CHEP06 - CMS internal refereeing process

Abstracts book

Table of contents

Abstract ID : 14

Managing Workflows with ShReek

The Shahkar Runtime Execution Environment Kit (ShREEK) is a threaded
 workflow execution tool designed to run and intelligently manage
 arbitrary task workflows within a batch job. The Kit consists of
 three main components, an executor that runs tasks, a control point
 system to allow reordering of the workflow during execution and a
 thread based pluggable monitoring framework that offers both event
 driven and periodic monitoring. Developed specifically to address the
 challenges of running High Energy Physics processing jobs in complex
 workflow arrangements, with highly varied monitoring needs, the
 ShREEK toolkit is in use at multiple HEP experiments, and can be
 adapted for a variety of other uses such as wrapping batch jobs to
 provide detailed interactive monitoring for administrators and users
 alike. In this presentation we will discuss the architecture of the ShReek system
and the experience using it
in several experiment workflows.

Primary authors : Dr. EVANS, David (FERMILAB) ; AFAQ, Anzar (FERMILAB)

Co-authors :

Presenter : Dr. EVANS, David (FERMILAB)

Track classification : Distributed Data Analysis

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Saturday 15 October 2005

Last modified on : Saturday 15 October 2005

Comments :

Abstract ID : 17

Configuring the CMS Software on OSG Sites

The configuration of complex HEP software applications with external dependencies across diverse grid sites
is difficult. In this paper we present development work on CMSSoftDB, which has been used to configure the
CMS software distribute on Open Science Grid sites. CMSSoftDB is a lightweight, python and XML based tool
that provides a uniform interface for dealing with CMS Application Software across different sites. An XML
based Database structure is used to track software installed on a site including site and version specific
details, as well as different methods of installation. This information is then used to generate site and version
specific runtime environments while exposing the same interface to users. The CMSSoftDB tool uses python to
manage and interact with the software metadata published to it. The tool requires only standard python
libraries and a small disk area to be installed in. It is a site-local command line or python API tool. CMSSoftDB
allows the setup for any executable at a given site to be expressed in terms of shell variables and commands
for a specific shell, including any scram details if required, and stores them in an XML DB. The user command
line API then allows the user access to that tool without having to know specifics of the site or software
version to use it. CMSSoftDB was originally developed to provide a uniform interface to CMS Software
installations for OSG sites, by allowing site and version specific information to be held in a site-local instance
of CMSSoftDB, which is made available to users in a common location. Generic OSG jobs can then go to any
site and set up the software needed in a uniform manner.

Primary authors : Dr. EVANS, David (FERMILAB) ; Mr. KIM, Bockjoo (University of Florida) ; Dr. HOLZMAN, Burt (FERMILAB) ; Mr. THOMAS, Michael (Caltech)

Co-authors :

Presenter : Dr. EVANS, David (FERMILAB)

Track classification : Software Tools and Information Systems

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Sunday 16 October 2005

Last modified on : Monday 17 October 2005

Comments :

Abstract ID : 18

On Demand, Policy Based Monte Carlo Production and Tracking, Leveraging Clarens, MonALISA and RunJob

We describe a set of Web Services, created to support scientists in performing distributed production tasks
(e.g. Monte Carlo). The Web Services described in this paper provide a portal for scientists to execute different
production workflows which can consist of many consecutive steps. The main design goal of the Web Services
discussed is to provide controlled access for (multiple) set(s) of users in different roles (e.g. scientists,
administrators, grid operators,...) to complex production workflows without the added trouble of updating,
configuring, and patching these ever evolving applications and keep the users focused on their core tasks
(running production), while experts at the tier2 centers keep the software up to date. Once users execute a
workflow they recieve a tracking number that is used to track the job status which is propagated through
MonALISA. Job anomalies can be further investigated using the JobMon service. The Web Services have been
implemented inside the Clarens Web Service framework. This Python (and Java) based framework provides,
amongst others, x509 authorization, access control and VO management for its services. The Web Services
discussed in this paper re-use several of these Clarens components in providing access control and usage
quotas. Initially the services described in this paper where developed to support users in Monte Carlo
production activities, however due to their generic design, can be used to expose other (potentially complex)
applications to users as will be shown in this paper.

Primary authors : Dr. BUNN, Julian (Caltech) ; Dr. EVANS, David (FERMILAB) ; HSU, Shih-chieh (University of California, San Diego) ; Dr. LEGRANS, Iosif (Caltech) ; Dr. VAN LINGEN, Frank (Caltech) ; Dr. LIPELES, Elliot (University of California, San Diego) ; Mr. MARTIN, Terrence (University of California, San Diego) ; Prof. NEWMAN, Harvey (Caltech) ; Dr. RANA, Abhishek (University of California, San Diego) ; Dr. STEENBERG, Conrad (Caltech) ; Dr. THOMAS, Michael (Caltech) ; Prof. WUERTHWEIN, Frank (University of California, San Diego)

Co-authors :

Presenter : Dr. BUNN, Julian (Caltech)

Track classification : Event Processing Applications

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Sunday 16 October 2005

Last modified on : Sunday 16 October 2005

Comments :

Abstract ID : 33

Development of the TIer-1 Facility at Fermilab

CMS is preparing seven remote Tier-1 computing facilities to archive and serve experiment data. These centers represent the bulk of CMS's data serving capacity, a significant resource for reprocessing data, all of the simulation archiving capacity, and operational support for Tier-2 centers and analysis facilities. In this paper we present the progress on deploying the largest remote Tier-1 facility for CMS, located at Fermilab. We will present the development, procurement and operations experiences during the final two years of preparation. We will discuss the development and deployment to support grid interfaces for the Worldwide LHC Computing Grid and the Open Science Grid on the same physical resources. We will outline the hardware selection and procurement and plans for the future to meet the needs of the experiment and the constraints of the physical facility. We will also discuss the successes and challenges associated with enabling a mass storage system to meet the various experimental needs at a significant increase in scale over what is currently achievable. Finally we will discuss the model to support US Tier-2 centers from the Tier-1 facility.

Primary authors : Dr. FISK, Ian (FERMILAB) ; Dr. BAUERDICK, Lothar (FERMILAB) ; Dr. BAKKEN, Jon (FERMILAB) ; Mr. FAGAN, David (FERMILAB) ; Mrs. GIACCHETTI, Lisa (FERMILAB) ; Mr. KAISER, Joseph (FERMILAB) ; Mr. MESSER, Tim (FERMILAB) ; Mr. STIEHR, Gary (FERMILAB) ; Dr. WENZEL, Hans (FERMILAB) ; Mr. WU, Yujun (FERMILAB)

Co-authors :

Presenter : Dr. FISK, Ian (FERMILAB)

Track classification : Computing Facilities and Networking

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Monday 17 October 2005

Last modified on : Monday 17 October 2005

Comments :

Abstract ID : 38

Distributing software applications based on runtime environment

Packaging and distribution of experiment-specific software becomes a complicated task when the number of versions and external dependencies increases. In order to run a single application, it is often enough to create appropriate runtime environment that ensures availability of required shared objects and data files. The idea of distributing software applications based on runtime environment is employed by Distribution After Release (DAR) tool. DAR allows to automatically replicate application's runtime environment based on the reference software installation. Assuming that software is relocatable, applications can be packaged into a completely self-consistent "darball" and executed on any computing node, which is binary compatible with the reference software installation. Such light-weight distribution can be used on opportunistic GRID resources to avoid excessive efforts of complete installation of experiment-specific software. For over three years, DAR tool has been successfully used by CMS for Monte-Carlo mass production, helping physicists to get results earlier. In version 2, DAR was completely redesigned, optimized, and enriched with new features, ready to meet future challenges. The paper presents general concept of the tool and new features available in DAR 2.

Primary authors : Mrs. RATNIKOVA, Natalia (FERMILAB) ; Dr. EVANS, David (FERMILAB) ; Dr. WILDISH, Tony (Princeton) ; DARMENOV, Nickolay (CERN)

Co-authors :

Presenter : Mrs. RATNIKOVA, Natalia (FERMILAB)

Track classification : Software Tools and Information Systems

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Monday 17 October 2005

Last modified on : Monday 17 October 2005

Comments :

Abstract ID : 39

Development of the Monte Carlo Processing Service for CMS

The Monte Carlo Processing Service (MCPS) package is a Python based workflow modelling and job creation package used to realise CMS Software workflows and create executable jobs for different environments ranging from local node operation to wide ranging distributed computing platforms. A component based approach to modelling workflows is taken to allow both executable tasks as well as data handling and management tasks to be included within the workflow. Job Creation is controlled so that regardless of the components used, a common self contained job sandbox and execution structure is produced allowing the job to be run on most batch systems via a submission interface. In this presentation we will discuss the architectural choices made in MCPS, the development status, and experiences deploying to both the European and U.S Grid infrastructure.

Primary authors : Dr. EVANS, David (FERMILAB) ; Mr. AFAQ, Anzar (FERMILAB) ; Mr. SEKHRI, Vijay (FERMILAB) ; Dr. VAN LINGEN, Frank (Caltech)

Co-authors :

Presenter : Dr. EVANS, David (FERMILAB)

Track classification : Event Processing Applications

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Monday 17 October 2005

Last modified on : Monday 17 October 2005

Comments :

Abstract ID : 40

Schema Independent Application Server Development Paradigm

The idea of an application database server is not new. It is a key element in multi-tiered architectures and
business application frameworks. We present here a paradigm of developing such an application server in a
complete schema independent way. We introduce a Generic Query Object Layer (QOL) and set of Database/
Query Objects (D/QO) as the key component of the multi-layer Application server alongwith set of tools for
generating such objects. In Query Object Layer each database table is represented as a C++ Object (Database
Object) and structured complex queries spanning multiple tables are written into Object Representations,
calling them Query Objects. All database operations (select/insert etc) are performed via these Objects. In
general, developments of such servers tend to pre-identify interesting join conditions and hardwire queries
for such Query Objects, for the ease of development. We have tried to enhance this concept by generalizing
creation of such Query Objects based on existing/defined relations among the tables involved in the join, like
foreign key relations, and any other user-defined join-condition. Also delaying and generalizing creation of
actual SQL Query till the execution time. This is an enormously complex task, joins with cyclic conditions and
multi-relations going to same table are hard to convert into Query Objects. The task is divided into three
major components. A SQL Parser that reads-in Table definitions and create C++ Objects (Database Objects). A
Query Object View Creator that generates Query Object according to existing and user-defined join conditions
for multiple tables. And Object Layer Algorithms that are generic enough to deal with any Dataset or Query
Object. In addition to this the whole fabric of Application server is tied by exchanging self describing objects
that do not need any changes in case of a schema change. The Business Logic Layer can be quickly built for
know set of operations, writtens as "Managers" and Client interface is done through data structures that can
also be semi-generated through SQL Parser. The process of adapting the system for a new schema is very
fast. The maintenance over head is also very low.

Primary authors : Mr. AFAQ, Anzar (FERMILAB) ; Dr. GRAHAM, Greg (FERMILAB) ; Dr. LUEKING, Lee (FERMILAB) ; Dr. VESELI, Sinisa (FERMILAB) ; Mr. SEKHRI, Vijay (FERMILAB)

Co-authors :

Presenter : Mr. AFAQ, Anzar (FERMILAB)

Track classification : Grid middleware and e-Infrastructure operation

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Monday 17 October 2005

Last modified on : Monday 17 October 2005

Comments :

Abstract ID : 44

Distributed CMS Analysis on the Open Science Grid

The CMS computing model provides reconstruction and access to recorded data of the CMS detector as well as
to Monte Carlo (MC) generated data. Due to the increased complexity, these functionalities will be
provided by a tier structure of globally located computing centers using GRID technologies. In the CMS
baseline, user access to data is provided by the CMS Remote Analysis Builder (CRAB) analysis tool which
enables the user to execute analysis applications on locally resident data using GRID tools independent of the
geographical location. Currently, mostly two different toolkits provide the needed functionalities, the
Worldwide LHC Computing Grid (LCG) and the OpenScience Grid (OSG). Due to infrastruture and service
differences between the two toolkits, analysis tools developed for one are frequently not immediately
compatible with the other.. In this paper, we will describe the development of additions to the CRAB tool to
run user analysis on OSG sites. We will discuss the approach of using the GRID submission of the CONDOR
batch system (CONDOR-G) to provide a sandbox functionality for the user's analysis job. For LCG sites, this is
provided amongst other things by the resource broker. We will discuss the differences of user analysis on LCG
and OSG sites and present first experiences running CMS user jobs at OSG sites.

Primary authors : Dr. GUTSCHE, Oliver (FERMILAB) ; KIM, Bockjoo (University of Florida)

Co-authors :

Presenter : Dr. GUTSCHE, Oliver (FERMILAB)

Track classification : Distributed Data Analysis

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Tuesday 18 October 2005

Last modified on : Tuesday 18 October 2005

Comments :

Abstract ID : 50

Automatic Publishing and Prepatation of CMS Simulated Datasets

Event Data is CMS is viewed as set of Datasets organized in a federation. Individual datasets are described by
the META Data associated with each dataset. The distributed nature of data production (MC Simulation) has
caused description of datasets scattered over several systems. This includes a central database called RefDB
(Reference DB) that holds a considerable portion of META Data required to describe a dataset and sites
parameters that host data. In order to create a usable federation, datasets needs to be described collecting all
scattered descriptions and then published for users to access. More specifically, individual files within
datasets needs to be cataloged and each run need to be described in META Data to correspond to related files.
The whole process of publishing is set of several cumbersome steps. CMSGLIDE is a tool built using the
workflow framework deployed for CMS production that creates the dataset. Dataset Publishing is also an
intermediate step for MC Production chain and the officilal CMS simulation production tool has a specific
dependency on CMSGLIDE. CMSGLIDE is integrated with other tools like the CMS publication and transfer tools
and is widely used within CMS. In the presentation we describe the development and operations of CMSGLIDE
and the integration of the tool into the CMS data management system.

Primary authors : Mr. AFAQ, Anzar (FERMILAB)

Co-authors :

Presenter : Mr. AFAQ, Anzar (FERMILAB)

Track classification : Distributed Event Production and Processing

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Tuesday 18 October 2005

Last modified on : Tuesday 18 October 2005

Comments :