ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Scuola di Scienze
Corso di Laurea in Fisica

Investigation of Petabyte-scale data transfer performances with PhEDEx for the CMS experiment

Relatore:
Prof. Daniele Bonacorsi

Presentata da:
Tommaso Diotalevi

Sessione II
Anno Accademico 2014/2015

Quo usque tandem abutere, Catilina, patientia nostra?
[Cicerone, Oratio ad Catilinam I]

Contents

Sommario

PhEDEx, il sistema di gestione dei trasferimenti di CMS, durante il primo Run di LHC ha trasferito all'incirca 150 PB ed attualmente trasferisce circa 2.5 PB di dati alla settimana attraverso la Worldwide LHC Computing Grid (WLCG). Questo sistema è stato progettato per completare ogni trasferimento richiesto dall'utente a spese del tempo necessario per il suo completamento. Dopo svariati anni di operazioni con tale strumento, sono stati raccolti dati relativi alle latenze di trasferimento ed immagazzinati in log files contenenti informazioni utili per l'analisi. A questo punto, partendo dall'analisi di una ampia mole di trasferimenti in CMS, è stata effettuata una suddivisione di queste latenze ponendo particolare attenzione nei confronti dei fattori che contribuiscono al tempo di completamento del trasferimento.

L'analisi presentata in questa tesi permetterà di equipaggiare PhEDEx con un insieme di utili strumenti in modo tale da identificare proattivamente queste latenze e adottare le opportune tattiche per minimizzare l'impatto sugli utenti finali.

Il Capitolo 1 fornisce una panoramica di LHC, con maggiore attenzione all'esperimento CMS.

Il Capitolo 2 descrive le basi del CMS Computing Model, il sistema di gestione dei dati e le relative infrastrutture.

Il Capitolo 3 offre una visione d'insieme sul problema delle latenze di trasferimento e ne descrive una suddivisione di diverse categorie.

Il Capitolo 4 analizza i dati prodotti dal sistema di monitoraggio di PhEDEx.

Abstract

PhEDEx, the CMS transfer management system, during the first LHC Run has moved about 150 PB and currently it is moving about 2.5 PB of data per week over the Worldwide LHC Computing Grid (WLGC). It was designed to complete each transfer required by users at the expense of the waiting time necessary for its completion. For this reason, after several years of operations, data regarding transfer latencies has been collected and stored into log files containing useful analyzable informations. Then, starting from the analysis of several typical CMS transfer workflows, a categorization of such latencies has been made with a focus on the different factors that contribute to the transfer completion time. The analysis presented in this thesis will provide the necessary information for equipping PhEDEx in the future with a set of new tools in order to proactively identify and fix any latency issues.

Chapter 1 provides a global view of CMS experiment at LHC.

Chapter 2 describes the basics of the CMS computing Model, with a focus on the data management system and related infrastructures.

Chapter 3 offers an overview of the transfer latency issue in CMS computing operations and describes a categorization of such latencies in different categories.

Chapter 4 gives an analysis of latency data produced by the PhEDEx monitoring system.

Chapter 1

High energy physics at LHC

1.1 A general view of the Large Hadron Collider

The Large Hadron Collider (LHC) [1, 2] is part of the CERN accelerator complex (see Figure 1.1) [3], in Geneva, and it is the most powerful particle accelerator ever built. The particle beam is injected and accelerated by each element of a chain of accelerators, with a progressive increase of energy until the beam injection into LHC, where particles are accelerated up to 13 TeV (by LHC design, its nominal center-of-mass energy).

The LHC basically consists of a circular 27 km circumference ring, divided into eight independent sectors, designed to accelerate protons and heavy ions. These particles travel on two separated beams on opposite directions and in extreme vacuum conditions (see Section 1.1.1). Beams are controlled by superconductive electromagnets (see Section 1.1.2) , keeping them in their trajectory and bringing them to regime.

Some of the main LHC parameters are shown in Table 1.1.

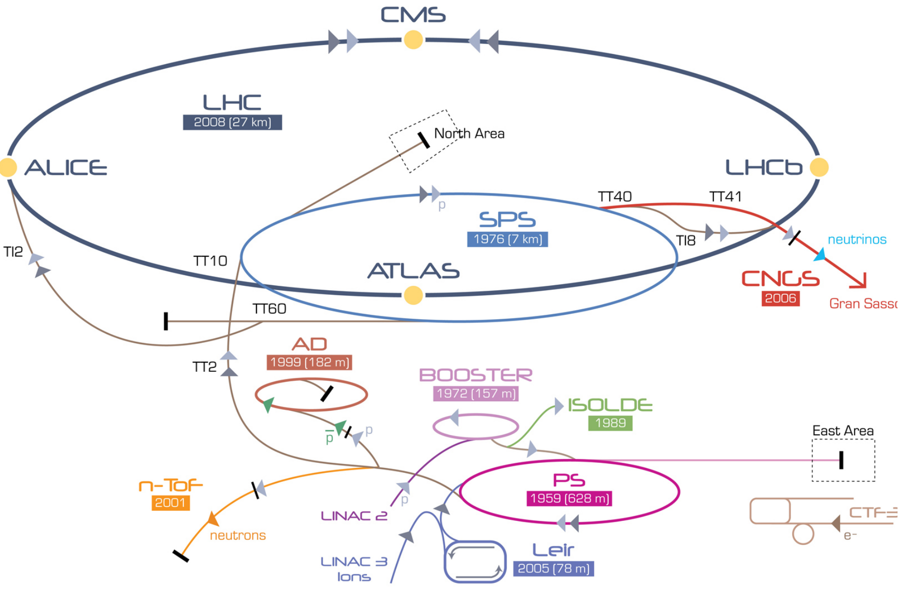


Figure 1.1: The accelerators chain at CERN.

Table 1.1: Main technical parameters of LHC.

Quantity	**value**
Circumference (m)	26 659
Magnets working temperature (K)	1.9
Number of magnets	9593
Number of principal dipoles	1232
Number of principal quadrupoles	392
Number of radio-frequency cavities per beam	8
Nominal energy, protons (TeV)	7
Nominal energy, ions (TeV/nucleon)	2.76
Magnetic field maximum intensity (T)	8.33
Project luminosity ($\mathrm{cm^{-2}\,s^{-1}}$)	10×10^{34}
Number of proton packages per beam	2808
Number of proton per package (outgoing)	1.1×10^{11}
Minimum distance between packages (m)	~7
Number of rotations per second	11 245
Number of collisions per second (millions)	600

1.1.1 The vacuum system

The LHC vacuum system [4] is, with more than 104 kilometers of vacuum ducts, one of the most advanced in the world. Basically it has two functions: the first is to avoid collisions between beam particles and air molecules inside the ducts, recreating an extreme vacuum condition (10^{-13}atm), as empty as interstellar space; the other reason is to cancel heat exchange between components who needs low temperatures in order to work properly and maximize the efficiency.
The vacuum system is made out of three independent parts:

- an isolated vacuum system for cryomagnets;
- an isolated vacuum system for Helium distribution line;
- a vacuum system for beams.

1.1.2 Electromagnets

Electromagnets [5] are designed to guide beams along their path, modifying single particles trajectories as well as align them in order to increase collision probability. There are more than fifty different kind of magnets in LHC, amounting to approximately 9600 magnets. Main dipoles, the bigger ones, generate a magnetic field with a maximum intensity of 8.3 T. Electromagnets require a current of 11 850 A in conditions of superconductivity in order to reset energy losses due to resistance. This happens thanks to a system of liquid helium distribution that keeps magnets at a temperature of about 1.9 K. At this incredibly low temperatures, below that required to operate in conditions of superconductivity, helium became also superfluid: this means an high thermal conductivity used consequently as a refrigeration system for magnets.

1.1.3 Radiofrequency cavities

Radiofrequecy cavities [1] [2] [6] are metallic chambers in which electromagnetic field is applied. Their primary purpose is to separate protons in packages and to focus them at the collision point, in order to guarantee an high luminosity and thus a large number of collisions.
Particles, passing through the cavity, feel the overall force due to electromagnetic fields and push them forwards along the accelerator. In this scenario, the ideally timed proton, with exactly the right energy, will see zero accelerating voltage when LHC is at nominal energy while protons with slightly different energies will be accelerated or decelerated sorting particle beams into "bunches". LHC has eight cavities per beam: each of which furnishes 2 MV at 400 MHz. The cavities work at 4.5 K and are grouped into four cryo-modules.
At regime conditions, each proton beam is divided into 2808 bunches, each containing about 10^{11} protons. Away from the collision point, the bunches are a few cm long and 1 mm wide, and are compressed up to 16 nm near the latter. At full luminosity, packages are separated in time by 25 ns, thus resulting in about 600 million collisions per second.

1.2 LHC detectors

Along the LHC circumference, the particles collide in four beam intersection points, in which the four main LHC experiments are built. Each experiment has its own detector, designed and built to gather the fragments of the large number of collisions and reconstruct all physical processes that generated them.
In particular, the four major experiments installed at LHC are:

- A Large Ion Collider Experiment (ALICE)
- A Toroidal LHC ApparatuS (ATLAS)
- Compact Muon Solenoid (CMS)
- Large Hadron Collider beauty (LHCb)

In addition, there are secondary experiments, among which:

- Large Hadron Collider forward (LHCf)
- TOTal Elastic and diffractive cross section Measurement (TOTEM)

In the following sections, the LHC detectors are briefly introduced, with a major focus on the CMS experiment.

1.2.1 ALICE

ALICE [7, 8] is a detector specialized in heavy ions collisions. It is designed to study the physics of strongly interacting matter at extreme energy densities, where

a phase of matter called "quark-gluon plasma" forms. At these conditions, similar to those just after the Big Bang, quark confinement no longer applies: studying the quark-gluon plasma as it expands and cools allows to gain insight on the origin of the Universe. Some ALICE specifications are illustrated in Table 1.2.
The collaboration counts more than 1000 scientists from over 100 physics institutes in 30 countries (updated to October 2014)

Table 1.2: ALICE detector specifications:

Dimensions	length: 26 m, height: 16 m, width: 16 m
Weight	10 000 tons
Design	central barrel plus single arm forward muon spectometer
Cost of materials	115 MCHF
Location	St. Genis-Pouilly, France

1.2.2 ATLAS

ATLAS [9, 10], is one of the two general-purpose detectors at LHC. Although its similarities with the CMS experiment regarding scientific goals, they have subdetectors based on different technology choices, and the design of the magnets is also different. Some specs are illustrated below in Table 1.3.
It is located in a cavern 100m underground near the main CERN site. About 3000 scientists from 174 institutes in 38 countries work on the ATLAS experiment (updated to February 2012).

Table 1.3: ATLAS detector specifications:

Dimensions	length: 46 m, height: 25 m, width: 25 m
Weight	7000 tons
Design	barrel plus andcaps
Cost of materials	540 MCHF
Location	Meyrin, Switzerland

1.2.3 CMS

CMS [11, 12], as well as ATLAS, is a general-purpose detector at LHC. Is built around a huge solenoid magnet with a cylindrical form able to reach a 4 T magnetic field. Its main characteristics are illustrated in Table 1.4. In the next chapter, CMS will be discussed more specifically.

Table 1.4: CMS detector specifications:

Dimensions	length: 21 m, height: 15 m, width: 15 m
weight	12 500 tons
Design	barrel plus end caps
Cost of materials	500 MCHF
Location	Cessy, France

1.2.4 LHCb

The LHCb [13, 14] experiment is specialized in investigating the slight differences between matter and antimatter by studying the quark bottom. Instead of ATLAS or CMS, LHCb uses a series of subdetectors to detect mainly forward particles: the first one is mounted near the collision point while the others are placed serially over a length of 20 meters.

Some specifications are illustrated below in Table 1.5. About 700 scientists from 66 different institutes and universities work on LHCb experiment (updated to October 2013).

Table 1.5: LHCb detector specifications:

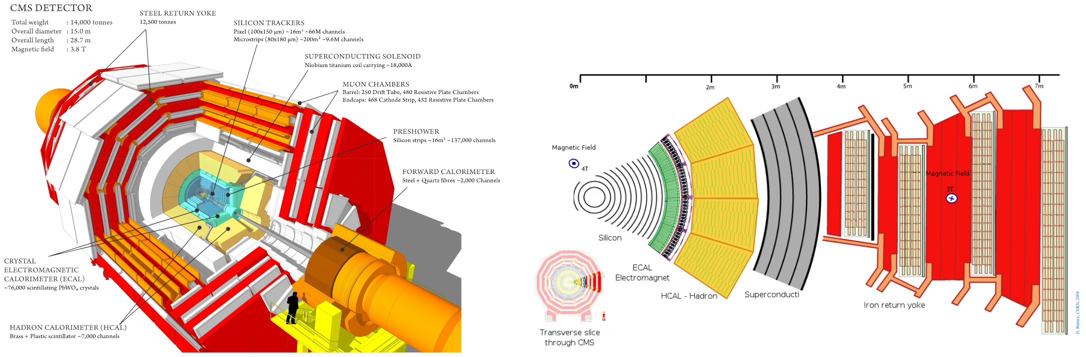
Dimensions	length: 21 m, height: 10 m, width: 13 m
Height	5600 tons
Design	forward spectometer with planar detectors
Cost of materials	75 MCHF
Location	Ferney-Voltaire, France

1.2.5 Other LHC experiments

Aside from ALICE, ATLAS, CMS and LHCb, a few details on LHC smaller experiments, LHCf and TOTEM, are given in the following. LHCf [15, 16] is a small experiment which uses particles thrown forward by p-p collisions as a source to simulate high energy cosmic rays. LHCf is made up of two detectors which sit along the LHC beamline, at 140 m either side of ATLAS collision point. They only weights 40 kg and measures (30 x 80 x 10) cm.

LHCf experiment involves about 30 scientists from 9 institutes in 5 countries (updated to November 2012).

TOTEM [17, 18] experiment is designed to explore protons cross-section as they emerge from collisions at small angles. Detectors are spread across half a kilometre around the CMS interaction point in special vacuum chambers called "roman pots" connected to beam ducts, in order to reveal particles produced during the collision. TOTEM has almost 3000 kg of equipment and 26 "roman pot" detectors. It involves about 100 scientists from 16 institutes in 8 countries (updated to August 2014)


CMS DETECTOR
Total weight : 14,000 tonnes
Overall diameter : 15.0 m
Overall length : 28.7 m
Magnetic field : 3.8 T
STEEL RETURN YOKE
12,500 tonnes
SILICON TRACKERS
Pixel (100x150 μm) ~16m² ~66M channels
Microstrips (80x180 μm) ~200m² ~9.6M channels
SUPERCONDUCTING SOLENOID
Niobium titanium coil carrying ~18,000A
MUON CHAMBERS
Barrel: 250 Drift Tube, 480 Resistive Plate Chambers
Endcaps: 468 Cathode Strip, 432 Resistive Plate Chambers
PRESHOWER
Silicon strips ~16m² ~137,000 channels
FORWARD CALORIMETER
Steel + Quartz fibres ~2,000 Channels
CRYSTAL ELECTROMAGNETIC CALORIMETER (ECAL)
~76,000 scintillating PbWO₄ crystals
HADRON CALORIMETER (HCAL)
Brass + Plastic scintillator ~7,000 channels
0m
2m
3m
4m
5m
6m
7m
Magnetic Field
4T
Silicon
ECAL
Electromagnet
HCAL - Hadron
Superconducti
Iron return yoke
Magnetic Field
2T
Transverse slice through CMS
D. Barney, CERN, 2004

(a) Section view.

(b) View along the plane perpendicular to beam direction.

Figure 1.2: Compact Muon Solenoid

1.3 The CMS experiment

The CMS main purpose is to explore the p-p physics at the TeV scale, including precision measurements of the Standard Model, as well as search for new physics. Its cylindrical concept is built on several layers and each one of them is dedicated to the detection of a specific type of particle.
About 4300 people including physicists, engineers, technicians and students work actively on CMS experiment (updated to February 2014)

1.3.1 The CMS detector: concept and structure

As mentioned above, the CMS detector is made up of different layers, as illustrated in Figure 1.2. Each of them is designed to trace and measure the physical properties and paths of different kinds of subatomic particles. Furthermore, this structure is surrounded by a huge solenoid based on superconductive technologies, operating at 4.4 K and generating a 4 T magnetic field.
The first and inner layer of the CMS detector is called Tracker [12, pp. 26-89]: made entirely of silicon, is able to reconstruct the paths of high-energy muons, electrons and hadrons as well as observe tracks coming from the decay of very short-lived particles with a resolution of 10 nm.
The second layer consists of two calorimeters, the Electromagnetic Calorimeter (ECAL) [12, pp. 90-121] and the Hadron Calorimeter (HCAL) [12, pp. 122-155] arranged serially. The former measures the energy deposited by photons and electrons, while the latter measures the energy deposited by hadrons.
Unlike the Tracker, which does not interfere with passing particles, the calorimeters are designed to decelerate and stop them.
In the end, a superconductive coil alternated with muon detectors [12, pp. 162-246] are able to track muon particles, escaped from calorimeters. The lack of energy and momentum from collisions is assigned to the electrically neutral and weakly-interacting neutrinos.

Tracker

The Tracker is a crucial component in the CMS design as it measures particles momentum through their path: the greater is their curvature radius across the magnetic field, the greater is their momentum. As stated above, the Tracker is able to reconstruct muons, electrons and hadrons path as well as tracks produced by short-lived particles decay, such as quark beauty. It has a low degree of interference with particles and a high resistance to radiations. Located in the inner part of the detector, it receives the greatest amount of particles. Interference with particles occur only in few areas of the Tracker with a resolution of about 10 nm.

This layer (Figure 1.3a) is entirely made of silicon: internally there are three levels of pixel detectors, after that particles pass through 10 layer of strip detectors, until a 130 cm radius from the beam pipe.

The pixel detector, Figure 1.3b, contains about 65 millions of pixels, with three levels of respectively 4, 7 e 11 cm radius. The flux of particles at this radius is maximum: at 8 cm is about 10^6 particles/cm·sec. Each level is divided into small units, each one containing a silicon sensor of 150 nm × 150 nm. When a charged particle goes through one of this units, the amount of energy releases an electron with the consequent creation of an hole. This signal is than received by a chip which amplifies it. It is possible, in the end, to reconstruct a 3-D image using bi-dimensional layers for each level.

The power absorption must remain at minimum, because each pixel absorbs about 50 µW with an amount of power not irrelevant; for this reason pixels are installed in low temperature pipes.

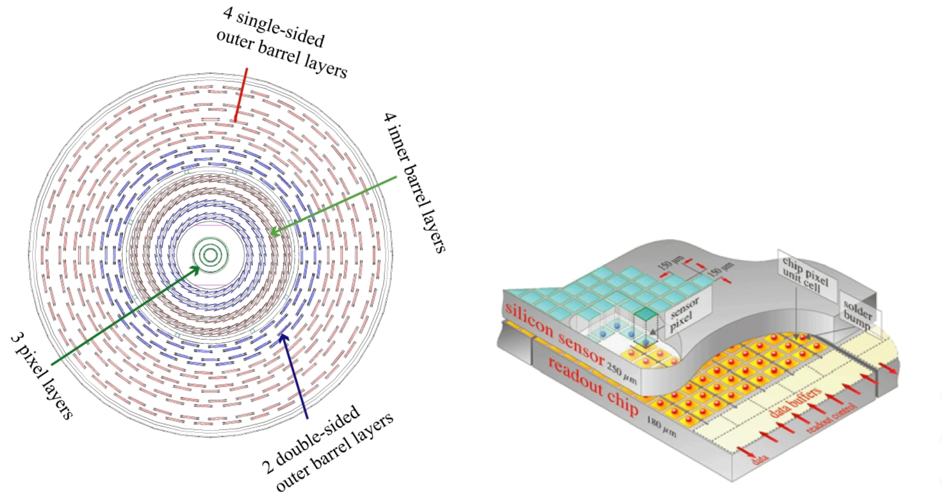
Strip detectors, instead, consist of ten levels divided into four internal barriers and six external barriers. This section of the Tracker contains 10 million detector strips divided into 15 200 modules, scanned by 80 000 chips. Each module is made up of three elements: a set of sensors, a support structure and the electronics necessary to acquire data. Sensors have an high response and a good spacial resolution, allowing to receive many particles in a restricted space; they simply detect electrical currents generated by interacting particles and send collected data. Also this section of the detector is maintained at low temperature (−20 °C), in order to "isolate" silicon structure damages due to radiations.

Calorimeters

In the CMS experiment there are two types of calorimeters that measure the energy of electrons, photons and hadrons.

Electrons and protons are detected and stopped by the electromagnetic calorimeter (ECAL). This measurement happens inside a strong magnetic field, with an high level of radiation and in 25 ns from one collision to another. This calorimeter is built with lead tungstate $PbWO_4$, and it is able to produce light in relation to particles energy. This material is essentially an high density crystal so that light production occur quickly and in a well defined manner allowing a rapid, precise and very effective detection thanks to special photo-detectors designed to work in high magnetic field condition.

The ECAL is divided into a cylindrical body called "barrel" and the two ends called



(a) Tracker layers from a point of view perpendicular to beams.

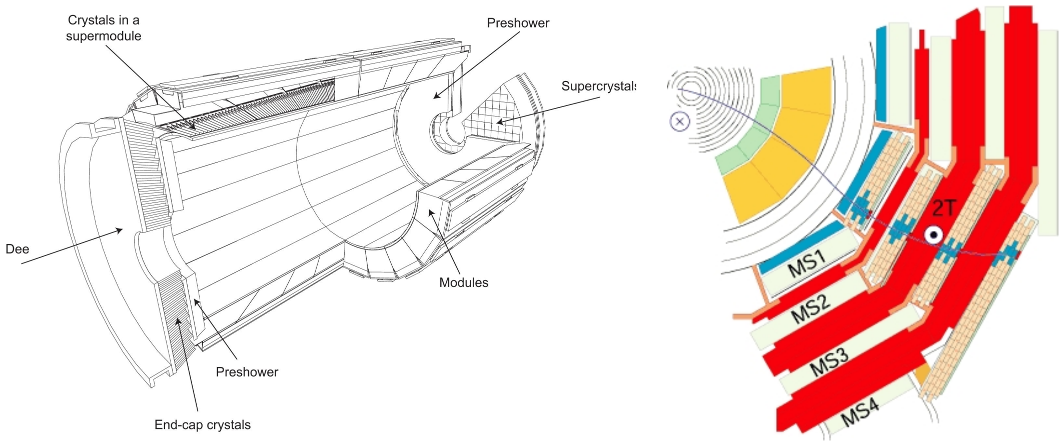
(b) CMS Silicon pixel detector.

Figure 1.3: Graphical depiction of the CMS Tracker subdetector.

"endcaps" (see Figure 1.4a) creating a layer between the Tracker and the other calorimeter.
Hadrons are instead detected by the hadron calorimeter (HCAL) specifically built for measuring strong-interacting particles; it also provides tools for an indirect measurement of non-interacting particles like neutrinos. During hadrons' decay, new particles can be produced which may leave no sign to any detector at CMS. To avoid that, HCAL is hermetic i.e. it captures each particle coming from the collision point, allowing detection of "invisible" particles, through the violation of momentum and energy conservation. The hadron calorimeter is made up with a series of layer highly absorbent and each time a particle produced from various decays crosses a layer, a blue-violet light is emitted. This light is than absorbed by optic cables of about 1 mm diameter, shifting wavelength into the green region of the electromagnetic spectrum and finally converted to digital data. The optic sum of light produced along the path of the particle it's a measure of its own energy, and is performed by specifically designed photo-diodes. Unlike ECAL, HCAL has two more sections called "forward sections", located at the ends of CMS, designed to detect particles moving with a low scattering angle. These sections are built with different materials, with the aim of making them more resistant to radiations since they receive much of the beam energy.

Muons detector

Muons detector are placed outside the solenoid because muons are highly penetrant and may not be stopped from the previous calorimeters. Their momentum is measured by recreating trajectories along four muons stations interspersed with ferromagnets ("return yoke"), shown in red in Figure 1.4b, and synchronizing data with those obtained from the Tracker. There are 1400 muons chambers: 250 "drift tubes" (DTs) and 540 "cathode strip chambers" (CSCs) tracing particles positions and acting at the same time as a trigger, while 610 "resistive plate chambers" (RPCs)



(a) ECAL layout showing crystal's modules.

(b) A muon, in the plane perpendicular to beams, leaving a curve trajectory into the four muon stations.

Figure 1.4: The *Electromagnetic CALorimeter* and the *muons detectors*.

form a further trigger system which decides quickly if save or delete acquired data.

1.3.2 Trigger and Data Acquisition

When CMS is at regime, there are about a billion interactions *p*-*p* each second inside the detector. The time lapse from one collision to the next one is just 25 ns so data are stored in pipelines able to withhold and process information coming from simultaneous interactions. To avoid confusion, the detector is designed with an excellent temporal resolution and a great signal synchronization from different channels (about 1 million).
The trigger system [12, pp. 247-282] is organized in two levels. The first one is hardware-based, completely automated and extremely rapid in data selection. It selects physical interesting data, e.g. an high value of energy or an unusual combination of interacting particles. This trigger acts asynchronously in the signal reception phase and reduces acquired data up to a few hundreds of events per second. Subsequently they are stored in special servers for later analysis.
Next layer is software-based, acting after the reconstruction of events and analyzing them in a related farm, where data are processed and come out with a frequency of about 100 Hz.

Chapter 2

The CMS Computing Model

The collisions between protons (or heavy ions) are called "events" and they constitute the base granularity of the computing models of each LHC detector. The collision rate *p*-*p* (about 10^9 Hz) is approximately equivalent to an amount of 1 PB per second of RAW data. This huge stream of data is initially skimmed through a trigger system, as described in Section 1.3.2, which reduces this amount down to a few hundreds of megabytes per second. At this rate, storage systems at the *CERN Computing Center* are able to save and archive this data stream.
Including physics simulations and data directly from detectors, LHC handles about 15 PB of data each year. Beyond that, each user of LHC (from different nations) must exploit this data, without being *physically* at CERN [19]. Requirements for this kind of computing system are:

- to handle an huge amount of data effectively;
- to allow access to data for thousands of users all around the world;
- to have enough resources for storage and treatment of *RAW data* (namely, CPU e storage);

These challenges apply to all stages of data handling, i.e. RAW data collection and archival, but also scheduled processing activities like production of simulated data with Monte Carlo techniques, data reprocessing, etc. Moreover the computing environment must be able to handle analysis processes, in the so called chaotic user analysis.
To address these challenges, the Worldwide LHC Computing Grid [20] (WLCG) was launched: each experiment has its own layer of applications, that sits on top of a middleware layer, provided by Grid projects in Asia, Europe and America (EGEE [21], EGI [22], OSG [23]).
Each experiment must adopt a proper Computing Model that consists of hardware and software resources, and mechanisms to make them work together coherently, in order to allow harvesting, distribution and analysis of this huge amount of data and the management of interactions between these components in real time. In addition to middleware projects mentioned above, each HEP experiment develops its own software designed to perform specific functions of that particular experiment (the so-called "application layer"). These applications must act coherently (and together

with other experiments or Virtual Organizations) in computing centers all around the world.
The CMS *Computing Model* [24, 25], of particular interest within this thesis, is described in the following sections, with particular attention to computing resources and a particular sector of the overall model: the Data Management.

2.1 CMS Computing Resources

Computing infrastructures through which Grid services operate are hierarchically classified into *Tiers* according to the MONARC model [26]: they are computing centers with storage capacity, CPU power and network connectivity that run different set of services for the LHC experiments (thus also resulting in different computing capacity). Each Tier is associated with a number: the lower the number, the greater its variety and demand in terms of storage, CPU and network. Additionally, also the required availability of the Tier is greater as the number decreases (starting with 24h/7 with Tier-0 and Tier-1, and going to 8h/5 at the Tier-2 level). This particular classification of Tiers is such that - depending on the national resources and strategic choices - a Tier-1 may support only one or more LHC experiments (e.g. the US Fermilab Tier-1 supports only the CMS experiment, while the Italian INFN-CNAF Tier-1 supports all four LHC experiments). Even the case of a center offering Tier-1 and Tier-2 functionality to a given experiment is allowed in the model (e.g. the France IN2P3 hosts a Tier-1 and a Tier-2 for CMS).
The CMS computing model on WLCG exploits the following resources:

- 1 Tier-0 Centre at CERN (T0);
- CMS Analysis Facility at CERN (CMS-CAF);
- 8 Tier-1 (T1), considering also the T1 functions conducted at CERN;
- 52 Tier-2 (T2) among which, only a smaller fraction - about 40-45 - are operational on a stable basis at all times (i.e. not in downtime or affected by hardware or software issues).

It is worth noting that in WLCG also the concept of Tier-3 (T3) exist: nevertheless, such centers do not sign any *Memorandum of Understanding* and they not guarantee any level of services. Usually Tier-3 structures are dedicated entirelly to user analysis and support for local communities of physicists. The majority of users mostly rely on T2s and T3s resources for distributed analysis, while T1s and the T0 are mainly dedicated to "scheduled" processing activities.

2.1.1 CMS Tiers

In the previous sections we introduced the WLCG Tier levels. Now we present the specific roles of CMS Tiers.

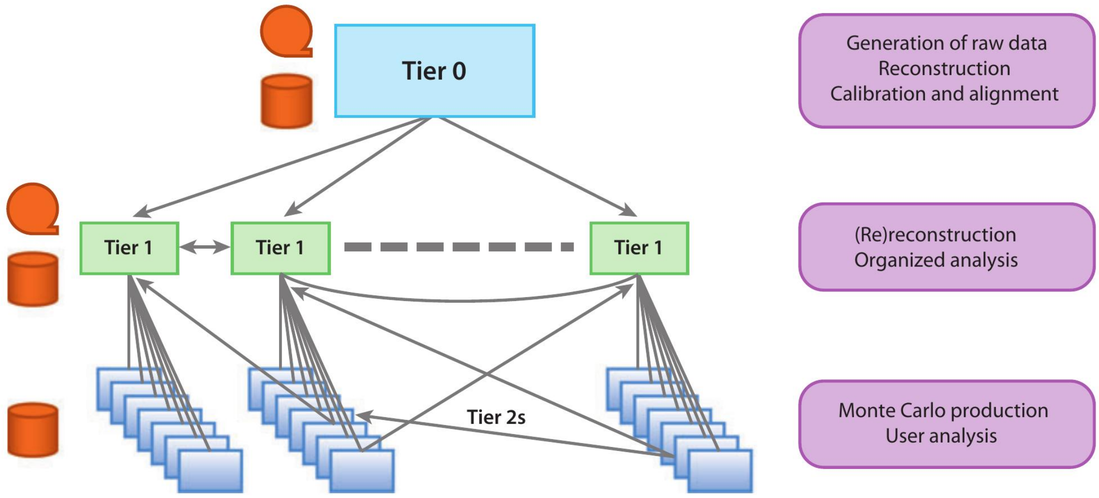


Figure 2.1: Exemplified Tiers structure of the CMS computing model [19].

Tier-0 and CMS-CAF

The Tier-0 and the CMS-CERN Analysis Facility (CAF) are located at CERN. Since 2012, the T0 computing capacity at CERN has been extended with the Wigner Research Center for Physics at Budapest: apart from the augmented resources, the CERN+Wigner solution offers a greater availability of T0 services in case of technical problem on one side of the infrastructure. The main role of Tier-0 for CMS is to receive RAW data from the CMS detector and store them on tape, to sort them into data streams and to start a first ("prompt") reconstruction of events. The CMS T0 hence classifies these data into 50 primary datasets and makes them available to be transferred to T1s. The latter is called the *hot copy*, while a backup *cold copy* is always stored at CERN. From T0 to T1, the data transfer throughput is fundamental: for this purpose a network infrastructure on optical fiber has been built, the *LHC Optical Private Network* (LHCOPN) [27], allowing performances as high as 120 Gbps.
The CMS-CAF provides support to low latencies activities and an asynchronous rapid access to RAW data , such as detector diagnostic, trigger performance services, calibrations etc...

Tier-1

Eight Tier-1 are located around the world (CERN, Germany, Italy, France, Spain, England, USA and Russia). The T1s perform several crucial roles: they accept data from the T0, they transfer such data to the T2s upon need, they offer custodiality of large volume of this data on tapes, they perform data re-reconstrucion, Monte Carlo simulations, etc. To perform such functions, the T1s must be equipped with remarkable CPU power, large volume of performant disk storage, an ample tape capacity and excellent network connections not only with the CERN T0 but also with T2s (typically more than 10 Gbps).

Tier-2

The 52 Tier-2 are 50%-50% devoted to Monte Carlo production and data analysis. To perform such functions, the T2s do not need tape space but excel in high-performance disk storage (used as caches for most interesting data for analysis) and CPU power, plus good network connections with the T1s (10 Gbps or more) and also with other centers, such other T2s and T3s worldwide.

2.2 CMS Data Model and Simulation Model

Data reconstruction workflow in CMS, through different stages of processing, consists of transforming information from RAW data to physical interesting formats for analysis. Typically, both on "prompt" reconstruction at T0 and subsequent reprocessing of data over time at T1, a similar application is executed producing more outputs which are "skimmed" into specific data-sets containing interesting events for particular types of analysis. The final stage is therefore made of derived data, which contains all the useful information for analysts. Simulation reconstruction workflow instead is based on a few steps: first a kinematics on the Monte Carlo events generator, then a simulation of the detector's response to generated interactions and finally the reconstruction where the single interaction is combined with pile-up events, for a real simulation of bunch crossing. This last step, adding events from previous and next collisions, may require a few hundreds minimum-bias events, making it very challenging in terms of I/O computing resources.

In the CMS data model, the ultimate reference formats for physics analysts are called AOD and AODSIM for real data and simulated data respectively. More information on intermediate formats can be found in 2.1. It is worth noting that CMS is currently introducing a new data format, called MiniAOD, which physics-wide contains the same information of the AOD but whose size is smaller than AODs, thus optimizing the disk/tape space utilization at Tiers. As this format is not yet in production, it will not be discussed further.

Table 2.1: CMS acquired data main formats

RAW	Raw Data as they came out of the detector.
ESD or RECO	Event Summary Data, they contain all information after reconstruction, including RAW content.
AOD	Analysis Object Data, they contain the reconstructed information on the physics object that are mostly used during the analysis.
AODSIM	Simulated Analysis Object Data, same as above but for Monte Carlo simulations.

2.2.1 CMS data organization

RAW data from CMS, are processed into "groups of data" called *datasets*. They represent a coherent set defined by criteria applied during its processing and contain also information on its processing history. Their dimension can vary, usually in a range of 0.1-100 TB. The datasets are internally organized in "fileblocks", which in turn are made of "files" (a fileblock may have 10-1000 files). Despite the dataset is a clean logical unit for a physics user, and despite the file systems of course deal with files, the building block of the CMS data management system is the fileblock (e.g. data transfers are organized and monitored on a fileblock basis).

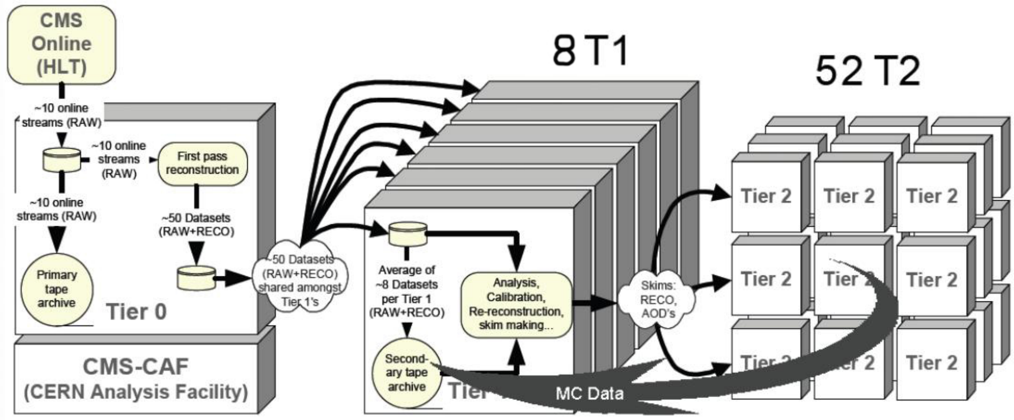


Figure 2.2: Data stream, MC and detector data, across Tiers [28, p. 13].

2.3 CMS computing services and operations

The original CMS Computing Model is *data-driven*: the jobs go where data is, and no data is moved in response to job submission. In the overall Model, two main areas can be identified: the first is the *CMS Workload Management System* [29], which takes care of everything related to the job submission and job management on the computing resources, and the second is the *CMS Data Management System* [30], which takes care of the data handling and access on the CMS resources. They are discussed in the following sections.
Regarding data location, instead, there isn't a central catalog for storing information of each file at CMS: the latter corresponds to a *Logical File Name* (LFN) and the existence or location of one file is known through a mapping (made and stored from grid *central services*) of this LFN with sites containing replicas of it. A further local mapping of LFN is carried by an internal catalog, the *Trivial File Catalog* (TFC), with the *Physical File Name* (PFN).

2.3.1 CMS Workload Management System

The *Workload Management System* (WMS) is based on grid middleware as well as on CMS-developed tools and solutions maintained by CMS. The WMS solutions take care of the job management as a whole, i.e. including both the scheduled processing (e.g. simulations) and the distributed analysis.

2.3.2 CMS Data Management System

The CMS Data Management System (DMS) is supported by both Grid and CMS services. Its purpose is to guarantee an infrastructure and tools for finding, accessing and transferring all kinds of data obtained at CMS. DMS main tasks are:

- preserve a *data bookkeeping catalog* which describes data content in physical terms;
- preserve a *data location catalog* which holds in memory location and number of data replicas;
- handle *data placement* and *data transfer*.

These tasks are executed by these principal components:

- **PhEDEx** [31, 32], the *data transfer and location system*. It handles data transport through CMS sites and tracks existing data and their location. This system will be analyzed later in this thesis.
- **DBS** [33], the *Data Bookkeeping Service*, a catalog of Monte Carlo simulations and experiment metadata. It records existing data, origin information, relations between datasets and files, in order to find a particular subset of events inside a dataset on a total amount of about 200 000 datasets and more than 40 millions of files.
- **DAS**, the *Data Aggregation System* [34], created to furnish users with a coherent and uniform interface for data management recorded on multiple sources.

All of this components are designed and implemented separately. They interact among them and with Grid users as web services.

2.4 PhEDEx

PhEDEx, standing for ***Ph****ysics* ***E****xperiment* ***D****ata* ***Ex****port*, handles CMS transfers via Grid in a secure, reliable and scalable way. It is based on an Oracle database cluster [35] located at CERN, the ***T****ransfer* ***M****anagement* ***D****ata****b****ase* (TMDB): it contains information about data replicas location and active tasks. It has two interfaces: a website [36], through which users may require datasets or fileblock transfer interactively and a web data service [37] made for PhEDEx interactions between other Data Management components. When a request is made through one of the interfaces, PhEDEx connects to TMDB for metadata and rewrites results once the task is completed.
TMDB is designed to minimize *locking contention* between different agents and demons executed by PhEDEx in parallel and is made to optimize cache use, avoiding

coherence problems on the inside. Agents that are executed centrally at CERN perform most of *data routing* work, calculating the less expensive path in terms of performance. This happens considering the performance of the network links used to connect destination site with the origin one, based on successful transfers between them in a certain time window.
Once the route is chosen, download agents receive necessary metadata from TMDB and start the transfer using specific plugins depending on Grid middleware. Every transfer success or data deletion is independently verified for each fileblock and, in case of failure, other agents are activated trying to complete the request. Performance data are constantly recorded on TMDB and can be displayed through PhEDEx dashboard.
However it is still common to observe transfer workflows that do not reach full completion due e.g. to a fraction of stuck files which require manual intervention [38]. A deeper study of this kind of situations is the focus of this thesis, and will be discussed in the next chapters.

2.5 The CMS Remote Analysis Builder

The CMS Remote Analysis Builder (CRAB) [39, 40] is a tool developed for distributed analysis [41] in CMS. It creates, submits and monitors CMS analysis jobs over the Grid. It is designed to relate with every Grid component, allowing full system autonomy and maximizing physicists analysis work. CRAB furnishes access to acquired data for each user regardless their geographical location.
Data analysis at CMS is, as previously mentioned, *data-location driven*; it means that user analysis is done where data is stored. The CRAB steps for distributed analysis are:

1. Locally execute analysis codes on samples in order to test their workflow correctness;

2. Select the amount of data required for analysis;

3. Start analysis: CRAB transfers the code into destination site and it returns the result as well as job's log.

Actually CMS is in a transition phase between version 2 and the new version 3 which implements a series of code optimization and new features. A first version of CRAB 3 [42] is already used by some CMS analysts, although most of them are still using CRAB 2: the migration towards version 3 will boost in 2015 and early 2016, at the start of Run 2. The CRAB3 architecture and workflow will be briefly described in the following. CRAB workflow steps from user's job submission to the publication of results are:

1. CRAB client submits user's request to CRAB server;

2. CRAB server places the request into Task (Oracle) Database (Task DB);

3. A CRAB server subcomponent called *Task Worker* always monitors over Task DB searching new requests;

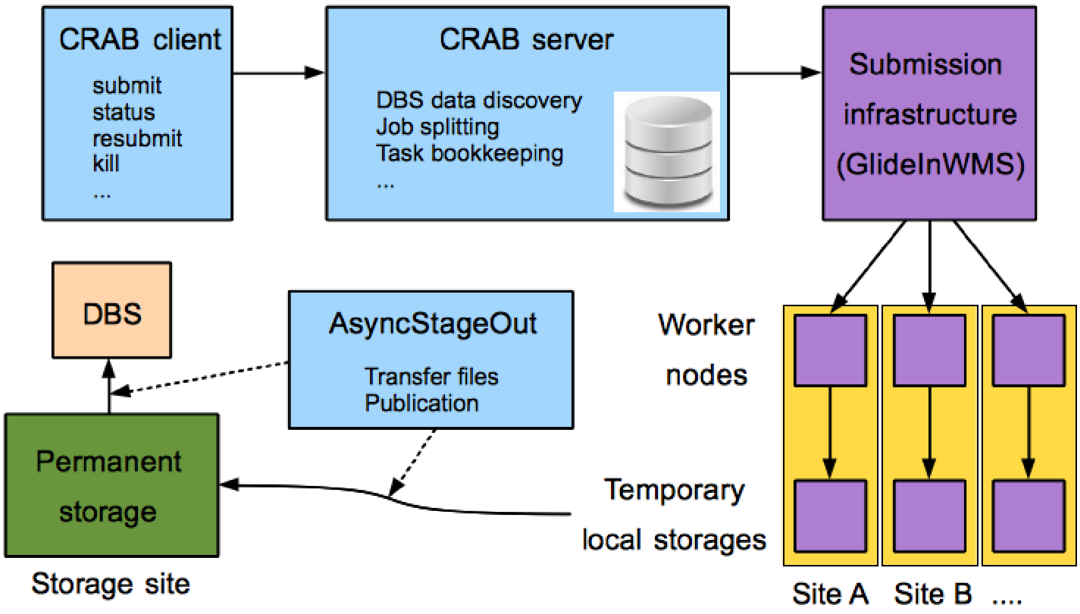


Figure 2.3: CRAB3 exemplified architecture [43].

4. When Task Worker receives new requests it forwards to a submission infrastructure (GlideInWMS);

5. GlideInWMS finds available Worker Nodes and puts jobs inside;

6. Once the Worker Node finishes the job, it copies output files into the *temporary storage* of the site;

7. AsyncStageOut service transfers output files from temporary storage into a permanent one and publishes it into DBS.

CRAB 3 simplified architecture is shown at Figure 2.3.

2.6 CMS Computing as a fully-connected mesh

CMS Computing Model since its creation has undergone a deep transformation. Initially it was very hierarchic, based on MONARC: each Tier-2 was connected to only one "regional" Tier-1 albeit with a certain flexibility [28, p. 20]. This idea is shown in Figure 2.1, where each T1 is connected only with a few T2 number.
The initial *network topology* (Figure 2.4) provided :

- Unidirectional data-flow from Tier-0 to Tiers-1;
- A data-flow between Tier-1 and one between each Tier-1 and its connected Tier-2;
- Each Tier-2 is connected with only one Tier-1 with incoming CMS data and outgoing Monte Carlo simulations ready for Grid distribution;
- Tier-2 connections were allowed but with a low-lewel expected load.

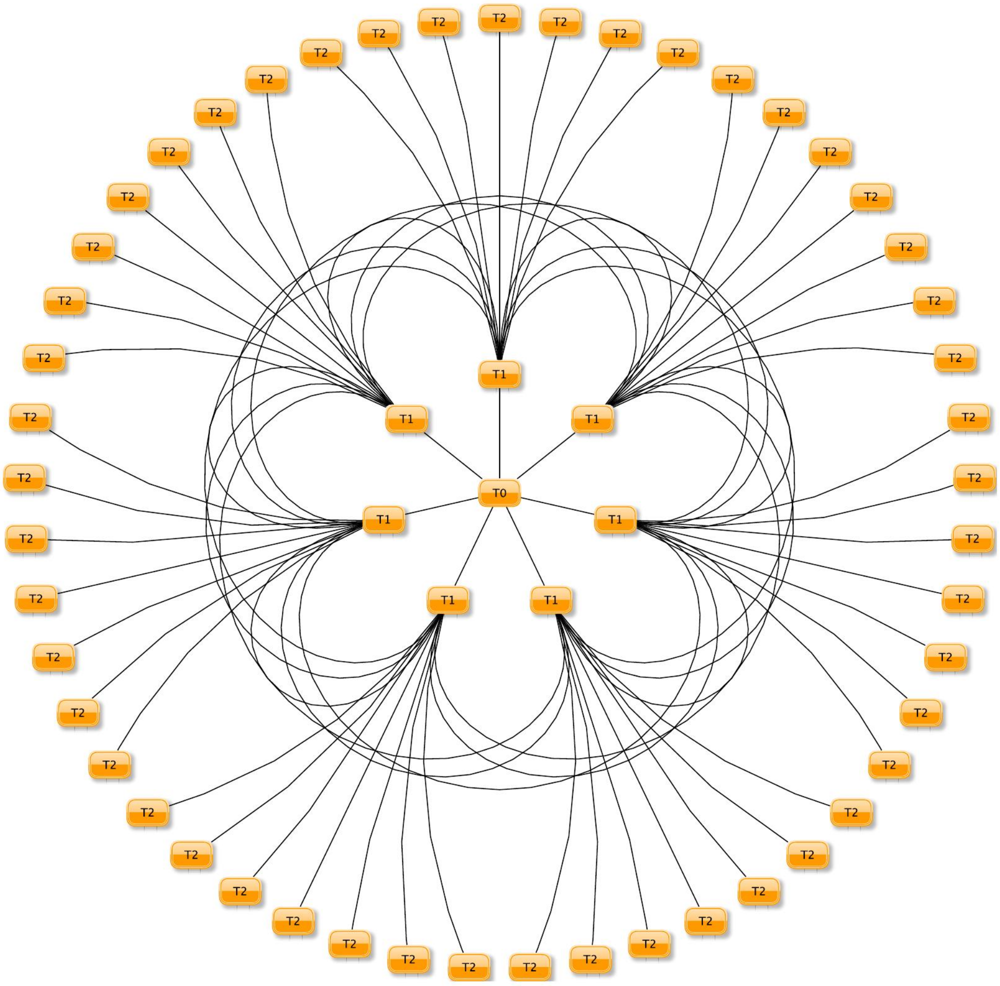


Figure 2.4: Original network topology of CMS Computing System. [44]

The amount of bi-dimensional links in such a model is a few hundreds.
During Run-1, however, data transfer flexibility has become an important part of the Computing Model, with a more dynamic architecture. As shown in Figure 2.5, links between Tier-2 were not irrelevant at all; they have opened the chance of accessing data quickly without passing through T1 sites; so more flexibility implies a much optimized use of resources and a quick way to process workflows.
A further evolution already started in Run-1 consists of reading data remotely, without copying them on a local drive allowing a much clearer data access [45, pp. 39-48]. The evolution of such a *fully-connected mesh* finds in bandwidth availability its strength. Now the set of transferred data at CMS reaches over 1 PB per week and most of the links are far from being saturated. Links number between Tier-2 has been increased until the current *network topology* [44, 46] (Figure 2.6); Tier-0 is directly connected with each Tier-1 and also most part of Tier-2, with about 60 link. Tier-1 and Tier-2 are almost all connected between them, bi-directionally, with about 100-120 link per site. The total amount of connection is about 2000.
A better use of this network will increase analysis speed and capacity [44, 46]. Right now, performance is based on: dataset completion time, failure rate of a certain

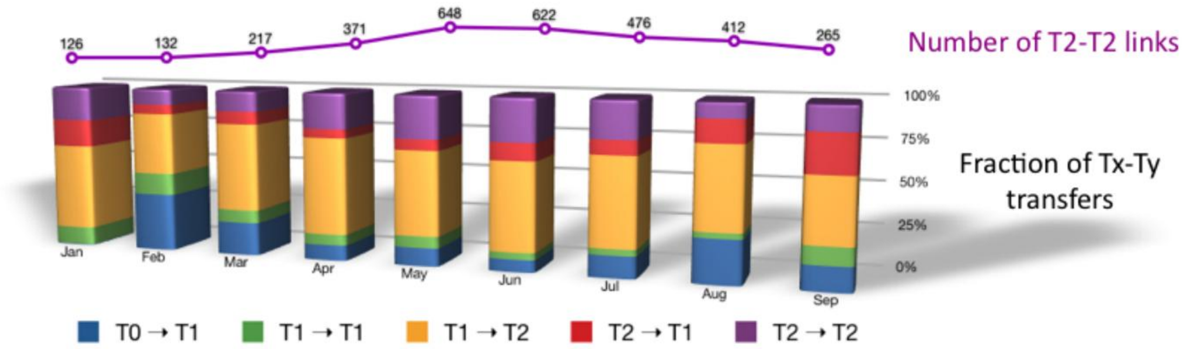


Figure 2.5: Total transferred volume percentage during 2010 [24].

link, completion time of a set of jobs and CPU efficiency.

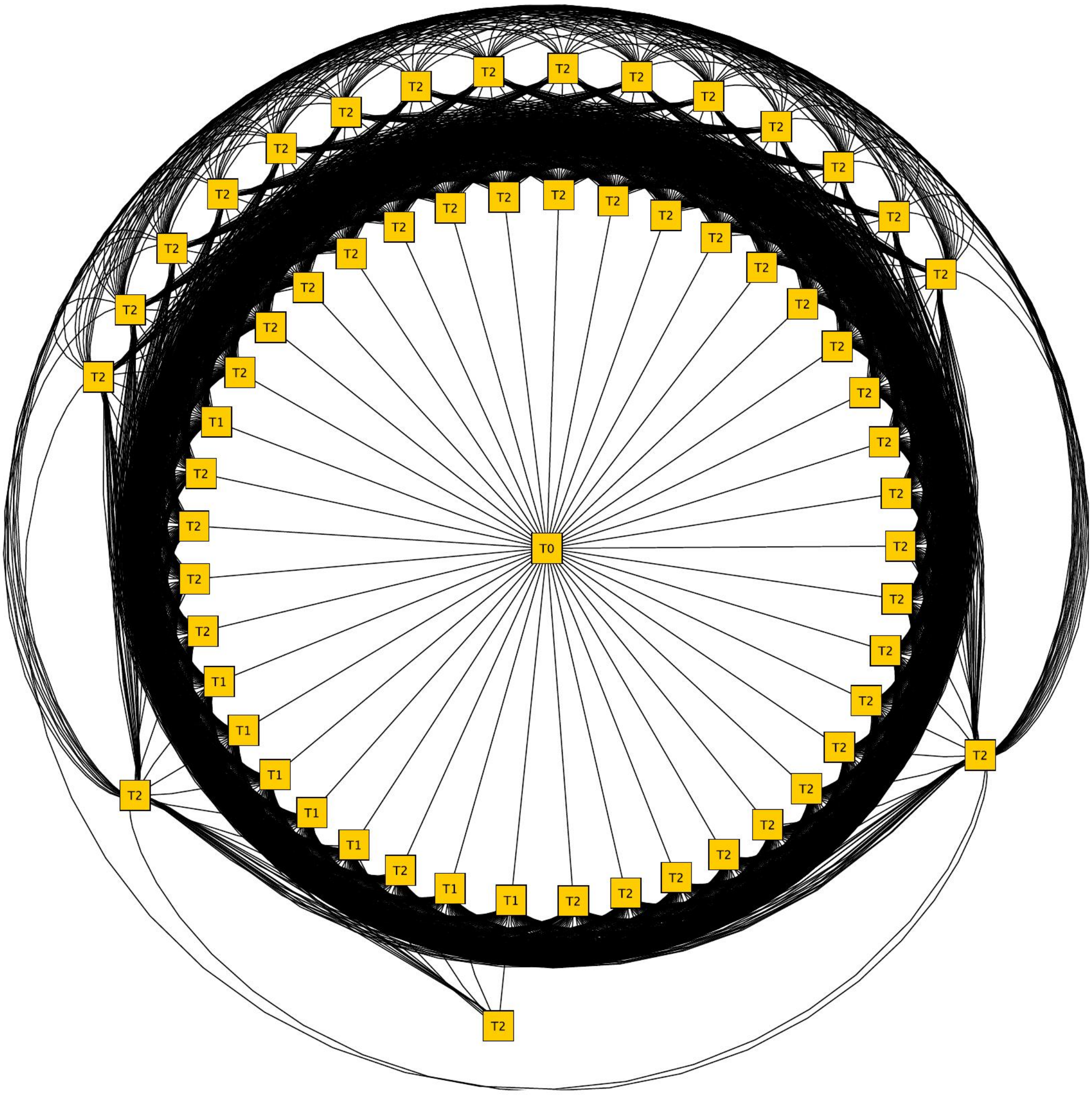


Figure 2.6: Actual Network topology at CMS Computing System.

Chapter 3

Reducing latencies in CMS transfers

PhEDEx is a reliable and scalable dataset replication system based on a central database running at CERN and a set of highly specialized, loosely coupled, stateless software agents distributed at sites (as stated in section 2.4).

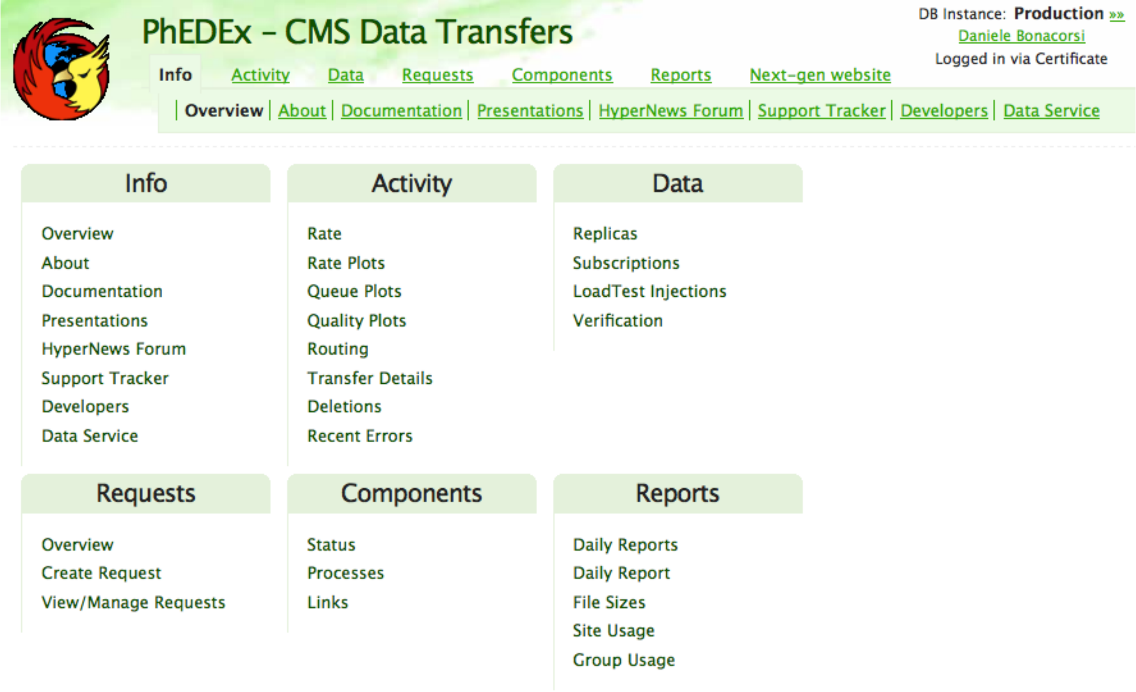


Figure 3.1: The CMS PhEDEx web interface.

Once logged in with a valid Grid certificate, the CMS PhEDEx web interface (Figure 3.1) make possible a wide set of actions, e.g. transfer subscription, monitoring of the transfer progresses and latencies, health controls of the software agents, etc... Originally PhEDEx was designed to perform transfers of massive volumes of data among WLCG computing. During Run-1 PhEDEx moved 150 PB and currently it is moving about 2.5 PB of data per week among 60 sites (Figure 3.2 and 3.3) [**Cap 1**, 38].

While the desired level of throughput has been successfully achieved, it is still common to observe transfer workflows that not reach full completion in a timely

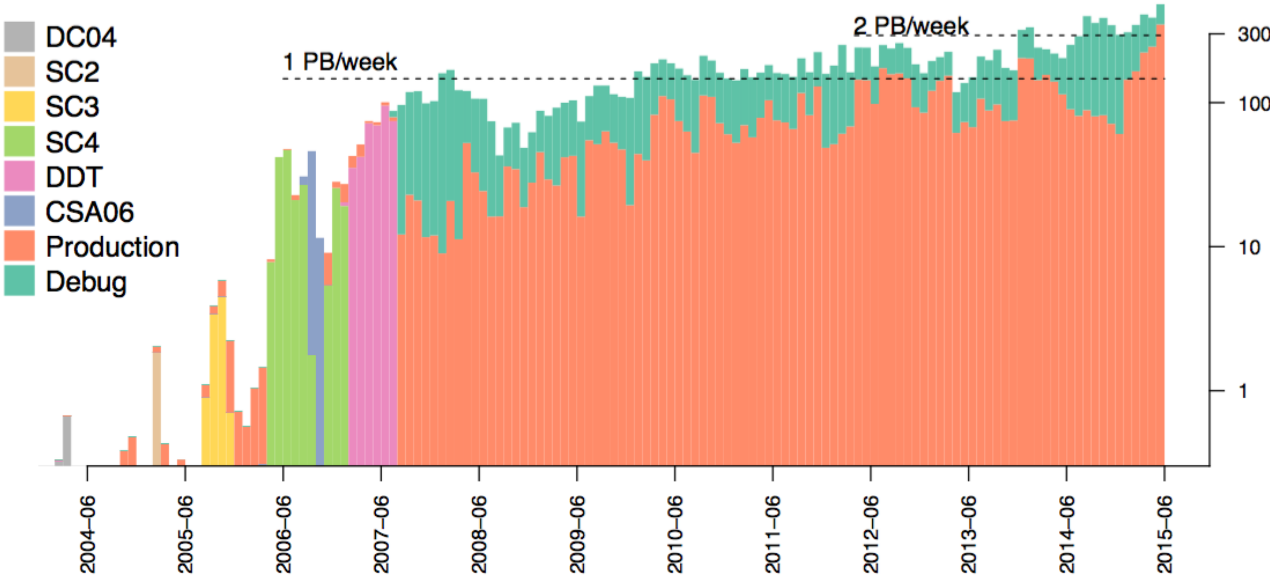


Figure 3.2: Overall data volume moved by PhEDEx since 2004. Note the logarithmic scale.

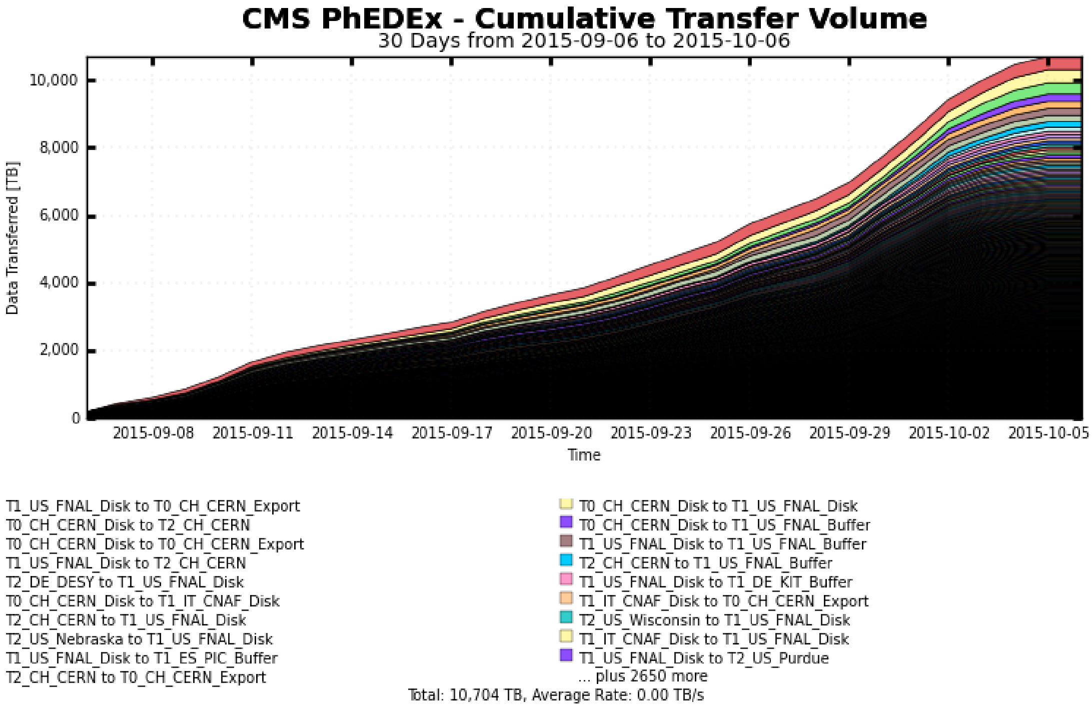


Figure 3.3: Cumulative transfer volume managed by PhEDEx among all Tiers in a 30 days period around September 2015. Each color represent a Tier-X to Tier-Y connection. Lower data volumes are even hardly visible as the total of active links is more then 2650.

manner due to a small fraction of stuck files that require manual intervention. This chapter will provide a general view of the latency issue in CMS transfers while actual analysis will be addressed in Chapter 4.

3.1 Can PhEDEx be improved?

The PhEDEx design aims at providing the highest possible transfer completion rate, although possible infrastructural unreliabilities, through fail-over tactics and automatic retrial. This particular choice however does not minimize the waiting time of individual users for the completion of their jobs.
For this reason, during several years of operations, a large set of data concerning latencies observed in all transfers between Tiers has been collected: their study allows a categorization of such latencies in order to attack them appropriately at the development level so to ultimately improve the system and increase its overall preformances.

3.1.1 Instrumenting PhEDEx to collect latency data

Data produced from the CMS experiment are collected into datasets which regroup similar physical information coming from particle collisions e.g. the final state of 4 muons events. Nevertheless the atomic unit for transfer operations in PhEDEx is the file block: an arbitrary group of O(100-1000) files in the same dataset. Initially all information were block-level organized and PhEDEx didn't record states of individual files: for this particular reason, in 2012, this system was instrumented to detect latency issues with historical monitoring tables, containing file-level informations, and equipped with useful tools to retrieve monitoring data for further analysis.
The first agent, called *BlockAllocator*, that is responsible for monitoring subscriptions, records the timestamps of the main events related to block completion i.e. the time when the block was subscribed and the time when the last file in the block was replicated at destination. The difference between these timestamps can be defined as the total latency for the entire block as experienced by users including manual intervention.
The second agent, called *FilePump*, that is responsible for collecting the results of transfer tasks, records a summary of each file history inside the block such as the time when the file was first activated for routing or the number of transfer attempts needed for the file to arrive at destination, as well as the source node of the first and last transfer attempts.
For performance reasons, after the transfer is complete, these live tables are stored into historical log tables: file-level statistics in $t\_log\_file\_latency$ (usually deleted after 30 days) and block-level statistics into $t\_log\_block\_latency$ (permanently recorded) [**Cap 2**, 38]. Those tables are integrated with additional events related to block completion such as:

- the time when the first file was routed for transfer;
- the time when the first file was successfully replicated at destination;

- the time when 25%/50%/75%/95% of the files inside the block were replicated at destination;
- the time when the last file of the block was successfully replicated at destination;
- the total number of attempts needed to transfer all files in the block;
- the source node for the majority of files in the block.

3.1.2 Cleaning data for latency analysis

Since 2012 the total amount of records collected about latencies has reached about 3 million block subscriptions. However before starting the analytics work presented in Chapter 4, this set of data has been cleaned and processed in order to remove uninteresting data and define useful observables. First of all there are ill data, i.e. records with missing or inconsistent data, mostly issued from test transfers and representing only the 5% of the total amount. There are also about 780000 that took place in blocks still open and growing in size. Furthermore 62000 transfer entries which have been suspended during their execution: these particular entries were removed because their treatment may render the whole process uselessly complex although being well-defined targets. It has been defined also a cutoff of 3 hours on the transfer time: seen the typical time scale of data transfers at CMS, if a transfer takes less than 3 hours from subscription to the completion we can, arbitrarily but sensibly, argue that it is not a candidate for having latency problems [**Cap 3**, 38]. Roughly 960000 items has been removed by this cutoff.
Although the total amount of subscriptions was nearly about 3 million, this cleaning procedure has allowed to remove about 2 million transfers, leaving a cleaner yet enough populated sample.

3.1.3 The "skew" variables

In order to determine the evidence of a latency effect, new variables have been introduced called "skew" variables. These variables show the transfer rate ratio between a small portion of time (such as the last 5% or the first 25%) and the X percent from the beginning or to the end of a transfer.
More precisely:

- *Skew x* variables:

$$Skew_X = \frac{\text{transfer rate of the LAST 5 percent of the files}}{\text{transfer rate of the FIRST X percent of the files}} \cdot \frac{X}{5} \tag{3.1}$$

  where $X = 25, 50, 75, 95$;

- *Skew Last X* variables:

$$SkewLast_X = \frac{\text{transfer rate of the LAST 5 percent of the files}}{\text{transfer rate of the LAST X percent of the files}} \cdot \frac{X}{5} \tag{3.2}$$

  where $X = 25, 50, 75$;

- *Reverse Skew X* variables:

$$RSkew_X = \frac{\text{transfer rate of the FIRST 25 percent of the files}}{\text{transfer rate of the FIRST X percent of the files}} \cdot \frac{X}{25} \tag{3.3}$$

  where $X = 50, 75, 95$;

- *Reverse Skew Last X* variables:

$$RSkewLast_X = \frac{\text{transfer rate of the FIRST 25 percent of the files}}{\text{transfer rate of the LAST X percent of the files}} \cdot \frac{X}{25} \tag{3.4}$$

  where $X = 5, 25, 50, 75$.

On a transfer ideally running at a constant rate all these variable would have value 1. Skew variables that significantly differs from unity can be considered a good hint of a transfer with latency issues. These considerations have led to a further data cleaning: only block with more than 5 files, a size larger than 300 GB and defined values for all the "skew" variables were left, being big enough to have a "bulk" and a "tail" [**Cap 3**, 38].
The analysis presented in Chapter 4 has been performed using roughly 42000 transfers entries which are the result of this skimming process.

3.1.4 Types of latency

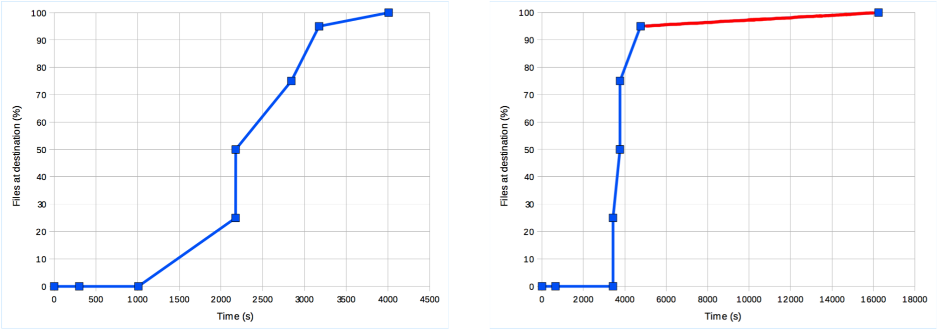
Transfers may be affected by latency issues in different ways. In this work, attention has been put on these particular types of latency: late stuck (already referred to as transfer "tail") and early stuck.

Late Stuck

Late stuck files are present when one or few files take much longer to get transferred than the rest of the block. In figure 3.4, sample plots of the number of files at destination as a function of time are shown. Different patterns and temporal scales in block completion are clearly visible [47]. Figure 4.32a shows a normal transfer trend: in this scenario, block completion is gradual from its subscription until the end in a reasonable time. Figure 4.32b, instead, shows a transfer with a final "tail": unlike the previous image, the last 5% (in red) requires a longer time than the rest of the block to complete, and probably only a manual intervention allowed its actual completion.
This type of latency can be observed by selecting transfers in which the time needed for moving the last 5% of bytes is larger than a given threshold δ. To prevent very large blocks from being included even if they have no real latency issues we may add to δ an offset which depends from the size of the block and a reference speed parameter v. In formulas

$$\Delta T_{last5\%} > \delta + \frac{S}{20v} \tag{3.5}$$

where $\Delta T_{last5\%}$ is the time of the last 5% of replicas and S the size of the block. Sensible values used for the parameters are $v = 5MB/s$ and $\delta = 10h$ [**Cap 4**, 38].
The CMS data production and processing tasks over WLCG may occasionally



(a) Percentage of files at destination as a function of time since subscription, for a block transferred with perfect transfer quality

(b) Percentage of files at destination as a function of time since subscription, for a block with a few permanently stuck files ("transfer tail")

Figure 3.4: Comparison between a block with perfect transfer quality (a) and a block with stuck files (b). More details on the differences among the two plot, in particular the red segment, are presented in the text below.

produce corrupted outputs. This happens when one or a few files in a datablock are missing or resulting from a non-handled failure during storage or they have wrong size/checksum. PhEDEx has been designed to deal with such data corruption events and is able to detect them (thanks to internal checking mechanisms and pre/post validation scripts) and tag the corresponding transfers as failing. Moreover PhEDEx tries to minimize the impact of the corrupted files on the data placement operations by re-queuing them while it keeps on transferring the rest of the data. As the transfers of the corrupted files keep on failing systematically, PhEDEx suspends them for a longer time and creating consequently a transfer "tail".

Reasons of such latency are heterogeneous but mostly due to storage problems, which explains why only a few files are affected from this issue. These late stuck can have a very important impact on CMS operations: in fact, transferred data is often useless to CMS analysts until its completion. Therefore it is quite important to find the stuck files and fix the underlying problem as soon as possible. Usually a manual intervention is required, consisting of either replacing the file or, if there is not any uncorrupted data, announcing it as lost [**Cap 5**, 38].

Among other analysis, chapter 4 will provide a study of these tails, showing the impact of these missing/corrupt files clearly visible at the last few percentages of the whole block transfer.

Early Stuck

Early stuck latency affects those blocks that begin with serious performance issues and start flowing properly only after some time, presumably once such issues have been fixed.

This type of latency can be observed by selecting transfers in which the time for the first replica is larger than a given threshold δ. To prevent very large blocks from being included even if they have no real latency issues we may add to δ an offset which depends from the average size of the files and the reference rate parameter v.

In formulas

$$\Delta T_{1st} > \delta + \frac{\langle S \rangle}{v} \tag{3.6}$$

where ΔT_{1st} is the time of the first replica and $\langle S \rangle$ the size in the datablock. The same values of v and δ shown above can be used [**Cap 4**, 38].
Early stuck situation, if not promptly identified, leads to serious consequences: only a quick problem identification, attach and fix can avoid to pile up delays and additional work load [**Cap 4**, 38]. The solution for this latency is not straightforward: it may require admin intervention at the source or destination site, or even an involvement by central operators.
A study of these early stuck is also provided in chapter 4.

Chapter 4

Analysis of latency data

In this chapter the analysis made to investigate the PhEDEx performances will be described. Studying this data has allowed to observe different patterns and trends depending on the type of latency that we have previously introduced and described (see Section 3.1.4). In particular, the definitions used for the different latency types (see eq. 3.5 and 3.6) represent the starting point of the entire analysis. Such definitions have been discussed and decided within a group of PhEDEx developers and operators with we started a collaboration, and also contributed together with a paper presented at the Conference on Computing for High Energy Physics 2015 (**CHEP 2015**)[38]).
The goal of our analysis is manifold: in the first place we want to verify if the definitions stated above are well suited to select samples with a certain degree of purity; secondly, we want to identify characteristic markers that reduce the variability and hence suggest either real-time monitoring action to be adopted by PhEDEx developers (e.g. new features, code improvements or refinements).

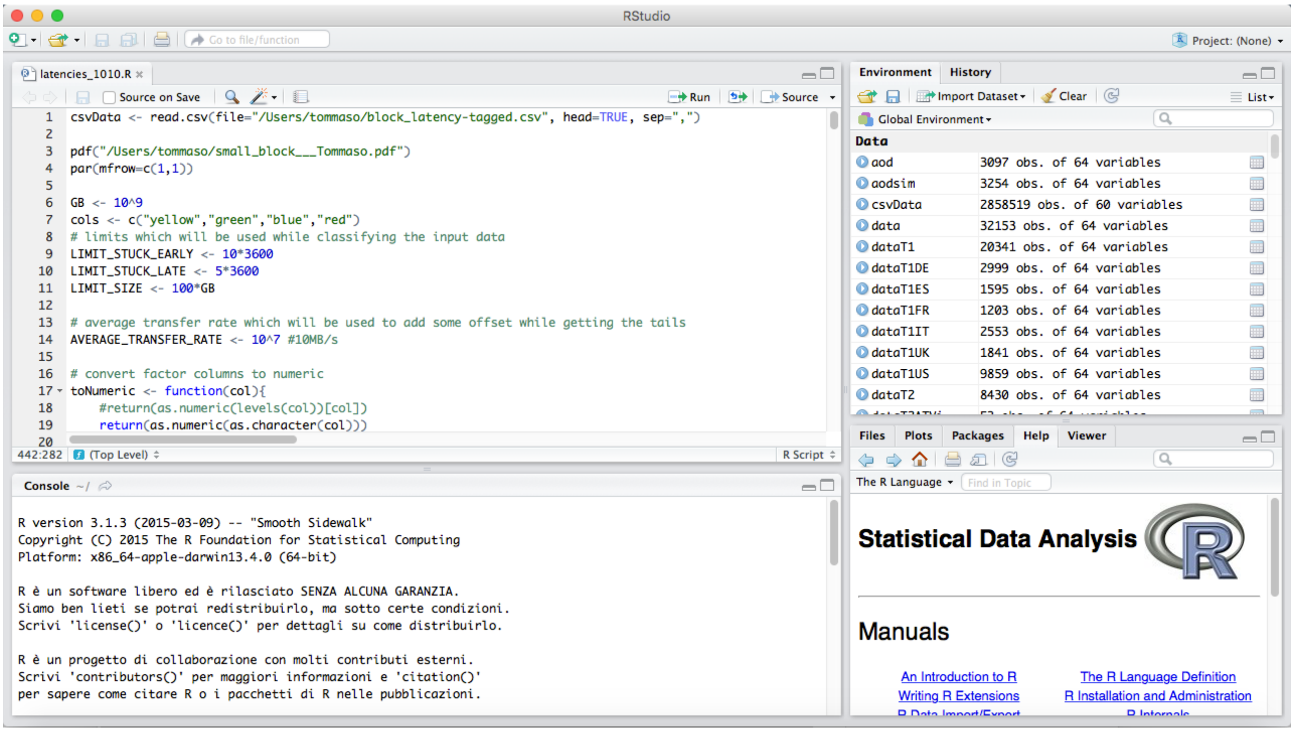
4.1 Data collection and elaboration

Initially data were collected inside a .csv file of enormous dimensions produced by the PhEDEx monitoring system (described in detail in Section 3.1.1) that required a lot of time in order to be compiled and executed. After an optimization work, this .csv file has been reduced to roughly 3 million entries (skimmed using the procedure described in Section 3.1.2) and information were better organized. We elaborated these data interactively using a R[1] script through an user interface called RStudio [48] (see Figure 4.1): using my laptop the compilation time required about half an hour.

Plots and histograms produced have been divided depending on the latency type i.e. Early Stuck and Late Stuck. At last, for the sake of clarity, another particular type of latency called "Stuck Other", untreated in Chapter 3, will be briefly explained.

Initially we noticed a particularity in the definition of the latencies typologies:

[1] **R** is a GNU programming language and software environment for statistical computing and graphics created by Ross Ihaka and Robert Gentleman at the University of Auckland.



```r
csvData <- read.csv(file="/Users/tommaso/block_latency-tagged.csv", head=TRUE, sep=",")

pdf("/Users/tommaso/small_block___Tommaso.pdf")
par(mfrow=c(1,1))

GB <- 10^9
cols <- c("yellow","green","blue","red")
# limits which will be used while classifying the input data
LIMIT_STUCK_EARLY <- 10*3600
LIMIT_STUCK_LATE <- 5*3600
LIMIT_SIZE <- 100*GB

# average transfer rate which will be used to add some offset while getting the tails
AVERAGE_TRANSFER_RATE <- 10^7 #10MB/s

# convert factor columns to numeric
toNumeric <- function(col){
    #return(as.numeric(levels(col))[col])
    return(as.numeric(as.character(col)))
```

```
Data
aod        3097 obs. of 64 variables
aodsim     3254 obs. of 64 variables
csvData    2858519 obs. of 60 variables
data       32153 obs. of 64 variables
dataT1     20341 obs. of 64 variables
dataT1DE   2999 obs. of 64 variables
dataT1ES   1595 obs. of 64 variables
dataT1FR   1203 obs. of 64 variables
dataT1IT   2553 obs. of 64 variables
dataT1UK   1841 obs. of 64 variables
dataT1US   9859 obs. of 64 variables
dataT2     8430 obs. of 64 variables
```

```
R version 3.1.3 (2015-03-09) -- "Smooth Sidewalk"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R è un software libero ed è rilasciato SENZA ALCUNA GARANZIA.
Siamo ben lieti se potrai redistribuirlo, ma sotto certe condizioni.
Scrivi 'license()' o 'licence()' per dettagli su come distribuirlo.

R è un progetto di collaborazione con molti contributi esterni.
Scrivi 'contributors()' per maggiori informazioni e 'citation()'
per sapere come citare R o i pacchetti di R nelle pubblicazioni.
```

Figure 4.1: RStudio user interface [48] used for the analysis of the .csv file produced by PhEDEx.

they are not independent i.e. some overlap exists between them. Table 4.1 shows an interesting detail: the sum of the three categories is greater than the sample itself; this implies a double-counting of some entries or rather a correlation between different latency typologies. In fact, looking at Table 4.2 we can notice (in red) an overlap of stuckLate latencies in the stuckEarly subset (about 10% of double-counting). So far, we have considered these categories as independent for the time being, in order to quickly attack the problem and gain more insight. We acknowledge this may not be optimal, and we foresee to expand the analysis in the near future.

In the next session, we will present the analysis and discuss key points of the Early Stuck and Late Stuck, separately.

Table 4.1: Count of the different typologies of latencies (stuckEarly, stuckLate, stuckOther). *csvData* represent the entire sample, *data* represent the sample after the skimming process, see text for more detail.

	csvData	data	stuckEarly		stuckLate		stuckOther		sum of stuck
	2858519	32153	17310	54%	5144	16%	12971	40%	35425

4.2 Early Stuck Analysis

The Early Stuck kind of latency - defined in Section 3.1.4 - affects those blocks whose transfer begins with serious performance issues and start flowing properly

Table 4.2: Overlap between stuckEarly, stuckLate, stuckOther.

	also in stuckEarly	also in stuckLate	also in stuckOther
stuckEarly	17310	3272	0
stuckLate	3272	5144	0
stuckOther	0	0	12971

only after some time.
A typical trend of a block with Early Stuck issues is shown in Figure 4.2: we can notice from the axis scales, that the first 25% in fact has a transfer rate of about 5 orders of magnitude smaller than the last 25%. Despite this plot seems obvious and easy to understand, it was never done before and may offer some useful information to PhEDEx operators when a transfer presents itself as an Early Stuck; having a chance to produce this plot in real time during CMS data transfers would allow to spot and promptly address data subscriptions that show illness since the very beginning, thus allowing PhEDEx operators to intervene and fix.

Another aspect to investigate in detail is shown in Figure 4.3: this histogram shows the time for the first file replica to appear at destination starting from the time when the subscription was inserted. We can notice a decreasing smooth trend with a maximum in the first hours, as expected.
However there are some particularities:

- using as lower limit for Early Stuck latencies $\delta = 10h$ in 3.6 (see Figure 4.4 to observe the cutoff) we assume as "problematic" nearly the 54% of the sample meaning that this type is predominant compared to the other stuck. This fact represent a problem because, if verified by further studies, means that latency issues for half the time occur at the very beginning of the transfer process thus making monitoring plots strongly recommended;
- there is a peak clearly visible in the 48h time interval (see Figure 4.5). This 48h peak originally made us presume that it was mainly due to Tier-2 centers: in fact, differently from Tier 1 sites, they do not work 24h/7 but only 8h/5 and e.g. a problematic subscription with latency issues made on Friday afternoon may not be solved until Monday morning causing a peak at roughly 48 hours. Instead Figure 4.6 and Figure 4.7 shows quite the opposite i.e. the 2 days peak is caused by Tiers 1. A plausible explanation of this phenomenon must be sought on the storage mechanisms of a Tier 1: custodial data, in fact, are stored on tapes that may require a tape robot related delay in order to retrieve the particular one that hosts the data requested for transfer. These delays apparently accumulate in a temporal windows of 48h, and create the peak as of Figure 4.6.

Another aspect to investigate in detail is shown in Figure 4.8: this plot shows a zoom on the first 24 hours in the transfer of the first file replica in a block. Each bin represent 1 hour, i.e. collects all the blocks whose transfer starts within 1 hour. A smooth behavior for increasing values of the first file replica time can be observed, with the exception of the increasing second bin which is analyzed in detail on Figure 4.9: this plot shows an increasing trend until the maximum located at about 30 minutes from job subscription. This is an indirect measurement of the actual PhEDEx infrastructure latency in starting a transfer task: the transfer are managed by the specific "FileDownload" software agent, which is stateless and queries the PhEDEx TMDB (see Section 2.4) to get information on the work to be done. These agents have a default cycle of roughly 20 minutes before rechecking and reactivating themselves, which means that it is highly probable that within subscription and the first file replica a time larger than 1 hour passes, explaining the behavior we indeed observe.

In Chapter 3, Section 3.1.3, we have introduced the "skew" variables: unlike the variables used for the previous investigations, these are by far more versatile and allow a wider scale latency investigation, but at the cost of sacrifying the simplicity of understanding and a certain level of intuitiveness. Many are the possible combination of these variables but not every single one of them is important to our goal: we have selected two of the most significant ones, and a bidimensional plot is shown in Figure 4.10: this plot shows SkewLast75 vs Skew25 in transfers with Early Stuck latency. Recalling equations 3.1 and 3.2 the explanation is quite simple: files in the block are initially stuck causing a low first 25% rate and producing high values of Skew25 (according to eq. 3.1), however the remaining 75% is quite fast which produces low values of SkewLast75 (according to eq. 3.2).

As data transfers having a given class of sites as sources and/or destination, as well as sources/destination being in specific nations may be affected by specific issues, a more in-depth analysis was performed in two directions: the first aimed at selecting only CMS Tier 1 (Tier 2) as source sites; the second aimed at distinguishing the transfers coming from the same Tier level but in different nations. Both investigations, separately, are presented and the results discussed in the following paragraphs.

Tier 1 as sources

We already introduced the Figure 4.6 which we now discuss further. The trend is quite smooth, with no exception, decreasing from a maximum at just few hours. It is indeed known that in most cases a T1 outbound transfer should start quickly for several reason (for more detail of Tier 1 facilities see Section 2.1.1). First a Tier 1 has plenty of tape server/drives resources, so actions may happen fluidly in parallel. Secondly a delay in exporting a file may come from the lack of a commissioned source-destination link, which is very rare for a Tier 1 as all outbound links have commissioned in advance.

A further investigation was made by separating the different Tier 1 by nation: the results are displayed in Figure 4.11 (Italian Tier 1), 4.12 and 4.13 (American Tier 1), 4.14 (German Tier 1), 4.15 (British Tier 1), 4.16 (French Tier 1), 4.17 (Spanish Tier 1). A clarification: I am not considering the Tier 1 functionality of

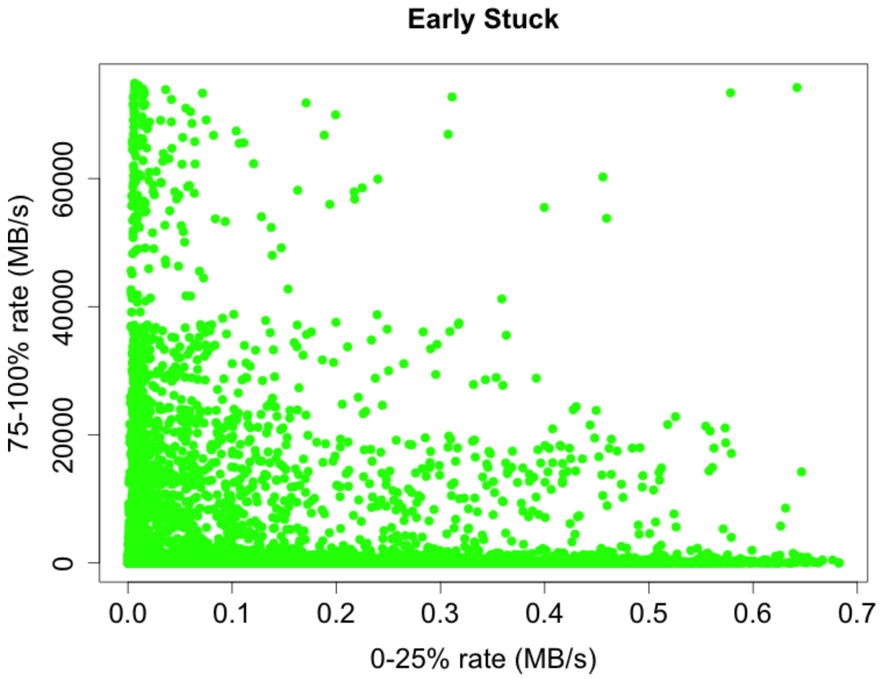


Figure 4.2: Transfer rate of the first 25% of the block vs rate of the last 25% for transfers suffering from a latency of the Early Stuck type according to eq. 3.6.

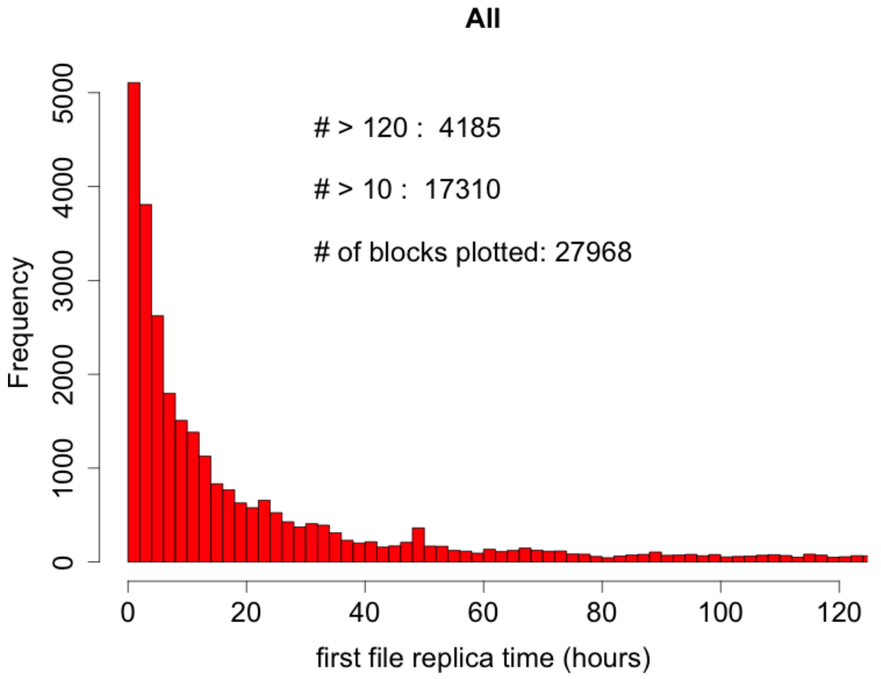


Figure 4.3: Time for the first file replica at the destination site for all data.

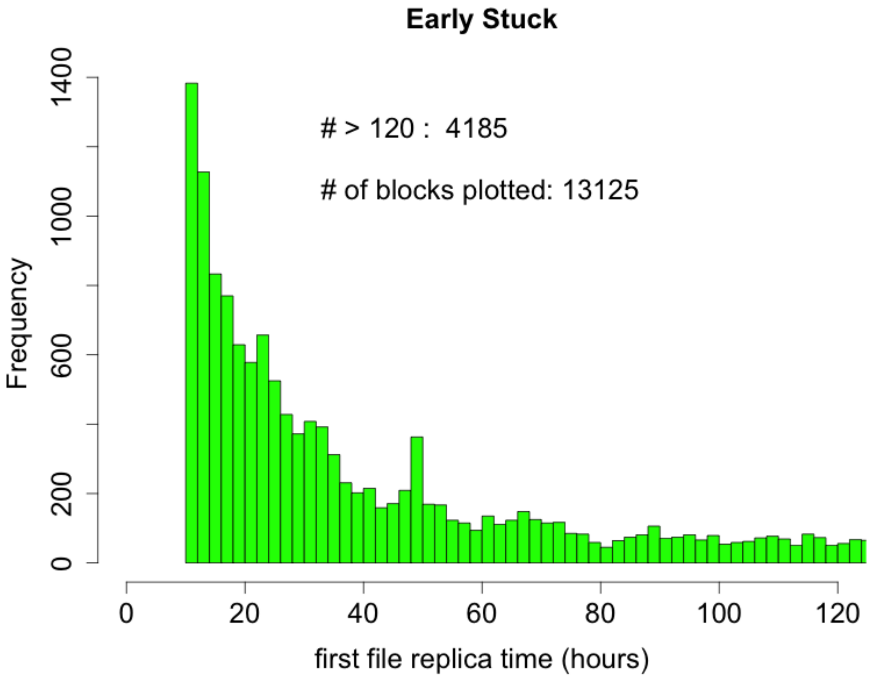


Figure 4.4: Time for the first file replica at the destination site for Early Stuck transfer (with more than 10h waiting time).

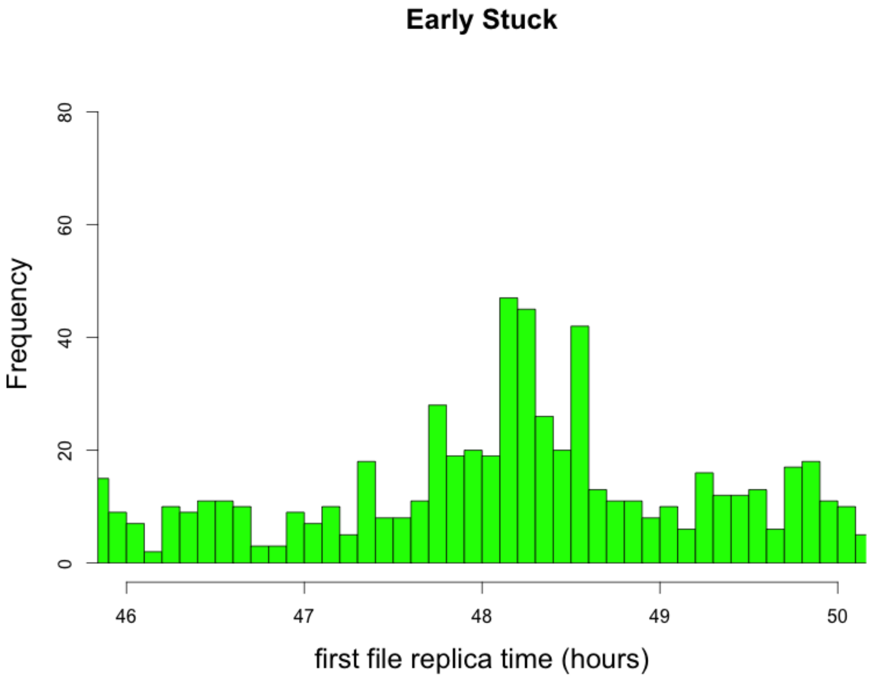


Figure 4.5: Magnification of the peak area of Figure 4.4).

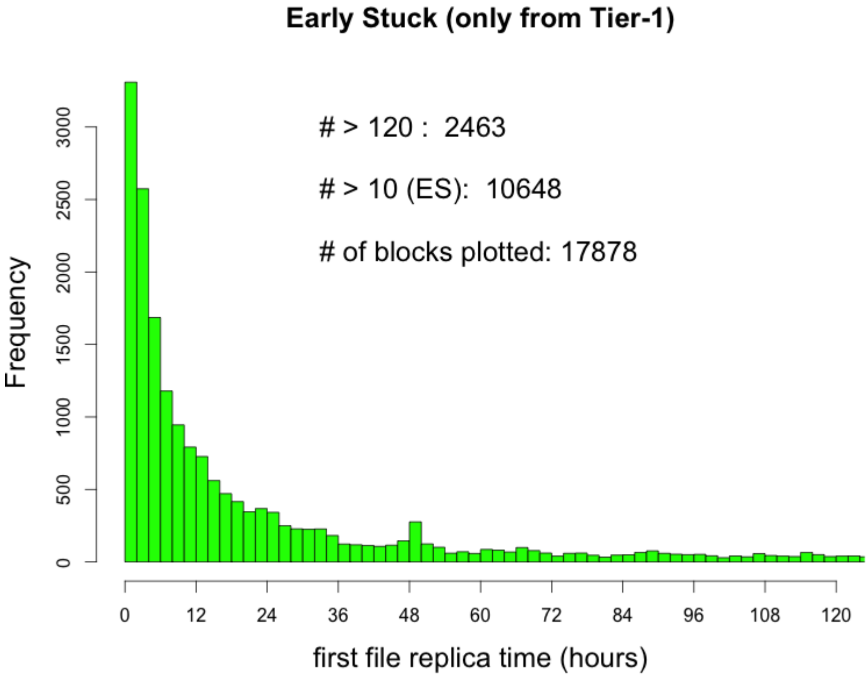


Figure 4.6: Time for the first file replica to appear at the destination site, when the source site is a Tier 1.

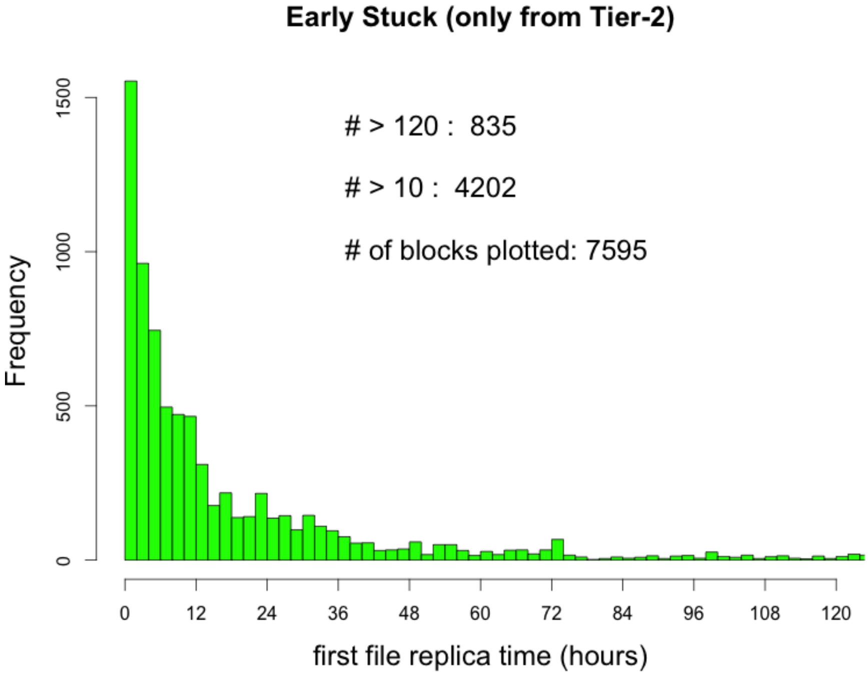


Figure 4.7: Time for the first file replica to appear at the destination site, when the source site is a Tier 2.

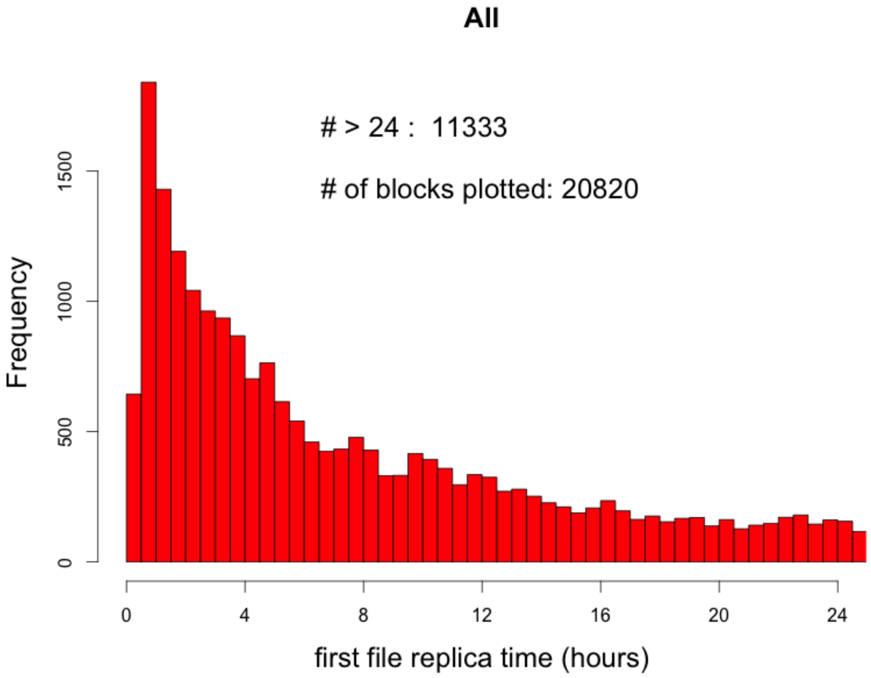


Figure 4.8: First day zoom for the transfer of the first file replica at the destination site, for all data in the .csv file.

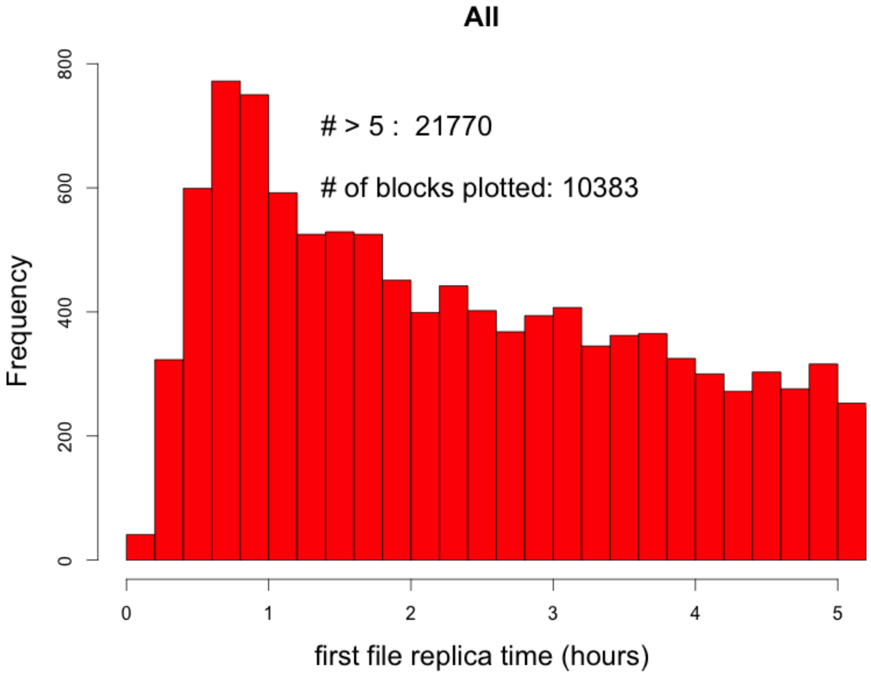


Figure 4.9: First 5 hours zoom for the transfer of the first file replica at the destination site, for all data in the .csv file.

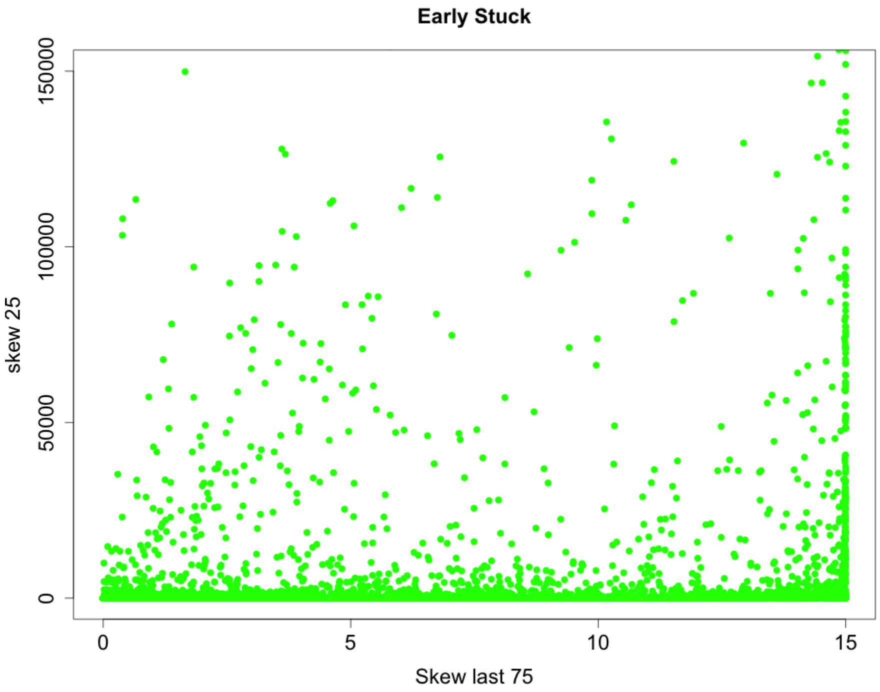


Figure 4.10: *Skew25* vs *SkewLast75* as defined in eq. 3.1 and 3.2, for transfers tagged as being Early Stuck according to eq. 3.6.

the PhEDEx site "T1_CH_CERN" (see Section 2.1.1) as it is collocated with the Tier 0 and it is definitely a peculiar Tier 1 site, making a direct comparison with other off-CERN national centers not appropriate. Some comments specific to each are provided in the following:

1. Italian Tier 1 (Figure 4.11): smooth curve as expected with no particular exceptions;

2. American Tier 1 (Figure 4.12 and 4.13): larger data traffic (as known, as the US Tier 1 alone is about 40% of the CMS Tier 1 resources), smooth curve as expected with a massive initial activity (depicted in detail in the 3-day zoom plot);

3. German Tier 1 (Figure 4.14): smooth curve as expected as the US and IT trend;

4. British Tier 1 (Figure 4.15): smooth curve with a slight final increase probably caused by some some tendency to accumulate backlog (a further analysis would be needed to understand if this is related to their specific storage solution, and this goes beyond the scope of this thesis);

5. French Tier 1 (Figure 4.16): smooth curve as expected, but exporting a definitely smaller data volume ;

6. Spanish Tier 1 (Figure 4.17): same considerations as the French Tier 1.

Table 4.3, at last, shows the fraction of Early Stuck entries divided by nation from Tier 1 sources.

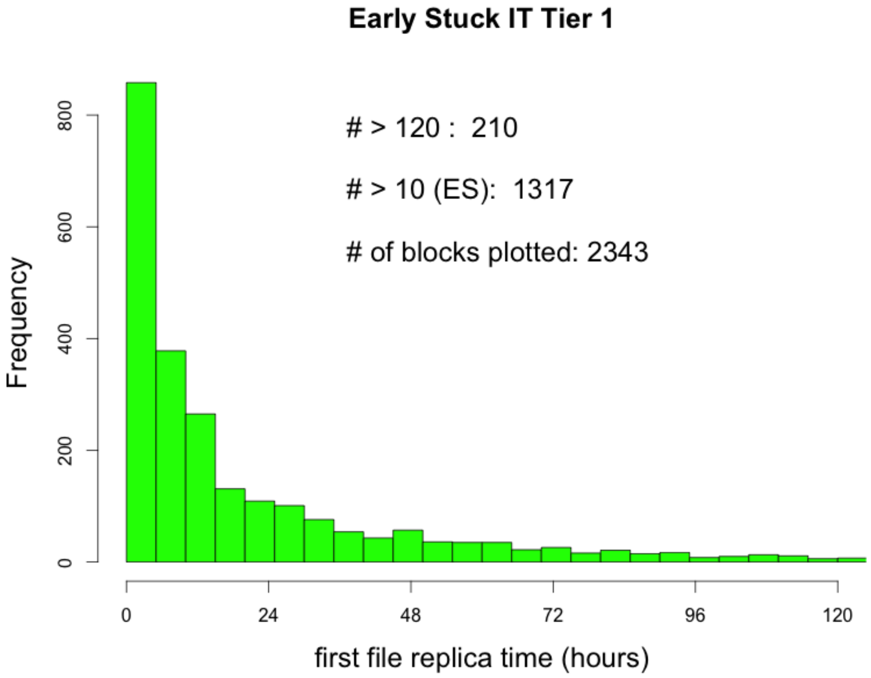


Figure 4.11: Time for the first file replica from IT Tier 1 source.

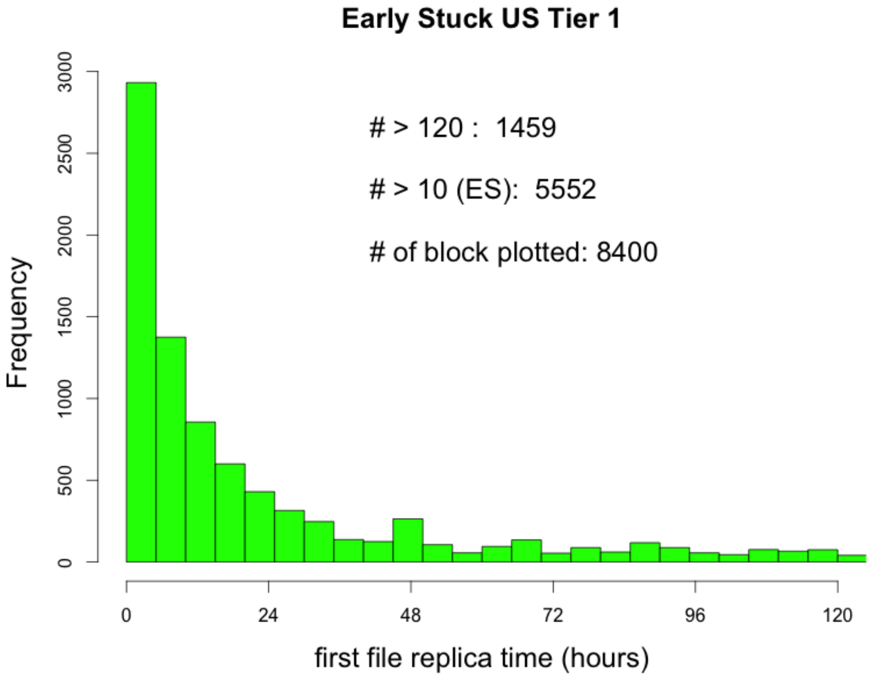


Figure 4.12: Time for the first file replica from US Tier 1 source.

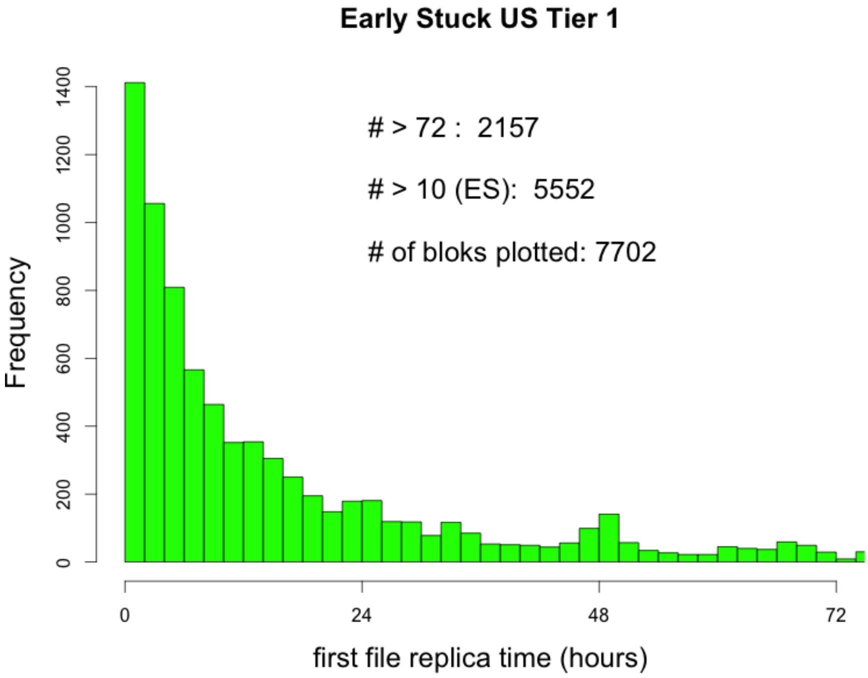


Figure 4.13: Time for the first file replica from US Tier 1 source (3-day zoom).

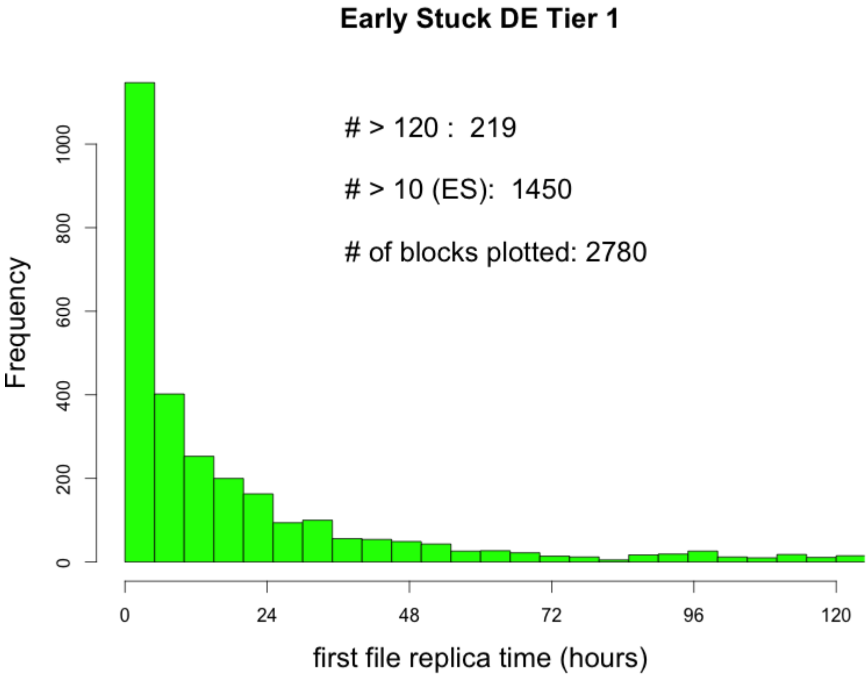


Figure 4.14: Time for the first file replica from DE Tier 1 source.

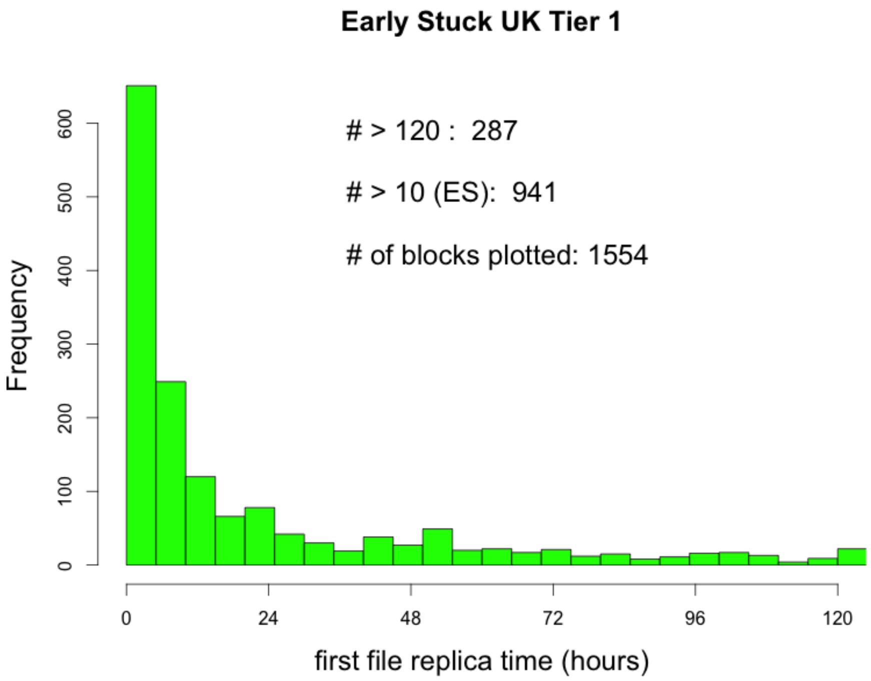


Figure 4.15: Time for the first file replica from UK Tier 1 source.

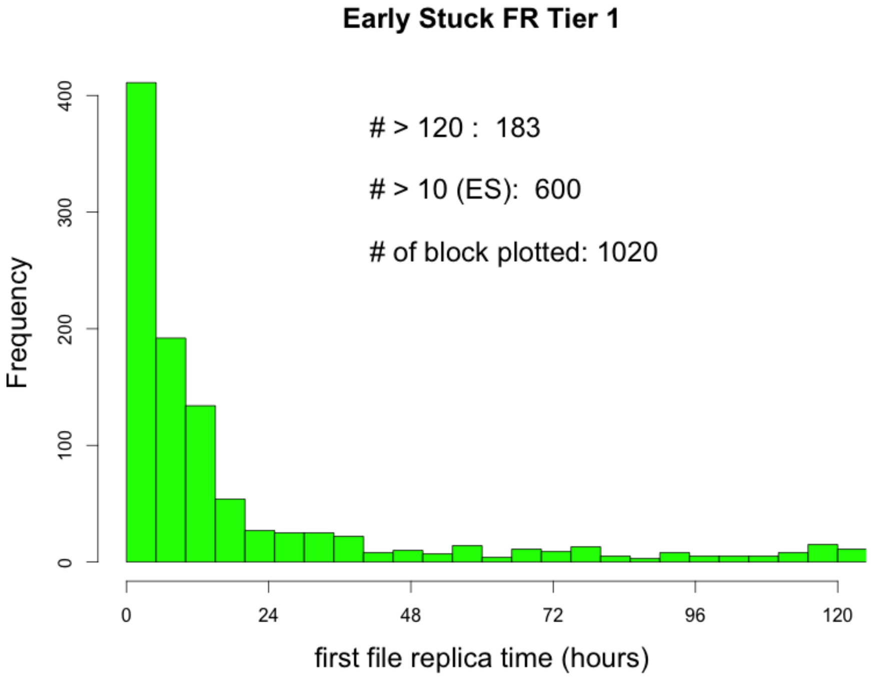


Figure 4.16: Time for the first file replica from FR Tier 1 source.

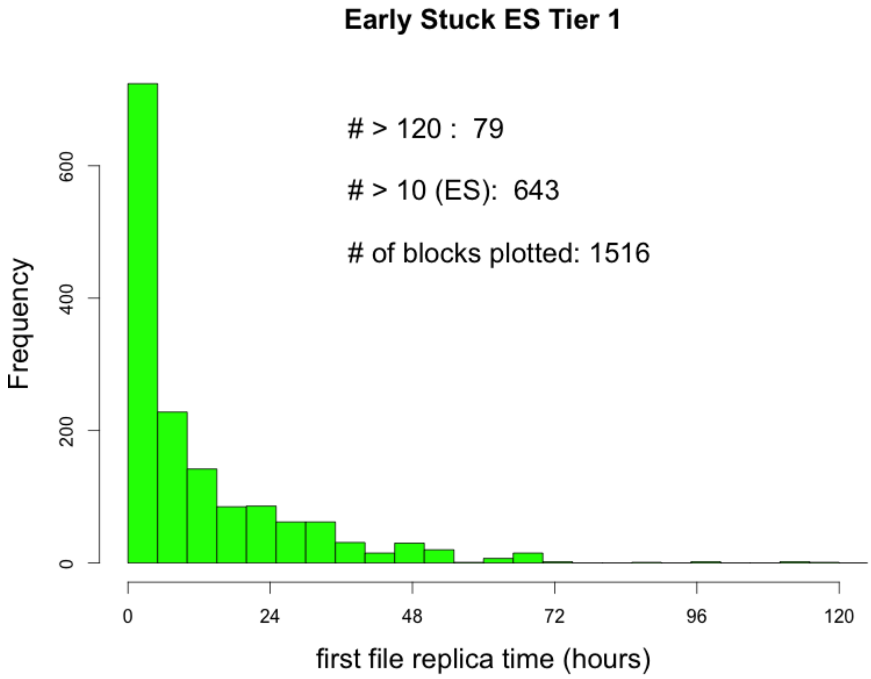
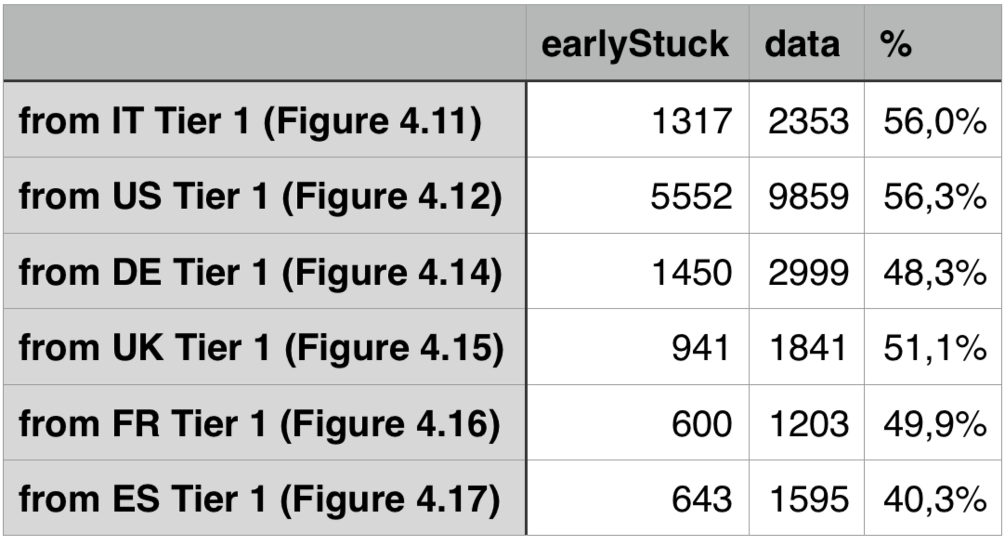


Figure 4.17: Time for the first file replica from ES Tier 1 source.



	earlyStuck	data	%
from IT Tier 1 (Figure 4.11)	1317	2353	56,0%
from US Tier 1 (Figure 4.12)	5552	9859	56,3%
from DE Tier 1 (Figure 4.14)	1450	2999	48,3%
from UK Tier 1 (Figure 4.15)	941	1841	51,1%
from FR Tier 1 (Figure 4.16)	600	1203	49,9%
from ES Tier 1 (Figure 4.17)	643	1595	40,3%

Table 4.3: Fraction of early stuck files and relative percentage for Tier 1 divided by nation.

Tier 2 as sources

We already introduced the Figure 4.7 which we now discuss further. While Tiers 1 are only 8 all around the world, Tiers 2 are more than 50 with quite a spread in location, size, service quality and performances. The most important thing is that within the first 2 days, most transfers just start, and only the queues remain to be dealt with. Nevertheless the curve showed in Figure 4.7 cannot be so relevant as it was for the Tiers 1.
For this reason, the corresponding results are shown on a more detailed study for each nation of interest (Figures 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.25, 4.26).
Table 4.4 shows the fraction of early stuck entries divided by nation from Tier 2 sources. An individual analysis is not necessary being very similar from each other. However some plots (RU Tier 2 4.19, US Tier 2 4.20, UK Tier 2 4.21) show a singular peak in the 72 hours area: this might be the convolution of the 48h scenario previously stated probably caused by a mix of time zone and human response.

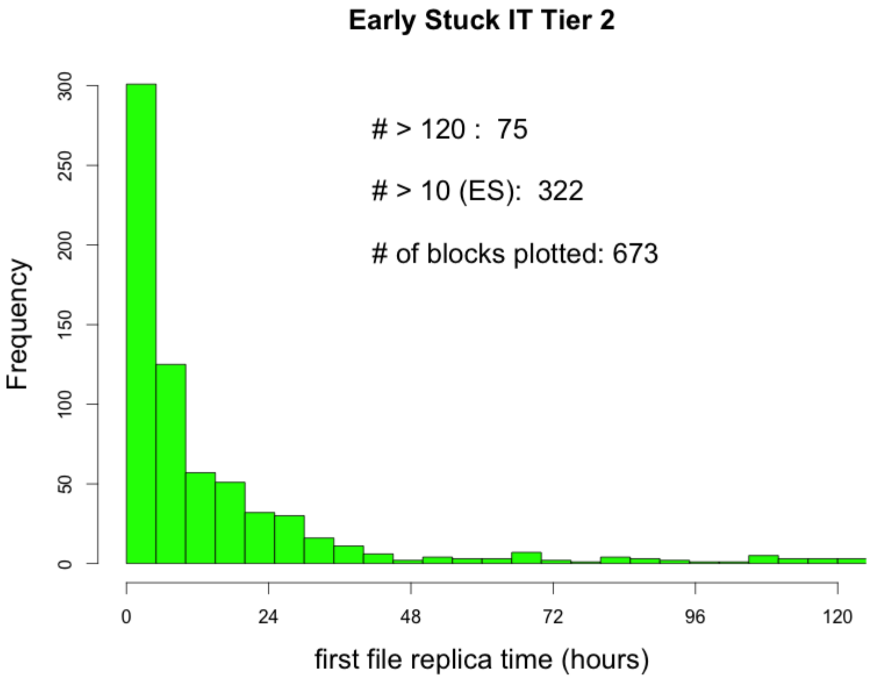


Figure 4.18: Time for the first file replica from IT Tier 2 sources.

4.3 Late Stuck analysis

This other kind of latency has already been defined in Section 3.1.4: it happens when one or few files take much longer to get transferred than the rest of the block. Figure 4.27 represent the typical trend of a block with a late stuck latency. The first 95% of the block, in fact, is transferred with a rate that is roughly 3 order of magnitude higher than the last 5%. In principle, if the transfer rate were flat, one would expect to see a bisector line.

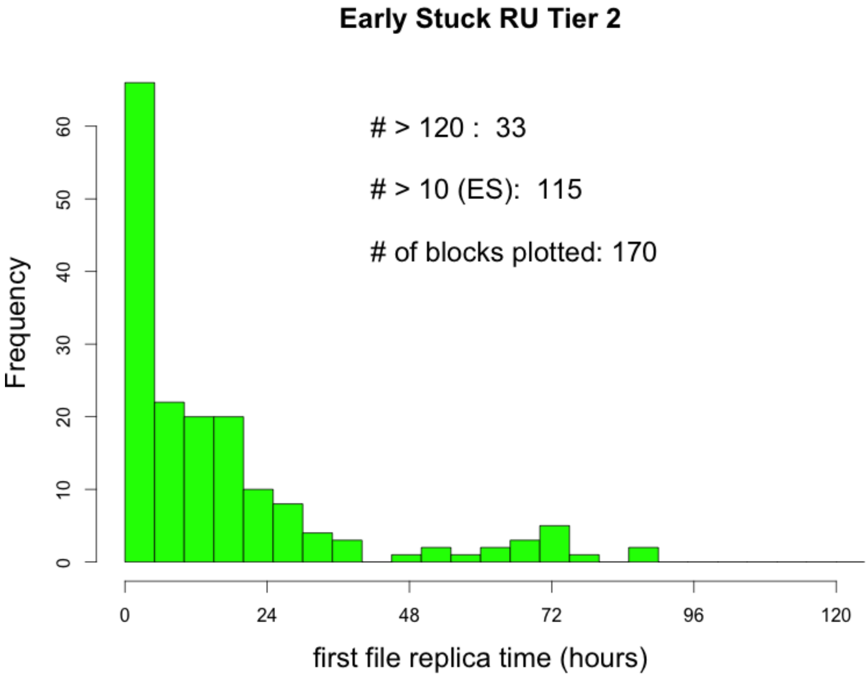


Figure 4.19: Time for the first file replica from RU Tier 2 sources.

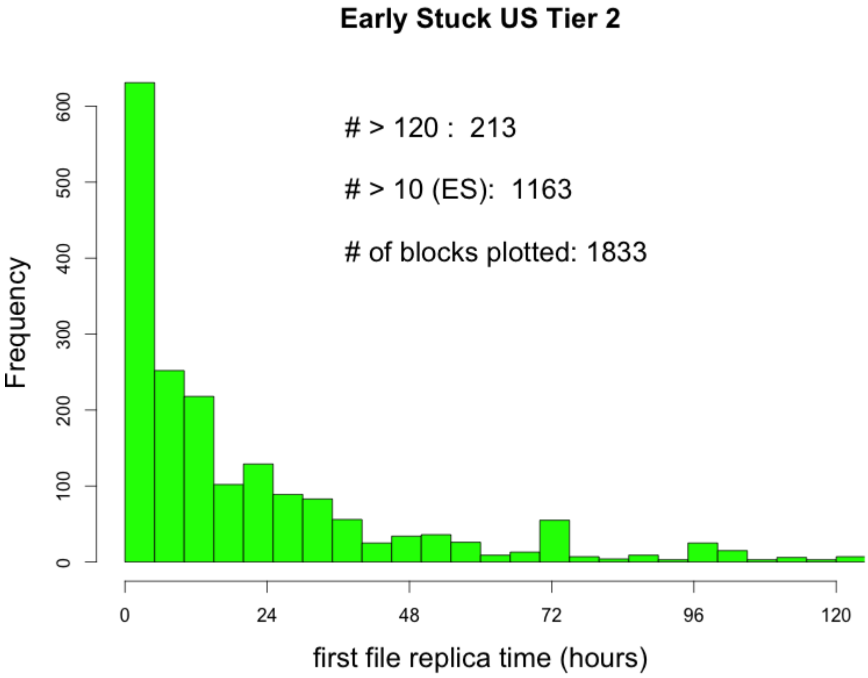


Figure 4.20: Time for the first file replica from US Tier 2 sources.

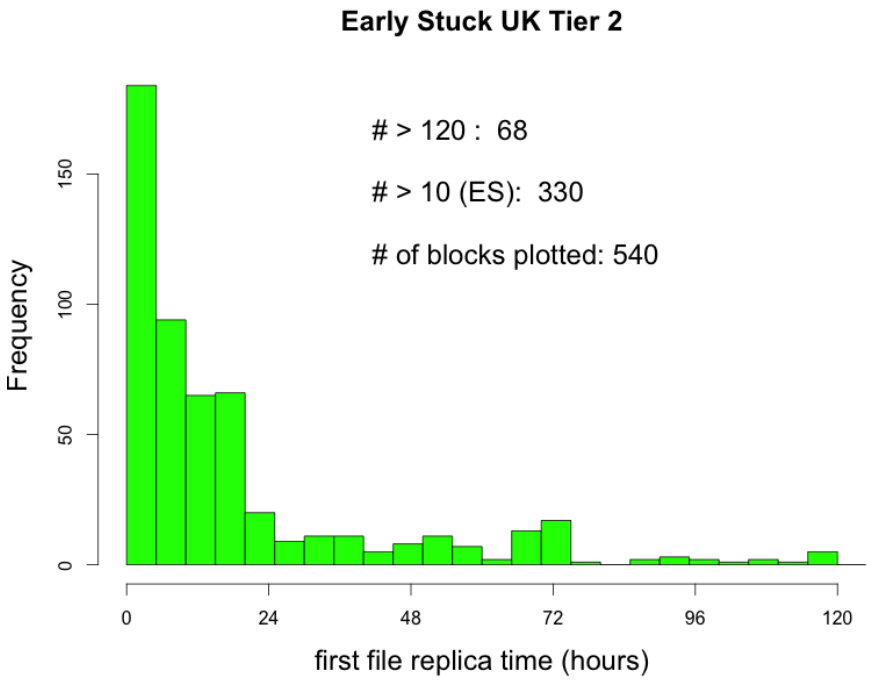


Figure 4.21: Time for the first file replica from UK Tier 2 sources.

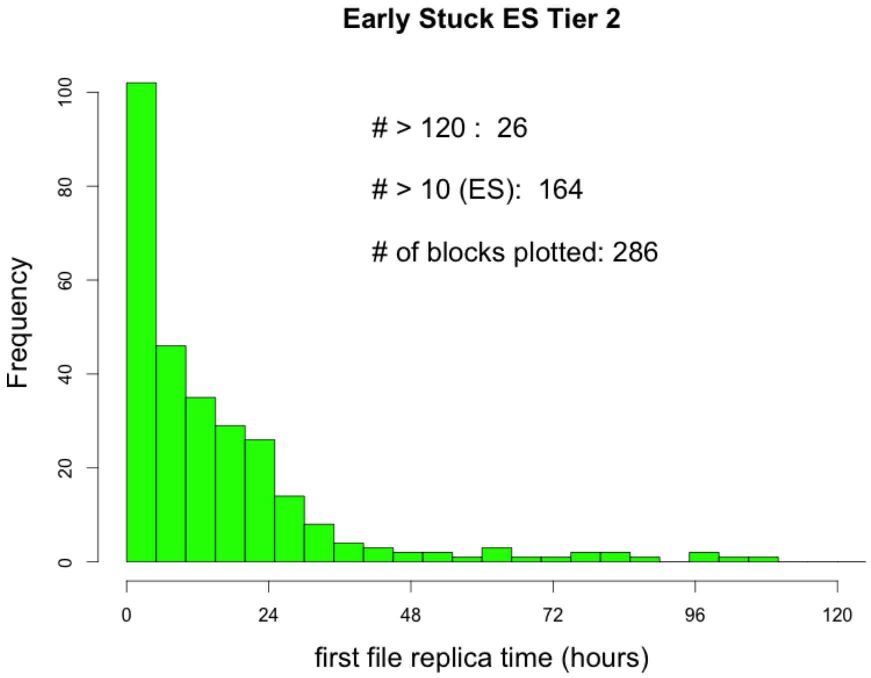


Figure 4.22: Time for the first file replica from ES Tier 2 sources.

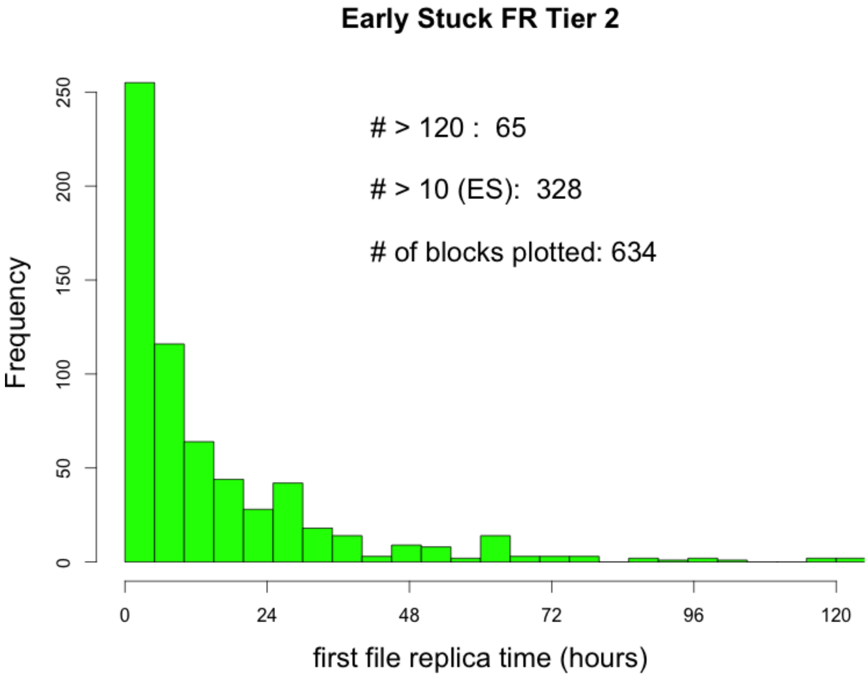


Figure 4.23: Time for the first file replica from FR Tier 2 sources.

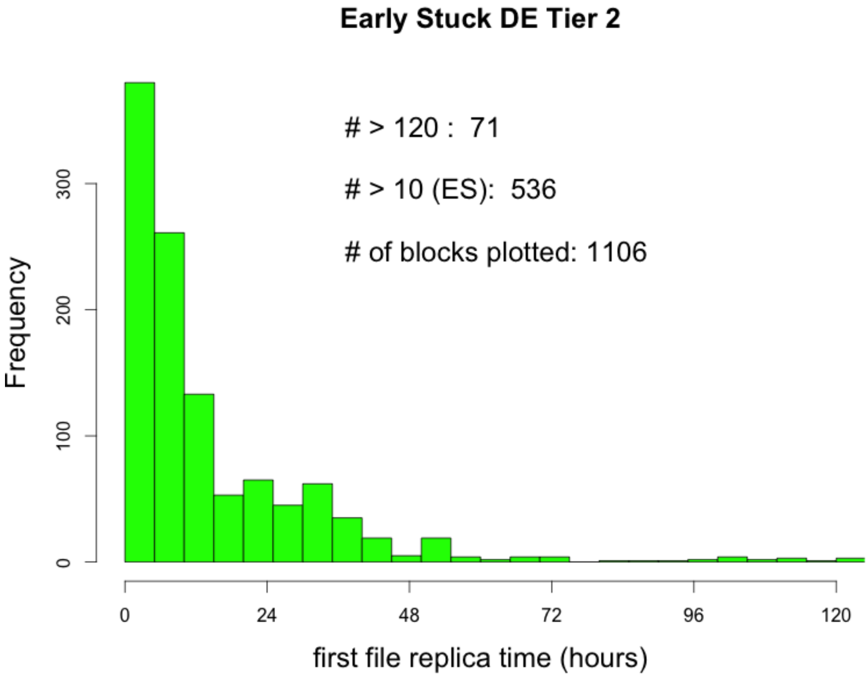


Figure 4.24: Time for the first file replica from DE Tier 2 sources.

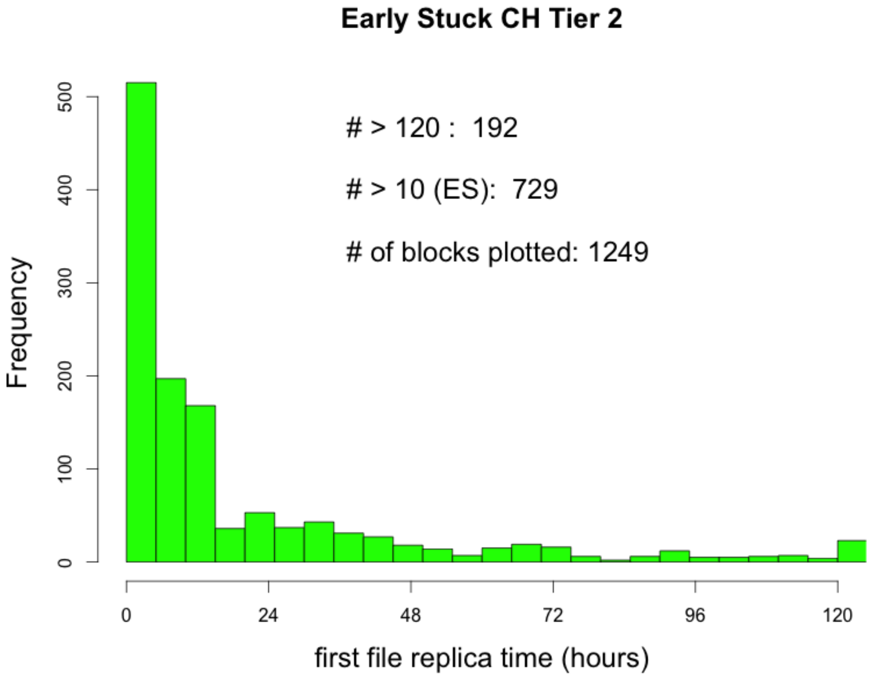


Figure 4.25: Time for the first file replica from CH Tier 2 sources.

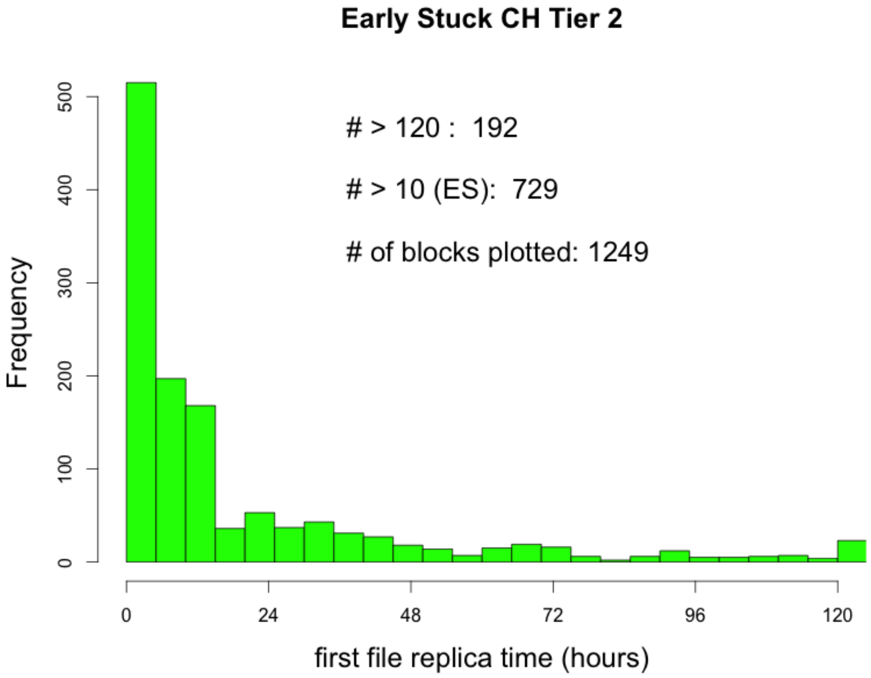
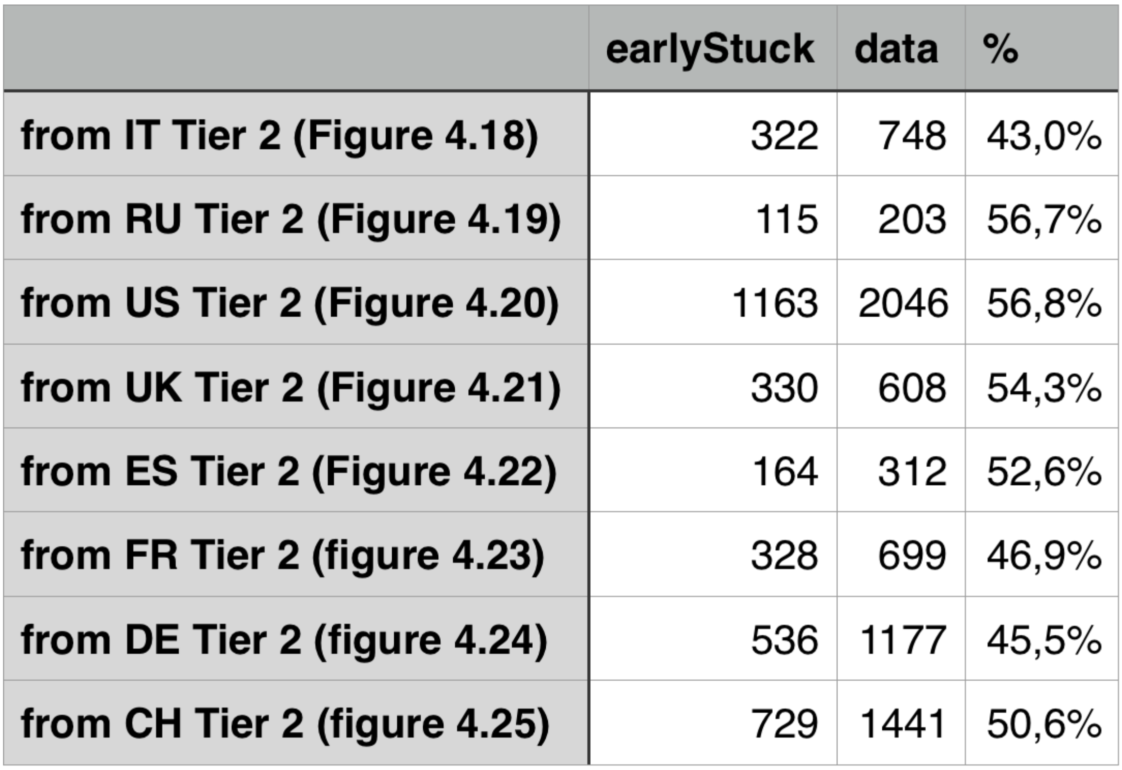


Figure 4.26: Time for the first file replica from CH Tier 2 sources



	earlyStuck	data	%
from IT Tier 2 (Figure 4.18)	322	748	43,0%
from RU Tier 2 (Figure 4.19)	115	203	56,7%
from US Tier 2 (Figure 4.20)	1163	2046	56,8%
from UK Tier 2 (Figure 4.21)	330	608	54,3%
from ES Tier 2 (Figure 4.22)	164	312	52,6%
from FR Tier 2 (figure 4.23)	328	699	46,9%
from DE Tier 2 (figure 4.24)	536	1177	45,5%
from CH Tier 2 (figure 4.25)	729	1441	50,6%

Table 4.4: Fraction of early stuck files and relative percentage for Tier 2 divided by nation.

In figure 4.28, a study of the Late Stuck transfers is shown. The hypothesis to check is that a large fraction of Late Stuck transfers are indeed tails in the transfer themselves due to blocks with one or more corrupted/missing files, whose transfer can hence only fail and it is tagged by PhEDEx as a transfer to be retried at a later stage. In fact, as already explained in Chapter 3, PhEDEx tries to maximize the throughput in every transfer task i.e. in case a file transfer fails in a transfer task, it passes to another file in the same task and comes back to the "queue" files only at the end: if one file cannot be transferred at all, it may be retried several times at the end of the task, thus resulting in a "tail" i.e. Late Stuck transfers. The figure shows the total transfer attempts versus the number of files in the block for all transfers: each point in the figure refers to a single block. It is meaningful that the visible line is the bisector line: each block with N files has been hit with at least N transfer attempts, i.e. if every single file transfer in a block of N files succeeds at its first attempts, a point at x=N and y=N is shown on the plot. All the points above this line are instead showing blocks for which one or more transfers had to be retried several times before succeeding (or being frozen and marked as "problematic" and hence requiring operators manual intervention). Two features can be observed. Firstly, blocks with fewer files tend to have a larger number of retrials, hence contributing more to the pool of Late Stuck transfers. Secondly, an unexpected larger number of transfer attempts is measured for blocks with exactly 100 files. Investigating the latter in more depth, a check with PhEDEx developers was performed, aimed at checking if special values of N are used in PhEDEx for some reasons, or if a configuration parameter in the CMS production tools is used that may cuts block size at N=100 files. No configuration parameter has been identified: the blocks with N=100 have been found to be only 1% of the total, with one exception: in the blocks created in 2011 (Summer11/Fall11/Run2011 production campaign, in the CMS jargon) there seems to be a larger fraction of blocks with exactly N=100 blocks, corresponding to about 5%. These 2011 blocks are hence populating the "N=100 peak" more than the rest, thus potentially implying that those blocks for some reasons may have been blocked for longer than the average. This is not particularly relevant in this work, so we can ignore the "N=100 peak" with no lack of generality. It must also be stated that in itself, the fact that we observe retrials as high as 1500 times is not a worry for PhEDEx: the system is designed to retry forever every once in a while, so if a file is blocked it will continue to fail. That number only depends on the reaction time of the CMS transfer team that needs to manually intervene and fix the root cause of the problem (e.g. by invalidating the corrupted file, so it stops being retried for transfer). It can be seen that only very few blocks actually have this issues of very high number of transfer attempts, thus indicating that this happened on a very small fraction of the overall CMS transfers: this may just indicate that a better monitoring/alarming procedure needs to be designed and put in place to address these cases.

In Figure 4.29 the same information as of Figure 4.28 is shown, despite restricting the sample only to the blocks that our classification labels as Late Stuck blocks. In this plot, as we are selecting only the Late Stuck blocks, if the aforementioned hypothesis were entirely correct one would expect (in principle) to see an absent bisector line, with all points somewhere in the upper side of the line. It can be observed that the most points are indeed in the upper part of the plot and the line is not entirely absent despite evidently less marked - thus suggesting that the hypothesis may be largely (while not entirely) correct indeed. Table 4.5 highlights the argument stated above. Another interesting observation is now possible: the fraction of Late Stuck seems to be evidently dominated by blocks with small number of files (less than about 50 or so). This may just be an effect of the fact that indeed the blocks with fewer files are a majority, hence they also dominate the Late Stuck population.
This plot may be useful to PhEDEx developers and an implementation (run-time maybe) will prevent block tails with a quick manual intervention.

Another important aspect to investigate is the time for the completion of the last 5% of the block, that is crucial for Late Stuck latencies. Figure 4.30 show this information computed for all data entries while Figure 4.31 is limited on data tagged as having Late Stuck issues: the latter in particular highlights the cut of 5 hours defined in Chapter 3 by eq. 3.5.

At last, differences between Tiers may be very important for latency issues as depicted in Figure 4.32 where a counting of block transfer, grouped bu source and destination Tier type, has been made for all transfer and for transfer tagged as Late Stuck. Notice that the contribution of Tier 0/1 sources is much lower in the sample with latency issues. As expected, the most critical connection is between

Tier 2 and Tier 2 where links may not be commissioned properly.

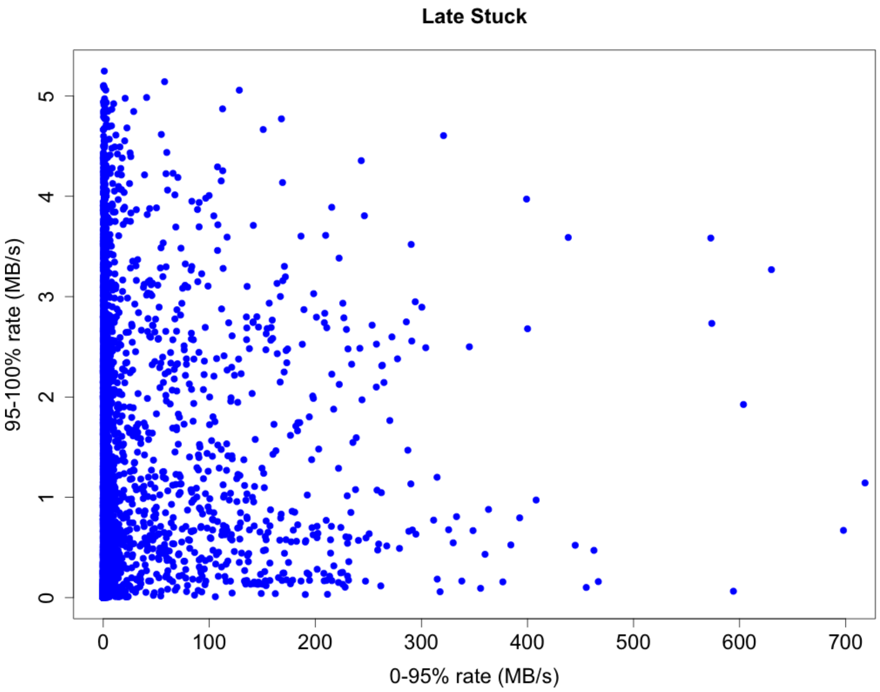
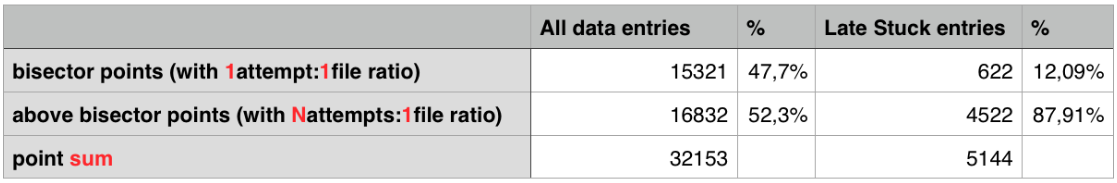


Figure 4.27: Transfer rate of the last 5% of the block vs rate of the first 95%. For transfers tagged as having latency tails according to eq. 3.5.



	All data entries	%	Late Stuck entries	%
bisector points (with 1attempt:1file ratio)	15321	47,7%	622	12,09%
above bisector points (with Nattempts:1file ratio)	16832	52,3%	4522	87,91%
point sum	32153		5144	

Table 4.5: Table showing the percentage of file belonging to the bisector line versus the percentage of file above the bisector line, for all data entries as well as Late stuck tagged entries.

4.4 Other types of latencies

Latencies are a very complex and manifold argument and even a categorization may result difficult. Equations 3.6 and 3.5 introduced in Chapter 3 introduce some sensible yet arbitrary parameters that approximate the definitions of Early Stuck and Late Stuck mentioned above. However there are latencies with do not belong to any of these categories for several reasons. We called them "Stuck Other" and , as Figure 4.34 show, they represent (in yellow) a considerable fraction of all latencies. Due to their complexity and variability, we will not analyze them in this thesis and we leave this for future work.

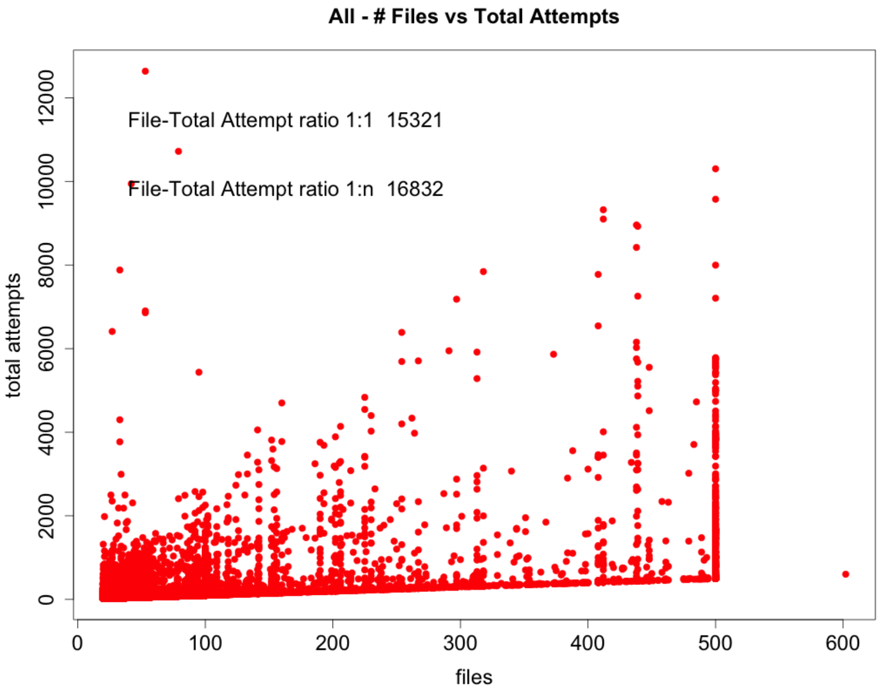


Figure 4.28: Bidimensional plot showing the number of file in a datablock vs the total attempts for that block, for all entries.

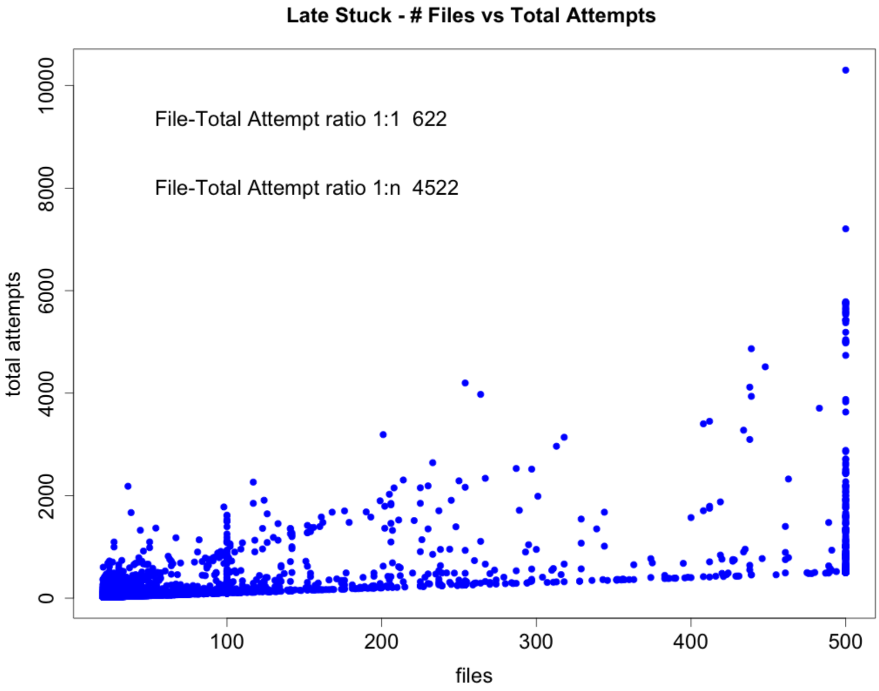


Figure 4.29: Bidimensional plot showing the number of file in a datablock vs the total attempts for that block, for data tagged has having Late Stuck latency.

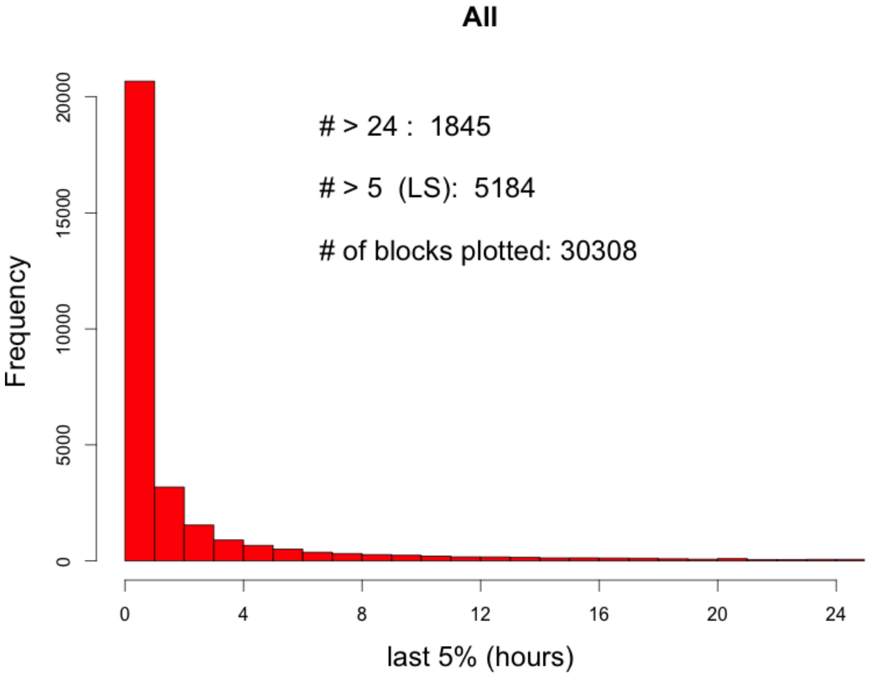


Figure 4.30: Time required for the last 5% replica for all data entries.

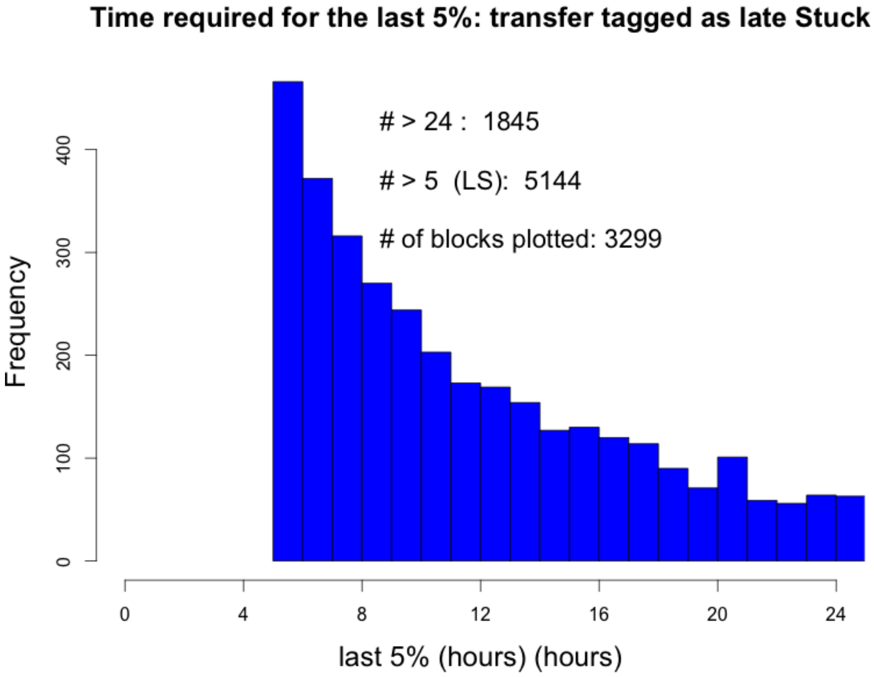
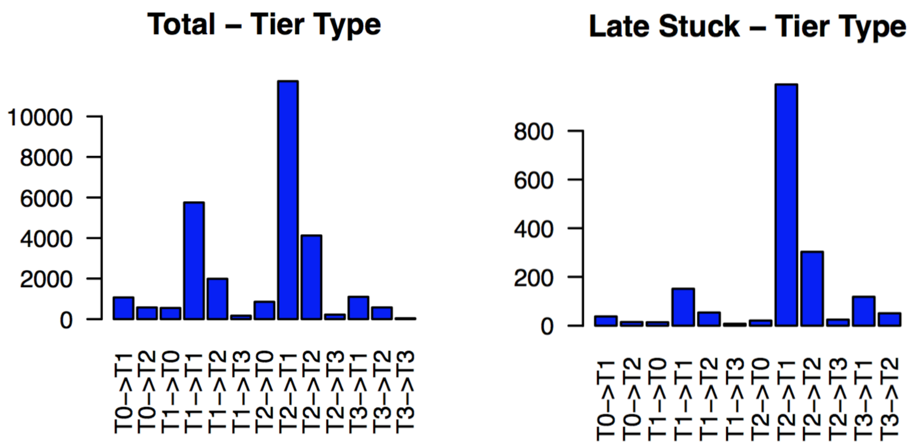


Figure 4.31: Time required for the last 5% replica for data tagged as having Late Stuck issue accordingly to eq. 3.5.



(a) Counting of block transfers for all transfer groped by source and destination Tier type.

(b) Counting of LateStuck-defined block transfers for groped by source and destination Tier type.

Figure 4.32: Counting of block transfers groped by source and destination Tier type.

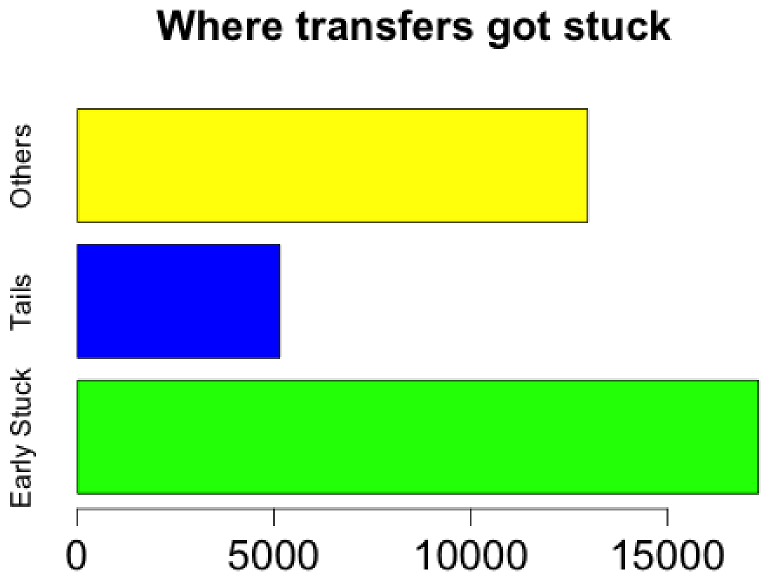


Figure 4.33: Number of data entries belonging to each latency category.

4.5 Implementation of latency plots for the CMS Computing shifts

All the variables of interest identified in this latency studies, and the vast majority of plots produced, could be implemented in the PhEDEx web interface (Figure 3.1) as already stated in this chapter. However, they have a value also in another aspect of the CMS Computing operations. The CMS Computing project overviews the CMS Computing shifts, aimed to enforce systematic and procedural controls over the overall computing infrastructure. It is a running activity since 2008 and running on a 24/7 basis, profiting of a (growing) team of >150 people worldwide. CMS Computing foresees a Computing Shift Person (CSP) on shift for 8 hours in a row (i.e. 3 shifters per day for a 24/7 coverage, in different continents to profit of the time zone and avoid the need of night shifts), plus a Computing Run Coordinator (CRC), an expert on-call for 7 days in a row. The CSP procedures are set in ad-hoc documentation, and constantly improving. The main duty is to monitor all CMS Computing system on specific monitoring pages, including PhEDEx ones. Warning and alarm are triggered to operators, site contacts, and expert on-call per activity. Monitoring tools and overview systems are widely improved by and used for regular computing shifts. The shift activity is hence the perfect place to exploit new, interesting monitoring information for streamlining always better the CMS Computing operations. Some of the plots produced in this thesis will be considered for inclusion in the PhEDEx monitoring system and in the standard monitoring sources for the CMS CSP shifters.

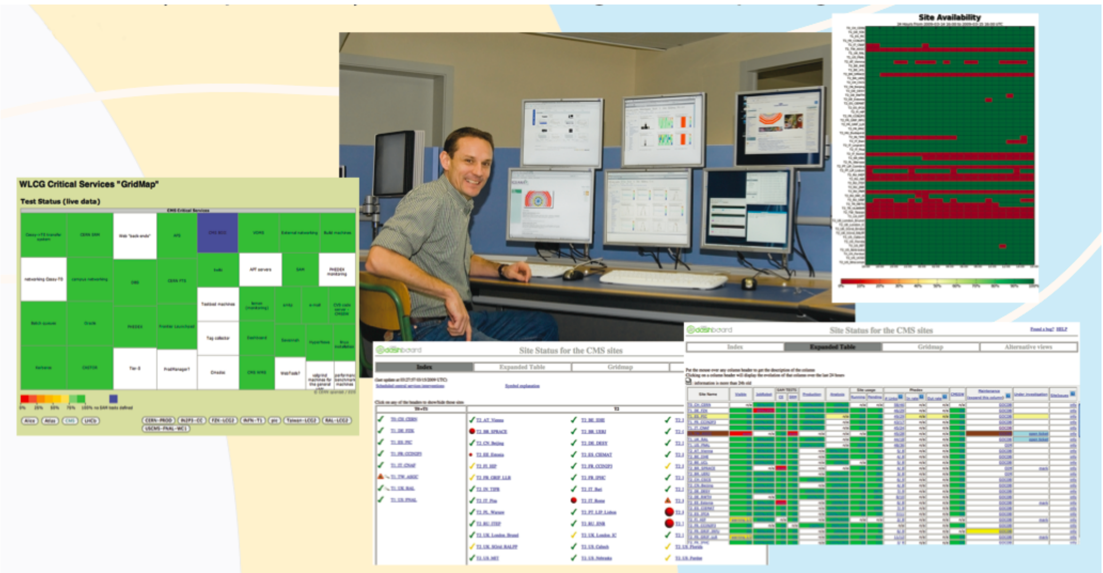

WLCG Critical Services "GridMap"
Test Status (live data)
Site Availability
Site Status for the CMS sites
Site Status for the CMS sites

Figure 4.34: Photo showing the CMS Computing Shift Person (CSP).

Conclusions

PhEDEx is one of the crucial components of the CMS Computing system. Thanks to data collection and filtering during Run-1, the CMS computing project has a large set of data concerning the latencies as observed in all transfers happening between all Tiers. This thesis investigated, on a Petabyte-scale, these data with the aim of increasing transfer performances hence improve the overall quality of the system itself.
Starting from a sample of roughly 3 million entries, we classified these latencies in 2 different categories: Early Stuck and Late Stuck. The first category (which represents the 54% of the analyzed sample) is populated by those blocks with serious performance issues that start flowing only after some time; the second category instead (which represents the 16% of the analyzed sample) is caused by some files that, for some reason, are corrupted and takes much longer to get transferred than the rest of the block. After a careful selection of the data deemed acceptable for analysis, some variables has been extrapolated e.g. times for the completion of a certain percentage of the block or completion rates. Also "skew" variables, not completely intuitive, have been defined ad-hoc for latency issues and selecting the most relevant to a particular case has required a collaboration with PhEDEx developers. Combining all of these information and using a programming language specific for statistic analysis called R, we finally suggested a series of interesting plots whose purpose could be their future production realtime during CMS operations and implementation in a dedicated section of the PhEDEx monitoring web pages: they would allow to spot and promptly address data subscriptions with latency issues since the very beginning thus allowing PhEDEx operators to intervene and quickly fix. A selection of the plots produced in this thesis are also being considered for inclusion in the PhEDEx official monitoring and have been considered to be useful enough to be exposed as the standard monitoring sources to the CMS CSP shifters, i.e. the Computing Shift Persons on shift for 8 hours in a row who offer a 24/7 monitoring coverage of the CMS Computing operations.
This is still a work in progress: approximations has been made and a certain degree of uncertainty exist; plans for the short term are in fact a better definition of these latency categories in order to produce purer plots and identify with more precision ill transfers for the upcoming Run 2. Plans for the long term are to implement, as stated above, these plots into CMS monitoring services for a better and faster control on WLCG transfers.
Some aspects and results of this work have also been presented at the 21st International Conference on Computing in High Energy and Nuclear Physics [38].

Bibliography

[1] Oliver Sim Brüning et al. *LHC Design Report*. Ed. by CERN library copies. Vol. 1, 2, 3. 2012. URL: http://ab-div.web.cern.ch/ab-div/Publications/LHC-DesignReport.html (cit. on pp. 1, 3).

[2] Lyndon Evans and Philip Bryant. "LHC Machine". In: *Journal of Instrumentation* 3.08 (2008). Ed. by IOPscience, S08001. URL: http://iopscience.iop.org/1748-0221/3/08/S08001 (cit. on pp. 1, 3).

[3] *CERN*. URL: http://www.cern.ch (cit. on p. 1).

[4] *CERN Engineering*. URL: http://home.web.cern.ch/about/engineering/vacuum-empty-interstellar-space (cit. on p. 2).

[5] *CERN Engineering*. URL: http://home.web.cern.ch/about/engineering/pulling-together-superconducting-electromagnets (cit. on p. 2).

[6] *CERN Engineering*. URL: http://home.web.cern.ch/about/engineering/radiofrequency-cavities (cit. on p. 3).

[7] *The ALICE experiment*. URL: http://home.web.cern.ch/about/experiments/alice (cit. on p. 3).

[8] The ALICE Collaboration et al. "The ALICE experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008). Ed. by IOPscience, S08002. URL: http://iopscience.iop.org/1748-0221/3/08/S08002 (cit. on p. 3).

[9] *The Atlas Experiment*. URL: http://home.web.cern.ch/about/experiments/atlas (cit. on p. 4).

[10] The ATLAS Collaboration et al. "The ATLAS Experiment at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3.08 (2008). Ed. by IOPscience, S08003. URL: http://iopscience.iop.org/1748-0221/3/08/S08003 (cit. on p. 4).

[11] *The CMS Experiment*. URL: http://home.web.cern.ch/about/experiments/cms (cit. on p. 4).

[12] The CMS Collaboration et al. "The CMS experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08004. URL: http://stacks.iop.org/1748-0221/3/i=08/a=S08004 (cit. on pp. 4, 6, 9).

[13] *The LHCb Experiment*. URL: http://home.web.cern.ch/about/experiments/lhcb (cit. on p. 5).

[14] The LHCb Collaboration et al. "The LHCb Detector at the LHC". In: *Journal of Instrumentation* 3.08 (2008). Ed. by IOPscience, S08005. URL: http://iopscience.iop.org/1748-0221/3/08/S08005 (cit. on p. 5).

[15] *The LHCf experiment*. URL: http://home.web.cern.ch/about/experiments/lhcf (cit. on p. 5).

[16] The LHCf Collaboration et al. "The LHCf detector at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3.S08006 (2008). Ed. by IOPscience. URL: http://iopscience.iop.org/1748-0221/3/08/S08006 (cit. on p. 5).

[17] *The TOTEM Experiment*. URL: http://home.web.cern.ch/about/experiments/totem (cit. on p. 5).

[18] The TOTEM Collaboration et al. "The TOTEM Experiment at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3.S08007 (2008). Ed. by IOPscience. URL: http://iopscience.iop.org/1748-0221/3/08/S08006 (cit. on p. 5).

[19] Ian Bird. "Computing for the Large Hadron Collider". In: *Annual Review of Nuclear and Particle Science* 61.1 (2011), pp. 99–118. DOI: 10.1146/annurev-nucl-102010-130059. eprint: http://dx.doi.org/10.1146/annurev-nucl-102010-130059. URL: http://dx.doi.org/10.1146/annurev-nucl-102010-130059 (cit. on pp. 11, 13).

[20] *WLCG Project*. URL: http://www.cern.ch/lcg (cit. on p. 11).

[21] *Enabling Grind for E-sciencE (EGEE)*. URL: http://www.eu-egee.org (cit. on p. 11).

[22] *European Grid Infrastructure (EGI)*. URL: http://www.egi.eu/ (cit. on p. 11).

[23] *Open Science Grid (OSG)*. URL: http://www.opensciencegrid.org (cit. on p. 11).

[24] Daniele Bonacorsi (for the CMS Computing Model). "Experience with the CMS Computing Model form commissioning to collision". In: *Journal of Physics*. Conference Series 331.7 (2010). Ed. by IOPscience, p. 072005. URL: http://iopscience.iop.org/1742-6596/331/7/072005 (cit. on pp. 12, 20).

[25] Daniele Bonacorsi (on behalf of the CMS Collaboration). "The CMS Computing Model". In: *Nuclear Physics B - Proceedings Supplements*. Conference Series 172.53-56 (2007). Ed. by Science Direct. URL: http://www.sciencedirect.com/science/article/pii/S092056320700552X (cit. on p. 12).

[26] *MONARC project*. URL: http://monarc.web.cern.ch/MONARC (cit. on p. 12).

[27] *LHCOPN*. URL: http://lhcopn.web.cern.ch (cit. on p. 13).

[28] G. L. Bayatyan et al. *CMS computing: Technical Design Report*. Technical Design Report CMS. Submitted on 31 May 2005. Geneva: CERN, 2005 (cit. on pp. 15, 18).

[29] M. Cinquilli et al. "The CMS workload management system". In: *Journal of Physics*. Conference Series 396.3 (2012). Ed. by IOPscience, p. 032113. URL: http://iopscience.iop.org/1742-6596/396/3/032113 (cit. on p. 15).

[30] M. Giffels et al. “The CMS Data Management System”. In: *Journal of Physics*. Conference Series 513.4 (2014). Ed. by IOPscience, p. 042052. URL: http://iopscience.iop.org/1742-6596/513/4/042052 (cit. on p. 15).

[31] Tony Wildish et al. “From toolkit to framework - the past and future evolution of PhEDEx”. In: *Journal of Physics*. Conference Series 396.3 (2012). Ed. by IOPscience, p. 032118. URL: http://iopscience.iop.org/1742-6596/396/3/032118 (cit. on p. 16).

[32] J. Rehn et al. “PhEDEx high-throughput data transfer management system”. In: *CHEP06* (2006). Ed. by GridPP. URL: http://www.gridpp.ac.uk/papers/chep06_tuura.pdf (cit. on p. 16).

[33] M Giffels, Y Guo, and D Riley. “Data Bookkeeping Service 3 – Providing event metadata in CMS”. In: *Journal of Physics: Conference Series*. Conference Series 513.4 (2014), p. 042022. URL: http://stacks.iop.org/1742-6596/513/i=4/a=042022 (cit. on p. 16).

[34] G Ball et al. “Data Aggregation System - a system for information retrieval on demand over relational and non-relational distributed data sources”. In: *Journal of Physics: Conference Series* 331.4 (2011), p. 042029. URL: http://stacks.iop.org/1742-6596/331/i=4/a=042029 (cit. on p. 16).

[35] *ORACLE*. URL: http://www.oracle.com/ (cit. on p. 16).

[36] R. Egeland et al. “The PhEDEx next-gen website”. In: *Journal of Physics*. Conference Series 396.3 (2012). Ed. by IOPscience, p. 032117. URL: http://iopscience.iop.org/1742-6596/396/3/032117 (cit. on p. 16).

[37] R. Egeland, C.-H. Huang, and T. Wildish. “PhEDEx Data Service”. In: *Journal of Physics*. Conference Series 219.6 (2010). Ed. by IOPscience, p. 062010. URL: http://iopscience.iop.org/1742-6596/219/6/062010 (cit. on p. 16).

[38] D.Bonacorsi et al. “Monitoring data transfer latency in CMS computing operations”. In: *CHEP 2015* (2015) (cit. on pp. 17, 23, 25–29, 31, 58).

[39] Giuseppe Codispoti et al. “CRAB: A CMS Application for Distributed Analysis”. In: *Nuclear Science Symposium Conference Record* N02.79 (2008). Ed. by IEEE. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4774652 (cit. on p. 17).

[40] *Software Guide on CRAB*. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideCrab (cit. on p. 17).

[41] Alessandra Fanfani et al. “Distributed Analysis in CMS”. In: *CMS-NOTE-2009-013, CERN-CMS-NOTE-2009-013* (2009). URL: http://inspirehep.net/record/875969 (cit. on p. 17).

[42] M. Cinquilli et al. “CRAB3: Establishing a new generation of services for distributed analysis at CMS”. In: *Journal of Physics*. Conference Series 396.3 (2012). Ed. by IOPscience, p. 032026. URL: http://iopscience.iop.org/1742-6596/396/3/032026 (cit. on p. 17).

[43] Andres Tanasijczuk. *CRAB3 architecture and task workflow*. Oct. 23, 2014. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/CRAB3TaskFlow (cit. on p. 18).

[44] Daniele Bonacorsi and Tony Wildish. "Challenging data and workload management in CMS Computing with network-aware systems". In: CMS-CR-2013-373 (2013). URL: `https://cds.cern.ch/record/1626815/` (cit. on p. 19).

[45] Ian Bird et al. "Update of the Computing Models of the WLCG and the LHC Experiments". In: *Nuclear Physics B - Proceedings Supplements* LCG-TDR-002 (Apr. 15, 2014). Ed. by CERN-LHCC-2014-014. URL: `http://cds.cern.ch/record/1695401/files/LCG-TDR-002.pdf?version=1` (cit. on p. 19).

[46] Daniele Bonacorsi and Anthony Wildish. *Challenging Data Management in CMS Computing with Network-aware Systems*. Tech. rep. CMS-CR-2013-426. Geneva: CERN, 2013. URL: `https://cds.cern.ch/record/1977895/` (cit. on p. 19).

[47] T Chwalek et al. "No file left behind - monitoring transfer latencies in PhEDEx". In: *Journal of Physics: Conference Series* 396.3 (2012), p. 032089. URL: `http://stacks.iop.org/1742-6596/396/i=3/a=032089` (cit. on p. 27).

[48] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2015. URL: `http://www.R-project.org/` (cit. on pp. 31, 32).