Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Corso di Laurea Magistrale in Fisica

Elastic Computing on Cloud Resources for the CMS Experiment

Relatore:
Prof. Daniele Bonacorsi

Correlatore:
Dott. Giuseppe Codispoti
Dott. Claudio Grandi

Presentata da:
Riccardo Di Maria

Sessione I
Anno Accademico 2014/2015

"I am on the edge of mysteries
and the veil is getting
thinner and thinner."

Louis Pasteur

Abstract

Al giorno d'oggi, la gestione dei dati e l'analisi dei dati in Fisica delle Alte Energie richiede una ingente potenza di calcolo e di storage. In particolare, LHC Computing Grid (WLCG), un'infrastruttura e insieme di servizi sviluppati e distribuiti da una vasta comunitá di fisici e informatici in tutto il mondo, ha dimostrato di essere un punto di svolta in termini di efficienza di analisi dati durante Run-I ad LHC, giocando un ruolo fondamentale nella scoperta bosone di Higgs.

Recentemente, il paradigma Cloud computing sta emergendo e raggiungendo un notevole livello di adozione da parte di molte differenti organizzazioni scientifiche e non solo. Cloud permette di accedere e utilizzare ingenti risorse computazionali, non di proprietá, condivise tra molte comunitá scientifiche. Considerando gli impegnativi requisiti della fisica LHC in Run-II ed oltre, la comunitá informatica LHC é interessato ad esplorare Clouds e vedere se possono fornire un approccio complementare - o anche una valida alternativa - alle soluzioni tecnologiche esistenti basate su Grid.

All'interno della comunitá di LHC, approcci Cloud sono adottati da numerosi esperimenti, ed in particolare l'esperienza dell'esperimento CMS risulta di rilevanza per questa tesi. Il Run-II ad LHC é appena iniziato, e le soluzioni basate su Cloud sono giá in produzione per CMS. Tuttavia, altri approcci di utilizzo Cloud vengono pensati e sono al livello di prototipale, come il lavoro svolto in questa tesi. Questo sforzo é di fondamentale importanza per fornire CMS delle capacitá elastiche e flessibili di accesso ed utilizzo di risorse di calcolo, necessarie per affrontare le sfide di Run-III e Run-IV.

Lo scopo principale di questa tesi é quello di presentare approcci Cloud all'avanguardia che consentono all'esperimento CMS di utilizzare risorse on-demand, allocate cioé dinamicamente in base alle esigenze. Inoltre, l'accesso diretto a queste risorse Cloud viene presentato come adeguato caso d'uso per far fronte alle esigenze esperimento CMS.

Il Capitolo 1 presenta una panoramica di Fisica delle Alte Energie ad LHC e dell'esperienza CMS in Run-I, nonché la preparazione per Run-II. Il Capitolo 2 descrive l'attuale modello di calcolo adottato da CMS, ed il Capitolo 3 fornisce approcci Cloud perseguiti e utilizzati all'interno della Collaborazione CMS. Il Capitolo 4 ed il Capitolo 5 discutono il lavoro originale e all'avanguardia svolto in questa tesi di sviluppo e di test di prototipi funzionanti per quanto riguarda l'estensione elastica di risorse di calcolo CMS su Clouds, ed il calcolo " as a Service " per la Fisica delle Alte Energie. Inoltre é dimostrato l'impatto di tale lavoro su un caso di utilizzo standard di fisica per CMS.

Abstract

Nowadays, data handling and data analysis in High Energy Physics requires a vast amount of computational power and storage. In particular, the world-wide LHC Computing Grid (LCG), an infrastructure and pool of services developed and deployed by a ample community of physicists and computer scientists, has demonstrated to be a game changer in the efficiency of data analyses during Run-I at the LHC, playing a crucial role in the Higgs boson discovery.

Recently, the Cloud computing paradigm is emerging and reaching a considerable adoption level by many different scientific organizations and not only. Cloud allows to access and utilize not-owned large computing resources shared among many scientific communities. Considering the challenging requirements of LHC physics in Run-II and beyond, the LHC computing community is interested in exploring Clouds and see whether they can provide a complementary approach - or even a valid alternative - to the existing technological solutions based on Grid.

In the LHC community, several experiments have been adopting Cloud approaches, and in particular the experience of the CMS experiment is of relevance to this thesis. The LHC Run-II has just started, and Cloud-based solutions are already in production for CMS. However, other approaches of Cloud usage are being thought of and are at the prototype level, as the work done in this thesis. This effort is of paramount importance to be able to equip CMS with the capability to elastically and flexibly access and utilize the computing resources needed to face the challenges of Run-III and Run-IV.

The main purpose of this thesis is to present forefront Cloud approaches that allow the CMS experiment to extend to on-demand resources dynamically allocated as needed. Moreover, a direct access to Cloud resources is presented as suitable use case to face up with the CMS experiment needs.

Chapter 1 presents an overview of High Energy Physics at the LHC and of the CMS experience in Run-I, as well as preparation for Run-II. Chapter 2 describes the current CMS Computing Model, and Chapter 3 provides Cloud approaches pursued and used within the CMS Collaboration. Chapter 4 and Chapter 5 discuss the original and forefront work done in this thesis to develop and test working prototypes of elastic extensions of CMS computing resources on Clouds, and HEP Computing "as a Service". The impact of such work on a benchmark CMS physics use-cases is also demonstrated.

Contents

Introduction

The successful LHC data taking in Run-I and the Long Shutdown 1 have led the LHC experiments to face new challenges in the design and operation of the computing facilities. The High Energy Physics data handling and data analysis requires a large amount of computing power and available resources, although the computing infrastructure for Run-II is dimensioned to cope at most with the average amount of data recorded. Anyhow, breakneck use cases could overload the infrastructure. As a matter of fact, usage peaks originating large backlogs have been already observed during Run-I with the result of delaying the completion of the data reconstruction and ultimately the data availability for physics analysis. The usage peaks are axiomatically common during data taking and the available computing resources are often not sufficient to deal with them. Moreover, the time required to absorb backlogs could be long-lasting, hindering the needs of the experiments.

The world-wide LHC Computing Grid (WLCG) infrastructure has demonstrated to be a game changer in the efficiency of data analyses during Run-I at the LHC However, it does not allow to dynamically allocate resources. Therefore, the CMS experiment is exploring the opportunity to access Cloud resources provided by external partners or commercial providers in order to cope with the several challenges. The feasibility has already been demonstrated as specific use cases have already been explored and Cloud-based solutions have been successfully exploited during LS1.

The underlying work of this thesis presents the proof of concept of the elastic extension of a CMS site, specifically the Bologna Tier-3, on external Cloud resources referring as "Cloud Bursting". The elastic extension is tested using real physics use cases, specifically the conversion of the CMS reconstructed events in a lightweight format suitable for the analysis, in order to provide the CPU efficiency of the newly instantiated resources, and the close to last step of the analysis for the Top Quark mass measurement in the all hadronic channel, performed in the Bologna CMS group.

Moreover, a direct access and integration of Cloud resources to the CMS Workload Management system is explored referring as "Computing-as-a-Service". This approach, already in use at the CMS Tier-0, has been expanded to the case of a generic CMS site and has been tested for the first time with the new CMS Workload Management tools. In this context, Cloud allows to access and use opportunistic computing resources as the

CMS experiment needed. Thus, this work turns out to be of paramount importance to be able to provide CMS with the capability to elastically and flexibly access and utilize on-demand computing resources.

The thesis is going to report the usability of the implemented models, together with an evaluation of the performances of the on-demand allocated resources. Furthermore, the technical challenges and the next steps toward a production system are discussed, along with the impact of such work on a benchmark CMS physics use-cases.

Chapter 1

High Energy Physics at LHC

1.1 The Large Hadron Collider at CERN

The *Large Hadron Collider* (LHC) [1] is the largest superconducting proton-proton and heavy ions collider, located at CERN (*European Organization for Nuclear Research*) [2]. The LHC is installed in the 26.7 km long tunnel that previously hosted LEP (Large Electron-Positron) Collider [3], at about 100 m beneath the Swiss-French border near Geneva. The purpose of the LHC is to test the Standard Model of Particle Physics, explore new energy frontiers and look for new physics.

The LHC was able to produce the first $\sqrt{s} = 900\ GeV$ $p-p$ collisions on November 23, 2009. After some pilot runs at $\sqrt{s} = 900\ GeV$ and $\sqrt{s} = 2.36\ TeV$, the 7 TeV center-of-mass energy was reached on March 30, 2010.
Finally, the center-of-mass energy was raised to $\sqrt{s} = 8\ TeV$ on April 5, 2012. This energy was maintained until the LHC was shut down for maintenance and upgrade.
In Spring 2015, the LHC has been restarted with a center-of-mass energy of $\sqrt{s} = 13\ TeV$.

The mechanism of protons (or heavy ions) production, injection, and acceleration up to 6.5 TeV proceeds in several steps (Figure 1.1).
In the very first step, the production of protons takes place through the ionization of hydrogen atoms. Hence, protons are accelerated in two steps by the Linear Accelerator up to 50 MeV, and by the Proton Synchrotron Booster up to 1.4 GeV.
The Proton Synchrotron accelerates the particles up to 26 GeV using 277 conventional electromagnets that push the protons to 99.9% the speed of light. Then, proton bunches reach the Super Proton Synchrotron, a circular particle accelerator with a circumference of 7 km, where they are accelerated up to 450 GeV.
Finally, protons are injected into the LHC in two separate pipelines in which they move in opposite directions. Here, the particles can be accelerated up to their maximum energy of 6.5 TeV. It has to be noted that the above pre-accelerator stages are critical in order to reach the $\sqrt{s} = 13\ TeV$ center-of-mass energy.

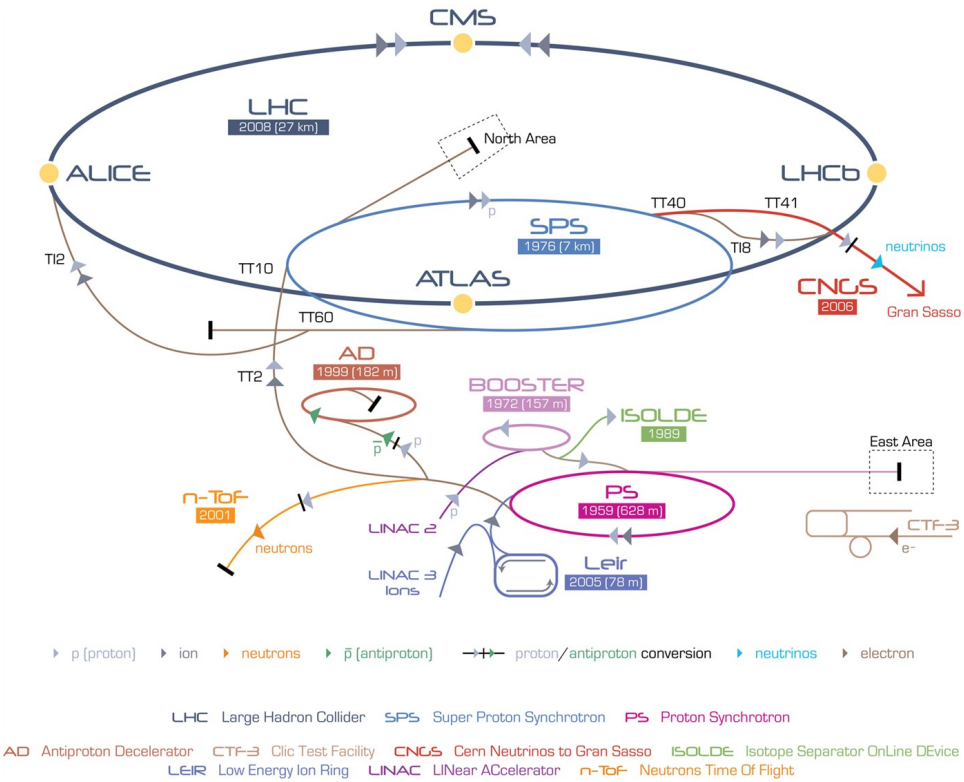

Figure 1.1: The LHC's injection chain composed of multiple smaller pre-accelerators.

Moreover, a vacuum system is necessary so the particles do not lose energy in the acceleration process due to impacts with the molecules that constitute air. The LHC Vacuum System is made up of 3 components: the insulation vacuum for Cryomagnets, the insulation vacuum for Helium distribution, and the beam vacuum.

The LHC consists in two adjacent parallel beam pipes separated by 194 *mm*, in which about 1400 bunches of protons circulate clockwise (Beam-1) and counterclockwise (Beam-2). The LHC experiments are located in the proximity of the interaction points of the two beam pipes. The four main LHC experiment detecting the final states of *p-p* (or heavy ions) collisions are:

ALICE *A Large Ion Collider Experiment*, the experiment especially designed to study the quark-gluon-plasma state of matter in *Pb-Pb* or *p-Pb* collisions [4];

ATLAS *A Toroidal LHC Apparatus*, a general purpose experiment [5];

CMS *Compact Muon Solenoid*, a general purpose experiment [6];

LHCb *LHC beauty*, the beauty-quark physics devoted experiment [7].

The LHC makes use of powerful superconducting magnets. A strong magnetic field B is needed in order to maintain the two beams in a circular trajectory. Since the colliding particles have the same charge, each beam must be provided with a magnetic field opposite respect to the other. Therefore, the LHC uses twin bore coil dipole magnets, hosting the two beam lines, that are installed inside the same mechanical structure to reduce the space required by the equipment (Figure 1.2). A powerful cryogenic system regulates the temperature at 1.9 K in order to favor the correct operation of the apparatus. The intensity of the magnetic field B necessary to maintain protons in a circular trajectory is given by:

$$B\ [\mathrm{T}] = \frac{p\ [\mathrm{TeV}]}{0.3\ r\ [\mathrm{km}]}$$

where p is the proton momentum and r is the LHC radius ($r \simeq 4.2\ km$).

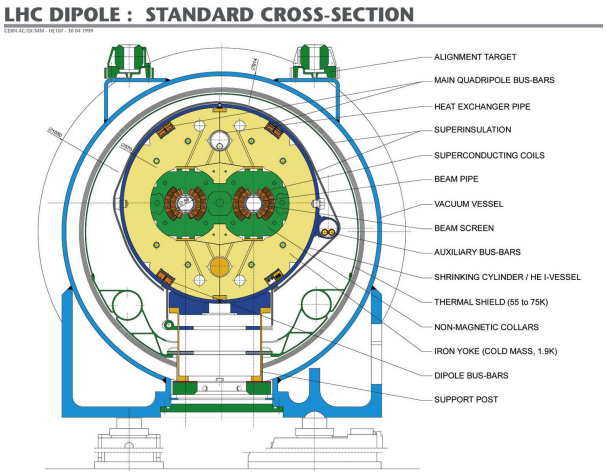

Figure 1.2: Diagram showing the cross-section of an LHC dipole magnet.

About $\frac{2}{3}$ of the beam line is equipped with 1,232 magnet coils each measuring 14.3 m in length, made up of copper-clad niobium-titanium cables. Over 96 *tons* of liquid Helium are used to keep the temperature down to the critical value of 1.9 K, and 392 quadrupole magnets are employed to focus the beams approaching the detectors.

The event rate produced, namely the number of events per second, is given by:

$$R = \mathcal{L}\ \sigma$$

where σ is the production cross-section of the physics process, and $\mathcal{L}$ is the luminosity, expressed by:

$$\mathcal{L} = f\ \frac{n_1 n_2}{4\pi\sigma_x\sigma_y}$$

where f is the bunch crossing frequency, n_1 and n_2 are the particles contained in Bunch-1 and Bunch-2 respectively, and σ_x and σ_y are the transverse dimensions of the beam.

A 25 ns bunch crossing (BX) interval is being adopted in Run-II, while it was 50 ns in Run-I (up to 2012). The ATLAS and CMS experiments are designed to operate at high luminosity ($10^{34}\ cm^{-2}\ s^{-1}$), while the instantaneous luminosity delivered to LHCb and ALICE are $10^{32}\ cm^{-2}\ s^{-1}$ and $10^{27}\ cm^{-2}\ s^{-1}$ respectively.

1.2 The CMS Experiment at the LHC

The LEP collider at CERN and the Tevatron [8] at Fermilab have provided remarkable insights into, and precision tests of, the Standard Model of Particle Physics. However, a number of questions remain unanswered on which LHC has to dwell.

Until 2012, the principal concern was the lack of any direct evidence for the Higgs boson, the particle resulting from the Higgs mechanism [9] which provides an explanation for the masses of elementary particles.
Although this concern was rejected by the ATLAS and CMS experiments [10, 11], other questions remain unanswered, including uncertainties in the behaviour of the Standard Model at high energies, the lack of any particle physics explanation for Dark Matter, and the reasons for the imbalance of matter and antimatter observed in the Universe.

The Compact Muon Solenoid (CMS) is one of two large general-purpose particle physics experiment at the LHC [12]. The CMS collaboration is formed by approximately 2317 people, representing 185 scientific institutes and 42 countries, who built and now operate the detector [13].
The goal of the CMS experiment is to explore and investigate a wide range of physics at the TeV scale. In fact, it allows to study the electroweak symmetry breaking due to the Higgs mechanism, and the properties of the recently found Higgs boson. In this way, these measurements along with very precise measurements of known electroweak and flavor physics phenomena could definitively confirm the Standard Model.
The CMS experiment allows to explore in depth QCD processes at extreme conditions of temperature, density and energy. Moreover, the search for physics beyond the Standard Model can be performed, which could involve the presence of Super-Symmetric particles, Z'/W' new heavy gauge bosons, or particles that could make up Dark Matter.

Accordingly, a manifold experimental concept is needed in order to achieve the aforementioned goals as discussed in the next Section.

1.3 The CMS Detector

The CMS Detector [14, 15] is 21.6 metres long, 14.6 metres in diameter, and its total weight is approximately 14,000 tonnes (Figure 1.3), and it is located in an underground cavern at Cessy in France, just across the border from Geneva.

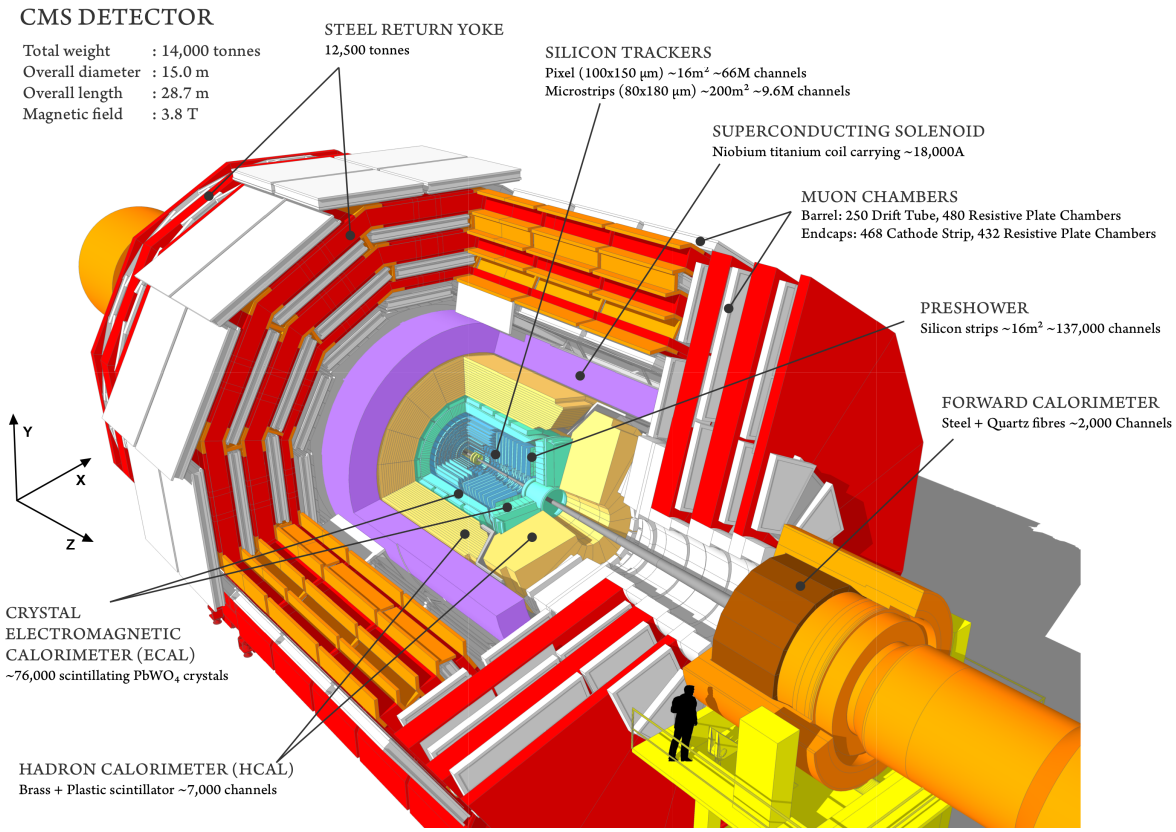

Figure 1.3: Representation of CMS and its different parts: the silicon Tracker (blue), the Electromagnetic Calorimeter (green-blue), the Hadronic Calorimeter (orange), the Magnet (purple), and the Muon chambers (white).

A proper reference frame has to be defined in order to describe physics quantities and detector geometry.

Coordinate Frame

The CMS interaction point represents the origin of a right-handed cartesian reference frame, defined as follows:

- the x-axis is horizontal, pointing towards the center of the LHC ring;
- the y-axis is vertical, pointing upwards;
- the z-axis is tangent to the beam line.

In this way, the $x-y$ plane results orthogonal to the beam pipe, while the z-axis defines the longitudinal direction.

The CMS Detector is characterized by a cylindrical structure and symmetry. Thus, cylindrical coordinates can be used in reconstruction algorithms, defined as:

r : distance from the interaction point in the transverse plane $x-y$, $r=\sqrt{x^2+y^2}$;

ϕ : azimuthal angle, measured from the x-axis in the transverse plane;

θ : polar angle, measured from the z-axis in the longitudinal plane $y-z$.

The cylindrical coordinates can also be used to define several useful variables:

particle momentum $p=\sqrt{p_z^2+p_T^2}$, $\quad p_T=\sqrt{p_x^2+p_y^2}$;

transverse energy $E_T=E\ \sin\theta$;

transverse mass $m_T=\sqrt{m^2+p_T^2}$;

missing transverse energy (MET) $E_T^{\ missing}=-\sum_i \vec{p}_T^{\,i}$;

rapidity $y=\frac{1}{2}\ln\frac{E+p_z}{E-p_z}$;

pseudo-rapidity $\eta=-\ln\left(\tan\frac{\theta}{2}\right)$.

The LHC operational regime turns out to be challenging for the CMS Detector. In fact, the expected rate is $R=\mathcal{L}\cdot\sigma_{pp}\simeq 10^9\ Hz$, considering the design luminosity $\mathcal{L}\simeq 10^{34}\ cm^{-2}\ s^{-1}$ and the expected $p-p$ cross-section $\sigma_{pp}\simeq 100\ mb$ at $\sqrt{s}=14\ TeV$. Moreover, the high instantaneous peak luminosity leads to pile-up effects, namely the overlapping of many events in the same data acquisition time interval. Out Of Time pile-up effects are due to the Run-II bunch crossing interval of 25 ns. This issue was challenging also at $\sqrt{s}=8\ TeV$, where the resulting pile-up was about 20 overlapped events. Finally, the radiations are able to cause damage and ageing of the sensitive detector materials in regions close to the beam pipe.

The features and characteristics of the CMS Detector and its Data Acquisition System (DAQ) are designed and developed in order to cope with these challenges. A two-level on-line trigger system selects physics signals and reduces the rate from $10^9\ Hz$ to 1 kHz. The sub-detectors are provided with high granularity to reduce the occupancy and high time-resolution to resolve multiple interaction vertices. Further, detectors and electronics devices are of hard-radiation type.
The achievement of the physics goals of the experiment requires proper particle identification and event reconstruction. In fact, high-quality physics objects with excellent characteristics are provided by CMS, such as:

- high MET and di-jet mass resolution, achievable with a hermetic and fine segmentation calorimeter structure;
- high electromagnetic (EM) energy resolution, precise di-electron and di-photon mass resolution in a wide η range;
- high-reliable and precise muon identification, muon charge determination, and high momentum resolution, yielding to high precise calculation of di-muon invariant mass in a wide range of angles and momentum.

The CMS Detector can be described dividing its structure into three main sections: *barrel*, *endcaps*, and *very forward regions*. The first section represents the central region, and it is composed of five "wheels", coaxial to the beam. Orthogonally to the beam axis, two structures hermetically close the *barrel* at both ends, identifying the second section. These structures are composed of three disks each. Finally, the *very forward regions* are made up of sub-detectors close to the beam axis to detect particles in a very-high pseudo-rapidity range.

A more detailed knowledge of the detector and its sub-system is provided in the following Sections.

1.3.1 Pixel and Tracker

The CMS experiment provides the largest tracker system ever built for a collider experiment, located close to the interaction point. The CMS Tracker [16] is composed of high granularity silicon pixel detectors in the inner region. On the other hand, silicon micro-strip detectors constitute the outer region. In this way, a clever charged particle track reconstruction and primary and secondary vertices reconstruction is allowed in a high particles density condition. Moreover, the strong magnetic field provided by the CMS Magnet [Ref. 1.3.4] allows precise momentum measurements.

The detector is characterized by a low occupancy due to its high granularity, a fast response, and a large redundancy obtained using many layers to collect more than 10 hits along the particle trajectory. In this way, several main goals are achievable, such as high efficiency in the whole η range down to very low p_T, good particle momentum reconstruction, efficient primary and secondary vertex reconstruction, and good pattern recognition. The latter consists in the recognition of all the hits produced by a single particle in the sensitive material.

Accordingly, the CMS Tracker (Figure 1.4) is entirely made of silicon detectors that cover the region $|\eta| < 2.5$ with a radius $r < 1.2\ m$ and for $|z| < 2.7\ m$, for $5.8\ m$ in length and a total surface of $210\ m^2$. The thickness (in radiation lengths X_0) of the silicon sensors varies as a function of the pseudo-rapidity, as for instance:

- 0.35 X_0 at small η;
- 1.8 X_0 in the transition region between barrel and endcap;
- 1.1 X_0 at $|\eta| \simeq 2.5$.

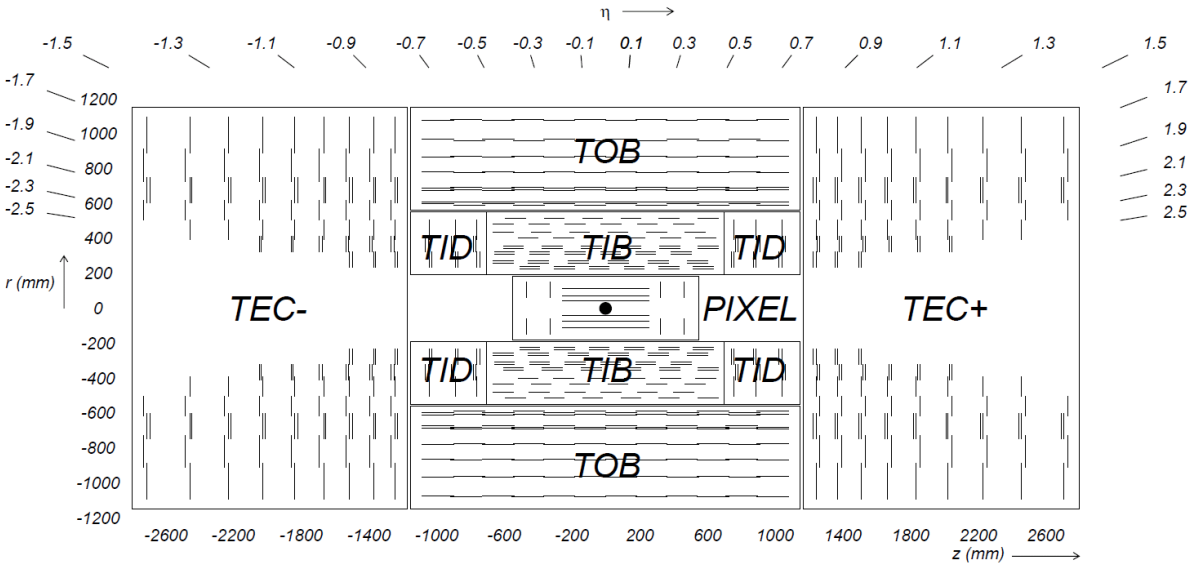

Figure 1.4: Schematic cross section through the CMS Tracker. Each line represents a Detector module. Double lines indicate back-to-back modules which deliver stereo hits.

The silicon detectors are categorized in *Pixel* and *Microstrip*. The Pixels provide very low occupancy, high resolution, and precise vertex reconstruction. Hence, these detectors are chosen to cover the region close to the beam pipe. On the other hand, Microstrips cover the more extended region outside the Pixel detectors, and allow to reduce the number of read-out channels, thus maintaining a good resolution.

Pixel Detector

The Pixel Detector is the closest detector to the interaction point, suffering a very high particle flux. It is made up of about 6.6×10^7 pixel cells, each of 100×150 μm^2, clustered in about 1,400 sensors for a total surface of 1.06 m^2. They are located as follows:

BPix 3 layers in the barrel region, each 53 cm long, at a radius $r = 4.4$ cm, $r = 7.3$ cm, and $r = 10.2$ cm respectively;

FPix 2 disks for each endcap, each made up of 24 blades in a turbine-like shape, at a radius $r = 7.3$ cm and $r = 15$ cm respectively.

A spatial resolution of 10 μm in the transverse plane $r - \phi$ and of 15 μm in the z-coordinate are achieved in the barrel region, while lower resolutions (15 μm and 20 μm respectively) are achieved in the endcap regions.

Silicon Strip Tracker

The Microstrip Detector region is divided as follows:

Inner Region (20 $cm < r < 55$ cm) is composed of 4 layers in the barrel (TIB, Tracker Inner Barrel) and 3 disks in each endcap (TID, Tracker Inner Disk);

Outer Region (55 $cm < r < 120$ cm) is composed of 6 layers in the barrel (TOB, Tracker Outer Barrel) and 9 disks in each endcap (TEC, Tracker EndCap).

A spatial resolution of $40 \div 60$ μm in the transverse plane $r - \phi$ and of 500 μm in the z-coordinate is achieved.

1.3.2 EM Calorimeter

The CMS Electromagnetic Calorimeter (ECAL) [17] is an homogeneous and hermetic calorimeter, preceded by pre-shower detectors in the endcap regions (Figure 1.5). It is made of more than 75,000 lead-tungstenate ($PbWO_4$) scintillating crystals that are able to detect the EM shower produced through Bremsstrahlung and pair production. In this way, the ECAL allows the identification of photons and electrons thanks to the energy they deposit in the material.

The scintillating material provides several crucial characteristics relating to the study of EM physics phenomena. It has high density ($\rho = 8.28$ $\frac{g}{cm^3}$), short radiation length ($X_0 = 0.89$ cm), and small Molière radius ($R_M = 21.2$ $\frac{X_0}{\epsilon_c}$ $MeV = 2.2$ cm). Furthermore, almost 80% of the light is collected by silicon avalanche photo-diodes in the barrel and vacuum photo-triodes in the endcaps due to very short scintillation time (25 ns) provided by this material. Hence, these characteristics allow the ECAL to have fine granularity and to be compact and fast.

The CMS ECAL is divided in two regions: Barrel ECAL and Endcap ECAL.

Barrel ECAL

The Barrel ECAL covers the pseudo-rapidity range $|\eta| < 1.479$. A thin-walled alveolar structure contains 61,200 crystals at a radius $r = 1.29$ m, each of which has a surface of 26 $mm \times 26$ mm and a length of 25.8 X_0 (or 230 mm). The crystals are mounted in a truncated pyramid geometry, tilted of 3° with respect to the axis from the interaction vertex, in both the ϕ and η directions in order to avoid cracks aligned with particle trajectories.

Endcap ECAL

The Endcap ECAL covers the pseudo-rapidity range $1.479 < |\eta| < 3$. The structure contains 7,324 crystals, each of which has a surface of 30 $mm \times 30$ mm and a length of 24.7 X_0 (or 220 mm), and are clustered in supercrystals.
The Endcap ECAL is preceded by a pre-shower detector in order to identify π^0 in the pseudo-rapidity region $1.653 < |\eta| < 2.6$. This latter is a sampling calorimeter and is made up of 2 layers, namely lead radiators to initiate the shower and silicon strip sensors to measure the deposited energy and the transverse shower profiles, for a total thickness of 20 cm.

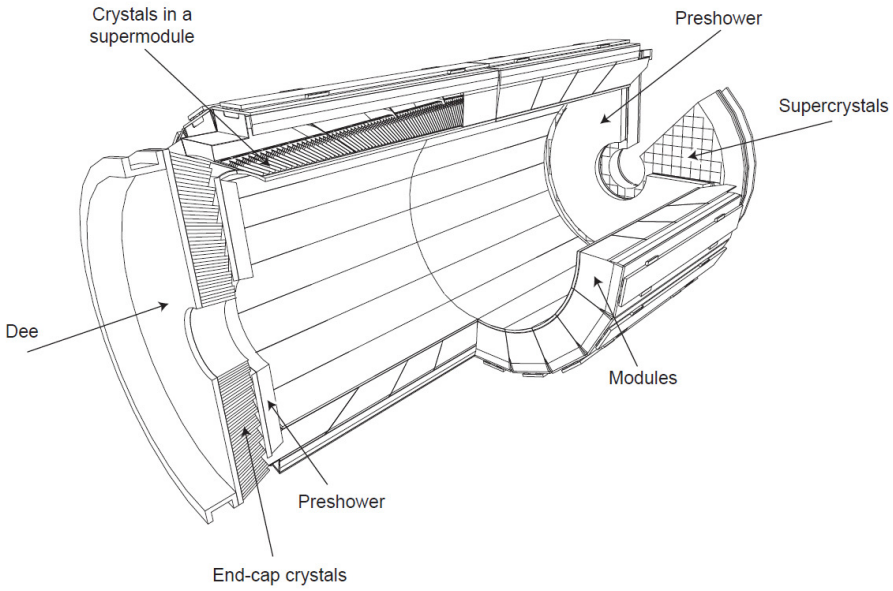

Figure 1.5: Layout of the CMS ECAL presenting the arrangement of crystal modules, supermodules, endcaps and the preshower in front.

The energy resolution of the CMS ECAL can be parametrized using terms: *stochastic* (S), *noise* (N), and *constant* (C).

$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{S}{\sqrt{E}}\right)^2 + \left(\frac{N}{E}\right)^2 + C^2$$

where E is the particle energy.
The stochastic term accounts for fluctuations in the number of photo-electrons produced and fluctuations in the shower-containment. The noise term accounts for electronics and pile-up noise. The constant term is related to the calorimeter calibration, and to the energy leakage of the crystals. Using test beams, the calibrated terms (with E expressed in GeV) turn out to be: $S = 2.8\%$, $N = 12\%$, $C = 0.3\%$.

1.3.3 Hadron Calorimeter

The CMS Hadron Calorimeter (HCAL) [18] is able to identify hadrons and measure hadron energy deposits in order to reconstruct jets and measure MET. This is achievable because it is hermetic up to $|\eta| = 5$.

The CMS HCAL is divided in two regions: Barrel and Endcap HCAL, and Forward HCAL.

Barrel and Endcap HCAL

The Barrel HCAL covers the pseudo-rapidity range $|\eta| < 1.26$, and the Endcap HCAL cover the pseudo-rapidity range $|\eta| < 3$. They are constrained between the ECAL and the Magnet, from $r = 1.77\ m$ to $r = 2.95\ m$.
The Barrel and Endcap HCAL are brass-scintillator sampling calorimeters. In fact, the brass is useful to obtain small shower dimension, being non-magnetic and having short interaction length (λ_0). Furthermore, they are coupled to hybrid photo-diodes using wavelength-shifting fibers.

Forward HCAL

The two Forward HCAL cover the pseudo-rapidity range $|\eta| < 5$ around the beam-pipe at $|z| = 11.2\ m$. They are designed to increase the hermeticity, and they present radiation-hard building materials due to the proximity to the beam line. In fact, steel plates are used as absorbers, while quartz fibers are used as active material (producing Cherenkov light at the passage of relativistic particles).

The Outer Calorimeter is added outside the magnet coil to improve the energy resolution of the Barrel HCAL, catching the tails of the hadron showers.

The depth of the HCAL is a function of the pseudo-rapidity, as can be seen from the following examples:

- 5.25 λ_0 at $|\eta| = 0$;
- 9.1 λ_0 at $|\eta| = 1.3$;
- 10.5 λ_0 at $|\eta| \simeq 5$.

The energy resolutions of the CMS HCAL for the different regions are:

Barrel HCAL $\frac{\sigma}{E} \simeq \frac{65\%}{\sqrt{E}} \oplus 5\%$

Endcap HCAL $\frac{\sigma}{E} \simeq \frac{85\%}{\sqrt{E}} \oplus 5\%$

Forward HCAL $\frac{\sigma}{E} \simeq \frac{100\%}{\sqrt{E}} \oplus 5\%$

where E is the particle energy expressed in GeV and $\oplus$ stands for sum in quadrature.

1.3.4 Magnet

The CMS Magnet [19] provides a significant bending power so that a precise measurements of the transverse momentum of charged particles is allowed, either in the tracker or in the iron yoke (Figure 1.6). It is made up of superconducting solenoidal coil providing a 3.8 T magnetic field. Hence, the charged particle tracks are bended in the tracker and a particle charge and momentum identification is provided.

The structure provides a length of 12.5 m with an inner diameter of 6 m, and a total weight of 220 $tons$. The solenoid is made of Niobium-Titanium (NbTi) cables wrapped with copper, and it is kept at a temperature of 4 K to maintain the superconducting mode. Moreover, it is located in a vacuum cylinder in order to isolate it from the outside. An external iron yoke is responsible for the return of the magnetic field, and completes the architecture above. The yoke is made of 5 layers in the barrel and 3 disks for each endcap region, with a total weight of $10,000$ $tons$. It has a length of 14 m and it is able to absorb almost all particles, except for muons and neutrinos.

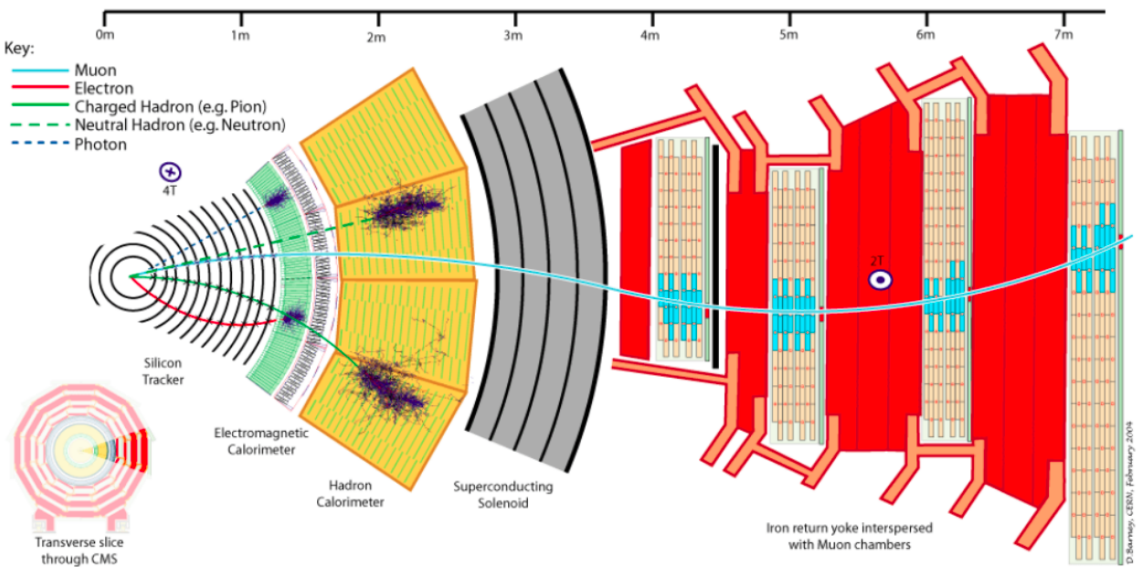

Figure 1.6: Transverse view of the CMS Detector, with the signature of muons (cyan line), electrons (red line), charged hadrons (green line), neutral hadrons (green dotted line), photons (blue dotted line).

1.3.5 Muon Detectors

The CMS Muon System [20] is the outer part of the CMS experiment and it is designed to allow muon identification. It is hosted by the return-yoke region of the superconducting Magnet [Ref. 1.3.4], and it consists of Drift Tubes (DTs) in the central region and Cathode Strip Chambers (CSCs) in the endcap regions. The redundancy is ensured by Resistive Plate Chambers (RPCs) matching with DTs and CSCs.
The magnetic bending power created by the return flux ($B \simeq 1.8\ T$) provides a standalone muon p_T measurement crucial for the trigger system.

The CMS Muon Detector covers the pseudo-rapidity region $|\eta| < 2.4$ and is entirely made up of gaseous detectors. They are based on the use of ionization electrons, that are created by the passage of charged particles in the gas, to produce signals.
Three different types of gaseous detectors are employed:

Drift Tubes are used in the barrel region and cover the pseudo-rapidity region $|\eta| < 1.2$;

Cathode Strip Chambers are used in the endcap regions and cover the pseudo-rapidity range $0.9 < |\eta| < 2.4$;

Resistive Plate Chambers cover the pseudo-rapidity region $|\eta| < 1.6$ in order to improve the DTs and CSCs performances and ensure redundancy.

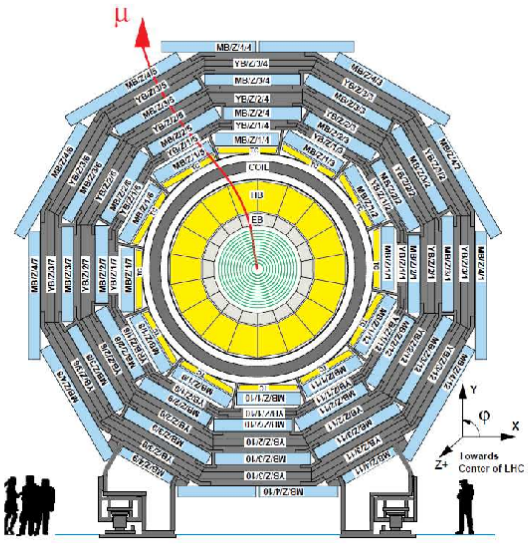

Figure 1.7: Transverse view of the CMS Barrel Muon System. Each wheel consists of 12 sectors formed by DTs (light blue) embedded in the yoke (gray).

The Muon System (Figure 1.8) has an overall space resolution of 250 μm in the $r-\phi$ plane and of 500 μm in the z-direction. Moreover, the reconstruction efficiency is close to 100%. The overall structure comprises two regions:

Muon Barrel holds 4 stations of DTs and RPCs, that are divided into 5 wheels in the z-direction (Figure 1.7);

Muon Endcap is made up of 4 disks orthogonal to the beam axis where CSCs and RPCs are located.

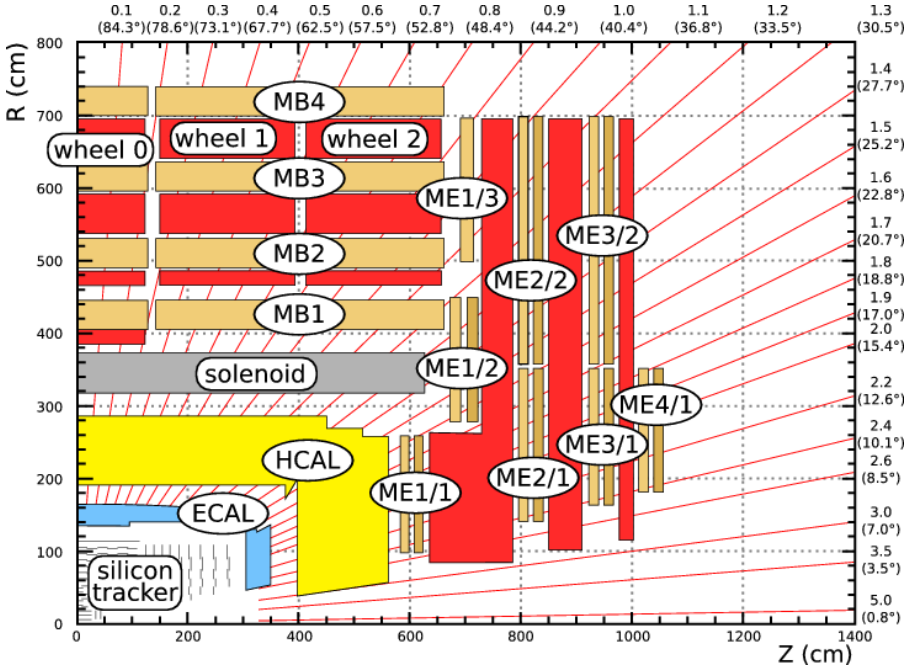

Figure 1.8: Longitudinal view of the CMS Muon System: DTs are colored in green, CSCs in blue, RPCs in red.

Drift Tubes

The Drift Chambers are made up of Drift Tubes and are located in the barrel region. They are organized in four stations:

MB1, MB2, MB3 contain 8 layers of drift cells used to measure the muon position in the $r-\phi$ plane, and 4 layers used to measure the z coordinate;

MB4 contain 8 layers of drift cells used to measure the muon position in the $r-\phi$ plane.

The drift cell covers an area of 4.2 $cm \times 1.3$ cm and is filled with a mixture of Ar (85%) and CO_2 (15%).

Cathode Strip Chambers

The multi-wire proportional chambers are located in the endcap regions, and have the cathode plane segmented in strips orthogonal to the anode wire in order to have a $2-D$ information about the muon position. The endcap regions easily holds CSCs due to their fan-shape.

The CSC is made up of trapezoidal panels mounted on 8 disks, 4 in each endcap, partially overlapping in the ϕ-plane to improve the coverage and efficiency. Furthermore, it is filled with a mixture of Ar (30%), CO_2 (50%), and CF_4 (20%).

Resistive Plate Chambers

The RPCs are placed in the barrel and in the endcap regions in order to ensure redundancy. They are made of four Bakelite planes which form two gaps filled with a mixture of $C_2H_2F_4$ (96.5%) and C_4H_{10} (3.5%). The presence of the double gap provides high efficiency with electric fields lower than that of single-gap chambers.
The RPCs ensure an excellent time resolution due to their 3 ns response time, and it is for this reason that they are used for triggering.

1.4 Trigger and Data Acquisition System

The CMS Trigger and Data Acquisition (DAQ) System [21, 22] is designed to collect and analyze the detector information every 25 ns, namely every bunch crossing, and select the events of potential interest for further analysis. The LHC provides 10^9 interactions per second, and each event is read out by roughly 10^8 CMS channels. Accordingly, a total amount of 1 MegaByte should be stored for each event.
In this way, the trigger turns out to be a fundamental part of the experiment, making a real-time selection of the events to store.

The CMS experiment adopts a multi-level trigger system: *Level-1 Trigger* (L1) and *High Level Trigger* (HLT). The first is based on custom hardware electronics, and reduces the rate roughly from 40 MHz to 100 kHz with a latency of 4 μs. The second performs software event building, event selection and reconstruction on commercial processors, reducing the rate roughly from 100 kHz to 1 kHz.

1.4.1 Level-1 Trigger

The L1 has to trigger events every bunch crossing (25 ns). It is based on a structure of sub-detectors in order to perform a first rough identification of particles (Figure 1.9), namely:

L1 Calorimeter Trigger focuses on electrons ($e^{\pm}$), photons, jets, and MET;

L1 Muon Trigger focuses on muons from the CMS Muon System;

L1 Global Trigger makes use of pre-defined algorithms in order to take final decisions.

The sub-detectors are organized in other several sub-structures in order to increase the performance of the entire L1 Trigger system. Moreover, the pipelined-shape of the general structure allows the temporary storage of the CMS Detector information in pipeline memories, for up to 4 μs from the collision.

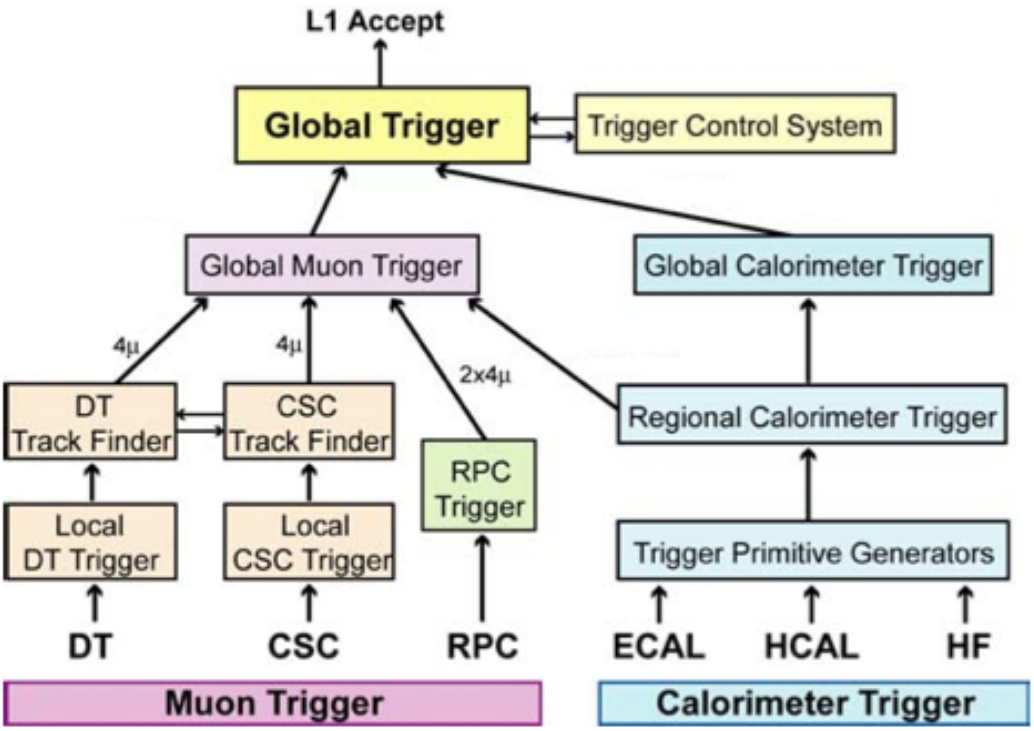

Figure 1.9: Level 1 Trigger decision flow before data is being transferred to the DAQ.

1.4.2 High Level Trigger

The HLT performs a refined reconstruction and selection of events relying on data filtered by L1 Trigger. At this stage, the events are assigned to specific datasets through the use of signatures.

The *streams* are the result of the selection process, and they contain the Detector information, the L1 Trigger and HLT results, ready to be subjected to the forthcoming reconstruction step.

1.5 CMS Data Taking: Run-I and Run-II

The Physics Program of the CMS experiment, as well as those of the other LHC experiments, is aimed at answering fundamental questions in Particle Physics:

- What is the origin of elementary particle masses?
- What is the nature of the Dark Matter we observe in the Universe?
- Are the fundamental forces unified?
- How does Quantum ChromoDynamics behave under extreme conditions?
- Do matter and antimatter properties differ?

1.5.1 LHC Run-I

In the LHC *Run-I* in 2011 and 2012, the collider reached a peak luminosity of $7.7 \times 10^{33}\ cm^{-2}s^{-1}$, more than 75% of its design luminosity. An integrated luminosity approximately of 25 fb^{-1} at $\sqrt{s} = 8\ TeV$ was delivered to each of the two general purpose experiments, ATLAS and CMS. In this way, a vast quantity of physics results have been yielded, allowing the CMS collaboration to summarize them in more than 300 publications.
The highlight has been the observation of a new particle of mass $\sim 125\ GeV$ by the ATLAS and CMS experiments [10,11] in 2012. This particle was identified as a *Higgs Boson* after detailed studies of its properties had been performed, partially providing an answer to the first question. The decays of the new boson to the gauge bosons of the Standard Model, W, Z, and photon, were established, each with more than 5 standard deviation significance. The couplings of the new boson to these particles have been determined using a combination of theory predictions for the decays and production, and they turn out to follow the mass dependence uniquely characteristic of the Higgs field. Further, in the search for the decay to fermions $\tau^+\tau^-$ and $b\bar{b}$, the corresponding couplings with the Higgs boson turned out to be consistent with Standard Model expectations.
A new analysis technique in the measurement of the total width of the Higgs boson was performed using off-shell Higgs production properties at masses of a few hundred GeV. In this way, CMS was able to constrain the Higgs boson width to 5.4 times the expected value in the Standard Model [23] of 4.1 MeV, a 200 times more stringent constraint than that reached in previous "direct" measurements.

On the other hand, many searches have been undertaken with the data taken in 2011 and 2012, improving several limits and precision measurements. The CMS collaboration have placed limits on a conspicuous number of physics quantities relating to well known Standard Model physics, rejecting valiant theories or giving birth to new ones. CMS was

also able to make precision measurements of rare decays that are well-predicted in the Standard Model, and to probe topics previously beyond the understanding of physics knowledge. As a case in point, the first cross sections relating processes of associated production $t\gamma$ have been recently presented. Moreover, the coupling to the *top* quark through the $t\bar{t}H$ process, appears to be within reach.

The Standard Model does not provide answers to the remaining questions, and many searches have been carried out with the data from 2011 and 2012 for many of them. Furthermore, many proposals on the existence of new physics have been put forward, which try to address at least some of questions posed at the beginning of this Section. However, it is crucial to rely on LHC *Run-II* data taking in order to provide answers to the questions of Physics.

1.5.2 LHC Run-II

In Spring 2015, the LHC has been restarted with a center-of-mass energy of $\sqrt{s} = 13\ TeV$ after the Long Shutdown 1 (LS1). In fact, the LHC is designed to operate with cycles of 3-years data taking interleaved with Long Shutdown period used to maintain and upgrade the collider (Figure 1.10).

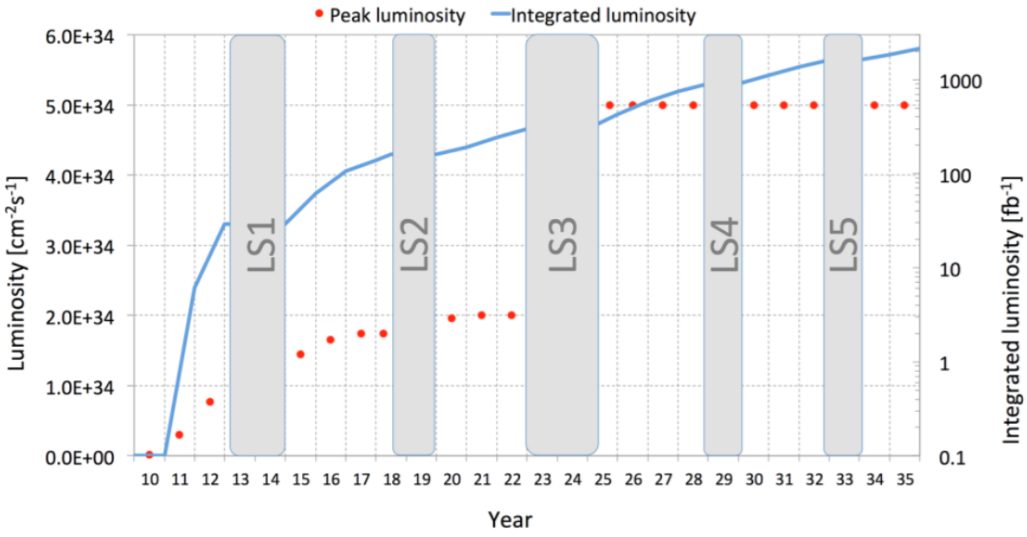

Figure 1.10: Projected LHC performance through 2035, showing preliminary dates for LSs of LHC and projected luminosities.

The LHC *Run-II* has been started with a beam energy of 6.5 TeV and a bunch crossing interval of 25 ns, to be eventually increased to 7 TeV in 2016. Moreover, the instantaneous peak luminosity will exceed $\mathcal{L} = 1 \times 10^{34}\ cm^{-2}s^{-1}$, providing an integrated luminosity approximately of 45 fb^{-1} per year.

1.5.3 Technical and Physics motivation for upgrades

The CMS experiment has to improve its detector ability in order to cope with the unprecedented LHC performances. The ageing of the detector materials could no longer deal with the precision required by the selection and measurement of the $p-p$ collisions final states. The number of overlapping events in the same data acquisition time interval is expected to be 50 after the Phase-I upgrade and over 100 after the Phase-II upgrade. These high values of pile-up increase the probability to fake the rate in tracking phase, and reduce the ECAL and HCAL energy resolution. Hence, these detectors needs more high granularity in order to be able to distinguish significant particle events from the pile-up ones. Moreover, it is necessary to take into account the effect of radiation on the sensitive detector material. A declining of the detector performances could be caused, leading to the impossibility to perform any event reconstruction.

The CMS physics program is very challenging, and aims to shed light on unanswered fundamental questions. It is going to provide precise measurements of known Standard Model processes, exploration of HEP processes, and study of very rare final states.
Furthermore, the study of the Higgs boson continues to be crucial, including precise measurements of the Higgs boson couplings, mass, J^{PC} and width, and the search for rare Standard Model and beyond the Standard Model decays. Radiative corrections to the Higgs should cause the mass to increase to very high values (TeV scale), and new physics could appear in order to cancel this growth. Deviations from perfect Standard Model behaviour because of its interaction with other forms of matter, *e.g.* Dark Matter, could answer some very fundamental questions, such as the origin of the matter-antimatter asymmetry of the Universe. Therefore, the detailed study of the 125 GeV Higgs boson is a scientific imperative that has to be pursued to a higher level of statistical precision than that of the LHC Run-I.

Finally, many searches have been undertaken with the data taken in 2011 and 2012, not revealing evidence of beyond the Standard Model physics. Therefore, search for new physics at higher mass scales is required with more statistics in order probe or not the existence of Super-Symmetric particles, or beyond the Standard Model physics.

1.5.4 Phase-I

The Long Shutdown 2 (LS2) will start in 2019, in order to apply the so-called *Phase-I* upgrade. At the end of the LS2 period, the LHC *Run-III* will start with an instantaneous peak luminosity of $\mathcal{L} = 2 \times 10^{34}\ cm^{-2}s^{-1}$. Furthermore, the integrated luminosity will be approximatively of $300 \div 400\ fb^{-1}$ per year.

The Phase-I upgrade provides improvement of the detector performances and resolution of high-radiation damage to the detector material. It consists of 3 sub-detectors upgrades, as follows.

Pixel Tracker Upgrade The current Pixel Tracker will be replaced with a new silicon Pixel Detector. It will be high efficient and have low mass, equipped with 4 barrel layers and 3 endcap disks, to provide 4 pixel hits in the full $|\eta| < 2.5$ acceptance region [24].

L1 Trigger Upgrade The L1 Trigger will be improved with higher granularity and additional processing capabilities to cope with increasing luminosity, energy, and pile-up. In fact, a substantial increase in the trigger threshold will be required in order to remain within the 100 kHz limit for which it was designed to operate. Moreover, the L1 Trigger will undergo an upgrade of the electronics of the Calorimeter Trigger, the Muon Trigger and the Global Trigger [25].

HCAL Upgrade The HCAL upgrade provides substitution of the photo-detectors and electronics to improve the measurement of jets and MET.
The Forward HCAL currently uses Photomultiplier Tubes to collect light from the absorber material and produce electronic signals. They will be replaced with multi-anode tubes.
The Barrel and Endcap HCAL use Hybrid Photo-Diodes transducers. They will be replaced by Silicon Photomultipliers (SiPM) to achieving better performances [26].

1.5.5 Phase-II

The Long Shutdown 3 (LS3) is planned to start in 2023, in order to operate a major upgrade of the collider, called *Phase-II* upgrade [27]. The upgraded operating regime will be called *High Luminosity LHC* (HL-LHC) precisely because the collider will be subjected to a significant increase in luminosity. At the end of the LS3 period, the HL-LHC will start with an instantaneous peak luminosity levelled at $\mathcal{L} = 5 \times 10^{34}\ cm^{-2}s^{-1}$, reaching an integrated luminosity of 3000 fb^{-1} per year.

The proposals for Phase-II upgrade are based on performance projections that consider real data taken during Run-I and radiation doses studies for HL-LHC.

Chapter 2

CMS Computing

The CMS experiment needs to collect, archive, process and distribute the data coming out of the LHC, and the CMS scientists worldwide need to access and analyze such data with low latencies. Efficient storage and data management systems, as well as performing workload management solutions, are crucial for the execution of a successful LHC program. These needs pose stringent requirements to the computing systems and resources needed, and on the specifics of the LHC experiments Computing Models to cope with such challenges. Only in terms of storage needs, for example, the LHC produces several tens of PetaBytes of data per year.

The introduction of innovative computing technologies has been required by the huge amount of data produced by the LHC operation, allowing to carry out analysis and computations that demand resources far beyond those typically available on computers used by users.

The Computing *Grid* paradigm has been chosen and further developed to allow a distributed set of computing centres and resources to cooperate and work together coherently in order to handle the LHC data. Each LHC experiment has developed a Computing Model to organize and manage storage and computing resources according to the experiment needs. It includes the set of all hardware and software components developed to cope with the collection, processing, distribution and access in end-users analysis of the huge amount of data produced. The management and interaction of each of these components are performed through instruments and services operated on a 24-*hours*-per-7-*days* basis.

This Chapter aims to present the CMS Computing Model, including computing resources organization and software development required for their management.

2.1 Computing Grid technology

The idea of creating an infrastructure of resources, services and tools based on distributed computing turns out as an implementation of the Grid paradigm in the HEP context, in which each experiment can add their own application layer using a single middleware common to all. The significant costs of maintaining and upgrading the necessary resources are more easily handled in a distributed environment, where individual institutes and participating national organizations can fund local computing resources and retain responsibility for these.

The Worldwide LHC Computing Grid (WLCG) Project [28, 29] is the global collaboration building, managing and preserving the data storage and analysis infrastructure required for LHC operation. Computing resources are provided through WLCG to all scientists all over the world in order to store, distribute and analyse the huge amount of LHC data. The WLCG Project collaborates and inter-operates with Grid development projects, network providers and production environments around the world, such as the European Grid Infrastructure (EGI) [30], that provides access to high-throughput computing resources across Europe using grid computing techniques in order to support international research in many scientific disciplines, or the Open Science Grid (OSG) [31], a national production computing grid infrastructure for large scale science, built and operated by a consortium of U.S.A. universities and national laboratories.

The managed resources accessible from anywhere in the globe lay the foundations to the basic concept of the Grid as a shared computing infrastructure suitable for problem solving in dynamic, multi-institutional virtual organizations [32]. The subdivision of the process into subprocesses executed in parallel is the simplest system to perform computations in a relatively small time. Each subprocess develops a different portion of data using the same code. The state of the art of this system is represented by the distributed computing infrastructure, used in several areas such as bioinformatics, medicine, chemistry, and obviously HEP. However, the advantages of a distributed computing infrastructure is not limited to perform parallel computations needed by the experiment. A better use of the resources among various nations is allowed, since the resources of a single nation are to be available to the whole world research.

The basic idea of the Grid is to implement software technologies that enable the user to transparently access the infrastructure. The latter handles the parallel submission of several processes at the enabled sites, avoiding the user to connect directly to each computing centre. Differently, the number of direct connections required would be so high to be unworkable in practice.
Furthermore, a worldwide communication network allows scientists belonging to the same research group not to be located in the same geographical area. It is a great way to promulgate knowledge and encourage collaborations between different research groups.

The Grid is founded on certain basic elements, each of which provides a specific service required for efficient operation of the infrastructure. The software layer, customizable at application level by the experiments, is provided by the middleware layer. The middleware infrastructure is made up of the logical elements of a Grid site, namely:

Computing Element (CE)
Service managing the user requests for computational power. The computing power available to the site is achieved by using clusters of computers organized in farms and managed by software tools, *e.g.* the batch systems. The CE handles the load coming from user job submissions, on queue in batch systems of distributed sites. In this way, it manages the running/pending jobs and all the needed communication with the related services in the Grid infrastructure.

Worker Node (WN)
The WN is the single compute node in the farm of the site. Hence, it is the actual perpetrator of the user's job. The execution of the jobs is regulated by scripts that allow to configure the environment, to automate the execution of the code, and to make available the output for the copy in local or remote storage sites.

Storage Element (SE)
Service allowing a user or a process to store and access data. Although the implementation of local access to files is different for each site, remote access is via common interface that uses SRM protocol (Storage Resource Manager). The SE provides the memory required to store data from the detector, data resulting from MC simulations and from users' analysis. Tapes and disks are used to store data: tapes as long-term secure storage media, disks for quick data access for analysis. Each SE has to have a substantial storage capacity, and ensure sufficient performance in data access and I/O.

User Interface (UI)
The UI is the machine enabling the user access to the Grid. By logging to the UI, the user can reach remote resources and has interactive access to execute own code. Generally, the UI does not require very high performance, since it has at the most to be able to execute the code on few locally copied data to verify proper operation.

Central Services
Services helping users to access computing resources. These services are intended to enable a proper management of resources. Even if they can be of different type and vary between different Computing Models, there are some basic functions that have to be able to perform, hereinafter in brief.

Workload Management System (WM System)
Set of services and tools that handle the lifetime of a processing workflow, by submitting and tracking the status of all individual jobs. The WM system is constantly updated on the Grid status. It has to decide which site is going to manage user processes in order to optimally distribute the workload on the infrastructure, taking into account the characteristics that each process has. If a user's analysis requires access to a particular dataset, the WM system has to assign it on a site that can access them efficiently. The proper functioning of the WM system is crucial in order to fully exploit the resources.

Information Systems
Services constantly updated on the status of the infrastructure. They provide information needed to decide the distribution of the workload to the WM system. Moreover, the CE and SE status are known with high accuracy.

Meta-Data Catalogues
Catalogue holding all data information like for instance file size, number of events, variables and other features required by the user. These information can also be accessed by programs that regulate the submission on the infrastructure.

Data Bookkeeping and Replica Location Systems
The Logical File Name (LFN) allows users and information systems to intuitively identify and to simply manage the data. However, each site has a different architecture so that data have different Physical File Name (PFN) compared to LFN. The data location service provides the tools that allow to know the sites where data is stored, and the information needed to access it.

File Transfer System (FTS)
Service that allows to copy data across different Grid sites. The FTS ensures optimized data transfer operations and an equitable use of the network. While the request for dataset(s) replication go through experiment-specific tools, the single underlying site to site file transfer operations are taken care of by FTS.

Virtual Organization Management System (VOMS)
The concept of Virtual Organization (VO) is a bearing wall of the Grid. A VO is a dynamical group of institutions or individuals held together by rules and conditions which determine the policy in terms of resource sharing. The WLCG services have a certain degree of security related to X.509 certificates which provide secure authentication for both the users and the services. The VOMS manages the authorization and contains all the Grid users, their information, covering roles, membership to groups, permissions and privileges [33]. Thus, it provides information to authorize users and controls the operations that the user can perform on the infrastructure.

Over the course of LHC Run-I, CMS has designed and deployed a solution for a global Grid-aware Storage Federation (SF) based on the *xrootd* [34, 35] technology. The new infrastructure turns out to be developed in parallel to the SE. Unlike the latter, it does not rely on a catalogue to locate the files but on a set of "redirectories". If the file is not found in the local storage, the redirector asks to the SE in its federation whether it has the file. If negative, the redirector asks to a higher level redirector. The process continues until either the file is found or the highest redirector does not find anything.

The computing centres worldwide are hierarchically organized into four types of *Tiers*, depending on the kind of services they provide [36–38]. The range of levels is from 0 to 3, and a lower Tier has generally more services, potentiality, availability and responsibility than a higher one [39]. Moreover, WLCG characterizes them on the basis of *Service Level Agreements* (SLAs) which specify, among other details, that activities running at Tier-1 and Tier-2 centres are supposed to be supported on a 24-*hours*-per-7-*days* and 8-*hours*-per-5-*days* basis respectively (Figure 2.1).

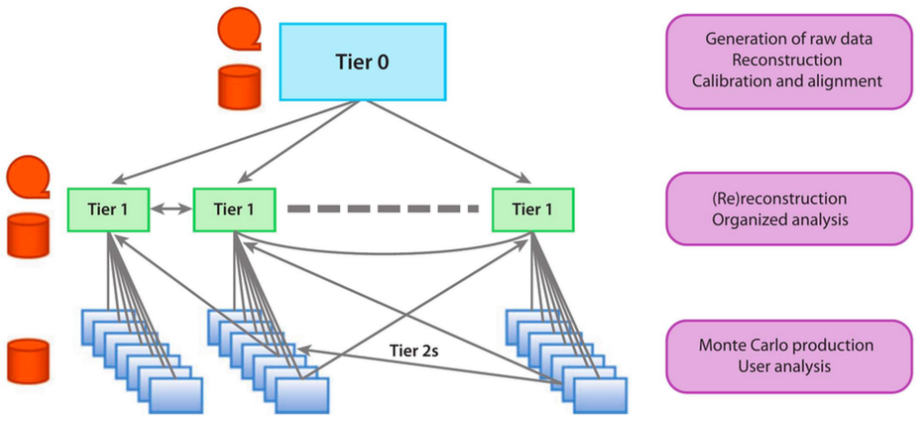

Figure 2.1: Scheme of the Tiered structure of the WLCG and main activities performed.

2.2 The CMS Computing Model

During data taking, the CMS Detector collects a huge amount of data that have to be accessible by scientists in order to carry out physics analysis. An appropriate computing system and tools are required to be able to execute at the needed scale and level of performance complex tasks as running distributed analysis or generation of MC simulations.

In the following Sections, the data model is described, and an overview of the necessary computing services is provided.

2.2.1 CMS Data Hierarchy

The CMS experiment requires tools to catalogue the data, to track the location of the corresponding physical data files on site storage systems, and to manage and monitor the flow of data between sites. Accordingly, higher level objects are defined in order to simplify the data management problem.

The basic structure is characterized by the *Event*, containing high and low level information relating to a single bunch crossing. Consecutive physical Events in a certain period of time are grouped in *LumiSections*. The time interval of a LumiSection is chosen so that the collider luminosity is assumed to be constant in this interval, and each LumiSection turns out to be related to a well-defined integrated luminosity. Therefore, the total integrated luminosity of a data sample used in a specific analysis can be determined starting from that of the analyzed LumiSections.
The *Run* is defined as a DAQ data-taking unit and contains an integer number of LumiSections. The time period of a Run is related to detector and beam conditions that have to be stable.

Datasets are collection of Events coming from one or more Runs that contain Events with specific characteristics. Physically, the data of a dataset are saved in *files*, which in turn are clustered in *fileblocks* in order to increase the system scalability. In this way, fileblocks turn out to represent the granularity of the CMS data management, as they are the smallest unit of data moved through the Grid. Note that the same Run can be located in different files so that Events of the same Run can be found in the same file or in different files.

The CMS Data Model classifies Event data in different data-*tiers* corresponding to the levels of processing.

RAW data

The RAW data are directly produced by the detector and contain signals and information from each of its components. Generally, these datasets provide low level information that are too detailed and difficult to manage to be used in the analysis. On the other hand, RAW data are the precious and unique experimental outcome of the actual physics particle collision as observed by the detector. Therefore, backup is immediately performed at CERN computing centre in order to host a safe copy of the data, and shortly after in at least one Tier-1 site on tape libraries. The RAW data are unique, and there would be no way to recover them once they were no longer accessible.

RECO data

The RECO data (or Event Summary Data, ESD) are obtained from processing the RAW data, and can in principle be used to perform an analysis. The processing involves the application of specific detector reconstruction algorithms and

compression algorithms, including detector-specific filtering and correction of digitized data, primary and secondary vertex reconstruction, tracking and particle identification. Thus, these reconstructed data provide high level information related to reconstructed event such as track, energy and momentum of the particles. Nevertheless, intermediate objects are defined in order to decrease the high processing required to move from RECO data to objects usable for the final analysis, *e.g.* AOD, PAT, etc.
The RECO data are periodically replaced with newer versions produced with the latest updated versions of the software or calibration constants.

Analysis Object Data (AOD)
The AOD is a data-tier that contains all and only the information necessary for the performance of most of the analyses. These datasets provide very high level information, such as four-vectors associated to types of particles involved in an event, resulting adequate for the physics analyses, and at the same time smaller in size and more manageable for the overall computing infrastructure.

Simulated Analysis Object Data (AODSIM)
The Monte Carlo simulations generate a number of events which is comparable to (actually, greater than) the number of actual physics events produced by the LHC. The AODSIM is the data-tier that contains such simulation-level information, and is used by the vast majority of CMS analysts together with the AOD data-tier.

MiniAOD
The MiniAOD is a high-level data-tier designed and deployed in Spring 2014. Thenceforth, it is used CMS-wide in order to serve the needs of the mainstream physics analyses with a data-tier of size further decreased with respect to full AODs. The production of MiniAODs is normally done centrally, and they are saved using PAT data formats keeping a small event size ($30-50\ kb/event$). The main contents of the MiniAOD are: high level physics objects (leptons, photons, jets, MET), the full list of particles reconstructed by the *ParticleFlow*, MC Truth information, and Trigger information.

Physics Analysis Toolkit (PAT): Data Formats
The PAT are datasets containing high level information that can be customizable by the user in order to foster the different analysis groups (Physics Analysis Group, PAG). This choice provides the possibility to define your own objects using the algorithms developed by groups that deal with the management of the various physics objects (Physics Object Group, POG), as Muon, Jets, Electrons, etc. Although the PAT can not be considered as a real data-tier, PATs are widely used by analysts and the code needed to create them is completely embedded within the analysis framework.

A typical CMS data reconstruction workflow is briefly discussed in the following. The information contained in the RAW data are subjected to several stages of processing until obtaining objects of interest for physical analysis. In particular, the prompt reconstruction performed by Tier-0 site and the subsequent data reprocessing performed by Tier-1 sites produce several output that are processed during a skimming phase. In this way, events of interest for certain types of analysis are clustered in specific dataset. Therefore, the final stage consists of derived data containing all the useful and necessary information for the final analysis.

Regarding a CMS simulation reconstruction workflow, it expects to run firstly the *kinematics* step based on various MC event generators, followed by the *simulation* step that provides the combination of single interaction with pile-up events. This is required in order to simulate the detector response as a result of generated interactions. Consequently, the computing resources turn out to be very stressed at I/O level due to hundreds of minimum-bias events required for the previous combination. Finally, the *reconstruction* step is performed in order to simulate reconstruction of a real bunch crossing.

The workflows involving the different CMS data-tiers is briefly presented and discussed in the following (Figure 2.2).
The events selected by the CMS HLT are temporarily stored by the online system, and are grouped in Primary Datasets. Therefore, the data are transferred to CERN Tier-0 for repacking (to produce RAW) and processing. Some streams have priority over others, depending on the importance given to the dataset or on the need for some data to arrive earlier than others.
The prompt reconstruction of events is performed at the Tier-0 site in quasi real-time, with a predefined latency, and in Run-II also at the Tier-1 sites. The RECO files contain all the additional information provided by the reconstruction of events, and their content is hence more "verbose" compared to the RAW format. The RAW and RECO files obtained are stored in the CERN computing centre, and a second copy is performed at Tier-1 sites. Basic improvements in the software, as well as better knowledge of calibration and alignment of the detector require re-reconstruction at least once per year, performed at the Tier-1 centres. At this level, a further reconstruction work and skimming of data is performed by Tier-1 sites creating the AOD, which are also distributed to Tier-2 sites.

Non-Event data

Aside from Event data formats, described in the previous part, CMS also handles several categories of so-called Non-Event data. These are used to store background information, such as the status of the detector or calibration constants, and some details are briefly summarized in the following.

The *Construction data* contain information concerning the construction of the detector. The *Equipment Management data* include information about geometry, position, and electronic equipment of all CMS subdetectors. The calibration constants and thresholds are stored in the *Configuration data*, that turn out to be the keystones for the reconstruction of events. The *Condition data* are the parameters describing status and monitoring of the detector during Run, and are used for the Data Quality Monitor (DQM), a procedure to check the quality of the data during the acquisition and define the data usable for physical analysis. They are produced both by online and offline applications and used by the HLT, subsequent reconstruction and analysis. Several databases provide access to these data, as follows.

Online Master Data Storage (OMDS)
It is directly connected to the detector, and allows to write the Configuration data and to receive the Conditions data.

Offline Reconstruction Conditions DB ONline (ORCON)
It is a copy of OMDS localized to the detector site, and it is used to synchronize information (Configuration data and Conditions data) for offline-use.

Offline Reconstruction Conditions DB OFFline (ORCOFF)
It contains the automatically synchronization between ORCON and Tier-0, and it is replicated on different computing centres in multiple copies, which are accessed by jobs requiring knowledge of the Non-Event data.

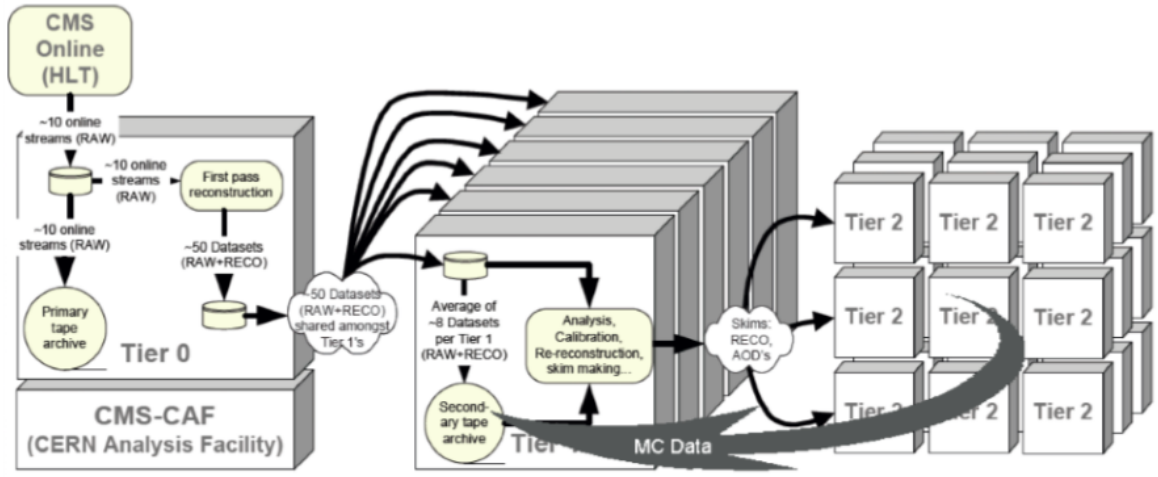

Figure 2.2: Data flow in the CMS Computing Model.

2.2.2 CMS Grid sites

The Grid infrastructure makes computing resources available to the CMS collaboration. The computing centres host and run different set of services for CMS, and several of them also may have hardware more powerful or reliable than others. Thus, they can be classified by role and characteristics in a hierarchical structure which originally comes from the MONARC model [40].

Tier-0

The CERN Data Centre in Geneva and the Wigner Research Centre for Physics in Budapest are the physical locations of the unique logical Tier-0 function.
The Tier-0 is exploited for the first reconstruction of events. It has to accept data from the DAQ [41], storing RAW data on tapes. In fact the CMS data model does not foresee the detector site to be equipped with storage resources. Instead, the architecture provides tape libraries at CERN in order to host a safe copy of the data.
Performing repacking and prompt reconstruction of events, it is absolutely crucial that the Tier-0 is able to provide sufficient computing power to complete operations at the same rate at which events arrive from the online system. Furthermore, it is responsible for alignment data processing and calibration of the detector, necessary for the data reconstruction.
Finally, the RAW and RECO data are distributed to at least one Tier-1 site in order to have another safe copy on the Tier-1 level as a whole. In this regard, a redundant optical network infrastructure, private to the LHC (called LHCOPN [42]), has been deployed among the LHC Tier-0 and Tier-1 sites, in order to guarantee and protect a high-performance data transfer traffic among Tier-0 and Tier-1 sites.

CMS-CERN Analysis Facility (CMS-CAF)

The CMS-CAF [43] is located at CERN and was designed to hold several latency-critical workflows including alignment and calibration, detector commissioning and diagnosis, and physics analysis. It combines flexible CPU resources with rapid access to the entire CMS data set and supports fast turn-around analysis when required and specialized functions related to the operation of the detector, such as performance monitoring. One of the advantages of the CMS-CAF is the proximity to the CERN Tier-0 facility, and then fast access to all the RAW data produced by the detector. The CMS-CAF also provides resources for interactive activity to CMS typical user located at CERN, as an additional resource for analysis.

Tier-1

A total of 13 LHC Tier-1 sites do exist worldwide. However, only 7 Tier-1 were available to CMS experiment in LHC Run-I and at the beginning of Run-II. Large scale and centrally organised activities are carried out at the Tier-1 level, and the data can be exchanged among all Tier-1 sites and to/from all Tier-2 sites.
One of the main Tier-1 function is to store and supply data to Tier-2 sites, and to deal with reconstruction of events using calibration constants regularly updated. Moreover, the Tier-1 sites are offering computing resources for other processing tasks, *e.g. skimming*, which is the creation of datasets where are gathered only interesting events for a given analysis.
A CMS Tier-1 provides a great amount of CPU resources for several very important tasks. It has to be highly reliable and have high connectivity, as might be the only site which store certain data. For this reason, the Tier-1 sites must be able to ensure the opportunity for other sites to copy data into their SE.

Tier-2

About 160 LHC Tier-2 sites are located worldwide but just about 50 were available to CMS experiment in LHC Run-I. These computing centres are usually placed at Universities or scientific Institutes.
The CMS Tier-2 sites provide services for data analysis and production of MC events. They provide a large amount of the total computing resources and many data are copied here from Tier-1 sites. However, the Tier-2 sites do not host any tape library, and their storage is entirely disk-based, hence they do not have custodial responsibilities.

Tier-3

The CMS Tier-3 sites offer very flexible computing capabilities despite there is no formal agreement with WLCG. These sites can perform tasks similar to Tier-2 sites, but with less obligations of responsibility and reliability. The possible shutdown for maintenance makes Tier-3 sites less reliable than lower sites, but they are very useful resources *e.g.* for local communities of analysis users. In fact, the Tier-3 sites provide interactive access to local users, allowing the direct job submission to the batch system or to the Grid.

2.3 Tools for CMS workflows execution

The Grid infrastructure takes advantage of several services in order to ensure the proper and efficient use of resources. A general description of these services can be found in Section 2.1. The focus of the following Sections is to present their actual implementation inside the CMS Computing Model.

The CMS Computing Model is mainly characterized by a *data-driven* paradigm: jobs run where the data is, and no data is moved in response to job submission. The two major sectors of the CMS Computing Model are: the *CMS Data Management System* [44] and the *CMS Workload Management System* [45].
These will be presented and discussed in the following Sections [Ref. 2.3.1-2.3.3].

2.3.1 CMS Data Management tools

The DM system comprises a set of tools designed *e.g.* to locate, access and transfer different types of CMS data, relying on both CMS and Grid services.
The data management elements in CMS are scalable, modular, and designed to work in a coherent manner. The main components are:

- Data Bookkeeping Service (DBS), a meta-data catalog describing which CMS data do exist, their location, plus plenty of additional information;
- PhEDEx, a reliable, scalable dataset replication system;
- Data Aggregation Service (DAS), designed to aggregate views and provide them to users and services.

This modular system allows the optimal use of appropriate underlying technologies.
Although the DM system components are designed and implemented separately they interact with each other and with the Grid users as web services.

Dataset Bookkeeping System For a proficient data utilization, a CMS physicist needs to have plenty of meta-data in addition to the CMS data, *i.e.* information on the datasets of interest as the name, size, number of events, parentage, relationships between data and collections of events, etc. Moreover, other general information have to be provided *e.g.* data quality information, and applications, software versions and configurations used for dataset processing. A component with meta-data catalogue functionalities need to be used in order to allow user to access these precious information.
The Dataset Bookkeeping System (DBS) is the meta-data catalogue within the CMS environment, and it is implemented with and Oracle database backend. It provides these and other information to users via standard SQL (Structured Query Language)

requests, that is a language designed to read, edit and manage data stored in a database. In fact, the DBS is also used by CMS Remote Analysis Builder (CRAB) - see later in this Chapter - to check that the specified dataset exists and is located in sites that meet the demands of the user's job.

Information on the location of the data are managed by PhEDEx, the CMS data transfer system.

Physics Experiment Data Export The CMS experiment poses very demanding requirements on a data management solution in terms of security, reliability and scalability. The Physics Experiment Data Export (PhEDEx) [46], a robust data management layer, has been specifically developed in order to manage the priority transfer of files from multiple sources to multiple sites. PhEDEx takes care of functions like serving data to a site that is asking for them, trigger migration to or recall from tapes, schedule multiple transfers on the same sites in optimized ways, etc. Automatic retrials and a variety of tactics to guarantee the data delivery to destination sites are some of the basic mechanisms in the PhEDEx design, since its original concept in 2003.
PhEDEx has the freedom to choose the source of the file to copy using algorithms based on the concept of storage overlay network. In this view, the nodes of the network are the data storage locations and the connections between them are the NREN links, and the algorithms are designed in order to optimize the overall transfer operations CMS-wide and to reduce the load on the infrastructure. The system is studied to evaluate the past history of CMS transfers over all links and to select the fastest and most reliable route to use in order to transfer the requested data from one PhEDEx node to another.

PhEDEx is based on a cluster of Oracle database [47] located at CERN, the Transfer Management Data Base (TMDB), which has two interfaces: an interactive website [48] allowing the transfer of dataset or fileblock, and a web data service [49] favouring the interaction between PhEDEx and Data Management elements.
Once a request has been made through one of the interfaces, PhEDEx connects to TMDB to obtain the necessary meta-data and updates the database after the task has been completed. The TMDB contains information about the location of data replicas and about active tasks. Moreover, it is the main source of information for the location of the data, provided to the DBS, and therefore it performs part of data location functionalities, as previously described.

In addition to the TMDB, the PhEDEx architecture comprises a set of specialized, stateless software agents which are run both centrally and on distributed sites. These agents share information about replica and transfer state using the TMDB as a blackboard [50]. In this way, they can access information *e.g.* about network routing, dataset subscriptions and on the status of the overall infrastructure. The PhEDEx topology has been designed to reflect the Tier structure typical of HEP experiments, thus connecting all Tiers (each with one or more PhEDEx nodes) altogether, and exploiting any

Grid-aware middleware transfer tool (like FTS) as a simple point-to-point file transfer mechanism [51]. In particular, concerning the routing functionality, the work of identification and optimization of data routing is performed by agents running at CERN, which take into consideration the performance of the link previously used to connect the target site with the source site containing the data to be replicated, based on the percentage of successful transfers on that specific source-destination route in a given time window. Therefore, one of the agents that run on each site, receives the necessary meta-data from TMDB and starts the transfer using specific plugins. The success of any transfer or any deletion of data in a Grid site is independently checked for each fileblock. In case of failure, other agents are activated in order to complete the request. All performance data are continuously recorded in TMDB and can be viewed through the PhEDEx dashboard.

Trivial File Catalogue The underlying database to PhEDEx provides information on the fileblocks location, and therefore also on the files location. Running at LFN level, all copies of the same file in different sites have the same logical name, and can be used as if they were the same physical file. It is therefore necessary to have a system that translate from LFN to PFN and vice versa for each site. In general, this could be achieved by using a catalog that is local to each site, maintained by the site itself, *i.e.* a Local File Catalogue (LFC). However, the CMS collaboration adopts a system based on a translation by algorithm: each site publishes the rules necessary to map the LFN into the PFN, and the algorithm operates the translation. Thus, the CMS LFC becomes trivial, taking the name of Trivial File Catalogue (TFC). Generally, a user does not need to know the PFN, as translation is performed automatically by the CMS software.

2.3.2 Grid services to support workload management

One of the great advantages of the Grid infrastructure is that a typical computing task can be easily split in many independent processes without needing to intercommunicate. In fact, the processes are performed in parallel over multiple machines reducing the total execution time, taking advantage of bulk operations provided by the middleware. Hence, the different jobs provide several outputs that have to be merged together for an efficient access and transferred to the destination sites.

The WM system distributes the workload on the infrastructure in order to maximize the efficiency and minimize the time required for the execution of jobs. An information service named Berkeley Database Information Index (BDII) [52] updates the WM system on status and characteristics of the computing sites, in order to decide how distribute the jobs. In the BDII, the status of the infrastructure as a whole is available, provided through automatic notifications for changes in the sites and a periodic control by the service itself.

The WM system manages submission requests of jobs accessing the BDII and receiving information about the Grid status. Thus, the jobs will be queued on the WM system global queue, to be distinguished from the batch system queue of the site, the local queue. Depending on the authorization level of the user, the management policies of the global queue are variable at the discretion of the WM system. However, the management policies of local queues are the responsibility of the sites. The sites may decide to implement them in a different way from each others, and a special authorized user has the ability to specify the management demand of the priorities on the local batch system. The management strategy of jobs can be done using different techniques:

gLite-WMS An algorithm has the task of making the choice of the site among those that meet the demands of the jobs, trying to event out the load on the infrastructure. If there are no compatible sites whose local queue has fallen below the threshold, the matchmaking procedure is repeated by WM system at predetermined intervals until reaching a time-out after which the submission is canceled. If the computing centre fails a considerable number of jobs of the same task, the site will be considered to be unreliable and the failed jobs will be sent to other sites.

The system is highly dependent on the live status of the infrastructure leading to an efficient distribution of jobs. However, serious drawbacks could come up if the site status is not promptly updated. In fact, if the information system does not truly reflect the conditions of the infrastructure, an incorrect use of resources can be expected. The latencies on WM system updating can lead to wrongly choose sites that have queues full of workloads, neglecting others that may have emptier queues and may accept and run the jobs.
In any case, several gLite-WMS instances can work in parallel with no need to exchange information, and the jobs are assigned to sites as soon as possible so that the gLite-WMS global queue is fictitious with no possibility to prioritize the jobs.

GlideIn-WMS The aim of GlideIn-WMS system is to minimize as much as possible the time of submission to the local queue of the chosen site. Notwithstanding, performances or site statistics are neglected to a certain extent.

The system is based on creating pilot-jobs with the same demands of the original ones, that are submitted to the CE sites. If one goes running, the others are eliminated from the local queues or used for a job in the global queue with compatible demands. In this way, a minimization of the time interval between the job submission on the UI and the actual running on a site is ensured.
However, the GlideIn-WMS relies on a central service that holds the global queue. Scalability and redundancy can be achieved by replicating the core services.

Recently, the CMS hybrid model has given way to a model based entirely on GlideIn.

The CMS job submission infrastructure (Figure 2.3) relies on the GlideIn-WMS that is mainly based on the *pilot job* approach. Several components of the HTCondor [53] Workload Management system are the basic architecture on which the GlideIn-WMS is built. During the submission to a traditional Grid site, the GlideIn-WMS factory submits pilot jobs, called *GlideIns*, to the CE, crating an HTCondor overlay batch system. The pilot job run starts an HTCondor *startd* daemon that allow to join a distributed HTCondor pool. Therefore, the node is able to accept work from the HTCondor pool, performing Grid Security Infrastructure (GSI) authentication and context switching at runtime using gLExec [54]. The user jobs turn out to be Condor jobs, submitted to the HTCondor user pool, even if this is hidden to the experiment software framework.

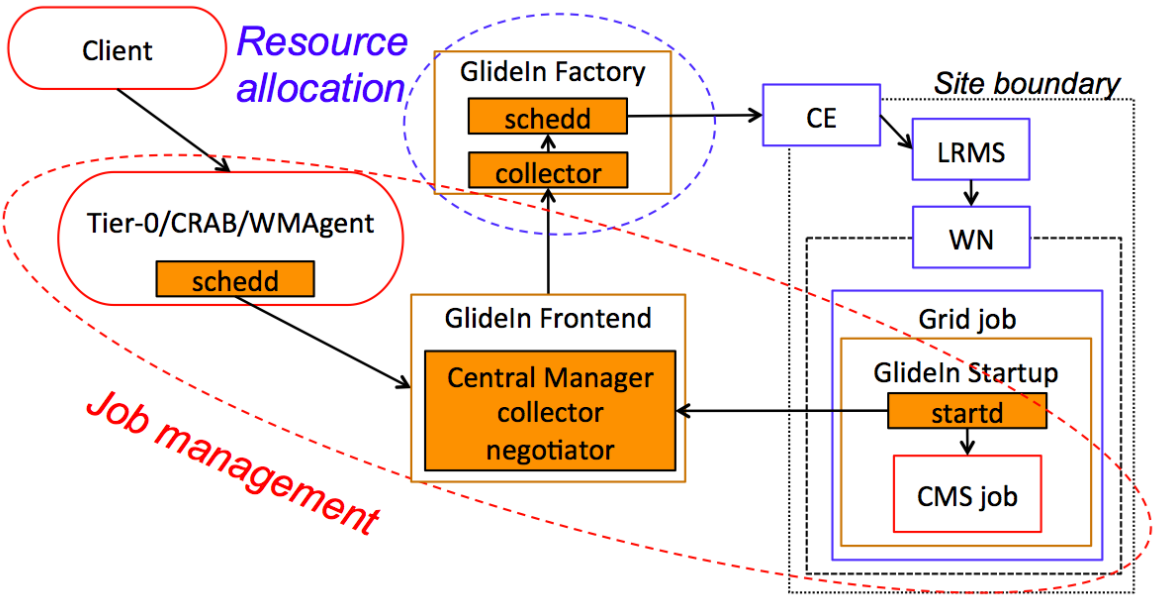

Figure 2.3: CMS Grid workflow using GlideIn to standard CE interface.

2.3.3 CMS Workload Management tools

The Request Manager system and the WMCore/WMAgent infrastructure

In the CMS Workload Management System, the overall infrastructure for "scheduled" processing, *e.g.* Monte Carlo production and data reprocessing tasks, has been designed to be scalable and efficient [55]. The centralized and automated management of all different workloads is done via a system called *Request Manager* [56]. It handles the management of all the necessary steps to prepare a sample from the acquired physical specifications, known in jargon as the *request*. Starting from such requests, the Request Manager creates the actual *workflow* to be executed, and hands over the task to the *WMCore/WMAgent* infrastructure, that is responsible for the actual execution of the task. It takes care of all the CMS-level operations necessary to the completion of a specific request, including all the necessary interaction with the underlying Grid workload management infrastructure.
A *WorkQueue* component divides a workload into *tasks* based on the specified priority and the type of workflow. Afterwards, it selects the best WMAgent instances to carry out that specific workload, ultimately creating the *jobs* that are run onto the WLCG computing sites. As from this design, the WMAgent framework is able to process different workflows in parallel, keeping others into its local WorkQueue, thus managing a potentially very large pool of production requests with different priorities.
In terms of monitoring, the WMAgent framework sends real-time information about jobs statuses to the CMS Dashboard [57]. Moreover, it publishes meta-data about the actually newly produced and available data to the CMS Dataset Bookkeeping Service (DBS) [Ref. 2.3.1], also interacting with the PhEDEx Data Management system [Ref. 2.3.1].

CMS Remote Analysis Builder

The CMS Remote Analysis Builder (CRAB) [58, 59] is a tool developed for CMS distributed analysis [60]. It is responsible for the creation, submission and monitoring of the CMS analysis jobs on the Grid, allowing transparent use of the infrastructure.
CRAB manages the connection with every single Grid component in order to avoid the user the in-depth knowledge of the complex Grid infrastructure, and to provide access to all data produced and collected by the experiment regardless of their Grid storage element.

CRAB performs users' distributed analyses on the Grid following the already mentioned data-location driven paradigm.
In fact, it elects the complete dataset on which perform the analysis, after it has performed the analysis code on local data samples in order to test its workflow. Therefore, CRAB carries the code at the site where the data are located, and it returns to the user the log of the jobs and the output of the analysis.

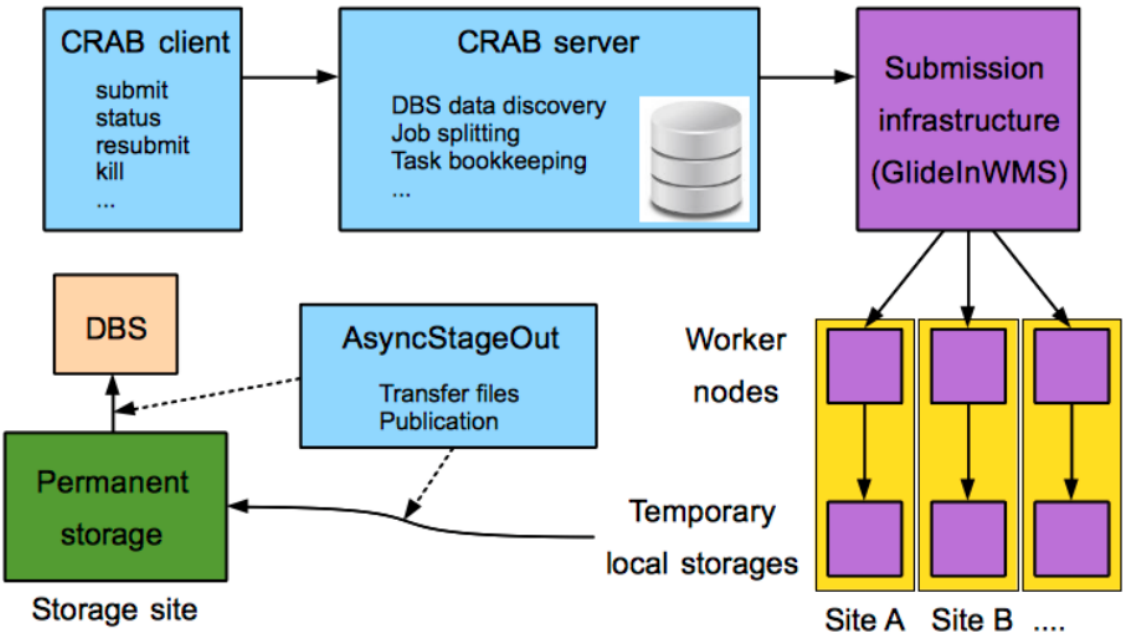

Figure 2.4: Simplified diagram of CRAB3 architecture.

Chapter 3

Cloud Computing in CMS

Cloud computing techniques enable a convenient on-demand access to a shared pool of configurable computing resources. They could be of different nature (*e.g.* networks, servers, storage, applications, and services) and can be efficiently provisioned and released with minimal management effort or service provider interaction.

Cloud computing has profoundly changed the modus operandi of both private and public institutions that have needs to access large computing power to achieve their goals. In fact, the problem of infrastructure management is largely overcome, allowing to buy virtual resources instead of maintain very expensive physical ones. The infrastructure provides constantly up-to-date resources in order to run newest software according to user's needs. The usage of the computing resource, as CPU, memory, and storage, defines the price in terms of time. In this way, the Cloud computing has turned out to profitable both for the company that provides the service and for user that buys it.

Particularly, Cloud computing can potentially reduce costs of hardware technologies from the user point of view. The economic benefit can be translated in more CPU power provided to the user, and in the possibility to make use of backup and recovery according to needs. In this case, the use case is handled by the Cloud provider, as well as software updates. A Cloud provider is able to use its resources at 100% of their potential whereas a privately owned resource is exploited only according to the needs of the user. The Cloud computing is characterized by high flexibility, as scalable access to computing resources is efficiently provided. Moreover, the resources turn out to be transparently accessible from anywhere in the globe, as the data are accessible through standard or high-performance networks.

However, the Cloud infrastructure could become a "double-edge sword" in specific use cases. The loss of data ownership arises as the main drawback considering that the user does not own physically the data, and the Cloud resource is certainly not located in situ. Additionally, security problems could take place as the user loses the responsibility on the security of the owned data. Thus, it is necessary to carefully verify the security of

the Cloud service provider. Moreover, other technical problems may occur, constraining the user at the mercy of the Cloud service provider.

3.1 Service and deployment models

Cloud computing providers offer their services according to several fundamental models, as follows.

Infrastructure as a service (IaaS)

An IaaS infrastructure refers to the most basic Cloud-service model. Data centres hosts large pools of hypervisors that support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements. The providers offer computing resources that most of the time are virtual machines, or general resources, billing IaaS services on a utility computing basis as cost reflects the amount of resources allocated and consumed. Additional resources could be offered, such as VM disk image library, IP addresses, VLANs, software bundles, firewalls, and load balancers. The Cloud users install operating-system and their application software on the Cloud infrastructure, patching and maintaining them.

Platform as a service (PaaS)

A PaaS infrastructure provides the user with a computing platform, typically including operating system, programming language execution environment, database, and web server. The cost and complexity of buying and managing hardware and software layers are overcome as users can efficiently develop and run their software solutions on the underlying Cloud platform. Moreover, computing and storage resources scale automatically to match the application demand so that the Cloud users do not have to allocate resources manually.

Software as a service (SaaS)

The SaaS business model provides Cloud clients to access application software and databases as on-demand, priced on a pay-per-use basis or using a subscription fee. The Cloud providers handle the infrastructure and platforms that run the applications, avoiding users to care about the infrastructure management. In this way, the need to install and run the application on the Cloud user's own computers is overcome, simplifying maintenance and support. The Cloud applications scalability can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand, and the relative load is handled by balancers. The applications are hosted centrally so that updates can be released without the need for users to install new software. Moreover,

the applications are multi-tenant (Cloud user organization) in order to accommodate a large number of Cloud users. However, the users' data are stored on the Cloud provider's server, leading to a possible drawback, *i.e.* there could be unauthorized access to the data. For this reason, users are increasingly adopting third-party key management systems to help secure their data.

The main adopted Cloud deployment models are briefly presented in the following.

Private Cloud A Cloud infrastructure operated solely for a single organization is called *Private*, and it is managed internally or by a third-party, and hosted either internally or externally. A Private Cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization about existing resources.

Public Cloud A Cloud infrastructure is called *Public* when the services are rendered over an open network for public usage. A Public Cloud architecture does not differ from a Private one, except for service security consideration because of possible non-trusted network communication.

Community Cloud *Community* Cloud shares infrastructure between several organizations from a specific community with common concerns as security, compliance, jurisdiction, etc. It is managed internally or by a third-party, and hosted either internally or externally.

Hybrid Cloud A Cloud computing service composed of some combination of Private, Public and Community Cloud services from different service providers is called *Hybrid*, and it offers the benefits of multiple deployment models. In fact, the extension of either the capacity or the capability of a Cloud service through aggregation, integration or customization with another Cloud service is allowed.

3.2 Use of Clouds in CMS

The idea behind the Cloud computing is about purchasing computing resources according to the needs, not resulting on the same footing respect to the sharing of computing resources among a partnership of institutions that characterizes the Grid. Hitherto in fact, it has been typically implemented by a lot of private companies.

Recently, Cloud resources have forced their way in the HEP field, being made available also to the CMS experiment. In this context, the Grid interface gives way to the Cloud interface in order to exploit these resources. Nowadays, the challenge for the HEP community is to get resources dynamically allocated, instead of accessing computing resources through the static allocation of Virtual Machines (VMs) as it happens in a

standard commercial Cloud. The CERN computing centre is already equipped with a Cloud infrastructure, called Agile Infrastructure (AI) [61]. This is designed to be the standard resource allocation system in LHC Run-II, and both the Tier-0 and CERN Analysis Facility (CAF) are designed to be provided on the AI. Moreover, Cloud implementation studies have already been performed on HEP institutes and commercial clouds, such as Amazon [62].

The CMS infrastructure recently built on the CMS CERN Cloud resources has to run all the CMS workflows and data handling, from scheduling processing to distributed user analysis. The Cloud resources usage has been explored and successfully exploited for Tier-0 deployment and the HLT farm re-usage during the LHC technical stops in LS1. In fact, the HLT has been optimized for offline computing tasks during LHC no data-taking periods. This turns out to be an important advantage for CMS as the HLT provides a computing capacity comparable in scale to the total offered by the CMS Tier 1 sites, when it is not running as part of the online system. In order to exploit it as explained, a Cloud layer has been overlaid on the HLT resource, making it accessible for general CMS use. optimized exploitation The nature of the site infrastructure (Cloud or Grid) is often hidden from CMS as the site continues to offer resources via a traditional Grid interface. However, Cloud interfaces are always more frequently exposed directly to CMS, that hence had to adapt its submission infrastructure. Automated Cloud systems virtualize computing resources and manage VMs in order to provide flexible and reliable resources to CMS. The CMS experiment interacts with several Cloud systems that rely mainly on an open source Cloud software system, called OpenStack [63]. Specifically, VM images are provided using the tool known as OZ [64]. The CMS SoftWare (CMSSW) is exported on the machines through CVMFS [65] via a set of *http* proxies. The CMS resources are accessed on Cloud using the same tools used on Grid, through the GlideIn-WMS service. In this way, the framework allows the user to submit jobs either on Cloud or on Grid with the requirement to change few parameters in the job description. The GlideIn-WMS has been modified to be able to submit Clouds compatible with Amazon EC2 as well as traditional Grid sites, as it will be explained in Chapter 5.

Chapter 4

Elastic Extension of CMS Computing Resources on External Clouds

In Spring 2015, the Large Hadron Collider has restarted after the first Long Shutdown (LS1), exploited for maintenance and upgrade activities that have led to unprecedented performances. Therefore, the LHC experiments have to face up to new challenges in the design and operation of the computing facilities.

The Run-II computing infrastructure is dimensioned to cope at most with the average amount of data recorded. Anyhow, breakneck use cases could overload the infrastructure as already observed in Run-I.
The usage peaks are axiomatically common during data taking, and they are inclined to originate large backlogs. The available computing resources are often not sufficient to deal with this problem, and the time required to absorb backlogs could be long-lasting, hindering the needs of the experiments. In this way, all data handling and processing activities would inevitably be delayed, causing problems in efficient data availability for physics analysis.

This state of the art has stimulated the CMS experiment to explore the use of Cloud resources. In case of commercial Clouds, they can be bought from external providers when needed in order to cope with usage peaks issues. The feasibility has already been demonstrated as specific use cases have already been explored and Cloud-based solutions have been successfully exploited during LS1 [Ref. 3.2].

The aim of this Chapter is to present the proof of concept of the elastic extension of the CMS-Bologna Tier-3 Site on an external Cloud infrastructure, implemented on OpenStack [63]. A newly designed LSF [66] configuration is used in what it can be considered a "Cloud Bursting" of a traditional CMS Grid Site.

4.1 Virtualization of CMS services

The virtualization is a useful procedure to instantiate additional units of the same service as needed. This is done until the point of saturation of the available hardware, making the continuous uninstallation and subsequent reinstallation unnecessary.

The virtualization of the CMS resources of the Bologna Tier-3 centre serves both as a CMS Grid Site and a Local Farm. In fact, the Bologna Tier-3 Worker Nodes generally deal with both use cases. Accordingly, they are going to be virtualized as hybrids and subsequently specialized.

The first point of focus is on the lightness of the images in order to avoid overloading the infrastructure. Accordingly, effective tests are able to be performed to cope up with the needs of the CMS experiment and local users.

CMS software installation

Software tools are retrieved from remote servers during the execution instead of being installed on the Virtual Machine (VM) image, allowing an image reduction of tens of GigaBytes. The CMS SoftWare (CMSSW) [67] access is provided by the CernVM File System (CVMFS) [68], a read-only network file system based on *http* and optimized to deliver experiment software in a fast, scalable, and reliable way. It was specifically developed to assist HEP collaborations to deploy software on the worldwide-distributed computing infrastructure and used to run data processing applications. In fact, files and meta-data are hosted on standard web servers, downloaded on demand and aggressively cached locally, using only outgoing *http* connections to avoid most of the firewall issues of other network file systems. In this way, a directory structure is transformed into a CVMFS "repository", a form of content-addressable storage.

Authorization and authentication

A custom installation of gLExec [54] packages is provided in order to enable the CMS Grid Site usage. The infrastructure is provided with a central authentication systems based on gLite Authorization Service (ARGUS). The CMS Grid user is authorized through the X.509 certificate by VOMS [Ref. 2.1], and authenticated by ARGUS that uses Grid Pool Account service to map each user to unique individual account.

The local user management relies on Lightweight Directory Access Protocol (LDAP) and Grid Pool Account services. The LDAP is a set of protocols that enables the hierarchical arrangement of corporate directory entries in a structure, which may reflect geographic or organizational boundaries.

Storage set-up

The storage solution used on the Tier-1 at CNAF and also on the Bologna Tier-3 resources is based on the IBM GPFS system [69]. It has been demonstrated that GPFS suffers from clients joining and abandoning the storage access cluster. For this reason, it has been decided to use a GPFS to NFS bridge on the hypervisor so that VM instances can be created and destroyed with the required elasticity without impacting GPFS. In this way, the GPFS access provides:

- user *home*;
- *local* area provided with *posix* access;
- *storm* area provided with *posix* read and *srm* write.

Batch System

The Load Sharing Facility (LSF) [66] is one of the most widely adopted batch systems worldwide, and a considerable experience can also be profited from within the HEP community and the WLCG Tiers. It is a system to manage large applications that can not be run interactively on a machine as they require too much CPU-time, memory or other system resources. For this reason, these large applications have to be run in batch, and they are called *batch jobs*.
The user makes a small file containing all the job specifications and the instructions to run the application, a so-called *batch job file*. It is similar to a shell script, except for the extra job specifications. Thus, the batch job file is submitted to the LSF system with the *bsub* command. The LSF takes care of batch management, handling all the job requests it receives into proper queueing systems. If there are enough system resources available for the job to complete, the LSF starts execution of jobs relying on job specifications.

The newly designed LSF used in this proof of concept has been subjected to central configuration accessible through remote mounting.

4.1.1 Kernel-based Virtual Machine and Kickstart

The Virtual Machines (VMs) are created from scratch using a virtualization infrastructure for the Linux kernel named KVM (Kernel-based Virtual Machine). The hosted VM monitor QEMU (Quick Emulator), that performs hardware virtualization, is used together with KVM in order to run VMs at near-native speed.

The installation method is based on Red Hat Kickstart to automatically perform unattended operating system installation and consistent configuration of several services that have to be provided.
The Kickstart configuration files can be built in three different ways:

- by hand;
- by using the GUI system-config-kickstart tool;
- by using the standard Red Hat installation program, called Anaconda [70].

Anaconda produces an *anaconda-ks.cfg* configuration file at the end of any manual installation. This file can be used to automatically reproduce the same installation or edited.

Thereafter, the VMs are managed through a desktop-driven virtual machine manager known as *virt-manager* (Red Hat Virtual Machine Manager), that uses the API *libvirt* to create and manage virtual machines.
Specifically, the Virtual Machine Manager allows to:

- create, edit, start and stop VMs;
- view and control each VM's console;
- see performance and utilization statistics for each VM;
- view all running VMs and hosts, and their live performance or resource utilization statistics;
- use KVM or QEMU VMs, running either locally or remotely.

Following these guidelines, UI [Ref. 2.1] machines are created from scratch, provided with packages, programs, and tools necessary for CMS users (*e.g. uibo-cms-04* and *uibo-cms-05*).
Moreover, the special use case PhEDEx is provided as a "weakened" UI, enhanced with specific data transfer tools. Here, the experiment software is installed on the *home* of a special user known as *t3phedex*.

In the same way, WNs [Ref. 2.1] are created from scratch taking into account several functions they should have. The WN configurations can be compared to those of the UI without any user package or graphic library, adding Grid Pool Account and gLExec [54] packages for the Grid usage.

4.1.2 Oz Template Description Language

An alternative installation method is based on the Oz Template Description Language (TDL) [71], that turns out to be very efficient regarding usage of the images on Cloud resources. TDL is an XML-based (Extensible Markup Language) language for creating image templates. TDL files define aspects of a VM image including operating system, installation settings, packages, and files. Oz interprets these TDL files and builds them into images, which can be pushed into Cloud providers.

Oz is a tool for automatically installing and customizing operating systems into files with only minimal up-front input from the user. It always uses the native operating system tools to do installs, ensuring that the created disk image is exactly the same as if the installation CD had been used on a bare-metal machine. For each type of guest operating system, Oz supports up to three operations: *operating system installation*, *operating system customization*, and *meta-data generation*, briefly described in the following.

Operating System Installation

The steps that Oz goes through to install an operating system are:

i. download the installation media;

ii. generate an automated installation file (*e.g.* kickstart);

iii. generate a modified installation media that includes the installation file;

iv. run the native installer in the KVM (or QEMU) guest;

v. at the end of installation, shutdown the guest.

Operating System Customization

Additional packages and files are installed into the operating system. This is always done as a separate step from installation, due to few reasons:

1. it reduces the chances of failure during the initial operating system install;
2. it uses the native tools (*e.g.* yum, apt-get) to do installation;
3. it allows customization of operating systems that were not initially installed via Oz.

The steps that Oz goes through to customize an operating system are:

i. modify the operating system disk image to allow remote access;

ii. start up the operating system in the KVM guest;

iii. run remote commands (*e.g.* ssh) to install packages and files;

iv. shut down the operating system;

v. undo the changes done in the first step.

Meta-Data Generation

The generated meta-data are represented in an XML file. The meta-data generation may be combined together with the customization step as an optimization. The steps that Oz goes through to generate an XML file are:

i. modify the operating system disk image to allow remote access;
ii. start up the operating system in the KVM guest;
iii. run remote commands (*e.g.* ssh) to discover the installed packages;
iv. shut down the operating system;
v. undo the changes done in the first step;
vi. output an XML file that lists the packages.

Oz supports a configuration file in the standard INI (initialization) format, as follows.

```
[paths]
        output_dir = /var/lib/libvirt/images
        data_dir = /var/lib/oz

[libvirt]
        uri = qemu:///system
        type = kvm
```

The *output_dir* key describes where to store the images after they are built. The *data_dir* key describes where to cache install media and use temporary storage, namely the work area that Oz uses. The *uri* key describes what *libvirt* URI Oz should use when manipulating the guests. The *type* key describes what type of guest Oz should use when creating *libvirt* guests, namely the type of virtualization to use.

Custom Automated Installation Files

As already mentioned, Oz generates a minimal operating system automated installation file in order to barely get the OS up and running. In this way, it is reduced the possibility of errors during the initial installation phase.
However, Oz allows to specify a custom automated installation file, under the condition that it has to do the following:

i. run all installation steps without prompting;
ii. additional steps or packages must not fail;
iii. automatically shut down the installer (operating system) at the end of installation.

4.2 Extension of the LSF batch queues using a VPN

The LSF does not natively support the dynamic allocation of Worker Nodes in a moment of need. Therefore, the main goal of the work project is to dynamically add new resources to an already busy farm. The newly designed LSF configuration allows the dynamic registration of new WNs to the LSF, thus resizing the standard site by bursting out to third parties. The nodes are added to the farm from *anywhere*, fully independently from their physical location. In this way, the pool of resources is enlarged, enabling the nodes to execute high-priority jobs that would otherwise queue up.

In this tuned configuration, the WNs are virtual machines instantiated on a remote site and made part of the farm. The amount of resources allocated can be elastically modelled to cope up with the needs of Grid and local users. In this prototype, they serve as an extension of the CMS-Bologna Tier-3 Farm, but the same approach could be followed by any CMS Tier level.

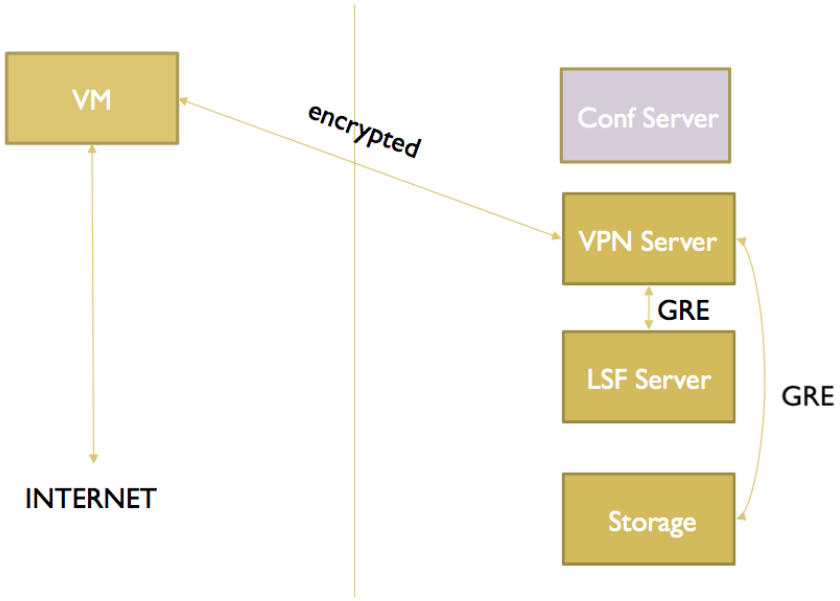

Figure 4.1: Schematic view of the dedicated LSF configuration.

Examining in depth the dedicated LSF configuration (Figure 4.1), each VM contacts the Configuration Server in the bursting site at boot, and it is subjected to an authorization phase. Afterwards, the Configuration Server sends a set of configuration files and commands allowing the VM to establish a VPN (*Virtual Private Network*) connection with a VPN server inside the computing centre. Meanwhile, the VPN server has established GRE (*Generic Routing Encapsulation*) tunnels with the machines (*e.g.* LSF Server, Storage) that have to be visible by the VM. This is required in order to allow the working and adjustment of proper routes.

The described implementation provides several advantages, as follows:

Efficiency the traffic is redirected to the farm for the minimum necessary;

Dynamism the resources management is provided through *boot* and *terminate* of the instances;

No Hypervisor constraint the virtual machine can be launched on whatever Cloud provider;

Few Requirements the virtual machine requires only one *rpm*.

It is necessary to take into account that the VPN connection is a "pure" one, as the VM and the OpenVPN server are visible to each other. However, it has no further effect on the network connectivity of the VM. In fact, it remains unchanged and behaves as if the VPN was not established for all addresses not internal to the computing centre. This will ensure that the computing centre is reached only by the proper traffic, avoiding the problems that may be caused by an increased network latency due to the geographical distance from the computing centre.

In this way, the new configuration allows computing centres to accept workloads larger than those they have been dimensioned to accept.

4.2.1 Validation of the system for physics analysis

The main goal of the CMS Computing Model is to enable the analysis of the data originated from the proton-proton and heavy-ion collisions in order to bring new discoveries and precise measurements in HEP.

The proof of concept here described is designed as an infrastructure for enabling and eventually easing the end-user physics analyses. For this reason, the best way to validate the full system is to chose one or more *real* analysis workflows and verify if the infrastructure is able to support such real use cases and if (for example) any significant loss of efficiency, or other drawbacks, can be observed in comparison with the benefit of having extra resources available.

In the context of the physics analysis performed inside the Bologna Physics group, the Top Quark mass measurement in the all-hadronic channel has been evaluated as a good candidate for exploiting the usability of the system both as a Grid site, for the most intensive tasks, and as a local farm for the final analysis.

The first task ever performed for the physics analysis, after having identified the datasets of interests, is a selection of the events based on the specific analysis, such as, in our case, selecting the events with a minimum number of jets and aggregating information related for instance to Monte Carlo simulation to the reconstructed physics objects. At this stage, usually performed profiting of the Grid infrastructure, the analysis

condensates the most CPU expensive tasks and reduces the size of the data to reasonably fit to a local, easy to access, storage.

This task was performed in CMS by the analysis group during Run-I. Gaining experience, for the Run-II, the physics groups are providing general requirements to allow the central production of the described MiniAOD data format. Although this step is thus performed centrally in Run-II, we used a standard MiniAOD creation workflow as a test workflow for the extension of the Tier-3. It ensures a reasonable CPU intensive task for the performance measurement and is exposed to all the Grid related features and issues as for instance the bandwidth for remote data access.

The second workflow is instead the final analysis task performed by the Bologna analysts before the actual Top Quark mass measurement. Data are accessed directly from the local storage, mostly by mean of jobs submitted through the batch system. More fine tuned selections, weighting and eventually calculations are performed and a final *n-tuple* data format is returned in small sized files, often accessed interactively for the production of the final results.

In summary, three possible test are thus performed in order to validate this proof of concept:

Type 1 : *"Hello World"*
A simple bash script that prints every second per 2 minutes, lasting enough to let the operator notice jobs running on the queues.

Type 2 : *User n-tuples creation for Top analysis*
User *n-tuples* creation for Top analysis; a close to final analysis with direct GPFS access on skimmed data. This allow also qualitative consideration on the direct storage access in Cloud environment.

Type 3 : *Standard MiniAOD CMS workflow creation*
A *xrootd* access to data is performed. Standard MiniAOD CMS workflow creation; a more CPU-intensive task, routine task for CMS computing centres, allowing a performance evaluation of the infrastructure; the data access is forced to be in "data federation", *e.g.* through the *xrootd* streaming protocol, exposing the jobs to network fluctuations and bottlenecks. The jobs are dimensioned to run each over one single file and lasting about 2 hours each.

4.3 Elastic management of the CMS Local Farm

The "burst" modality allows to cope with ever-increasing requests of computing resources. Periods of normal usage are usually interspersed with peak usage periods where resources usage greatly increases. Traditional non-commercial scientific computing centres are not able to cope with this. In fact, the peak usage can not be absorbed without generating excessively long queues and therefore jeopardizing the usage of the computing centre for other users.

Here, the underlying theme is to present the dynamical extension of the Bologna Tier-3 Local Farm before implementing the use case on an external Cloud as presented in Section 4.4. The architecture of the Bologna Tier-3 Farm is described in the following.

The Bologna Tier-3 Farm

- 3 Hypervisor (vmbo-t3-[01-02-03]) hosting virtual machines:
  - 6 User Interface machines
  - PhEDEx
  - 2 test machines
- 2 CE (cebo-t3-[01-02])
- 1 ARGUS server (cebo-t3-03)
- 2 Top-BDII (sgbo-t3-[01-02])
- 4 User Interface machines (2 CMS + 2 ATLAS)
- 1 SE (sebo-t3-01)
- 40 Worker Nodes
- GPFS Cluster
- GridFTP server (*Gridftp-storm-t3.cr.cnaf.infn.it*)

The standard site has been resized in order to obtain an extension of the CMS-Bologna Tier-3 Farm for the local usage. Dynamic registration of new WNs to the LSF is allowed using the extension mechanism [Ref. 4.2], and the amount of resources allocated are elastically mouldable for the needs of local users. In this tuned configuration, the WNs are virtual machines instantiated on the site and made part of the Local Farm. In this way, the new LSF configuration gives new resources to the CMS-Bologna Tier-3 Farm, which hence can accept workloads larger than those it has been dimensioned to accept.

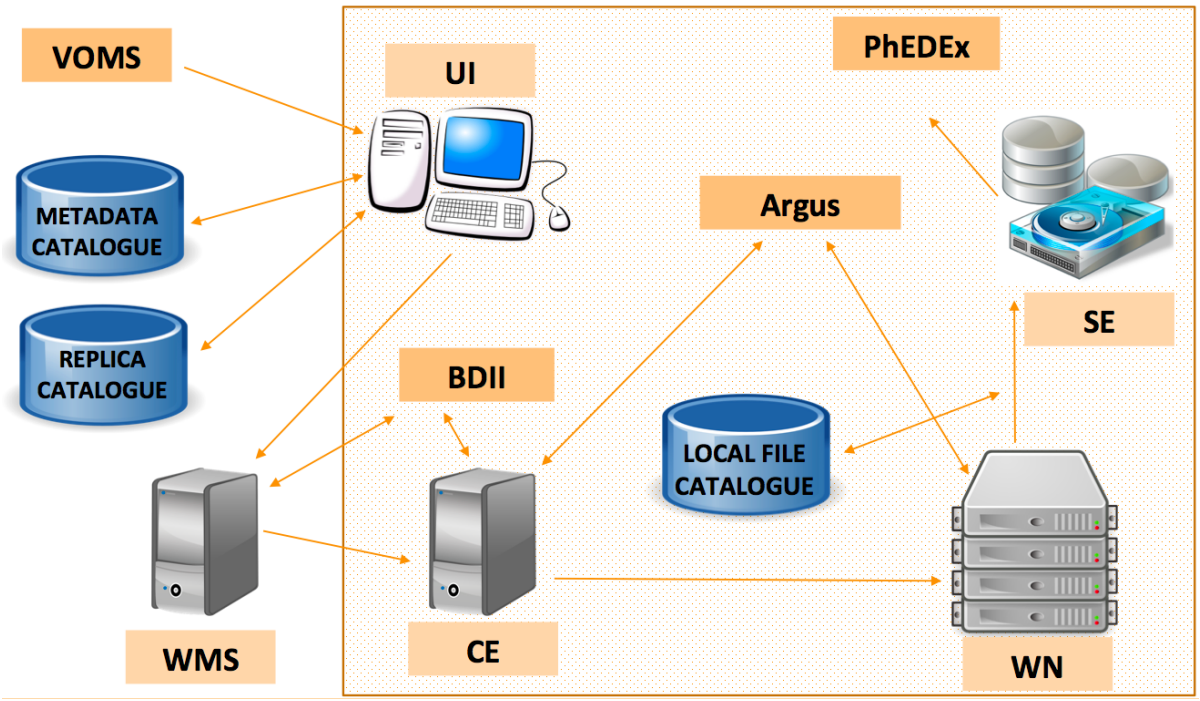

Figure 4.2: Standard CMS Site architecture.

4.3.1 Extension on Cloud OpenStack (Havana) infrastructure

In this approach, the WNs are instantiated on the CNAF-Cloud infrastructure, an *"Infrastructure as a Service"* (IaaS) [Ref. 3.1] based on OpenStack-Havana, even if its validity is independent from the third-part provided infrastructure. Therefore as before, the dynamic registration of new WNs to LSF is performed [Ref. 4.2], resizing standard site by bursting out to the CNAF-Cloud resources. In this tuned configuration, the WNs are virtual machines instantiated on the CNAF-Cloud site and made actually part of the Local Farm, and they turn out to be transparently accessible from *anywhere* outside. Note that the images need finite tuning to run properly in OpenStack, as follows:

- addition of cloud-aware packages;
- removing local storage access;
- implementing dynamic resizing of the partition disk;
- redefinition of network configuration (DHCP);
- any further contextualization.

The elastic extension on the external OpenStack-based Cloud infrastructure is performed through three activities:

- virtualization of the local resources and dynamic allocation;
- dynamic extension of the Bologna Tier-3 CMS Farm in external OpenStack resources;
- direct access to external OpenStack resources.

The first activity is considered as introductory to the others and it has been already treated [Ref. 4.1-4.2], while the third activity is reported in Chapter 5 as a possible use case relating to direct access/integration of OpenStack resources by the CMS Workload Management System.

4.3.2 Tests for the Local Farm extension approach

Accordingly to the Local Farm approach, the basic functionalities have been tested through *"Hello World"* jobs and subsequently through a workflow belonging to the Top Quark mass measurement analysis.

Elastic management of the Local Farm

The tests have been performed under the following setup:

- WNs statically instantiated on the Tier-3 Hypervisor
- WNs included in a test queue (T3_TEST) of the Tier-3 LSF batch system
  - tested with 100 "Hello World" (*Type 1*)
  - tested with 100 analysis jobs (*Type 2*)
- WNs included in a private Master LSF, instrumented for the dynamic extension of the queues
  - tested with 100 "Hello World" (*Type 1*)
  - tested with 100 analysis jobs (*Type 2*)

The system has positively responded to simple *Hello World* jobs. However, the access to the local storage (user *home* and data *area*) turns out to be a real bottleneck.

Extension on Cloud infrastructure

The tests have been performed under the following setup:

- WNs instantiated in OpenStack
- WNs included in a test queue (T3_TEST) of the Tier-3 LSF batch system

  - tested with 100 "Hello World" (*Type 1*)
  - tested with 100 analysis jobs (*Type 2*)

- WNs included in a private Master LSF, instrumented for the dynamic extension of the queues
  - tested with 100 "Hello World" (*Type 1*)
  - tested with 100 analysis jobs (*Type 2*)

In conclusion, the dynamic extension of the nodes in the local Tier-3 cluster turns out to be perfectly working. As a matter of principle, the use case concerning direct access to user *areas* is feasible for the argued approach. However, the study of a solution for direct access to local storage is required. In fact, the system's shortcomings regarding optimization of storage management lead to an impractical use of the mentioned use case on remote services. In particular, GPFS turns out to be a problematic issue in a dynamic environment as it suffers from clients joining and abandoning the storage access cluster, and a direct plug is unworkable. For this reason, it was decided to use a GPFS to NFS bridge on the hypervisor as no-scalable solution.

On the other hand, the Cloud OpenStack-Havana infrastructure turns out to be the natural extension to acquire external opportunistic resources. However, a similar result can be obtained remaining within the same farm in order to acquire internal opportunistic resources.

Performance tests for elastic management of the Local Farm

The performance tests for the elastic management of the Local Farm have been performed through *Type 3* workflow in order to have a CPU-intensive usage of the infrastructure. It is important to underline that a *xrootd* access to data is performed.

The use case of the dynamic extension of the Bologna Tier-3 Local Farm has required several preliminary tests before implementing the use case on an external Cloud. The actual tests have been performed through the submission of the same CMS job executed a certain number of times both on the standard WNs and on a virtual WN. The virtual WN has been instantiated on the Tier-3 Local Farm as a virtual service. The virtual WN image will also be used for the instantiation on the OpenStack infrastructure except for network configuration.

The submission of a single job executed several times has provided valuable tests on the measurement of jobs efficiency as a ratio of CPU-time to Wall-Clock-Time. The Figure 4.3 refers to the comparison test providing the standard execution efficiency of a single job performed several times. The test is carried out submitting to the standard production WNs of the Bologna Tier-3 Site. The result is a mean execution efficiency of 0.932 ± 0.014. However, no distribution is able to properly fit the resulted distribution.

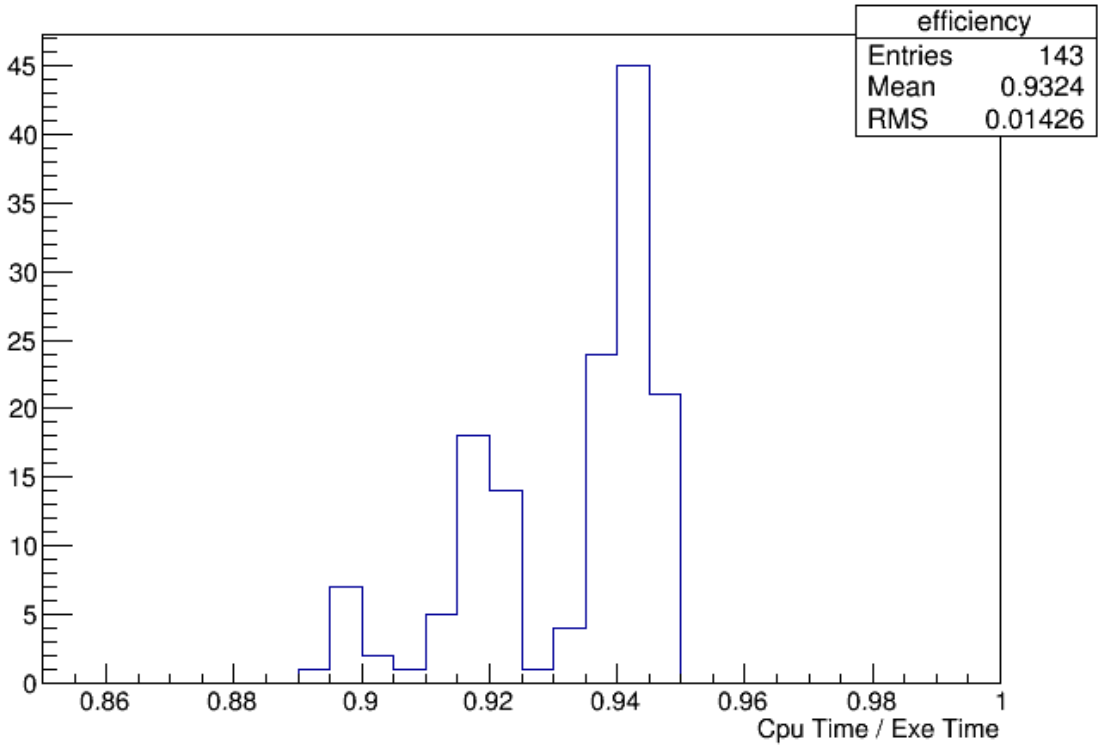

Figure 4.3: Efficiency for Single Job - standard WNs.

In particular, the distribution presents three peaks not profusely understood, leading to an efficiency of roughly 0.90, 0.92, and 0.94 respectively.
A consideration about different performance for file access could be pursued as systematic effect frequently observed during entire measurement. As a matter of fact, the Bologna Tier-3 Site can not be treated as a regulated-for-tests environment. In particular, the WNs are accessed by several user jobs of very different nature, causing an unpredictable usage of memory CPU and I/O. The usage of the Tier-3 resources in a time-regulated environment as in this case could still lead to a decrease of job efficiency. In light of this consideration, the distribution presents two lower peaks that do not add information when comparing with the virtual WN case.

The Figure 4.4 shows the distribution of jobs execution efficiency for the virtualized WN in the Tier-3 Site. The result is a mean execution efficiency of 0.944 ± 0.005, and the distribution perfectly fits to the normal (Gauss) distribution.

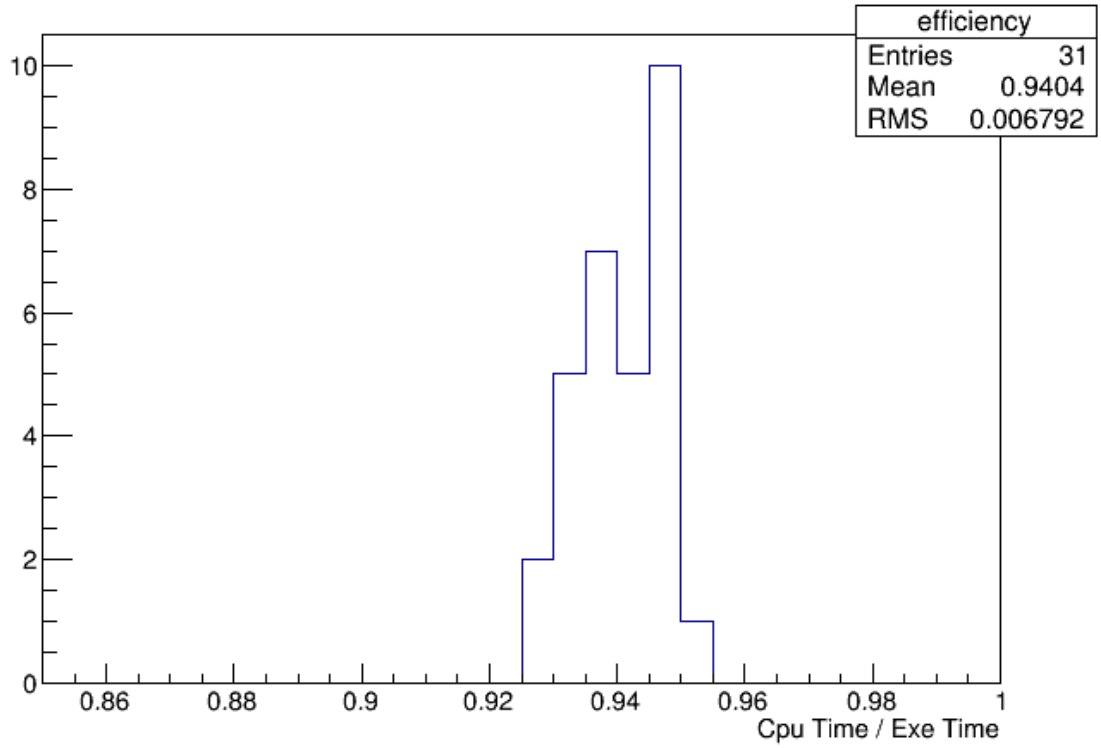

Figure 4.4: Efficiency for Single Job - virtual WN.

Comparing the distributions relating to the two different submission approaches, the distributions are consistent with each other, except for the peaks shifted of the standard approach that have already been discussed. The result is a fitting acceptable within 2σ significance level between the tests performed.

To conclude, the virtual WN approach turns out to be proficient as much as the standard one, and the dynamic extension of the nodes in the local Tier-3 cluster proves to be perfectly working.

4.4 Dynamic extension of the CMS-Bologna Tier-3 Grid Site as "Cloud Bursting"

The possibility to dynamically extend the computing resources is crucial for the HEP experiments and is becoming more and more appealing for many e-Science fields. According to available funds, some experiments require to acquire external resources. Contrariwise, other experiments have to rely on opportunistic resources, eventually provided by Cloud infrastructures. In this way, the maintaining of proprietary farms is avoided in order to cope with occasional needs.

The approach implemented for the Local Farm has been exploited in order to achieve the "Cloud Bursting" of the CMS-Bologna Tier-3 Grid Site. The dynamic extension of the computing resources is provided by the CNAF-Cloud OpenStack-Havana infrastructure, thus using external opportunistic resources. The dynamic registration of new WNs, that are actually virtual machines, to LSF is performed, resizing standard site by bursting out to CNAF-Cloud resources, even if the approach is independent from the third-part provided resources.

4.4.1 Tests for Cloud Burst approach

Accordingly to the Grid usage approach, the basic functionalities have been tested through basic Grid jobs. The tests have been performed under the following setup:

- WNs instantiated in OpenStack
- WNs included in a test queue (T3_TEST) of the Tier-3 LSF batch system
  - tested with direct job submission to the CE (*Type 3*)
- applied the dynamic extension to the Tier-3 LSF batch system
  - tested with direct job submission to the CE (*Type 3*)
- WNs included in the official queue (T3_BO) of the Tier-3 LSF batch system
  - tested with standard CRAB2 and CRAB3 job submission (*Type 3*)

The Master LSF has been reconfigured in order to allow the dynamic extension of the nodes. A test-queue of the Master LSF is used to include WNs, which are tested with direct Grid submission. Afterwards, the nodes are added to the LSF queue published for Grid submission. The performance functionalities have been tested through a workflow belonging to the Top Quark mass measurement analysis in real usage conditions:

- submission of hundreds jobs for each test;
- submission to WNs included in the production queue (T3_BO);
- use of the Master LSF Server.

The submission of the Top Quark skimming workflow has provided valuable tests on:

- access to remote data through *srm*/*xrootd*;
- copy of the results with standard Grid command in the destination storage;
- measure of jobs efficiency as a ratio of CPU-time to Wall-Clock-time.

Performance tests for Cloud Bursting of the Grid Site

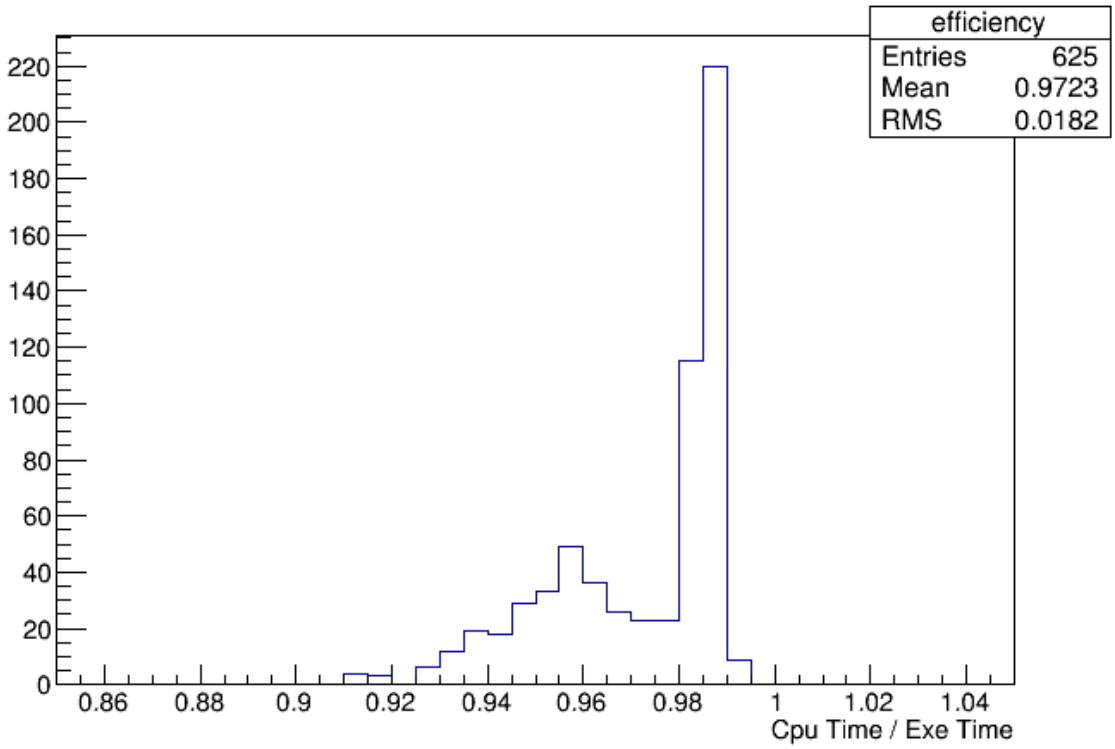

Figure 4.5: Efficiency for T3-Burst - standard WNs.

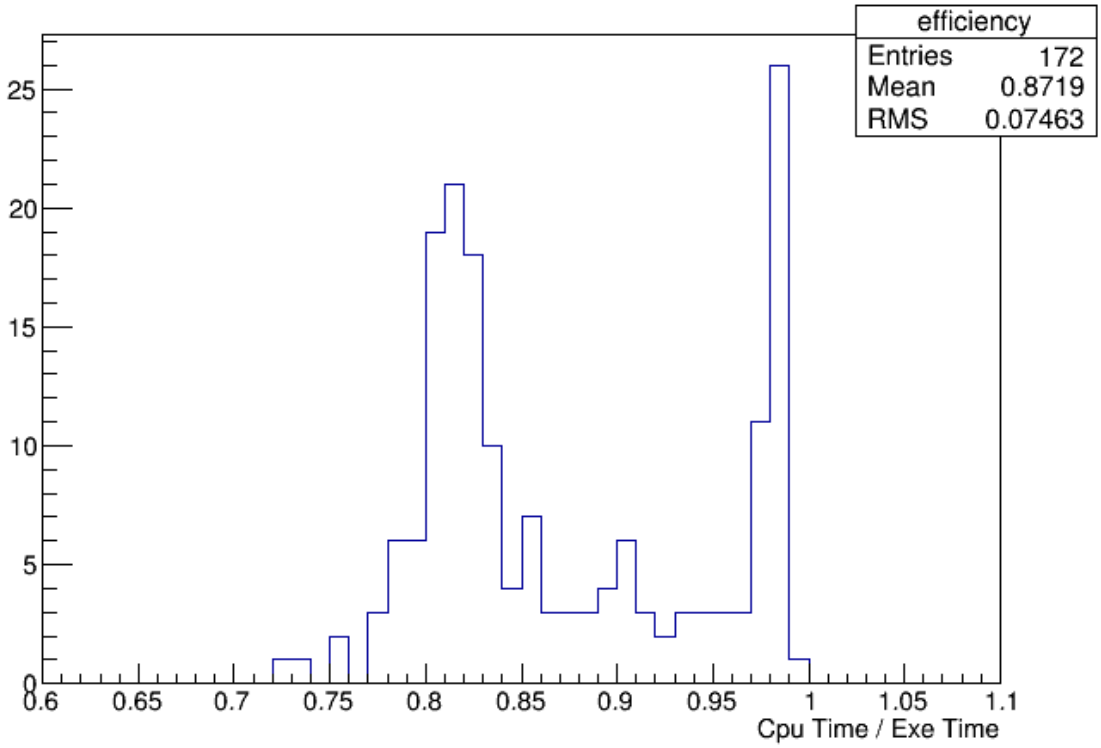

Figure 4.6: Efficiency for T3-Burst - OpenStack WNs.

In light of positive results regarding preliminary tests, the Bologna Tier-3 Grid Site has been subjected to actual tests concerning the dynamic extension on Cloud resources in order to validate the underlying proof of concept. A standard-usage test (Figure 4.5) is provided in order to compare it with the Cloud Bursting use case.
Several thousand CMS Grid jobs have been submitted to the Bologna Tier-3 Grid Site that includes the 40 standard WNs and the 6 Cloud WNs. The Cloud WNs are CMS-type and have been instantiated on the CNAF-Cloud OpenStack-based infrastructure as a virtual services, based on the virtual WN image previously instantiated on the Tier-3 Local Farm. The execution time of this job type is two hours both on the standard WNs and on the Cloud WNs. The results show that the submitted jobs are distributed among all WNs in proportion to the resources availability in the two infrastructure, and no failed execution on Cloud WNs has been registered.

The submission of the Top Quark skimming workflow has provided valuable tests on the measure of jobs efficiency as a ratio of CPU-time to Wall-Clock-Time. The Figure 4.6 refers to submission to all Cloud WNs performed with concurrency of other users accessing the same dataset. The distribution is spread on a wide range, and presents

several peaks with a mean of 0.87 ± 0.07. The concurrency of users accessing the same data decreases the mean efficiency in comparison to standard submissions, and causes the presence of several peaks, as discussed during preliminary tests.

Comparing the distributions relating to the two different submission approaches, the behavior of the jobs on the various types of WNs is similar enough not to be noticed by the end-user. The Cloud bursting approach turns out to be proficient as much as the normal usage of the Grid site, successfully validating the underlying proof of concept that allows the Grid site to obtain opportunistic Cloud resources efficiently.

The Local Farm approach issue is largely overcome in this context, since the access to the local storage is no more performed. Moreover, the Grid usage approach does not use *local users* but only Grid Pool Account. As a matter of fact, the Cloud Bursting of the Bologna Tier-3 Grid Site turns out to be the natural evolution of this proof of concept and the solution proposed for the production peaks absorption.

To conclude, it turns out that the CMS-Bologna Tier-3 Farm using LSF as a local batch system can be efficiently and dynamically expanded on external Cloud-aware resources without a serious loss of performances. Hence, the resources dynamically allocated can be used for the normal operation of the site.

Chapter 5

Cloud Computing: HEP Computing "as a Service"

The CMS experiment has to face up to new challenges in the design and operation of the computing facilities to cope with the ever-increasing physics analysis requests. Breakneck use cases, like for instance common usage peaks during data taking, could overload the infrastructure saturating available resources. Anyhow, a large amount of available allocated resources could remain unused during specific no data-taking periods.

The use of on-demand allocated Cloud resources has been explored by the CMS-Bologna Cloud Group in order to dynamically exploit temporarily-unused available resources. In this way, the same physical hardware resource can be used for several different use cases only for the time necessary.
However, the focus of the principal concept is to properly employ potential opportunistic resources provided (or bought) from external partners or commercial providers.

The use of on-demand Cloud resources has already been explored and successfully exploited by the CMS experiment for Tier-0 deployment and the HLT farm re-usage during the LHC technical stops in LS1 [Ref. 3.2]. As already amply demonstrated in Chapter 4, the CMS-Bologna Cloud working group has successfully covered this field with the "Cloud Bursting" mechanism.
Here, the purpose is to demonstrate a suitable use of direct access to external OpenStack-based Cloud resources.

Therefore, the aim of this Chapter is to present the proof of concept of the use of on-demand allocated Cloud resources in the Bologna Tier-3 CMS Site as an alternative to the extension of the Grid site presented in Chapter 4.

5.1 Workflow Management

The CNAF OpenStack infrastructure has been the subject of a prototype test regarding the use of on-demand Cloud resources directly accessed by the CMS Workload Management System as if belonging to the Bologna Tier-3 CMS Site. In particular, a suitable use of direct access to the external OpenStack-based Cloud resources has been explored, demonstrating no loss of efficiency in comparison with standard Grid access.

In this context, it is important to take into account that the final CMS user would not perceive any changes in the job submission phase. The user continues to interface with the infrastructure through CRAB tools, contacting the related CRAB server that is in charge of the communication with the GlideIn-WMS. On the other hand, several important changes occur considering the GlideIn-WMS side.

Referring to Figure 2.3, the GlideIn-WMS contacts the Computing Element of the site that handles the load and queues on site's batch systems, pending the users' jobs and managing the communication with the infrastructure. Specifically, the CE sorts the load of the infrastructure to several WNs at its own discretion, trying to even out the work between WNs.

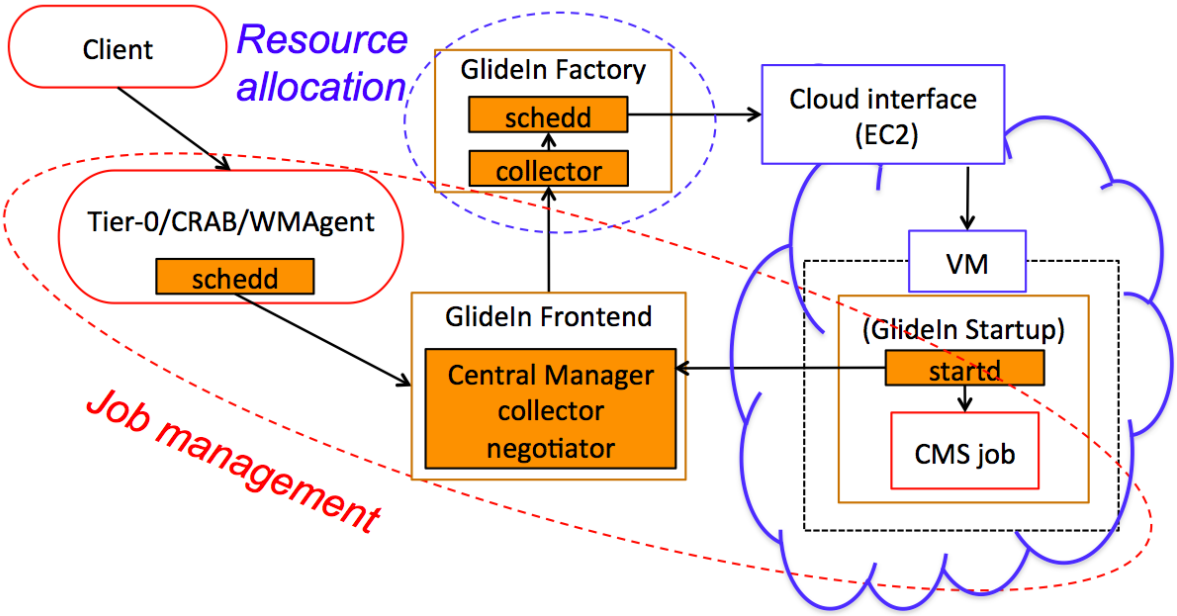

Figure 5.1: CMS Grid workflow using GlideIn to OpenStack EC2 interface.

Henceforward, the communication with the standard CE interface is substituted by the communication with the EC2 (*Amazon Elastic Compute Cloud*) interface of the OpenStack-Havana infrastructure (Figure 5.1). EC2 allows scalable deployment of applications by providing a Web service through which GlideIn-WMS can boot an AMI-type

image to create a virtual machine called "instance". It is important to note that an AMI (Amazon Machine Image) is a special type of virtual appliance used to instantiate VMs within the EC2 environment, and it serves as the basic unit of deployment for services delivered using EC2.

The instance contains any software desired, and in this context it is based on a CMS Worker Node image. In this way, EC2 allows GlideIn-WMS to elastically instantiate virtual machines running the *HTCondor startd* processes that are able to fetch users' jobs. Moreover, the GlideIn-WMS can create, launch, and terminate these VMs as needed, hence the term "elastic".

Therefore, the Cloud infrastructure turns out to be the proper environment to use of on-demand and opportunistic resources.

5.1.1 GlideIn-WMS interfacing

Originally, the GlideIn-WMS has no connection with the Cloud OpenStack infrastructure, requiring a proper link that has to be established between GlideIn-WMS and the EC2 interface. Therefore, several information have to be transmitted to the GlideIn-WMS:

- EC2-AMI ID of image;
- EC2 credentials of user;
- flavor to use for VM instance;
- EC2 endpoint.

The EC2-AMI ID refers to the CMS WN image uploaded on the CNAF-Cloud OpenStack-Havana infrastructure and available to the CMS project tenant. It can be acquired using *euca-describe-images* command, and it contains several information that have to be registered on the GlideIn-WMS for the communication, such as:

Image ami-00000094;

Name WN-BO-T3-SL66-v1.1;

ID 1bc627f1-041a-4bde-8601-ad6fc4b57f59;

Checksum 090575455d31db27b66cc299e99bb629.

The EC2 credentials refers to the project manager of the CMS tenant on the OpenStack infrastructure. In particular, the GlideIn-WMS needs *AccessKeyID* and *SecretAccessKey* of the user account supervisor to be used for the submissions. Therefore, EC2

user keys provides the OpenStack infrastructure with proper authentication in order to create VMs.

The *flavor* refers to specific characteristics that VM instance has to have during creation from the image. EC2 offers CMS project tenant with certain available flavors, as follows.

Name	VCPUs	Root Disk	Total Disk	RAM
m1.tiny	1	1 GB	1 GB	512 MB
m1.small	1	10 GB	10 GB	2,048 MB
m1.medium	2	20 GB	20 GB	4,096 MB
m1.large	4	40 GB	40 GB	8,192 MB
m1.xlarge	8	80 GB	80 GB	16,384 MB

Considering the availability of the CMS project tenant on the CNAF-Cloud OpenStack-Havana infrastructure, the instantiation of CMS WNs is alternately provided as:

- 3 VMs with 8 VCPUs and 16,384 MB RAM;
- 6 VMs with 4 VCPUs and 8,192 MB RAM.

The EC2 endpoint refers to the *port* of the OpenStack infrastructure (*e.g. http://cloud ctrl02-e.cloud.cnaf.infn.it:8773/services/Cloud*) that has to be contacted by the GlideIn-WMS in order to create the connection.

Additional operations have to be performed on GlideIn-WMS in order to establish a proper connection between *users* and *frontend* (*e.g.* reachable through host *t2-gwms-02.pd.infn.it.*), as follows:

- enable access via *gsissh* to the GlideIn-WMS Frontend;
- create user accounts;
- enable user DNS;
- configure entry of the site (T3_IT_Bologna);
- open GlideIn-WMS to submission host (*e.g.* UI) to elude firewall.

5.1.2 Dynamic Allocation of Worker Nodes

The proper interfacing of the GlideIn-WMS to EC2 interface has been established as explained in Section 5.1.1.
On the other hand, few requirements are needed in order to interface EC2, as follows:

- opening of several EC2 *ports* in order to allow GlideIn-WMS connection;
- prior registration of images on CMS tenant in the CNAF-Cloud infrastructure;
- inclusion of GlideIn bootstrap *rpm* in the VM image.

As already mentioned, the virtualization of the CMS resources of the Bologna Tier-3 centre allows to instantiate virtual machines as Worker Nodes function. In this approach, the WNs are instantiated on the CNAF-Cloud infrastructure even if its validity is independent from the third-part provided resources.

The Cloud site hosts a pre-built image and the GlideIn-WMS is then able to use the EC2 interface to request that a VM be built from this image, the batch slot is installed and configured as a Condor executing machine. A *contextualization* process can be performed on the VM once built in order to install site-dependent packages or high-level application. Note that it is unworkable to have a unique "certified" image provided with a specific local customization that is usable on any site. Thus in this context, the contextualization turns out to be not possible even though some specific site customizations scripts can be defined at the GlideIn-WMS level. Finally, an HTCondor *startd* is started on the machine allowing to join the distributed HTCondor pool. In this way, user jobs run on the HTCondor overlay batch system, whose size changes according to the resource availability in the Cloud site.
The GlideIn work in Cloud can be summarized as follows:

- GlideIn startes as service when the VM is created;
- GlideIn *rpm* pre-script creates running unprivileged user;
- download startup script;
- perform checks defined in the GlideIn-WMS Frontend;
- download and execute *HTCondor startd*;
- *startd* runs multiple single core jobs and/or multi-core jobs to "fill" the machine;
- GlideIn stops its execution in case of error or when there is no more work;
- VM stopped and deleted.

Naturally, the GlideIn lifetime turns out to be the VM lifetime. However, it is possible to configure the GlideIn lifetime at the GlideIn-WMS level, and configuration is additionally possible for:

Maximum Time : time that the VM can be idle before shutting down;

Retire Time : no new user jobs are accepted after this time;

Job Max Time : after Retire Time is exceeded, user jobs can run at most for this time otherwise they are killed;

Maximum Wall Time : maximum overall time for the VM.

It has to be noted that the GlideIn-WMS creates a different SSH key for each requested virtual machine so that if a key is compromised, only the concerned VM is affected.

5.1.3 Forefront CRAB3 submission

The job submission tool for CMS (CRAB) has recently undergone upgrading from version 2 to 3. The work of this thesis has provided the very first submission via CRAB3 towards a Cloud infrastructure. In particular, this was performed connecting the CERN Integration Test Bed (ITB) GlideIn-WMS to the EC2 interface of the CNAF Cloud. It is important to note that this is the very first CMS-wide test ever done in the experiment, and thus attracting large interest from the CMS research teams working on Cloud topics.

This activity was possible thanks to the daily collaboration with the CERN-ITB Group. In particular, the background setup of Test Bed has been performed solving several issues an tuning specific configuration. The debugging has been the critical part of the whole work, and several misconfigurations of the GlideIn have been successfully corrected and recorded to the attention of the ITB Group. The configuration set up for this thesis is the basis for what will be used CMS-wide regarding the direct submission to EC2 interfaces using CRAB3.

5.2 Prototype testing with Top physics workflows

The use of on-demand allocated Cloud resources has been the subject of several tests in order to validate the underlying proof of concept. The effective tests have been performed using CRAB2 through the submission of typical CMS jobs to the CNAF-Cloud infrastructure, occurred contacting the CMS-Padova GlideIn-WMS. Moreover, the very first tests CMS-wide have been performed concerning the usage of CRAB3 using the CERN-ITB GlideIn-WMS.

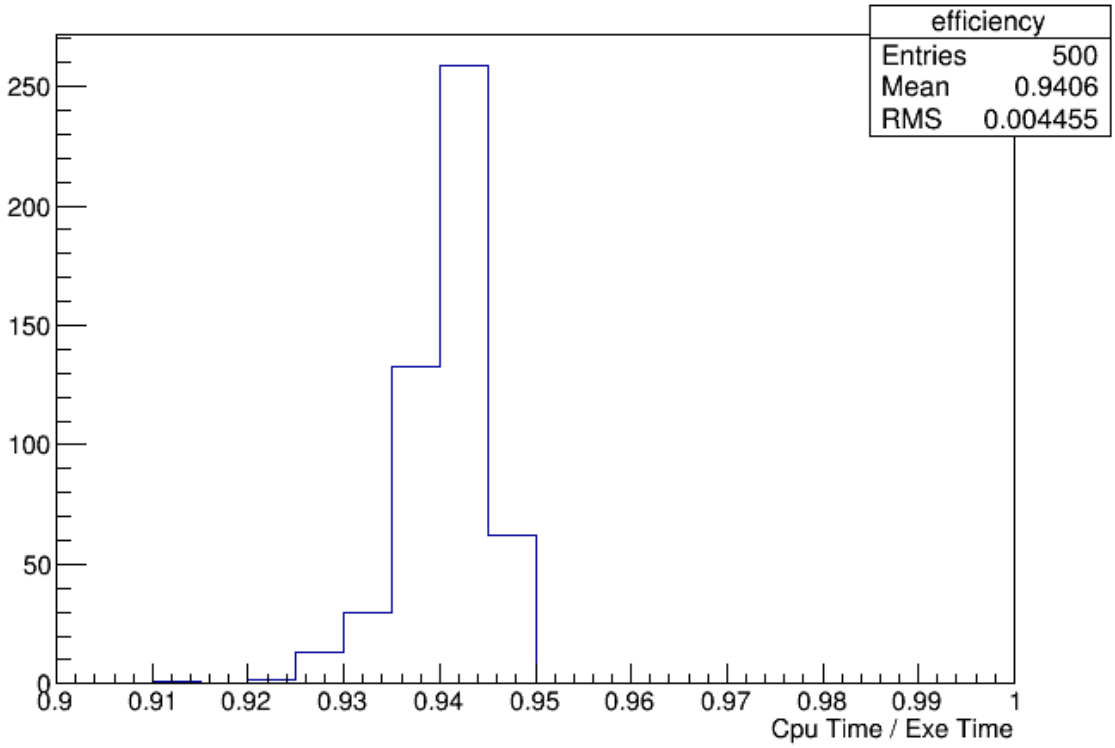

Figure 5.2: Efficiency for CRAB2 - standard WNs.

The submission of the Top Quark skimming workflow (*Type 3*) has provided valuable tests on the measure of jobs efficiency as a ratio of CPU-time to Wall-ClockTime. The Figure 5.2 refers to the comparison test providing the standard execution efficiency of a Grid submission to the CMS-Bologna Tier-3 Site. The test is carried out using 500 jobs executed on the standard WNs of the Tier-3 site. The result is a mean execution efficiency of 0.941 ± 0.004, and the distribution perfectly fits to the normal (Gauss) distribution. Accordingly to Figure 5.3, a measure of jobs efficiency following submission to the CNAF-Cloud infrastructure is performed. The test is carried out using 500 jobs executed on the virtual WNs instantiated on Cloud resources managed by CMS-Cloud tenant. It is important to note that the jobs are actually the same as before and the data are accessed from the same locations using the same protocols. The distribution is spread on a range wider than the previous one, and its mean is located at 0.908 ± 0.015. However, this could be due to different latencies of file access, as frequently experienced over the course of the entire proof of concept. Finally, the efficiency turns out to be compatible with the reference test within 2σ.

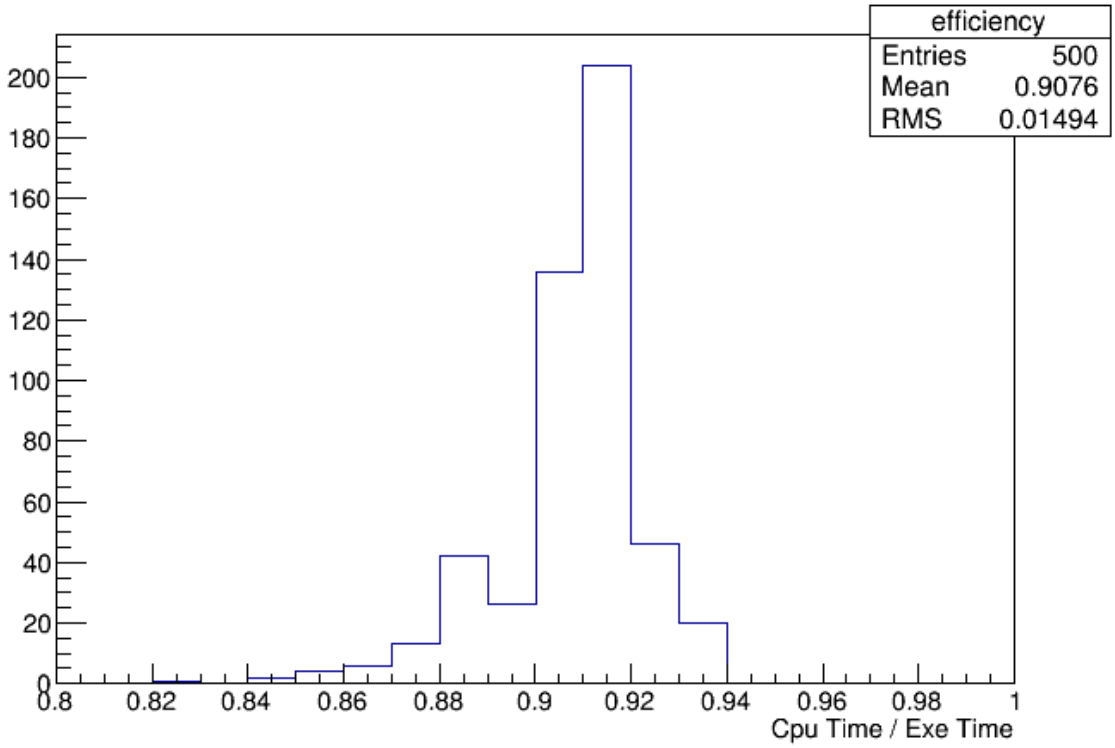

Figure 5.3: Efficiency for CRAB2 IaaS - OpenStack WNs.

Comparing the distributions related to the two different submission approaches, a shift to lower efficiency characterizes the execution of jobs in Cloud environment. However, the shifting is not statistically significant in the context of the collected statistic. In conclusion, the Cloud approach turns out not to be introducing significant inefficiencies at this scale.

As can be observed in Figure 5.4, the very same conclusion applies for the CRAB3 case although the submission is limited to 100 jobs given the limited availability of the CERN-ITB GlideIn-WMS, small dimensioned and used for CMS-wide official tests.

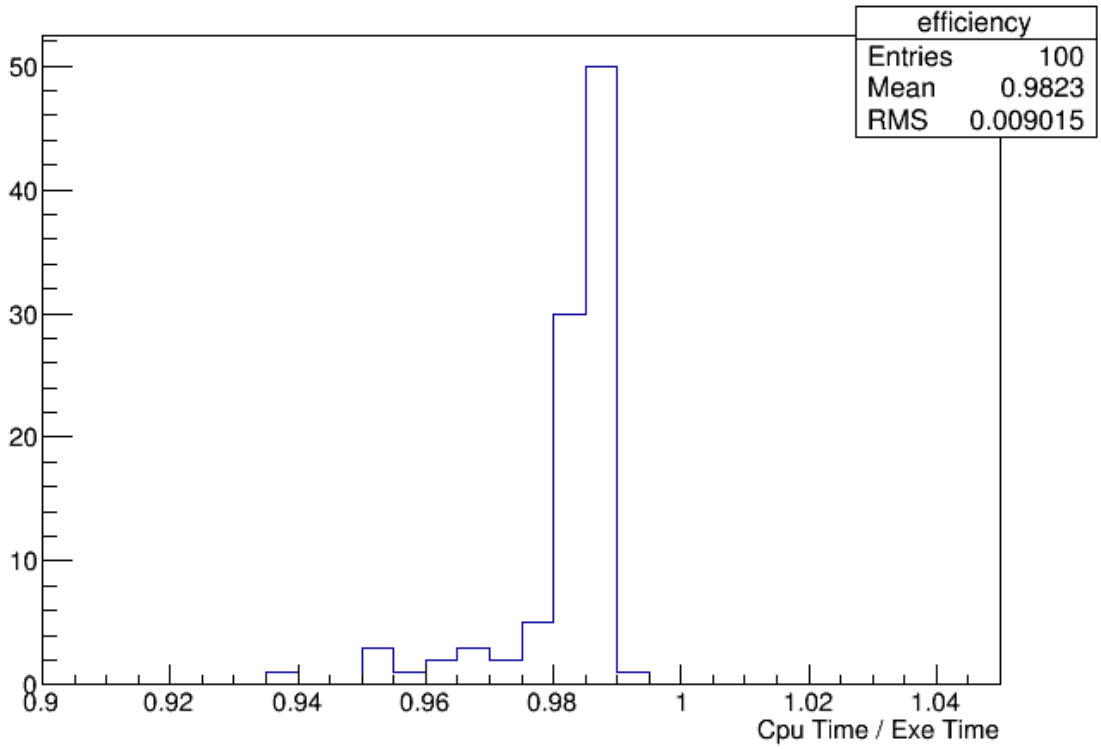

Figure 5.4: Efficiency for CRAB3 IaaS - OpenStack WNs.

Conclusions

After the successful LHC data taking in Run-I and in view of the future Runs, the LHC experiments are facing new challenges in the design and operation of the computing facilities. In order to cope with these challenges, CMS is exploring the opportunity to access Cloud resources provided by external partners or commercial providers. Cloud allows to access and utilize not-owned large computing resources providing CMS with the capability to elastically and flexibly access and use the computing resources. The possibility to dynamically extend the computing resources is crucial for the HEP experiments that often have to rely on opportunistic resources, eventually provided by Cloud infrastructures.

The work of this thesis has presented the proof of concept of the elastic (dynamic) extension of the CMS-Bologna Tier-3 site on the external OpenStack-based CNAF-Cloud infrastructure. The focus has been on the "Cloud Bursting" of the CMS Grid site using the newly designed LSF configuration that allows the dynamic registration of new worker nodes to LSF otherwise not natively designed. The "burst" modality allows to cope with ever-increasing requests of computing resources, avoiding to jeopardize the usage of the computing centre for users during high requests of computing availability. The dynamic registration of new Worker Nodes to LSF is performed, resizing the standard site by bursting out to the CNAF-Cloud resources. In this approach, the dynamically added worker nodes instantiated on the OpenStack infrastructure are transparently accessed by the LHC Grid tools and at the same time they serve as an extension of the farm for the local usage. In this way, the amount of resources allocated have been elastically modelled to cope up with the needs of CMS experiment and local users. The elastic extension has been tested using real physics use cases, specifically the conversion of the CMS reconstructed events in a lightweight format suitable for the analysis. Thus, the CPU efficiency of the newly instantiated resources has been tested, along with the close to last step of the analysis for the Top Quark mass measurement in the all hadronic channel. In conclusion, the dynamic extension of the nodes in the local Tier-3 cluster turns out to be perfectly working, and the CMS-Bologna Tier-3 Farm using LSF as a local batch system can be efficiently and dynamically expanded on external Cloud-aware resources without a serious loss of performances. The allocated resources have demonstrated to be reliable as much as the standard Grid resources. In particular, no failures in the job

execution have occurred, leading to performances comparable with the ones recorded in the standard Grid environment. Hence, the resources dynamically allocated can be used for the normal operation of the site. On the other hand, the Cloud OpenStack-Havana infrastructure turns out to be the natural extension to acquire external opportunistic resources, although similar results can be obtained remaining within the same farm in order to acquire internal opportunistic resources.

Moreover, the work of this thesis has explored the direct access and integration of CNAF Cloud resources to the CMS Workload Management system, leading to the suitable use case of HEP "Computing-as-a-Service". This approach, already in use at the CMS Tier-0, has been thus expanded to the case of a generic CMS site and has been tested for the first time with the new CMS Workload Management tools. Differently from the implemented HLT use case, proper security procedures (*gLExec*) has been used to handle the user on the execution host.

The work of this thesis has also provided the very first submission via CRAB3 towards a Cloud infrastructure, resulting the first CMS-wide test. In particular, this was performed connecting the CERN Integration Test Bed (ITB) GlideIn-WMS to the EC2 interface of the CNAF Cloud. The background setup of Test Bed has been performed solving several issues an tuning specific configuration. The debugging has been the critical part of the whole work, and several misconfigurations of the GlideIn have been successfully corrected and recorded to the attention of the ITB Group.

To conclude, the Cloud environment have allowed to perform standard physics task without significant loss of performance in comparison to the Grid environment and no failures have been registered due to the infrastructure.

A natural prosecution of this work could be to apply the discussed approaches at larger scales profiting the CNAF Tier-1 infrastructure in Bologna. In particular, the latter could provide a more performance access to the storage.

Moreover, commercial resources could provide a wider testbed for the work of this thesis, leading to the possibility to export on them the discussed approach.

Bibliography

[1] *LHC Design Report*, CERN 2004-003

[2] *CERN*, http://home.web.cern.ch

[3] *The Large Electron-Positron Collider*, http://home.web.cern.ch/about/accelerators/large-electron-positron-collider

[4] S. Beole et al., *ALICE technical design report: Detector for high momentum PID*, CERN-LHCC-98-19

[5] The ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST 3 (2008) S0B003

[6] The CMS Collaboration, *The CMS experiment at the CERN LHC*, JINST 3 (2008) S08004

[7] The LHCb Collaboration, *LHCb technical design report: Reoptimized detector design and performance*, CERN-LHCC-2003-030 (2003)

[8] *TEVATRON* http://www.fnal.gov/pub/tevatron/tevatron-accelerator.html

[9] Higgs P. W., *Phys. Lett.*, 12 (1964) 132

[10] ATLAS Collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, Phys. Lett. B, 716 (2012) 1-29

[11] CMS Collaboration, *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC*, Phys. Lett. B, 716 (2012) 30-61

[12] The CMS Collaboration, *The CMS experiment at the CERN LHC*, JINST 3 (2008) S08004

[13] https://cms.web.cern.ch/org/list-cms-institutes

[14] The CMS Collaboration, *The CMS Physics Technical Design Report, Volume I: Detector Performance and Software*, CERN/LHCC 2006/001, CMS Technical Design Report 8.1, CERN (2006)

[15] The CMS Collaboration, *The CMS Physics Technical Design Report, Volume II: Physics Performance*, CERN/LHCC 2006/021, CMS Technical Design Report 8.2, CERN (2006)

[16] The CMS Collaboration, *The CMS tracker system project: Technical Design Report*, CERN/LHCC 1998/006, CMS Technical Design Report 5, CERN (1998); CMS Technical Design Report 5, Addendum CERN/LHCC 2000/016 (2000)

[17] The CMS Collaboration, *The CMS electromagnetic calorimeter project: Technical Design Report*, CERN/LHCC 1997/033, CMS Technical Design Report 4 (1997)

[18] The CMS Collaboration, *The CMS hadron calorimeter project: Technical Design Report*, CERN/LHCC 1997/031, CMS Technical Design Report 2 (1997)

[19] The CMS Collaboration, *The Magnet Projet Technical Design Report*, CERN/LHCC 97/10, CMS Technical Design Report 1, CERN (1997)

[20] The CMS Collaboration, *The CMS muon project: Technical Design Report*, CERN/LHCC 1997/032, CMS Technical Design Report 3 (1997)

[21] The CMS Collaboration, *The TriDAS Project Technical Design Report, Volume 1: The Trigger Systems*, CERN/LHCC 2000/38, CMS Technical Report 6.1 (2000)

[22] The CMS Collaboration, *The TriDAS Project Technical Design Report, Volume 2: Data Acquisition and High-Level Trigger*, CERN/LHCC 2002/26, CMS Technical Report 6.2 (2002)

[23] CMS Collaboration, *Constraints on the Higgs boson width from off-shell production and decay to Z-boson pairs*, Phys. Lett. B 736 (2014) 64

[24] The CMS Collaboration, *CMS Technical Design Report for the Pixel Detector Upgrade*, CMS-TDR-11 (2012)

[25] The CMS Collaboration, *CMS Technical Design Report for the Level-1 Trigger Upgrade*, CMS-TDR-12 (2012)

[26] The CMS Collaboration, *CMS Technical Design Report for the Phase 1 Upgrade of the Hadron Calorimeter*, CMS-TDR-010 (2012)

[27] The CMS Collaboration, *Technical Proposal for the upgrade of the CMS Detector though 2010*, CMS UG-TP-1 (2011)

[28] J. D. Shiers, *The Worldwide LHC Computing Grid (worldwide LCG)*, Computer Physics Communications 177 (2007) 219-223

[29] *WLCG*, http://lcg.web.cern.ch/lcg/

[30] *European Grid Infrastructure*, http://www.egi.eu/

[31] *Open Science Grid*, http://www.opensciencegrid.org

[32] I. Foster, C. Kesselman, S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, Intl. J. Supercomputer Applications (2001)

[33] *Virtual Organization Membership Service*, http://toolkit.globus.org/grid_software/security/voms.php

[34] *xrootd* http://xrootd.org

[35] *xrootd* https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookXrootdService

[36] D. Bonacorsi, *WLCG Service Challenges and Tiered architecture in the LHC era*, IFAE, Pavia, April 2006

[37] The CMS Collaboration, *The CMS Computing Model*, CERN LHCC 2004-035

[38] The CMS Collaboration, *The CMS Computing Project Technical Design Report*, CERN-LHCC-2005-023

[39] I. Bird et al., *LHC Computing Grid - Technical Design Report*, CERN/LHCC 2005-024 (2005)

[40] MONARC Members, *Models of Networked Analysis at Regional Centres for LHC Experiments (MONARC)*, PHASE 2 REPORT, CERN/LCB 2000-001 (2000)

[41] G. Bauer et al., *The data-acquisition system of the CMS experiment at the LHC*, Journal of Physics, Conference Series 331 02202 (2011)

[42] *LHCOPN* http://lhcopn.web.cern.ch/lhcopn/

[43] O. Buchmueller et al., *The CMS CERN Analysis Facility (CAF)*, J. Phys.: Conf. Ser. 219 052022 (2010)

[44] M. Giffels et al., *The CMS Data Management System*, Journal of Physics, Conference Series 513.4 (2014)

[45] Cinquilli et al., *The CMS Workload Management System*, Journal of Physics, Conference Series 396.3 (2012)

[46] *PhEDEx* http://cms-project-phedex.web. cern.ch/cms-project-phedex

[47] *ORACLE* http://www.oracle.com/

[48] R. Egeland et al., *The PhEDEx next-gen website*, Journal of Physics, Conference Series 396.3 (2012)

[49] R. Egeland, C.H. Huang, T. Wildish, *PhEDEx Data Service*, Journal of Physics, Conference Series 219.6 (2010)

[50] D. D. Corkill, *Collaborating Software: Blackboard and Multi-Agent Systems and the Future*, Proceedings of the International Lisp Conference, New York, New York (2003)

[51] *The European DataGrid*, http: //eu-datagrid.web.cern.ch/eu-datagrid/

[52] *Berkeley Database Information Index*, https://twiki.cern.ch/twiki/bin/view/EGEE/BDII

[53] *HTCondor*, http://research.cs.wisc.edu/htcondor/

[54] D. Groep, O. Koeroo, G. Venekamp, *gLExec: gluing grid computing to the Unix world*, J. Phys.: Conf. Ser. 119 062032 (2008)

[55] S. Foulkes, J. Linacre, V. Spinoso, A. Lahiff, G. Gomez-Ceballos, M. Klute, A. Mohopatra, E. Fajardo, O. Gutsche, *A new era for central processing and production in CMS* (2012)

[56] S. Foulkes, D. Hufnagel, M. Mascheroni, M. Norman, Z. Maxa, A. Melo, S. Metson, H. Riahi, S. Ryu, D. Spiga, E. Vaandering, S. Wakefield, R. Wilkinson, M. Cinquilli, D. Evans, *The CMS workload management system* (2012)

[57] *The CMS Dashboard* http://dashboard.cern.ch/cms/index.html/

[58] Giuseppe Codispoti et al., *CRAB: A CMS Application for Distributed Analysis*, Nuclear Science Symposium Conference Record N02.79 (2008)

[59] *Software Guide on CRAB* https://twiki.cern.ch/twiki/bin/view/ CMSPublic/SWGuideCrab

[60] Alessandra Fanfani et al., *Distributed Analysis in CMS*, CMS-NOTE- 2009-013, CERN-CMS-NOTE-2009-013 (2009)

[61] Ramn Medrano Llamas et al., *Commissioning the CERN IT Agile Infrastructure with experiment workloads*, J. Phys.: Conf. Ser. 513 032066 (2014)

[62] *Amazon Elastic Compute Cloud*, http://aws.amazon.com/ec2/

[63] *OpenStack*, http://www.openstack.org/

[64] *OZ*, https://github.com/clalancette/oz/wiki

[65] *CVMFS*, http://cernvm.cern.ch/portal/filesystem

[66] S. Zhou, J. Wang, X. Zheng, P. Delisle, *Utopia: A load sharing facility for large, heterogeneous distributed computing systems*, Technical Report CSRI-257, Computer Systems Research Institute, University of Toronto (1992)

[67] D. Futyan, R. Mankel, C. Paus, *The CMS computing, software and analysis challenge*, J. Phys.: Conf. Ser. 219 032008 (2010)

[68] P. Buncic, C. Aguado Sanchez, J. Blomer, L. Franco, A. Harutyunian, P. Mato, Y. Yao, *CernVM - a virtual software appliance for LHC applications*, J. Phys.: Conf. Ser. 219 042003 (2010)

[69] F. Schmuck, R. Haskin, *GPFS: A Shared-Disk File System for Large Computing Clusters*, Proceedings of the FAST 2002 Conference on File and Storage Technologies, Monterey, California, USA (2002)

[70] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Installation_Guide/ch-kickstart2.html

[71] https://github.com/clalancette/oz/wiki/Oz-template-description-language