 Filling the plot

3.1 Dataset selection

Currently, we consider all datasets that are currently on, or have been on, Tier 2 sites (Tier 1s are excluded). This information is gathered from two places. First, we ask Phedex for all datasets which are currently on Tier 2s. Then, we use the `deleterequests` Phedex API to determine all datasets which have been deleted during the relevant time interval. We consider all datasets in the union of these two sets. Finally, we double-check using the Phedex transfer/deletion histories that it actually was on at least one site during the interval. If the dataset passes this check, a corresponding entry is made in the histogram, as discussed below.

3.2 Binning

Having computed these variables for each dataset, the popularity plot may be made. The histogram is filled for each dataset by choosing the following bin-value:

$$\frac{N_{\text{accesses}}}{N_{\text{files}} \cdot \langle N_{\text{replicas}} \rangle} \tag{2}$$

The factor of N_{files} in the denominator is due to the fact that a single request to a dataset actually consists of a series of requests to each file in the dataset. Dividing by N_{files} ensures that this quantity is the same for small and large datasets. The entry is given weight:

$$\langle N_{\text{replicas}} \rangle \cdot \text{size} \tag{3}$$

For ease of comparing plots made under different conditions, the bin-value is normalized to the length of the time interval (in Figure 1, the unit of time is months). Finally, the plot is normalized to have an integral of unity. The un-normalized integral can be thought of as a measure of "average data volume" during the interval, since it can be computed as:

$$\sum_{\text{datasets}} \langle N_{\text{replicas}} \rangle \cdot \text{size} \tag{4}$$

Finally, it should be noted that there are two special bins in Figure 1. The very last bin is the overflow bin. The very first bin contains those entries for which $N_{\mathrm{accesses}} = 0$ exactly. It is important to recall that the histogram is only filled with datasets that were on at least one site during the relevant interval. Thus, the very first bin shows the fractional volume of datasets which were on disk, but not accessed, during an interval.

Acknowledgements

We congratulate our colleagues in the CERN accelerator departments for the excellent performance of the LHC and thank the technical and administrative staffs at CERN and at other CMS institutes for their contributions to the success of the CMS effort. In addition, we gratefully acknowledge the computing centres and personnel of the Worldwide LHC Computing Grid for delivering so effectively the computing infrastructure essential to our analyses. Finally, we acknowledge the enduring support for the construction and operation of the LHC and the CMS detector provided by the following funding agencies: BMWFW and FWF (Austria); FNRS and FWO (Belgium); CNPq, CAPES, FAPERJ, and FAPESP (Brazil); MES (Bulgaria); CERN; CAS, MoST, and NSFC (China); COLCIENCIAS (Colombia); MSES and CSF (Croatia); RPF (Cyprus); MoER, ERC IUT and ERDF (Estonia); Academy of Finland, MEC, and HIP (Finland); CEA and CNRS/IN2P3 (France); BMBF, DFG, and HGF (Germany); GSRT (Greece); OTKA and NIH (Hungary); DAE and DST (India); IPM (Iran); SFI (Ireland); INFN (Italy); NRF and WCU (Republic of Korea); LAS (Lithuania); MOE and UM (Malaysia); CINVESTAV, CONACYT, SEP, and UASLP-FAI (Mexico); MBIE (New Zealand); PAEC (Pakistan); MSHE and NSC (Poland); FCT (Portugal); JINR (Dubna); MON, RosAtom, RAS and RFBR (Russia); MESTD (Serbia); SEIDI and CPAN (Spain); Swiss Funding Agencies (Switzerland); MST (Taipei); ThEPCenter, IPST, STAR and NSTDA (Thailand); TUBITAK and TAEK (Turkey); NASU and SFFR (Ukraine); STFC (United Kingdom); DOE and NSF (USA).

References