Journal of Physics: Conference Series **219** (2010) 072036 doi:10.1088/1742-6596/219/7/072036

Dynamic Virtual AliEn Grid Sites on Nimbus with CernVM

A. Harutyunyan[1, 2], P. Buncic[3], T. Freeman[4], K. Keahey[4]

[1]Armenian e-Science Foundation, Yerevan, Armenia
[2]Yerevan Physics Institute after A.I. Alikhanyan, Yerevan, Armenia
[3]CERN, Geneva, Switzerland
[4]University of Chicago, Chicago IL, USA

E-mail: hartem@mail.yerphi.am, Predrag.Buncic@cern.ch, tfreeman@mcs.anl.gov, keahey@mcs.anl.gov

Abstract. We describe the work on enabling one click deployment of Grid sites of AliEn Grid framework on the Nimbus 'science cloud' at the University of Chicago. The integration of computing resources of the cloud with the resource pool of AliEn Grid is achieved by leveraging two mechanisms: the Nimbus Context Broker developed at Argonne National Laboratory and the University of Chicago, and CernVM - a baseline virtual software appliance for LHC experiments developed at CERN. Two approaches of dynamic virtual AliEn Grid site deployment are presented.

1. Introduction

Infrastructure-as-a-Service (IaaS) providers allow users to easily acquire on-demand computing and storage resources. They provide each user with an isolated environment in the form of Virtual Machines which can be used to run services and deploy applications. This approach, also known as 'cloud computing', has proved to be viable for a variety of commercial applications. Currently there are many IaaS providers on the market. The biggest of them is Amazon with its 'Amazon Elastic Computing Cloud (Amazon EC2)' service [1].

The question arises whether we can dynamically provide cloud resources and elastically integrate them into the pool of resources available to CERN ALICE experiment [2] to satisfy the time-varying needs of scientists. Furthermore, can we do it in such a way that no change is visible to the user, i.e. the users do not need to change the ways in which they use the system?

In this contribution we show how cloud computing resources can be used within the AliEn Grid framework [3, 4], developed by CERN ALICE experiment, for performing simulation, reconstruction and analysis of physics data. We deploy baseline virtual software appliance for the LHC experiments developed by the CernVM project [5] on the resources of Science Clouds that use the Nimbus project [6, 7] to enable deployment of virtual machines (VMs) on remote resources. We further also use Nimbus tools for one click deployment of dynamically configurable AliEn Grid site on the Science Cloud of the University of Chicago.

An introduction to AliEn, Nimbus and CernVM is given in Chapter 2. Chapter 3 describes two approaches of virtual site deployment and in Chapter 4 we present timing measurements of dynamic site deployment. Chapter 5 provides a short summary.

2. AliEn, Nimbus and CernVM

2.1. AliEn

AliEn is a lightweight framework, developed by the ALICE experiment, to handle execution of jobs performing simulation, reconstruction and analysis of physics data. It operates like a global queue system which schedules jobs onto numerous distributed sites.

The framework is built from open source components using a combination of Web Services and distributed agents that communicate with each other using the SOAP protocol [8]. AliEn services can be divided into 2 categories: central services and site services. Central services perform Virtual Organization (VO) wide tasks such as user authentication (Proxy Service), job management (Job Manager Service), job scheduling (Job Broker Service), etc. Site services provide interfaces to computing and storage resources on the sites as well as perform auxiliary tasks (e.g. monitoring of those resources). Examples of site services are Computing Element (CE) and application package manager (PacMan). Site services must be deployed on all sites serving the needs of the ALICE experiment, whereas central services must be deployed only in one place. For the detailed description of AliEn framework please refer to [3] and [4].

2.2. Nimbus

The Nimbus project provides an open source, extensible IaaS implementation supporting Web Service Resource Framework (WSRF) [9] as well as the Amazon EC2 interfaces (the Virtual Workspace Service (VWS)) as well as end-user services that make cloud computing easy-to-use for the end user. An example of the latter is the Context Broker service [7] that enables secure establishment of security and configuration context over hosts spanning a distributed environment. The project mainly targets scientific community and offers a platform for experimentation with features for scientific needs. Detailed information on Nimbus project can be found at [6] and [7]

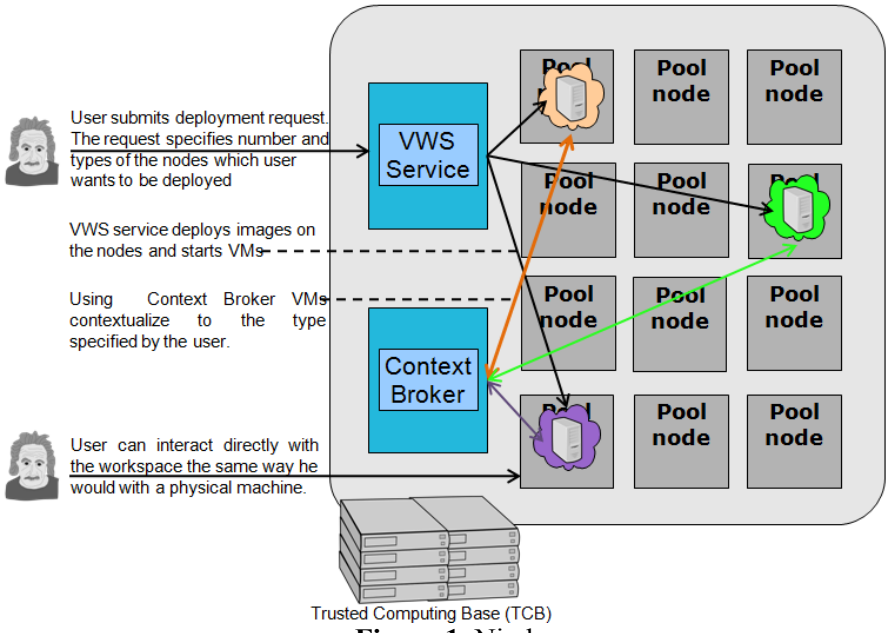

Figure 1. Nimbus.

2.3. CernVM

The CernVM Virtual Software Appliance is a thin Virtual Machine that contains just enough of Operating System to run the application frameworks of the four LHC experiments. Its Operating

System, based on rPath Linux2, fits into a compressed file smaller than 100 MB. The experiment software stack is brought into appliance by means of a file system specifically designed for an efficient and 'just in time' software distribution. In this model, the client downloads only necessary binaries and libraries as they get referenced for the first time. By doing that, the amount of software that has to be downloaded in order to run the typical experiment tasks in the Virtual Machine is reduced by an order of magnitude. For detailed information about CernVM please refer to [5].

3. Virtual site operation

To integrate computing resources provided by Nimbus into ALICE VO we have implemented two scenarios. In the first scenario (that we will call the 'classic' scenario), we dynamically deploy an AliEn "virtual site": we first deploy VMs supporting site services, then, start Worker Nodes (WNs) on the same cloud and, using the site services, get jobs for execution from central AliEn task queue (detailed information about 'classic' scenario is given in Section 3.1). The second approach uses CernVM Co-Pilot, which does not require deployment of site services on the cloud (detailed information about CernVM Co-Pilot is given in Section 3.2). In both scenarios we use images provided by CernVM project.

The question arises, about what happens to the job, if the WN on which it was running, gets terminated. AliEn has the mechanism for detection of such cases. There is a central service which checks jobs' status update times and in case there is job, whose last status update was performed more than 12000 seconds ago that job is marked in the database as failed. If during that job submission user who submitted it has requested automatic resubmission than the failed job will be automatically scheduled for execution again.

3.1. Deployment of a 'classic' AliEn site

To deploy a virtual site on a cloud we need to deploy following services:

- ClusterMonitor - routes messages from AliEn site services to central services. All site services communicate to central services and the configuration database through the site Cluster Monitor. (runs on services node)
- PackMan (Package Manager) – installs application packages (e.g. ROOT, AliRoot) required by the jobs on the site (runs on services node)
- MonaLisa – monitors site services (runs on services node)
- JobAgent (JA) – delivers the input of the data required by the job, starts the job and registers the output of the job in the AliEn file catalogue. (runs on WNs)

There is no need to run a CE service, which serves as an interface to the local batch system and is typically used to start the JA on the WNs. In our setup JAs are automatically started on the WNs whenever WNs are booted. We do not deploy Storage Element (SE), a service for accessing local mass storage systems, because the whole site running on a cloud is supposed to be started and stopped very often, and thus the data which potentially could be kept in the SE would not be available most of the time. That is why the data produced on the virtual site is kept on the SEs of other sites. For the detailed description of AliEn services see [4].

The step by step site deployment procedure is as follows:

1. Deploy the workspaces (1 services node, which will run ClusterMonitor, PackMan and MonaLisa and required number of WNs, which will run JA) using the image provided by CernVM.
2. Transfer credentials (X.509 certificate + private key) to all newly created workspaces
3. Transfer site configuration information (e.g. name of the SE where the data produced by the site must be kept) to the nodes.

4. Configure services node (e.g., NFS server, which must export the directory where the PackMan service will install the application software).
5. Configure WNs (e.g., add to configuration files the address of the services node)
6. Start the ClusterMonitor, MonaLisa and PackMan services
7. Start JA service on WNs

Using the Nimbus Context Broker [7] allowed us to perform all these steps (and thus deploy an AliEn site) with a single command. The command gets as an input a cluster description file and the VM image which must be used to start each of the nodes. Cluster description file is written in XML and specifies number and type ('service' or 'WN') of nodes as well as site configuration parameters.

When the site deployment command is executed Nimbus launches the VMs and 'contextualizes' them to the type specified in cluster description file. Contextualization is done in the following way: Upon launching VMs a lightweight agent running on each VM contacts Nimbus Context Broker and gets the contextualization information. This information is passed to the set of scripts (so-called contextualization scripts), which are launched to configure appropriate services inside VMs and start them.

Once the deployment and contextualization are completed, JAs contacts the Job Broker central service which fetches jobs from the ALICE task queue and sends them to JAs for execution.

3.2. Deployment of AliEn site using CernVM Co-Pilot

In this case AliEn site services are not deployed on a cloud. Instead we take the approach of deploying just the worker node VMs on the cloud. To implement that we have created CernVM Co-Pilot Agent and CernVM Co-Pilot Adapter services. Co-Pilot Agent runs on the WNs and uses Jabber/XMPP instant messaging protocol [10] to communicate with Co-Pilot Adapter, which runs outside the cloud and provides the functionality previously provided by AliEn site services. Co-Pilot Agent and Adapter were developed specifically to ease deployment of Grid sites on computing clouds. The use of Jabber/XMPP allows scaling of the system in case of a high load on the Adapter node by just adding new Adapter node(s) to the existing messaging network.

The steps for site deployment are following:

1. Deploy workspaces using an image provided by CernVM
2. Transfer credentials (agent username and password) to the newly created workspaces
3. Start Co-Pilot Agent service on the workspaces

Like in case of a 'classic' site, here we also use Nimbus Context Broker to deploy AliEn site with a single command. The command gets as an input cluster description file and the VM image which must be used to start the nodes. Cluster description file is written in XML and specifies number of nodes which need to be started. The Agent is preconfigured with the Jabber server hostname and the Jabber ID of Co-Pilot Adapter, which are the only configuration parameters it needs. Co-Pilot Agent then contacts Co-Pilot Adapter service and requests a job to execute. Upon receiving job request from an Agent, Adapter contacts AliEn Job Broker central service which fetches jobs from ALICE task queue and sends them to Adapter, which in turn forwards jobs to Agents. When the job is done, Agent reports results to Adapter which in turn forwards them to AliEn central services. Fig. 3 represents CernVM Co-Pilot setup.

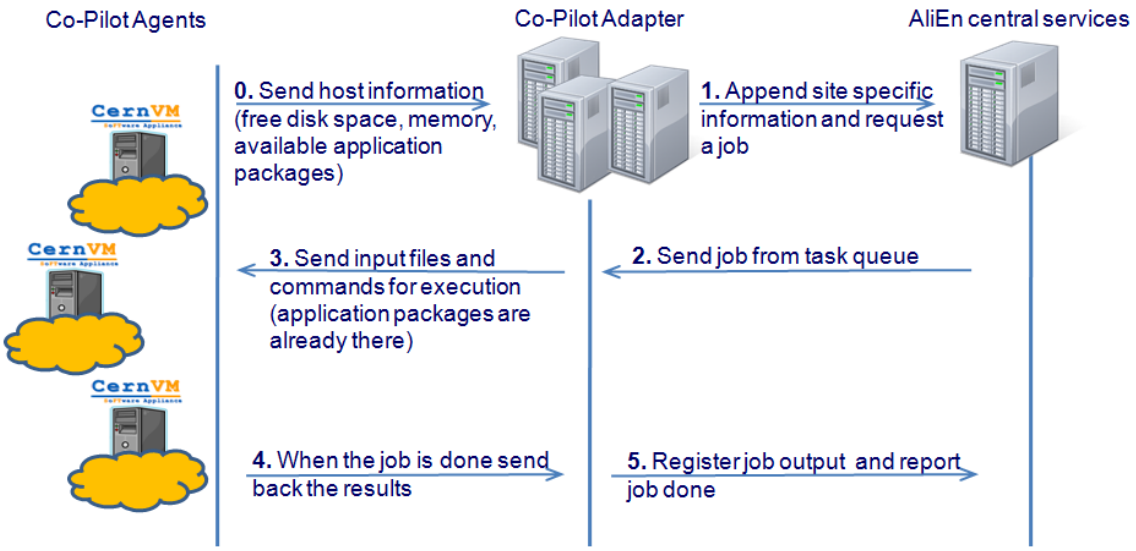

Figure 3. CernVM Co-Pilot.

3.3. Comparison of these two approaches

The 'classic' approach is very straightforward and does not require any new development neither in AliEn nor in Nimbus. However, it has some drawbacks. First thing is that we need a separate workspace (and in case of high load few of them) to run site services. Their deployment is time consuming and if we could exclude them from the setup we would potentially have more workspaces for running WNs. Another thing is that in this case the application software is brought to the cloud by PackMan service and is made available to WNs through NFS server. This is not optimal, because CernVM image already provides the application software and it would be cleverer and faster if WNs could use it (that would also allow elimination of PackMan service from the deployment), but that would require modification of AliEn JobAgent service code. The 'classic' approach can be used to easily integrate cloud resources into other Grid frameworks.

The Co-Pilot approach does not require deployment of service nodes and thus potentially allows running more jobs on the same number of virtual workspaces (since some of them will be used as worker nodes rather than service nodes). Besides it uses application software available from CernVM and does not require any package management from Grid middleware whatsoever. Currently Co-Pilot Adapter can fetch jobs only from AliEn. However it can be extended to communicate with any pilot job framework, e.g. PanDA - distributed production and analysis system [11] used by CERN ATLAS [12] experiment, or Dirac [13] – Grid solution used by CERN LHCb experiment [14]. Co-Pilot agent does not have anything AliEn-specific and is written in a way, that running jobs fetched from other frameworks should not require extra development.

4. Virtual site deployment timing measurements

For both types of dynamic sites we have measured the time which elapses between a) launching site deployment command and start-up of node(s) on the cloud, b) launching site deployment command and requesting of job(s) by WNs and c) requesting job(s) by the WNs and assignment of jobs to them by AliEn. To perform the measurements we have deployed sites with different number of worker nodes: from 1 to 10. For each number of WNs, 3 deployments have been performed, so 30 deployments were done for 'classic' site (launching 165 WNs overall) and 30 deployments for Co-Pilot site (again, launching 165 WNs overall). For each deployment timings have been measured, and afterwards mean times have been calculated. The deployment commands have been launched from a machine located at CERN to the Nimbus Science Cloud of the University of Chicago.

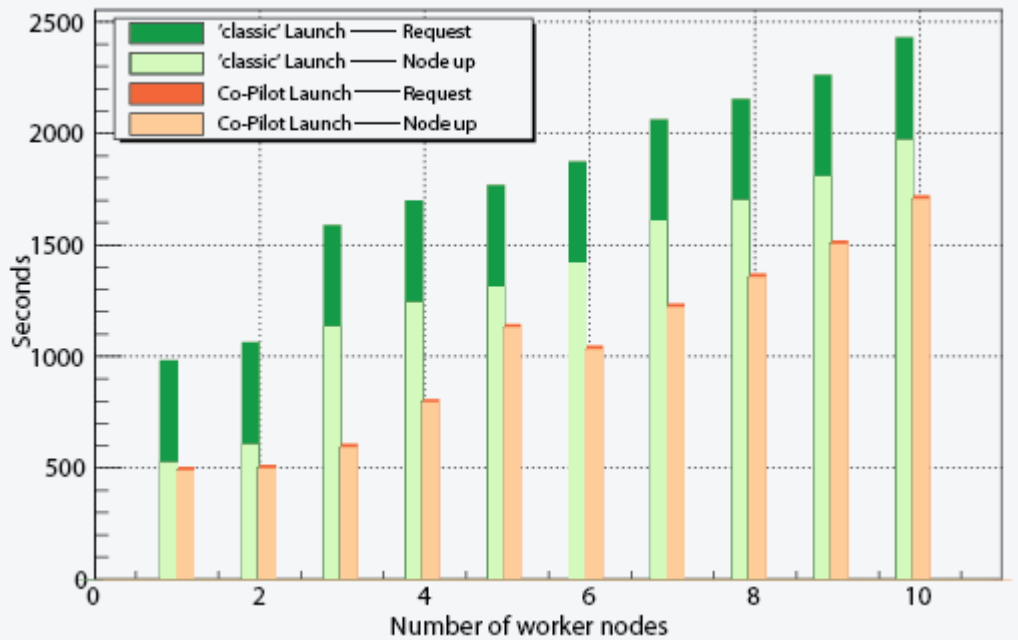

Figure 4. Virtual site deployment timings

The plot on Fig. 4 represents mean times which elapse between:

- launching of site deployment command and start-up of node(s) on the cloud (light green bars for 'classic' sites and light orange bars for Co-Pilot site)
- launching of site deployment command and requesting of job(s) by WNs (dark green bars for 'classic' sites and dark orange bars for Co-Pilot site)

It is seen from the plot that the time of worker node deployment on the cloud is in general proportional to the number of WNs being launched: this is because more nodes produce higher load on Nimbus services.

It is also seen, that for the same number of WNs node start-up duration is longer in case of 'classic' site: this is explained by the fact that in case of 'classic' site in addition to WNs we launch also a separate node for running AliEn site services.

The time span between worker node start-up and job request does not depend on the number of worker nodes and varies slightly. It is about 400 seconds in case of 'classic' site and about 15 seconds in case of Co-Pilot site. This time is spent to start and initialise site services on the services node and the JA service on WNs in first case, and to start and initialise Co-Pilot agent in second case.

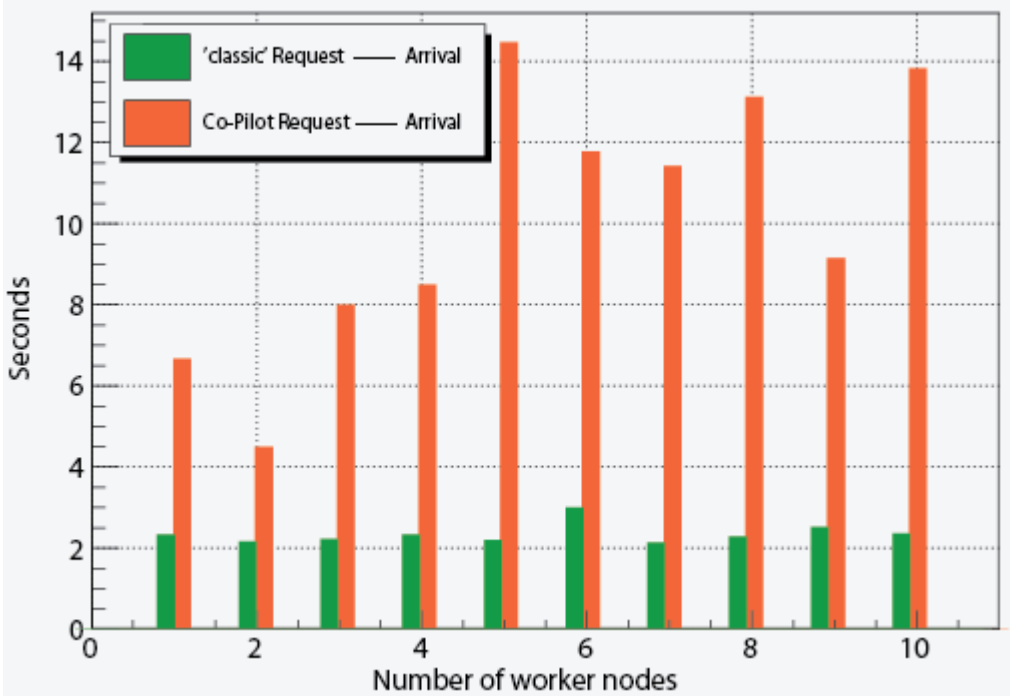

Figure 5. Job request and arrival timings

The plot on Fig. 5 shows mean times which elapse between requesting of job(s) by the WNs and assignment of jobs to them by AliEn (dark green bars for 'classic' sites and dark orange bars for Co-Pilot site).

In case of 'classic' site the mean time does not exceed 5 seconds, and in case of Co-Pilot site it varies from 3 to 30 seconds. The reason for this is that Co-Pilot Adapter is currently serving requests sequentially, whereas AliEn services are processing several requests simultaneously.

The table below shows minimums, maximums, means and standard deviations of measured time intervals for all 165 WNs of 'classic' site (launched during 30 deployments) and 165 WNs of Co-Pilot site (launched during 30 deployments).

	Minimum	Maximum	Mean	Standard deviation
'classic' site				
Launch – Node up	529	2031	1566.915	350
Launch – Job request	982	2489	2010.188	350.558
Job request – Job Arrival	1	5	2.381	0.82
Co-Pilot site				
Launch – Node up	476	1906	1241.679	383.219
Launch – Job request	491	1921	1256.588	383.184
Job request – Job Arrival	3	30	11.327	7.717

Table 1. Min., Max., Avg. and SD of measured variables

Minimum values of 'Launch – Node up' and 'Launch – Job request' were recorded during the deployment of sites with 1 WN and maximums were recorded during the deployment of sites with 10 WNs. The large variations of these intervals have been expected, because the load produced on the

Nimbus VWS naturally grows with the number of workspaces which are required to be launched simultaneously.

5. Summary

Integration of cloud computing resources provided by Nimbus with existing Grid infrastructure of CERN ALICE experiment using the CernVM Virtual Software Appliance has proved to be viable. Two scenarios of dynamic virtual AliEn Grid site deployment on the cloud were elaborated. Both scenarios enable the one click deployment of virtual machines on the Nimbus cloud at the University of Chicago, which then join the AliEn Grid and get jobs to execute.

One of the noteworthy achievements of this work is that the integration of the cloud computing into the existing Grid infrastructure is transparent for the end users and does not change their perception of the system.

6. Acknowledgements

This work has been supported by Google Summer of Code 2008 program. Nimbus project is supported by NSF grants #0721867, #0509466, and, in part, by the DOE SciDAC-funded CEDPS project. The work of A. Harutyunyan was also partially supported by Swiss Fonds "Kidagan" and Calouste Gulbenkyan Foundation. We are very thankful to Carlos Aguado-Sanchez for CernVM related support.

6. References

[1] Amazon Elastic Compute Cloud (Amazon EC2) http://aws.amazon.com/ec2/
[2] ALICE Collaboration, Technical Proposal, CERN/LHCC 95-71
[3] P. Buncic, A.J. Peters, and P. Saiz, "The AliEn system, status and perspectives" ECONF C0303241 (2003) MOAT004
[4] P. Buncic, A.J. Peters, P. Saiz, and J.F. Grosse-Oetringhaus, "The architecture of the AliEn system", CHEP 2004, Interlaken, Switzerland (2004) 440
[5] P. Buncic et al., CernVM - a virtual appliance for LHC applications, Proceedings of Science, PoS(ACAT08)012, 2009
[6] Keahey, K., I. Foster, T. Freeman, and X. Zhang, Virtual Workspaces: Achieving Quality of Service and Quality of Life in the Grid. accepted for publication in the Scientific Progamming Journal, 2005
[7] K.Keahey and T.Freeman. Contextualization: Providing one-click virtual clusters. In eScience 2008, 2008.
[8] WSRF – Web Service Resource Framework http://www.globus.org/wsrf/
[9] SOAP – Simple Object Access Protocol http://www.w3.org/TR/soap/
[10] XMPP – Extensible Messaging and Presence Protocol http://www.xmpp.org
[11] T. Maeno, PanDA: distributed production and distributed analysis system for ATLAS, J. Phys. Conf. Ser. 119 (2008) 062036.
[12] G. Aad et al. [ATLAS Collaboration], The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 (2008) S08003.
[13] J. P. Baud, P. Charpentier, J. Closier, R. Graciani, A. Maier, and P. Mato-Vila. DIRAC Review Report. Technical Report LHCb-2006-04 COMP, CERN, 2006.
[14] LHCb Reoptimized Detector Technical Design Report, LHCb Collaboration, LHCb TDR 9, CERN LHCC-2003-030