Single top quark production at the LHC: Data processing and cross section measurement
Lee, H.C.

Citation for published version (APA):
Lee, H. C. (2013). Single top quark production at the LHC: Data processing and cross section measurement
The “offline data processing” here refers to processing the data into a format that is required for physics analysis. It is distinguished from the “online data processing” in the trigger system of the ATLAS detector. For instance, simulation and event reconstruction discussed in previous chapters are two major steps of the offline data processing. Given the amount of data, their executions are challenging in terms of computing.

In this chapter, we address the computing challenges of running the offline data processing on large amounts of data. In Section 4.1 we introduce the platform, the World-wide LHC Computing Grid (WLCG), on which the offline data processing is performed. We discuss in Section 4.2 the approaches and the tools developed by the ATLAS experiment to tackle the computing challenges in processing and managing massive amounts of data. Based on the accounting information collected during the operation, performance and efficiency will be shown in Section 4.3.

### 4.1 The World-wide LHC Computing Grid

The LHC experiments are international collaborations, thousands of physicists around the world expect to access to the experimental data as soon as they are produced. Estimations [104, 105, 106, 107] showed that large computing power and storage capacity are needed to make the LHC data available for physics researches. Having data and computing resources located centrally leads to issues of scalability and unaffordable management cost.

Given the fact that institutes in the collaboration facilitate computing resources to support the LHC experiments, a system that can integrate and make use of those
distributed resources will provide a scalable approach for solution, spread management cost over institutions, and to some extent, encourage collaborations.

With resources managed by different administration domains, a number of issues have to be addressed to build such a system. First of all, a security mechanism has to be established and enforced to protect individuals and facilities from vulnerabilities in sharing resources over their administration boundaries. Secondly, the system, as a whole, has to remain stable while facilities may attach or detach themselves dynamically to the system. Furthermore, the integration has to go beyond one technology given the resource heterogeneity. Aiming for addressing these issues, the grid computing technology [108] is adopted for developing the largest distributed computing system in the world, the WLCG.

4.1.1 Computing resources

The computing resources of the WLCG consist of distributed computing clusters and storage facilities. They are presented in units of Computing Element (CE) and Storage Element (SE). Depending on the funding, resources on the WLCG are operated under three organizations of European Grid Infrastructure (EGI) [109], Open Science Grid (OSG) [110] and the Nordic Grid (NORDUGRID) [111].

A CE refers to a group of CPUs managed by a job scheduler. To use the CPU power, computing tasks are prepared as user jobs submitted to the job scheduler. Job priority as well as resource allocation are managed by the job scheduler based on the administrative policies defined by the data center. Depending on the choice of the data center, various job schedulers are operated on the WLCG.

A SE refers to a storage facility managed by a storage manager. There are several storage managers made available for the WLCG collaboration. For instance, Castor [112] and dCache [113] are designed to integrate with tape-based archival systems, while DPM [114], Storm [115] and EOS [116] focus more on disk-only storage facilities. Each technology has its own data access protocol and develops its own approach for managing the scalability and reliability of the service under intensive data access.

A grid site is defined as a collection of CEs and SES located, usually but not necessarily, at a data center. Given the grid sites are different in resource capacity and network connectivity, they need to be organized in such that resources are used properly for their best performance. After analyzing possible scenarios of organizing grid sites for the LHC computing, the most feasible model proposed by the study [117] is to structure the grid sites in a tiered hierarchy. Each tier is then given a special function in the LHC computing. There are four tiers in the hierarchy:

- **Tier-0** is the data center at CERN which provides the main computing power for prompt data processing and the archival facility for the collision data.
4.1. The World-wide LHC Computing Grid

- **Tier-1 sites** facilitate backup storages for the collision data. As a consequence of that, the main responsibility of Tier-1 sites is to reprocess the collision data whenever necessary. With relatively larger computing capacity, Tier-1 sites also contribute to the simulation and data analysis activities.

- **Tier-2 sites** provide auxiliary computing powers for the simulation. A fraction of data is also replicated to Tier-2 sites to support data analysis activities.

- **Tier-3 sites** are small computing facilities. Resources at Tier-3 sites are mostly reserved for the data analysis activities locally within institutions. Thus, they are not officially accounted as part of the WLCG resources.

The tiered hierarchy indicates not only the resource capacity and the level of the operational support a site has to provide, but also the requirements of the network connectivities between grid sites. In general, Tier-0 and Tier-1 sites facilitate more resources with more advanced operational support. They are also connected with high-speed, private optical networks [118] to ensure reliable and performant data flow during the data taking. Large resource capacity and advanced network connectivity at Tier-2 and Tier-3 sites are certainly useful; but they are generally not mandatory.

![Figure 4.1: Distribution of the WLCG grid sites around the globe. The Tier-0 at CERN is shown in yellow. Tier-1 sites are marked in blue; while Tier-2 sites are in red.](image)

According to the accounting by July 2013 [119], the WLCG consists of 13 Tier-1 and 143 Tier-2 sites. They are operated by 153 data centers across 37 countries in the world. Together with the Tier-0 at CERN, the WLCG provides approximately 430 thousands of CPUs, 2031 petabytes of online (disk) storage and 271 petabytes of archive (tape) storage. The distribution of grid sites around the globe is shown in Figure 4.1. Most of them are located in Europe, North America and Asia.
4.1.2 Grid middleware

The grid middleware is a software layer aiming at connecting distributed resources of the WLCG into one computing system. It addresses the issues of sharing resources across administration boundaries, providing basic functionalities for managing distributed data and computing jobs in a dynamic and secured manner.

The foundation of the grid middleware is a security infrastructure based on a concept called the “virtual organization”. Built on top of this foundation are the three middleware components: the “information system”, “data management system” and the “resource management system”.

Virtual organization

Introduced by Ian Foster et al in 2001 [120], a Virtual Organization (VO) is defined as an organization of distributed individuals and institutions sharing resources with each other in order to achieve a common goal or to solve certain scientific problem. The sharing goes beyond simple file exchange to a level of direct computing and storage access. The grid computing technology is to provide a secure, flexible and coordinated way of sharing a collection of dynamic resources given the setup of the VOs.

On the WLCG, four VOs are established corresponding to the four LHC experiments. The arrangement makes virtual boundaries between experiments in sharing the same computing infrastructure. It also protects data produced by one experiment from being accessible to others. Within an experiment, managers can decide how resources are shared and prioritized among collaborators without affecting other experiments. For the grid sites supporting more than one experiment, site managers can also organize resources effectively to meet different requirements from the experiments.

Technically, each VO maintains a Virtual Organization Membership Service (VOMS) to control the resource sharing. Every collaborator acquires a VO membership by registering personal identity in the corresponding VOMS. The access to the resource belonging to a VO can only be authorized when the issuer of the access is recognized as a member of the VO.

Within each VO, specific roles referring to different physics groups, computing activities or institutions are further defined for a fine-grained control of the resource sharing. For example, a group of people working for computing operation in the ATLAS experiment are associated with the so-called “production” role which has higher resource usage priority than other members in the ATLAS VO.
4.1. The World-wide LHC Computing Grid

**Information system**

Given that the computing resources are not centrally controlled, the information system is a mission-critical component on the WLCG. The information system, to some extent, defines the scope of the grid as every resource taking part of the grid has to present itself in the system. The system provides functionalities allowing users and services to discover existence and characteristics of the presented resources available in a certain moment for subsequent management or use.

The WLCG information system has a structure of three levels. The fundamental building block used in this hierarchy is the Berkeley Database Information Index (BDII) which is essentially a database of the Lightweight Directory Access Protocol (LDAP). Each CE and SE is co-located with a resource level BDII providing information concerns only the resources themselves. A site level BDII running at each grid site aggregates information from the SEs and CEs associated to the site. The top-level BDII collects all the information from the site-level BDIIIs and hence contains information about all grid services. Multiple instances of top-level BDII are deployed at various Tier-1 centers in order to provide a fault tolerant, load balanced grid information system. Practically speaking, the difference between resource level, site level and top level BDIIIs is just information content and scope.

The information system is bootstrapped from the site registration in the Grid Operations Center Database (GOCDB). When a site registers, it enters into the GOCDB the LDAP URL to its site level BDII. A list of LDAP URLs generated by the GOCDB is downloaded by the information provider running on the top level BDII and used to query all the site level BDIIIs. The result is then used to populate the top level BDII.

**Data management system**

Data on the grid are stored in files and spread over distributed storage facilities. In order to organize them and to be able to access to them regardless of their physical locations, the data management system of the grid middleware provides functionalities for managing storage space, indexing data files and transferring files among storage facilities.

Given that data within the SEs are managed by various storage systems, it is necessary to specify an uniform interface as well as the functionalities to be provided by different storage systems for meeting the data management requirements of the LHC experiments. After analyzing the data management use cases of the LHC experiments, the Storage Resource Management (SRM) specification is designed for this purpose.

For example, moving data between two SEs requires several actions at both SEs. Before the file transfer can start, the file at source SE may need to be staged from a nearline storage (e.g. a tape library) to a online disk. A necessary amount of storage
space at the destination SE also needs to be reserved. Upon the finish of the transfer, a proper integrity check has to be made as data can be corrupted due to various issues. Once the transfers are confirmed to be complete, both source and destination SEs are notified to continue the necessary post-processes of management, such as assigning lifetime to the new replica, migrating files to backend storage. Those management works are encapsulated in the SRM specification.

Files on the grid are identified at different level using different naming scheme. The Site URL (SURL) is used to name a file within a SE. With SURL, files in a SE can be conveniently identified without having to know where and how they are stored in the underlying storage facility. For indexing identical copies (replicas) of a grid file on various SEs, the Globally Unique IDentifier (GUID) is introduced. When a new file is created on the grid, it is assigned by the system with a GUID. Replicas of the file will acquire new SURLs; while they share the same GUID. As the GUID is a 40-bytes hashed string, a more user-friendly alternative to GUID is the Logical File Name (LFN). A GUID can be associated with multiple LFNs to create aliases referring to the same file and its replicas. Figure 4.2 illustrates the relations between SURL, GUID and LFN. Those relationships are managed by an indexing service called the LCG File Catalogue (LFC).

![Diagram](image)

**Figure 4.2:** Three different naming schemes of a grid file and their relations.

Bookkeeping large scale data movement and recovering transfer failures are tedious works for applications and users. In addition, the amount of data transferred between two SEs at a given time has to be controlled in order to achieve better throughput and network utilization. To enable the control and ease the bookkeeping work, the File Transfer Service (FTS) was introduced to deal with all necessary complexities. Through the FTS, file transfers are scheduled on predefined “channels” connecting source and destination SEs of the transfers. Network utilization between source and destination SEs is then throttled by controlling the number of concurrent transfers on the channels. The FTS also tracks transfer progresses and retries failed transfers.
Resource management system

The “resource” we are referring to here is the computing power provided by the CE. The uniform interface for managing jobs on the computer cluster behind a CE was originally defined by the Globus Toolkit. It was succeeded by the Computing Resource Execution And Management (CREAM) interface in 2011. The defined functionalities include job submission, deletion and status checking as well as necessary file staging mechanisms for input and output sandboxes.

With multiple CEs, it’s necessary to find the one that is capable to execute a given job. In addition, bookkeeping on distributed jobs on multiple CEs as well as retrying jobs upon failures are common management tasks that can be done automatically in a systematic way. A system offering those functionalities is the so-called “workload management system”. In the WLCG grid middleware, two workload management systems are popularly used nowadays. Condor [121] interacts mainly with CE for collecting resource information and uses its own resource advertising mechanism based on ClassAds [122] for resource selection and job brokering. Built on top of Condor, the gLite Workload Management System (gLite WMS) cooperates with the grid information system providing more dynamic and complex algorithms for resource-job matching.

4.1.3 Operational dynamics

One distinct feature of the WLCG is its living structure. Availability of resources can change dynamically due to infrastructure maintenance, hardware and software upgrades and random, unscheduled events that are expected to happen at anytime during the application runtime.

The GOCDB is provided as a system where sites may declare scheduled and unscheduled downtimes. A study [123] based on the site availability statistics tracked by the GOCDB showed that there were nearly 15,000 site interventions in the EGEE grid during 7 years of the operation. Among those interventions, 64% are scheduled for a planned intervention and the other 36% are unscheduled due to unexpected incident on services. The typical duration of interventions is below 3 hours, but in some occasions, may take up to weeks. The weekly accumulated downtime across all grid sites is fluctuating between 100 and 200 days indicating the availability of resources is dynamically changing at rather large scale. Some interventions are well planned and announced few weeks in advance, but large majority of interventions is registered very late meaning that in most of cases users are only notified shortly before interventions start.

Large-scale applications not taking the operational dynamics into account can suffer by failures, resulting in serious degradation on performance. Thus, developments have to take place on the application level to account the dynamics.
4.2 Offline data processing on the Grid

The grid middleware discussed in previous section connects distributed computing resources and provides basic functionalities for file and workload managements. As the grid middleware is designed to be generic for applications, it leaves two optimization issues to be addressed by the application running on top of it. Given an application workflow, the first issue concerns how data should be organized in such that the storage usage and the data-processing throughput are both optimized. Secondly, the application should also develop strategies to deal with possible failures due to the operational dynamics which is not fully addressed by the grid middleware. Developments for addressing these issues need to orchestrate grid middleware services in an application-specific way.

Focusing on the ATLAS offline data processing as one of the WLCG application, we will discuss the developments in the ATLAS experiment for processing and managing large amount of data on the grid. We give firstly an overview of the computing activities involved in the workflow and discuss the challenges arising from the amount of experimental data.

4.2.1 Activities and challenges

Figure 4.3 illustrates the entire ATLAS data processing workflow. The offline data processing is a pipeline of three computing activities drawn in solid-lined boxes in the figure. They are activities for “simulation”, “event reconstruction” and “group production”.

![Figure 4.3: The schematic diagram of the ATLAS data processing pipeline. The three computing activities involved in the offline data processing are shown in solid-lined boxes.](image)

The activity for simulation involves computations for generating physics events and simulating their interactions with the detectors as discussed in Section 2.2. Given the
detector complexities, this is the most CPU intensive step of the whole data processing pipeline. The output of this process is stored as raw data objects (RDO) providing comparable information to the detector readouts from real collisions.

The activity for event reconstruction applies the algorithms discussed in Section 3.1 on both simulated and data events to reconstruct physics objects from detector signals. A detailed output of the reconstruction is written to an event summary data (ESD) file; while less detailed summary is written to an analysis object data (AOD) file. The reconstruction is expected to be repeated several times as physicists have better understanding of the detector or improvements on the reconstruction algorithms.

Physics analyses can already start with data in AOD and ESD formats as the event information is provided inclusively for the majority of analyses. In practice, analyses in a physics group utilize only a subset of the event information. The activity for group production is to process data in AOD and ESD formats into more lightweight ones in which only the event information necessary for analyses in a physics group is preserved. The lightweight data formats are distinguished in three types. DAOD and DESD are essentially the same formats as AOD and ESD with reduced event information, while D3PD is a format with event information stored in a n-tuple structure to be discussed later in Section 5.1.

Computing challenges

Generally speaking, each computing activity of the offline data processing performs as a parameter-sweep application in which the same data processing program runs independently over data events. Thus, the computation of each activity is carried out by independent computing jobs executed in parallel on the grid. Given the amount of data to be processed, managing these parallel jobs through the pipeline is a challenging task.

One challenge is to manage a massive amount of computing jobs distributed on the grid. For instance, there were about a billion collision events collected by the ATLAS detector in 2011. For preparing physics analyses on this amount of events, more than 30 million computing jobs were performed on the grid for the simulation and the event reconstruction\(^1\). Given the operational dynamics of the grid resources, jobs can fail with various reasons causing the pipeline to be blocked. Furthermore, maintaining the software packages required by those jobs on more than 150 grid sites is also not trivial.

Yet another challenge is to organize and maintain a large amount of data throughout the data processing pipeline. For instance, the execution of a job relies on the availability of the outputs from its predecessor jobs. An issue arising here is about optimizing data movements between jobs to achieve a best data processing throughput out of

\(^1\)The number is given by querying the job historical view dashboard [124] on the total number of jobs submitted for the “MC production” activity in the context of the “MC11_7TeV” project.
Chapter 4. Offline Data Processing

the pipeline. Furthermore, outputs from jobs need to be organized according to their management properties such as the privilege and the lifetime. The accessibility of outputs should be also made available to the whole ATLAS collaboration as soon as the they are produced on the grid.

Physics studies relies on a successful offline data processing. To tackle the computing challenges of the offline data processing, certain technical approaches and methodologies have been developed by the ATLAS computing group through years of preparation. We will discuss these developments in the contexts of data management, workload management, software distribution and operation.

### 4.2.2 Data management

Building on top of the data management system provided by the grid middleware, the ATLAS data management concentrates mainly on two tasks. One is to decide how to organize data within a storage and among storages, the other is to develop an efficient way for transferring data between grid sites. The development results in the ATLAS data management system called Don Quijote2 (DQ2). The architecture and technical implementation of the DQ2 system are discussed in [125]. Hereafter we focus on the key features implemented in this system.

**Dataset**

The data of the ATLAS experiment consists of events. They are stored in files to be managed by the grid middleware. For organizing the data by experimental properties, i.e. events collected in a period of run or generated by simulation for certain physics process, the “dataset” is introduced to group files in which events with the same experimental property are stored. Dataset is also provided as the granularity of the ATLAS data management given the fact that how data should be managed within the collaboration is closely related to their experimental properties.

In view of data management, files are grouped into datasets according to their management purpose. A file can be part of multiple datasets at the same time in the system when it is involved in various data management activities. For example, the 3 data files on site A in Figure 4.4 are grouped into dataset 1 as a primary grouping; while two of them are also grouped into dataset 2 for the purpose of moving part of the dataset 1 to site C. Associations between datasets and files are maintained internally by the DQ2 system which also translates dataset managements into file-level operations in the grid middleware layer.

In addition to the dataset-file association, each dataset replica has two important properties for data management. The first one is the replica status indicating the data completeness of a dataset at a given site. A dataset replica is “complete” if all the consisting files are presented at a site or “incomplete” when some files are missing or
4.2. Offline data processing on the Grid

![Diagram showing dataset replication and storage classes]

**Figure 4.4:** An illustration showing that 3 data files at site A are grouped into two datasets. It also shows two replication operations where dataset 1 is replicated to site B and dataset 2 to site C. The datasets are replicated to specific storage classes according to their retention policies. The replica of dataset 1 at site B is incomplete since file 3 is not yet transferred to the site.

still to be transferred. In Figure 4.4, the replica of dataset 2 at site C is complete, while the replica of dataset 1 at site B is incomplete since file 3 is not yet available there.

The second property of a dataset replica indicates the data retention policy at a given site. A “custodial” replica must be kept available as long as the dataset is still valid in the system. A “primary” replica can only be deleted upon decisions from managers. Temporary copies are marked as “secondary” replicas and they can be removed whenever necessary to reclaim storage space.

**Storage class**

Datasets need to be managed differently according to their retention policy, ownership and access permission. For instance, raw datasets from the ATLAS detector are required to remain available until the end of the ATLAS experiment, while temporary datasets should be removed regularly if they are no longer needed.

In the ATLAS data management, storage class refers to space on SE satisfying certain set of data management requirements; the SE implements such a class by providing the necessary technologies and services. For example, by placing datasets into a space on SE of storage class “ATLASDATATAPE”, the system makes sure that those datasets will be stored permanently on a tape archive.
Figure 4.4 shows one example where the datasets are replicated to a different storage class according to their retention policy. The custodial version of dataset 1 is located at the “ATLASDATATAPE” at site A, while a primary replica of it is stored at the “ATLASDATADISK” at site B. Dataset 2 is created for a temporary purpose therefore its replication to site C is placed to a storage class considered as a scratch space, the “ATLASSCRATCHDISK”.

**Grid site association**

Data transfer is more efficient and performant if flowing through a network path with better reliability and larger bandwidth. In order to define and utilize the optimal network paths among grid sites, the Tier-1 and Tier-2 sites are grouped into associations following the tiered architecture of the WLCG. Each association consists of one Tier-1 site with several satellite Tier-2 sites. Tier-2 sites are usually associated with a Tier-1 in the same country as the network connectivity within a country is usually better than across countries. For those Tier-2 sites not having a Tier-1 in the same country, the association is carefully made to have decent network connectivities between the associated Tier-1 and Tier-2 sites. Given the number of the ATLAS Tier-1 sites, 10 associations are defined\(^2\).

Following the site association, data transfers between associations is by default restricted to flow through their corresponding Tier-1 sites connected with an advanced network. One consequence of it is that, as illustrated in Figure 4.5, data transfers between two Tier-2 sites “i” and “iv” should take a detour over their associated Tier-1 sites “A” and “B”. For the Tier-2 sites where good network connectivity is available, this restriction is loosened to allow the direct data transfers between Tier-2 sites across associations. These transfers are shown in Figure 4.5 as the lines labelled with “T2D”.

Site association helps to distribute the load of managing massive file transfers globally. In the setting of the ATLAS data management, each Tier-1 site hosts a WLCG FTS to handle the file transfers within and into its representing association, while the transfers to and from the Tier-0 site are managed by another FTS instance right at CERN.

Site association also forms the structure around which the management of the ATLAS computing activities is organized. We will discuss later how this structure plays a role in data distribution, workload management and operation.

\(^2\)Each association is nicknamed as “X-cloud” where “X” runs over a list of CA, DE, ES, FR, IT, ND, NL, TW, US and UK, corresponding to the ATLAS Tier-1 sites at Canada, Germany, Spain, France, Italy, Nordic countries, the Netherlands, Taiwan, the United States and the United Kingdom, respectively. The choice of “cloud” here is unfortunate, as it is sometimes confused with the “cloud computing” which is completely different.
4.2. Offline data processing on the Grid

![Diagram of data transfers among Tier-0 and Tier-1 and Tier-2 sites]

**Figure 4.5:** A simplified diagram illustrating the possible data transfers among Tier-0 and two associations of Tier-1 and Tier-2 sites. Transfers are presented as the arrowed lines between sites. The direction of the data flow is indicated by the arrow, while the line style refers to the FTS instance by which the transfer is managed. Lines labelled with “T2D” are direct data transfers to a large Tier-2 site (i.e. site ii) without going through its associated Tier-1 site (i.e. site A).

**Data distribution**

As soon as the data is produced by the offline data processing jobs, it has to be distributed to grid sites. Data distribution serves the purposes of delivering data to physicists around the globe, making backup copies of data, and implementing the data flow for the data processing pipeline. To which sites should a dataset be replicated? How many replicas of a dataset should be maintained? These are questions to be addressed by the data distribution model of the ATLAS data management.

Following the static site association, data distribution among sites in early operation (before 2011) was carefully planned in advance. In practice, this is an exercise of deciding number of copies to be made available on the grid and calculating what fraction of data a site should receive based on the storage capacity it offers. This approach is simple and works fine as long as the amount of ATLAS data is lower than the grid storage capacity; but when the amount of data grows significantly, it turns...
out to be an inefficient resource utilization as not all datasets are equally popular. Not only the static data pre-placement can filled up the storage with copies of unpopular data, it can also lead to a situation of unbalanced resource utilization in data analysis. A site may receive lots more user jobs than the other just because some popular datasets were transferred to it by chance.

With the direct data transfer enabled between Tier-2 sites from two associations (the lines labelled with “T2D” in Figure 4.5), a more dynamic data distribution scheme is introduced in 2011. When a dataset is produced, it’s only replicated up to a reduced number of copies or even not replicated at all. Further data replication can be triggered dynamically by the usage of it. If the dataset is highly requested by user (i.e. lots of jobs request to run on the same dataset), a background process will replicate it automatically to a new location to reduce the load on the original replica. Since the dataset popularity is time dependent, the automatically replicated copies are considered to be deletable after certain period of time when space is needed.

Apart from the pre-placement and dynamic replication, datasets can also be transferred upon users’ requests. This on-demand data transfer provides a flexibility allowing physics groups to plan an ad-hoc data distribution for certain data processing purpose. Individual users can also replicate datasets to a site close by for interactive data analysis. To avoid possibly unorganized transfers generated by on-demand requests, requests have to be approved by managers before turning into actions. Additional copies of data created this way is also taken into account for future data replication and processing on the grid.

In addition, data distribution are also triggered on-demand by the ATLAS workload management system for managing the data flow of the data processing pipeline.

### 4.2.3 Workload management

The ATLAS offline data processing activities are organized as computing tasks. A computing task, in its abstract form, consists of a data processing algorithm, an input dataset to be processed, and a resulting output dataset. A computing task breaks down into workloads, each of them runs the data processing algorithm over certain files belong to the input dataset, producing output files becoming part of the output dataset.

For managing the workloads in conjunction with the ATLAS data management, a data-centric workload management system called the PanDA Production ANd Distributed Analysis system (Panda) is developed. In contrast to the generic resource management systems in the grid middleware, workload management within the Panda system takes into account aspects that are specific for the ATLAS computing, for instance, the data processing pipeline, the grid site association, and the software availability. Since the architecture and technical implementation of the Panda system have been
4.2. Offline data processing on the Grid

discussed elsewhere in [126], we will just highlight the key features of the system below.

**Workload brokering**

In the ATLAS workload management system, each computing task is given a priority depending on how urgent the output of it is needed by physics groups and its relative position in the entire data processing chain. The task is then assigned to a site association providing that all input files have at least one replica within the association, and the corresponding Tier-1 site will have sufficient storage capacity for the output files by the moment the whole task is finished. When there are more than one site associations capable to run the task, the one with the least estimated waiting time for the task to start is chosen.

Once the task is assigned to a site association, the workloads are distributed to the sites within the association taking into account more detailed resource information. That is by considering the resource requirements of the workloads, software availability, the local storage capacity, computer node capacity, site downtime, the estimated waiting time and the locations of input files.

The workloads are assigned to a grid site by putting them into a queue corresponding to the site, waiting for necessary provisioning works to be done for executing the workload. One of the provisioning works is to dispatch input files to the site’s local storage if some of them are not presented at the site. Remember that the data transfer in ATLAS acts only on datasets, moving input files requires to create temporary datasets and replicate them to the site where the workload will be executed.

The workload is activated only when all input files are presented at the site. This is to avoid massive and uncoordinated data access through the WAN, which is known to be problematic and inefficient. It also opens up a possibility to utilize the native data access protocols supported by the storage technology for performance.

**Pilot-job framework**

Workload execution on the grid is carried out by “grid jobs”. Generally speaking, a grid job consists of a workload and some setups on the computer node, such as inspecting the configuration of the computer node and setting up the runtime environment, to ensure the workload can be executed properly on a remote resource. In a typical approach, a grid job is configured in a way that those setups are attached to a specific workload before it is submitted to a grid site. In the ATLAS computing, a different approach is taken. In this approach, the setups on the computer node are detached from the workload and generalized into a lightweight grid job called the “pilot” job. Pilot jobs are submitted constantly to the grid sites. Upon the finish of the setup works,
they pull activated workloads from the PanDA system and execute the workloads on the computer nodes.

The pilot-job framework has several advantages for scalability and performance. Firstly, the pilot jobs ensure the computer nodes are well prepared and available immediately for executing the workloads; thus, the failure rate is reduced. Secondly, the pilot jobs can be instrumented to perform workload bookkeeping or result validation making the system scalable. Moreover, as the pilot jobs are light-weighted, job submission and bookkeeping are less complex. The bookkeeping on the pilot jobs actually becomes less important and can be even ignored in the sense that the pilot job status is independent to the workload status that is the main concern of the users.

Furthermore, with the detachment between workload and pilot job, the system separates the scheduling of workloads from the scheduling of grid jobs, the later can be largely uncertain given the job scheduling policy varies among grid sites. A benefit of it is that when, for example, a workload’s priority has to be increased, the system can quickly re-schedule it on the workload level and attach it to the next available pilot job, resulting a shorter turnaround time of the arrangement.

The pilot-job framework relies on a steady but not too aggressive supply of the pilot jobs. A shortage of pilot jobs can generate a huge backlog of workload execution, while the computing resources can be overwhelmed by lots of “do-nothing” pilot jobs if they are too many compared to the actual workloads to do. In the pilot-job supplier (a.k.a. the pilot factory), communications with the workload manager are established to achieve a better balance in submitting pilot jobs to the grid sites.

4.2.4 Software distribution

The ATLAS software has to be made available on a grid site for executing workloads. Considering the analysis phase of a large physics experiment continues for several years after the last data is acquired, the software will have a long lifetime. In addition, every 6 months there is a new release, which provides the basis of the software used in the next round of simulation, event reconstruction and the following data analysis. How to distribute a release efficiently to all grid sites and maintain various releases globally are key issues to be addressed in this context.

Before 2011, the deployment of a ATLAS software release was done by submitting sets of high priority jobs to the grid sites to run a software installation process. It assumes that the storage for software (a.k.a. the software area) is prepared and shared by all the computer nodes within a site; therefore, the installation job only needs to be done once on one of the computer nodes. The installation job runs a procedure consisting of fetching software packages from a central repository, installing packages to the software area, configuring packages with respect to the site specification, and finally validating the installation. Sites passing the validation will receive a new software
tag published to the grid information system indicating the availability of the new software release.

The software area is provided differently across grid sites in terms of space and technology. Sites running out of space cannot receive any new releases. In this case, additional jobs have to be submitted to those sites for removing few older releases. The removed releases may still be used by the ATLAS community; removing them from a site implies resource reduction for certain data processing activities. Furthermore, new release installation at the site is delayed by the turnaround time of the deletion jobs. Technology used for building the shared software area also affects the site scalability and job efficiency. For instance, the widely used NFS version 2 from 1990s wasn’t designed in the first place for handling the scale of the computer clusters nowadays, causing a limitation factor to the number of computer nodes a site can expand without overloading the shared software area.

In 2011, a virtual filesystem called the CVMFS [127] is adopted to deliver software to distributed computer nodes in a fast, scalable and reliable way. The design of the CVMFS exploits the following properties:

- Software release has fixed content. Once a release is made and deployed, none of its consisting files will be modified.
- For one single data processing job, only part of the files and libraries from one release are used.
- A more recent software release is usually used more often than earlier releases; or there are usually some releases more popular than others.

Instead of installing software releases physically at sites, they are installed centrally in a very large software repository. Using the CVMFS, files added to this repository appear immediately in all computer nodes on the grid; while their contents are only fetched when they are opened on the computer nodes.

Given the fact that fetching file contents from the central repository is simply a reading process, the system employs the reliable and scalable HTTP protocol, by which the load on the central software repository can be easily delegated to the proxy servers provided by the grid sites. The file contents used by previous jobs are cached on the proxy server as well as the computer node’s local disk. The following jobs using the same contents will benefit from reduced latency for content fetching. Integrity of file contents between the central repository and caches are maintained by the CVMFS system.

CVMFS provides immediate availability of software releases since it only needs to be installed physically once at the central repository and the delivery to the site is triggered by the actually usage. It also removes the scalability limitation coming from having large amount of computer nodes sharing a common software area within a site.
4.2.5 Operation

The “operation” here refers to the management of the systems and services we discussed before to ensure that the outcome of the offline data processing meets the requirement from the physics community. The requirement can be a data delivery deadline, workload priority, or how data should be distributed among grid sites.

The ATLAS offline data processing is executed on a complex system built upon distributed services. The scale of data to be processed for enabling physics analysis also pushes the system to its limitation. Any issue concerning resource and service reliability can result in significant delay of physics analysis, affecting consequently the quality of physics measurement. In order to deliver the data required by physics analysis in time, the operation of the ATLAS computing plays an essential role for a smooth offline data processing. We will discuss in the following how the operation is organized.

Operational hierarchy

Given the layered structure and the scale of the system, the operation of the ATLAS computing is structured in a three-level hierarchy as illustrated in Figure 4.6. From top-down, they are “central operation”, “regional operation” and “site operation”.

![Figure 4.6: The operational hierarchy of the ATLAS offline data processing.](image)

In terms of the scope, the central operation is more workflow oriented making sure the requirements from the physics community are fulfilled and the data is delivered in time, while the site operation is more resource oriented concerning more about the accessibility and reliability of the computing resources at the grid sites. Being aware of the data processing requirements and the resource capability, the regional operation in the middle coordinates necessary configurations to achieve better data processing performance within a grid site association.

Following the distributed nature of the grid, operation also takes place in a distributed manner. The central operation is performed centrally by a small group of experts.
In contrast, the site operation is rather distributed as it has to be carried out by system administrators at the grid sites. The distributed efforts are communicated and coordinated via several means, such as regular meetings, mailing lists, ticketing systems, in order to achieve a smooth and successful operation for the data processing.

**Operational shifts**

Since the offline data processing runs 24 hours a day and 7 days a week, shifts are introduced to achieve round-the-clock monitoring and issue reporting to the operation units mentioned above. A central operation shift is setup to monitor the first-pass data processing at CERN as well as the following data distribution to Tier-1 and Tier-2 sites. In addition, the Atlas Distributed Computing Operations Shift (ADCoS) is organized in three different timezones: Asia-Pacific Timezone (00-08 CET), the European Timezone (08-16 CET) and the American Timezone (16-24 CET), overseeing the data processing activities taking place at Tier-1 and Tier-2 sites. Shifters are supervised by on-call experts.

When an issue is observed, shifters and on-call experts provide first-line investigation, trying to identify the cause of the issue. Once the cause is identified, they make contact with relevant operation units through well-defined tools (e.g. a ticketing system) for necessary operational actions to resolve the issue. Before the site functional test (discussed below) is introduced, shifters are also responsible for excluding malfunctioning sites and services from being used by the data processing, and including them back once issues are resolved.

**System monitor and functional tests**

In a distributed system, the system monitor helps operation to spot issues happening at a remote site or service. It also provides auxiliary information for finding the cause of an issue. Monitoring the data processing activities requires integration of information provided by services in the grid middleware layer, the DQ2 data management system, and the PANDA workload management system. The dashboard system [128] is designed to be the common framework for retrieving, processing, integrating and displaying the information for various operational purposes. Based on the dashboard system, a collection of monitoring pages [129] has been developed in the context of the ATLAS distributed computing.

Given the latency of collecting and processing distributed information for system monitor, before an issue appears on the monitor, it may have already caused massive failures. In addition, excluding/including sites and services for being used by the data processing is a routing work that can be automated to reduce manual interventions from the shifters and on-call experts. Thus, functional tests are introduced to run in the background of the actual data processing activities, providing early detection.
of failures due to malfunctioning sites in particular. The tests cover a range from as basic as transferring a dummy file to as advance as running a data processing job at a site. By assessing regularly the “baseline” services provided by sites, issues detected by the functional tests are taken into account by an automatic mechanism to exclude malfunctioning sites from being used by the actual data processing activities. A notification is also sent automatically to the responsible operation units to draw their attentions on the observed issues. Later on, when excluded sites pass again the tests, they are automatically included again for the data processing. The HammerCloud framework [130] is developed for generating the tests and reporting problematic sites.

4.3 Performance and efficiency

The ATLAS offline data processing has been running in production mode on the WLCG since the first ATLAS data taking in 2009. The success has already been shown by its capability and efficiency in delivering data for various physics researches to take place, and proven to be essential for the research of this thesis.

Based on the bookkeeping and accounting information collected during years of operation, we will demonstrate the performance and efficiency that has been achieved in managing data and workloads of the offline data processing.

4.3.1 Data management

Figure 4.7(a) demonstrates that ATLAS is capable to produce and manage data in petabyte scale. A notable increase on the data production rate around March 2010 corresponds to the first data taking on collisions at 7 TeV. With a continuous LHC operation from 2010 to 2012, the data grows in a steady rate of 30-40 petabytes per year. By the end of 2012, the total amount of ATLAS data on the grid storages has been accumulated to more than 120 petabytes.

Concerning the number of datasets, Figure 4.7(b) shows that there are more than 5 million datasets managed by the DQ2 system. The number of datasets is in general increasing over the time; nevertheless, the fluctuations show that datasets are also removed regularly, in particular a more intensive dataset deletion can take place at the beginning of each year to make room for the new data to be produced in the coming year.

The monthly average of the data transfer throughput as well as the success rate between March 2009 and the end of 2011 are shown in Figure 4.8. The throughput is calculated as total amount of data being transferred in a month divided by the length of the month; while the success rate is defined as the ratio between successful and total transfers in a month. The transfer throughput before March 2010 was relatively lower
4.3. Performance and efficiency

![Figure 4.7: Monthly evolution of the total amount of the ATLAS data on the grid storages in terms of (a) size in petabytes and (b) number of datasets. Both the size and number of datasets include dataset replicas.](image)

as the data was produced in a lower rate with the 900 GeV collision energy. With the site exclusion mechanism implemented in early 2010 to prevent data transfers to malfunctioning grid sites, the data transfer success rate was improved from 80% to 90%.

![Figure 4.8: Monthly evolution of the throughput and success rate of the ATLAS data transfer activities from March 2009 to December 2011.](image)

Starting from the 7 TeV data taking in March 2010, the ATLAS data management has been running a steady data transfer rate of 2 gigabytes per second over years with an
overall efficiency of 90%. In certain occasions when data are more intensively produced, one can see the system is capable to achieve a throughput up to 4 gigabytes per second without the degradation on the efficiency. The significant drop of success rate in November and December 2010 was caused by the loads from both an intensive event re-reconstruction on full 2010 data and the first data taking on heavy-ion collisions (as one can see the throughput was pushed up to 5 gigabytes per second), and an unexpected power cut at CERN in December.

Data transfer can fail due to transient issues such as temporary high load on the SE or persistent issues such as disk failure. Failed transfers are retried by the FTS. For transient issues, they usually succeed with the following attempts. For those not managing to succeed with the retry mechanism, operational efforts are involved to investigate the main cause and resolve the issues.

### 4.3.2 Workload management

The ATLAS offline data processing has achieved a remarkable scale. The accounting in Figure 4.9 shows that the data processing in 2011 has been carried out by 78 million grid jobs consuming more than 46 thousands years of wall-clock time for the computation.

![Pie chart showing the scale of ATLAS offline data processing activities on the WLCG](image)

**Figure 4.9:** The scale of the ATLAS offline data processing activities on the WLCG in terms of (a) total number of completed grid jobs and (b) the accumulated wall-clock time consumption in year 2011.

With the responsibility of simulating the interactions between particles and the complex ATLAS detector, the simulation is not surprisingly the most time consuming process among all activities. The event reconstruction and group production also utilize a significant amount of computing power. In addition, one notes from the figure a special type of “output merging” process taking less than 1% computing power. It is the process responsible for merging small output files into larger ones, recognizing
that managing and accessing larger files on the grid storage is more efficient and cost-effective.

During the workload execution, there is an overhead for staging input/output data from grid storage to the computer node. The averaged CPU utilization fractions of the grid jobs shown in Figure 4.10 reflect to this I/O overhead concerning different data processing activities. The CPU utilization fraction of a grid job is calculated as the fraction of the CPU time consumption over the elapsed time of the job. As expected, the simulation is the most CPU intensive process. For the event reconstruction and group production, the CPU utilization fraction reaches more than 80%. As an I/O intensive process, the output merging have significant I/O overhead and utilize just 35% of the CPU.

![Figure 4.10: CPU utilization fraction of the grid jobs corresponding to different data processing activities. All completed jobs shown in Figure 4.9(a) are taken into account.](image)

Figure 4.11 shows that the success rate of the grid jobs is in general higher than 90% as the workloads are centrally prepared and tested carefully before they are launched in scale. By analyzing the error codes collected by the PANDA system, job failures are categorized in Figure 4.12.

The largest category of “input/output error” indicates that jobs failed mostly during the input and output staging. This type of failure usually has connection with issues on the SE. It can be due to high load on the SE, issues with the network communication between the SE and the computer node, or hardware problems of the SE resulting in missing or corrupted input files.

The second largest type of failure is caused by the “runaway jobs”. Runaway jobs are those killed either by the PANDA system when they are running longer than expected or by the batch system behind the CE as their resource usage exceeds the limitation. Hardware problems on the computer node can cause the workload execution to idle, resulting in long running jobs. Jobs can run over the resource limitation because they require longer wall-clock time or more memory to process certain data events with more complex structure in it, or due to a bug in the data processing software. In
Chapter 4. Offline Data Processing

![Bar chart](image)

**Figure 4.11:** Fractions of grid jobs in different status corresponding to different data processing activities. All completed jobs shown in Figure 4.9(a) are taken into account.

![Pie chart](image)

**Figure 4.12:** Categorization of grid job failures based on the error code and job logs in the PANDA system. All failed jobs in Figure 4.11 are taken into account. See text for the explanation on the categories.

In addition, jobs losing contact with the PANDA system due to network issues are also counted into this category.

Failures during the execution of the workload fall mostly into the following two categories. The “runtime failure” considers those with error codes recognized by the system; others are counted into the “undocumented error”. Both types of failures are somehow related to the configurations of the data processing algorithms and hence, they require careful inspections from the operation and the algorithm developers.

The type of “software missing” sums up the failures in loading necessary software release or libraries for executing the workload, while the workloads not being processed more than 3 days after their creation are counted as the “expired jobs”. “DDM failure” refers
4.4 Summary

to the case when jobs fail to cooperate with the DQ2 system for data management such as replicating output datasets to a target SE. Rarely happened problems are categorized as “misc”.

Job failures due to transient issues can be recovered automatically by the retry mechanism built in the PANDA system. For the rest of failures, operational efforts are required to coordinate necessary fix to cure the problem.

Conclusion and outlook

With the developed system, the offline data processing has been performed in large scale using the grid technology. Massive data transfers and workload executions are managed with a 90% success rate, though operational efforts are required to overcome the remaining 10% of failures. Due to the dynamics of the grid and the complex of the system, some issues are unfortunately unavoidable. Lessons learned from the operation [131] provide valuable input for developing advanced services in the system to work around those issues efficiently without the need for manual intervention.

Although the offline data processing has been successfully delivering data in time for physics analysis, new computing challenges are waiting ahead when the LHC will be restarted in 2015. Not only the machine will reach its design capacity, the trigger rate of the ATLAS detector is also targeted at 1 kHz (about 2 times faster than data taking in 2011 and 2012) to give big advance to physics studies. Therefore, the amount of computing resources required for the offline data processing is foreseen to be significantly larger than it is today [132]. Apart from requesting more resources from the collaborating grid sites, new systems aiming for better resource utilization in the data and workload managements are under development. Possibilities of exploiting additional computing resources provided by, for example, the super computer or the cloud computing system (see Chapter 5), are also of interest. Various R&D activities have been conducted in the context of the ATLAS computing to investigate the use of those resources for the offline data processing in the future.

4.4 Summary

The offline data processing is a necessary step to prepare data for all physics analyses of the ATLAS experiment. Given the high collision rate of the LHC and the remarkable efficiency of the ATLAS detector, it requires a world-wide collaboration to tackle the computational challenges arising from the massive amount of data to be handled.

How the distributed computing resources are integrated for the LHC computing were introduced. The integration results in the WLCG system in which the middleware layer provides necessary protocols and interfaces allowing applications to access distributed resources seamlessly as a single infrastructure.
We also discussed the data and workload management systems developed specifically for the ATLAS experiment in order to achieve an ultimate goal of the offline data processing: delivering data in time for physicists world-wide to carry on for researches and discoveries.

The goal, as demonstrated, has been achieved successfully given the fact that a remarkable number of physics studies has been carried out within the ATLAS collaboration since 2009. Without the achievement, the Higgs boson would not have been found and our search for single top quark production would not be possible. Building on this achievement, new developments are taking place now in the ATLAS computing, aiming for addressing new challenges in 2015.