PD2P : PanDA Dynamic Data Placement for ATLAS

T. Maeno [1], K. De [2], S. Panitkin [1], for the ATLAS Collaboration

[1] Brookhaven National Laboratory, NY, USA
[2] University of Texas at Arlington, TX, USA

tmaeno@bnl.gov

Abstract. The PanDA (Production and Distributed Analysis) system plays a key role in the ATLAS distributed computing infrastructure. PanDA is the ATLAS workload management system for processing all Monte-Carlo (MC) simulation and data reprocessing jobs in addition to user and group analysis jobs. The PanDA Dynamic Data Placement (PD2P) system has been developed to cope with difficulties of data placement for ATLAS. We will describe the design of the new system, its performance during the past year of data taking, dramatic improvements it has brought about in the efficient use of storage and processing resources, average wait time for user analysis jobs, and plans for the future.

1. Introduction

The PanDA Production and Distributed Analysis System is the ATLAS workload management system for processing user analysis, group analysis and production jobs. In 2011 more than 1400 users have submitted jobs through PanDA to the ATLAS grid infrastructure. The system processes more than 2 million analysis jobs per week. Analysis jobs are routed to sites based on the availability of relevant data and processing resources, taking account of the non-uniform distribution of CPU and storage resources in the ATLAS grid. The data distribution has to be optimized to fit the resource distribution, and also has to be dynamically changed to meet rapidly evolving requirements for analysis use cases. The PanDA Dynamic Data Placement (PD2P) system has been developed to cope with difficulties of data placement for ATLAS. PD2P is an intelligent subsystem of PanDA to distribute data by taking the following factors into account: popularity, locality, the usage pattern of the data, the distribution of CPU and storage resources, network topology between sites, site operation downtime and reliability, and so on. We will describe the design of the new system, its performance during the past year of data taking, dramatic improvements it has brought about in the efficient use of storage and processing resources, average wait time for user analysis jobs, and plans for the future.

2. Overview of The PanDA System and Analysis Workflow

Figure 1 shows a schematic view of the PanDA system [1]. Jobs are submitted to the PanDA server. The PanDA server is the main component which provides a task queue managing all job information centrally. The PanDA server receives jobs into the task queue, upon which a brokerage module operates to prioritize and assign work on the basis of job type, priority, software availability, input data and its locality, and available CPU resources. The autopyfactory pre-schedules pilots to OSG and EGEE/EGI grid sites using Condor-G [2]. Pilots retrieve jobs from the PanDA server in order to run the jobs as soon as CPU slots become available. Pilots use resources efficiently; they exit immediately

if no job is available and the submission rate is regulated according to workload. Each pilot executes a job on a worker node (WN), detects zombie processes, reports job status to the PanDA server, and recovers failed jobs. Ref.[3] describes the details on pilots. For NDGF, the ARC control tower [4] retrieves jobs from the PanDA server and sends the jobs together with pilot wrappers to NDGF sites using ARC middle-ware.

Each end-user submits a user task (job set) that is split to multiple job subsets according to localities of input datasets, job statistics, and available CPU resources at sites. A dataset is a collection of files and the ATLAS Distributed Data Management (DDM) system [5] replicates datasets. A job subset is sent to a site where input datasets are available, i.e., if input datasets are distributed over multiple sites there will be multiple job subsets and they will be sent to multiple sites. Each job subset is composed of many jobs. One important constraint in the ATLAS computing model is that analysis jobs read or transfer input files from the local storage element at each site, i.e., analysis jobs themselves don't read or transfer files over the wide area network. This is mainly because analysis jobs are typically I/O intensive and run on many files. Therefore, input datasets are pre-placed at sites before analysis jobs get started. As a consequence, job distribution is tightly affected by data distribution.

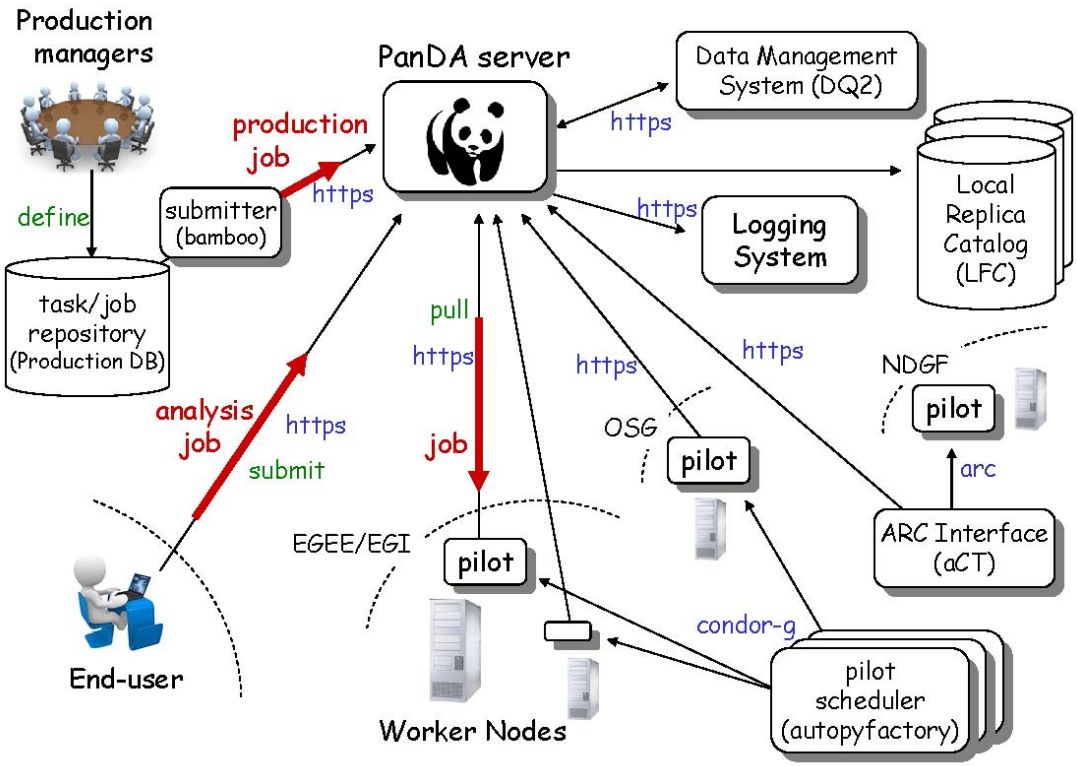

Figure 1. Schematic view of the PanDA System

3. Issues and Goals

In the ATLAS computing model, analysis jobs can run at Tier-0, Tier-1 or Tier-2 site. There are about 100 sites available to users. The amount of CPU resources and storage capacities are not uniform among sites. ATLAS is producing huge amounts of data and thus no site can hold all data. Each user has their own analysis requirements which are being changed rapidly for various analysis use cases. Because job distribution is tightly correlated to data distribution, data distribution has to be dynamically optimized to fit the non-uniform CPU/storage resource distribution and to meet rapidly

evolving user requirements. ATLAS had used a policy-based data distribution model where two copies of datasets were pushed to Tier-2 sites as soon as they were produced. The model had been working well but had suffered some drawbacks. Tier-2 sites had been filling up too rapidly, most datasets copied to Tier-2 sites were never used, and job distribution was unbalanced. The root cause was that user patterns of data usage were unpredictable and changeable.

PD2P has been developed to overcome those problems. The main goals are as follows: Dataset replications are triggered by user requests to run jobs. Popular datasets have many replicas, i.e. the number of dataset replicas is defined based on usage. Users should not experience delay due to data movement. Also any changes should not affect on-going critical analysis activities.

4. Implementation

PD2P has two algorithms, one is for replications at Tier-1 sites and the other is for Tier-2 sites. The intention is that Tier-1 sites are used as data repository while Tier-2 sites are used more for analysis job execution.

4.1. General Policies

The following general policies apply to both algorithms:

- PD2P considers only official datasets.
  Although users can submit jobs with private datasets those datasets are not replicated by PD2P.
- Replication policies for data types are defined by the ATLAS computing model.
- Copies are made at sites where enough disk space is available.
- PD2P is triggered only when end-users submit jobs to sites that are dedicated to analysis activities.
  Other types of jobs such as production or site testing jobs are ignored.
- Only online sites and dataset replicas at online sites are used. Also each Tier-2 site can be configured as to whether or not PD2P is used.
  For example, once HammerCloud [6] or Site Status Board [7] blacklists sites based on site status and/or downtime, PD2P doesn't use those sites or replicas at those sites.

4.2. Tier 1 Algorithm

Primary copies of ATLAS data are placed at Tier-1 sites based on the Memorandum of Understanding (MoU) share which specifies the contributions expected from the corresponding region. PD2P makes secondary copies at Tier-1 sites when the following conditions are satisfied for each input dataset:

- PD2P didn't make a replica of the dataset during the past week to a Tier-1 site.
- The total number of job sets which used the dataset is 10 to the power of X, where X is an integer larger than 0.
- The number of dataset replicas at Tier-1 sites is less than int($\log_{10}(N_{used})$), where N_{used} stands for how many times the dataset was used per job set.

One Tier-1 site is selected based on MoU share for each dataset and a replication request is sent to DDM. When a copy is made at a Tier-1 site, another copy is made at a Tier-2 site at the same time based on MoU share. The idea is to have popular datasets not only at Tier1 sites but also at Tier2 sites.

4.3. Tier 2 Algorithm

The Tier-2 algorithm is executed independently of the Tier-1 algorithm. The following parameters are used; N_r is the number of the dataset replicas at Tier-2 sites, N_J is the number of jobs waiting in the queue to use the input dataset, and N_S is the number of job sets that use the input dataset and have at

least one waiting job in the queue. The Tier-2 algorithm makes two additional copies at Tier-2 sites when the following conditions are satisfied for each input dataset:

- There is no replica within Tier-2 sites, or not enough replicas are available while many jobs are waiting, i.e. $N_r = 0$, or $N_S > 2$ and $N_r < \mathrm{int}(\log_{10}(N_J/200))$.
- N_r is less than 5.
- No more than two copies are concurrently being replicated.

One Tier-2 site is selected based on MoU share. The other Tier-2 site is selected by using

- $List_d$: a list of Tier-1 and Tier-2 sites where the input dataset is already available
- $List_c$: a list of Tier-2 sites that have fast FTS channels to one of the sites in $List_d$
- W : the weight per site which is calculated using the number of active worker nodes at the site, site reliability, numbers of running and queued jobs at the site, and the number of replicas made by PD2P for the last 24 hours

The weight W is calculated for each site in $List_c$ and the Tier-2 with the largest W is used. The site reliability is defined every 1 month based on job and data transfer failure rates. The idea of making two copies is to have one copy quickly available at a reliable site while distributing another copy, even if slowly, by following MoU share.

4.4. Rebrokerage

PD2P relies on future reuse of data for its effectiveness. The data copy triggered by the initial job is not used unless subsequent jobs reuse it. Also, the initial job remains at the original site although a new copy was replicated at free sites by PD2P. In order to increase reuse of PD2P replicas, the rebrokerage mechanism has been implemented to periodically reassign jobs to other sites if they are waiting in the queue for a while.

4.5. Deletion

DDM takes care of data deletion. Once access to a dataset replica has decreased to zero the replica gets deleted. Ref [6] describes the DDM deletion service in detail.

5. Results

Figure 2 shows cumulative evolution of disk usage in US Tier-2 sites. We can see an exponential rise of the disk space utilization in April and May 2010 after LHC startup, with a much slower rise since PD2P was deployed in June 2010 even though luminosity was growing rapidly. PD2P represents a large improvement in terms of disk usage efficiency and manageability over the policy-based data distribution model. Data placement policy has since evolved, as a result of extensive optimization, to a hybrid combination of PD2P based automated replication and limited policy-based distribution of data known to be most popular, in order to rapidly engage Tier-2 sites fully in the analysis of new data.

In 2012, PD2P made 9k dataset copies at Tier-1 sites while making 72k copies at Tier-2 sites. Figure 3 shows how PD2P made dataset copies in 2012 at Tier-1 sites (left chart) and at Tier-2 sites (right chart). The distribution at Tier-1 sites was roughly proportional to MoU share, as expected. The distribution at Tier-2 sites was well balanced, which means PD2P used Tier-2 resources broadly.

Figure 4 shows how many times dataset copies were reused after PD2P replications. 45% of datasets were never reused while others were well reused. This implies that users' interest is largely unpredictable and volatile and the request-based model of PD2P suits user behaviour.

Figure 5 shows average wait time and execution time for analysis jobs in 2011. Average wait time was almost constant in 2011 while average execution time grew up to 90 min. More detailed discussion is available in Ref [8].

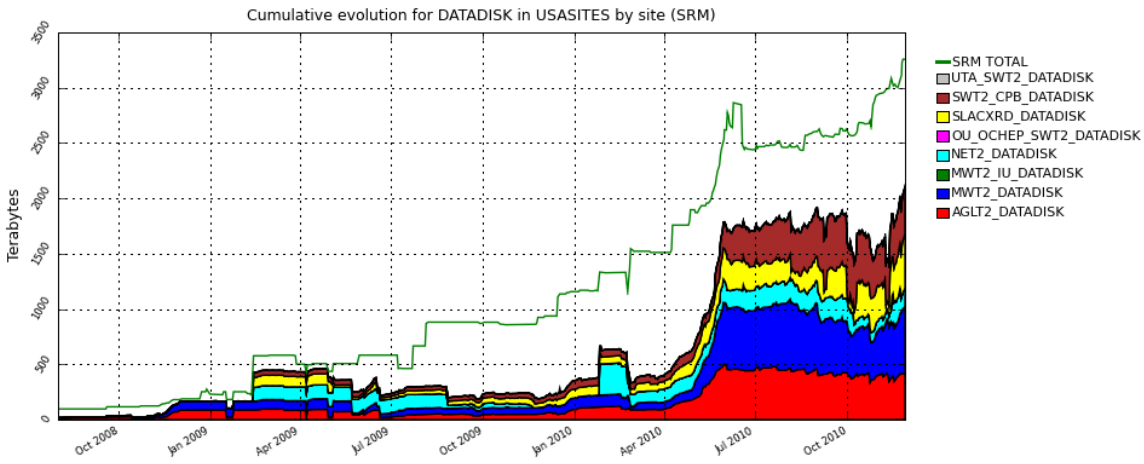

Figure 2. Cumulative evolution of data flow to US Tier-2 sites

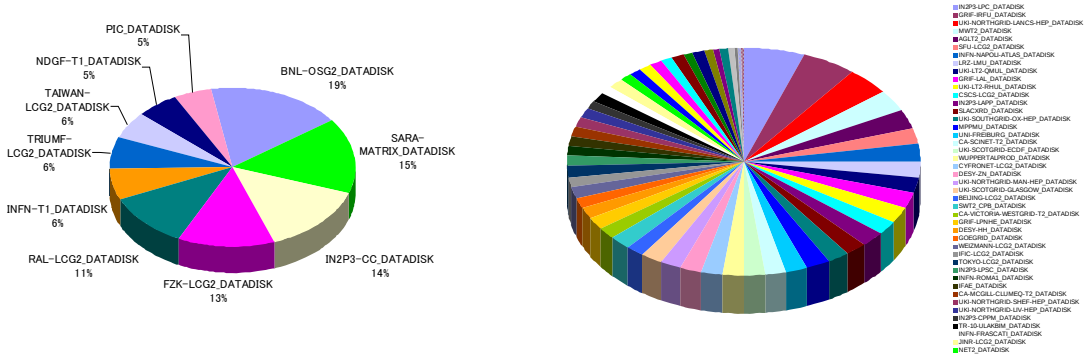

Figure 3. Copies made by PD2P in 2012 at Tier-1 sites (left) and Tier-2 sites (right)

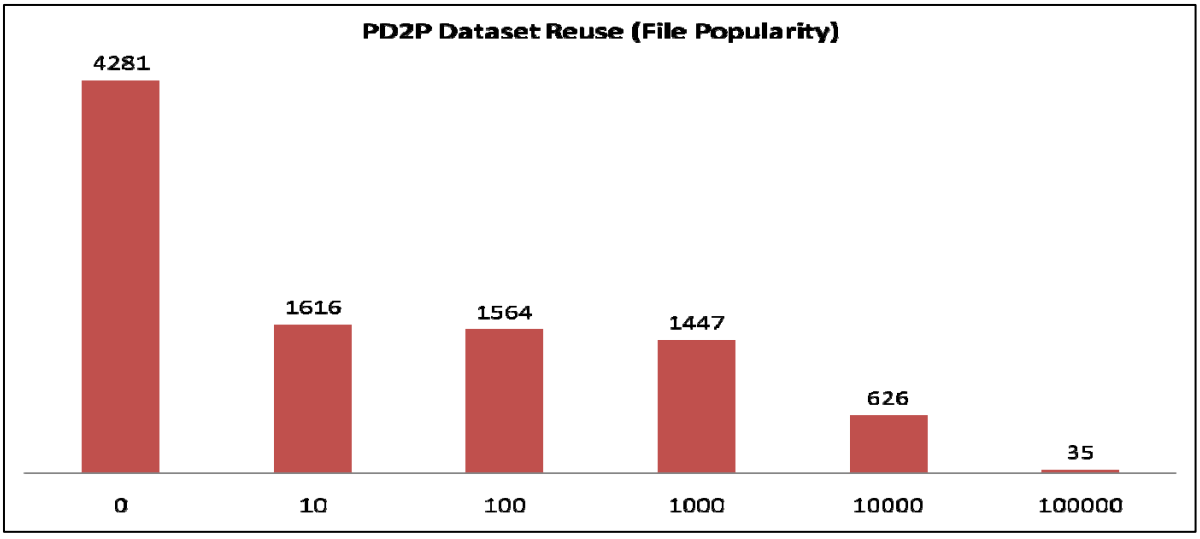

Figure 4. Frequency of Reuse for PD2P datasets

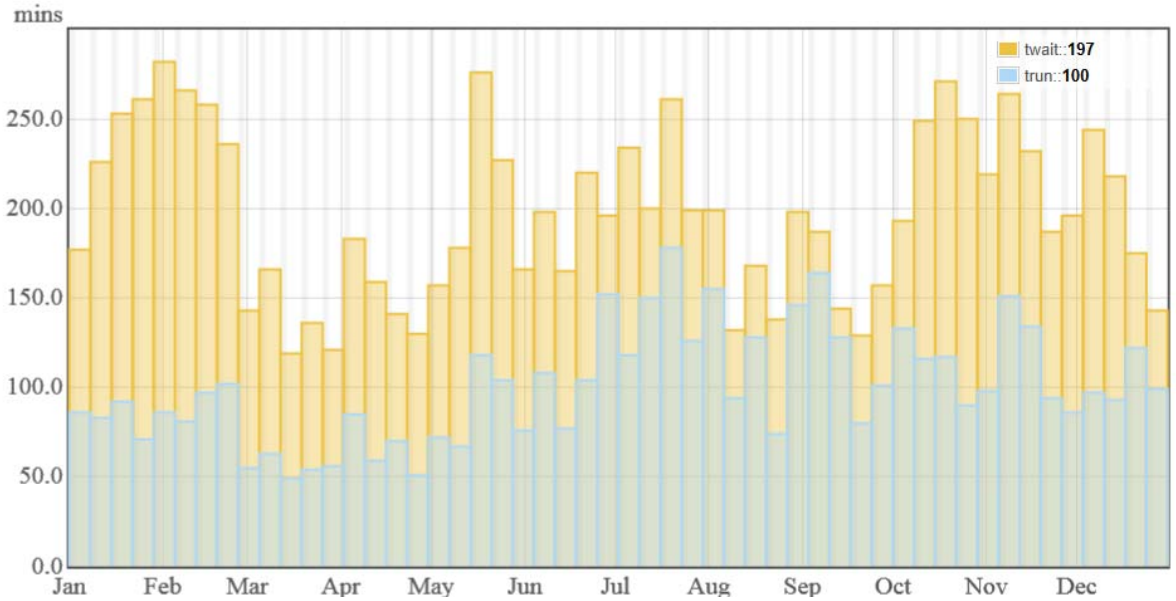

Figure 5. Average wait time (yellow) and execution time (blue) for analysis jobs in 2011

6. Future Plans

PD2P uses Tier-1 sites as primary data repositories while using Tier-2 sites as temporary storage for cache data. With PD2P, caching is performed at the dataset level. The federated xrootd system [9] will extend the caching scheme from the dataset level to the file level. Furthermore, ATLAS intends to evaluate a more fine-grained approach to caching, below the file level, by taking advantage of a ROOT-based caching mechanism [10] as well as efficiency gains in ROOT I/O implemented by the ROOT team that minimizes the number of transactions with storage during data read operations. The brokerage of the PanDA system will be improved to have cache-awareness.

7. Conclusions

The PD2P system has been developed to cope with the challenges of data placement to effectively serve the ATLAS analysis community amid constrained storage and processing resources. PD2P shows a large improvement in terms of disk usage efficiency, while distributing ATLAS data (and thereby analysis processing) at Tier-1 and Tier-2 sites broadly. ATLAS's caching scheme will be extended from PD2P's dataset level to more fine-grained levels in the future, based on I/O developments presently underway.

References

[1] Maeno T., *Overview of ATLAS PanDA Workload Management*, J. Phys. Conf. Ser. **331** (2011)

[2] Caballero J., *AutoPyFactory: A Scalable Flexible Pilot Factory Implementation*, in Proc. of the

19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[3] Nilsson P., *The ATLAS PanDA Pilot in Operation*, J. Phys. Conf. Ser. **331** (2011)
[4] Filipcic A., *arcControlTower, the sytem for Atlas production and analysis on ARC*, J. Phys. Conf. Ser. **331** (2011)
[5] Garonne V., *The ATLAS Distributed Data Management project: Past and Future*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[6] Legger F., *Improving ATLAS grid site reliability with functional tests using HammerCloud*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[7] Iglesias C. B., *Automating ATLAS Computing Operations using the Site Status Board*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[8] Panitkin S., *A Study of ATLAS Grid Performance for Distributed Analysis*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[9] Lothar B., *Using Xrootd to Federate Regional Storage*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[10] ROOT : http://root.cern.ch/