CHEP 06

Abstracts book

http://indico.cern.ch/conferenceDisplay.py?confId=048

Table of contents

Abstract ID : 276

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 : oral presentation

Submitted by : Dr. FISK, Ian

Submitted on Friday 11 November 2005

Last modified on : Friday 11 November 2005

Comments :

Abstract ID : 277

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) ; FAGAN, David (FERMILAB) ; GIACCHETTI, Lisa (FERMILAB) ; KAISER, Joseph (FERMILAB) ; MESSER, Tim (FERMILAB) ; STIEHR, Gary (FERMILAB) ; Dr. WENZEL, Hans (FERMILAB) ; 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 Friday 11 November 2005

Last modified on : Friday 11 November 2005

Comments :

Abstract ID : 278

Development of the Monte Carlo Production 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) ; AFAQ, Anzar (FERMILAB) ; 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 Friday 11 November 2005

Last modified on : Friday 11 November 2005

Comments :

Abstract ID : 279

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 infrastructure 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 : GUTSCHE, Oliver (FERMILAB) ; KIM, Bockjoo (University of Florida)

Co-authors :

Presenter : GUTSCHE, Oliver (FERMILAB)

Track classification : Distributed Data Analysis

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Friday 11 November 2005

Last modified on : Friday 11 November 2005

Comments :

Abstract ID : 280

Benchmarking AMD64 and EMT64

We report on the ongoing evaluation of new 64 Bit processors as they become available to us. We present the
results of benchmarking these systems in various operating modes and also measured the power consumption.
To measure the performance we use HEP and CMS specific applications including:
the analysis tool ROOT (C++), the MonteCarlo generator Pythia (FORTRAN), OSCAR (C++) the GEANT 4 based
CMS detector simulation program and ORCA(C++) the CMS event reconstruction program.
Processors we tested include: single and dual core AMD Opteron AMD64 processors at various clock speeds
Intel Xeon EMT64 processors AMD Athlon AMD64 processors

Primary authors : Dr. WENZEL, Hans (FERMILAB) ; FURUKAWA, Mako (University of Nebraska-Lincoln)

Co-authors :

Presenter : Dr. WENZEL, Hans (FERMILAB)

Track classification : Computing Facilities and Networking

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Friday 11 November 2005

Last modified on : Friday 11 November 2005

Comments :

Abstract ID : 293

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 along with 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, written 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 : AFAQ, Anzar (FERMILAB) ; Dr. GRAHAM, Gregory (FERMILAB) ; Dr. LUEKING, Lee (FERMILAB) ; Dr. VESELI, Sinisa (FERMILAB) ; SEKHRI, Vijay (FERMILAB)

Co-authors :

Presenter : AFAQ, Anzar (FERMILAB)

Track classification : Grid middleware and e-Infrastructure operation

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Friday 11 November 2005

Last modified on : Friday 11 November 2005

Comments :

Abstract ID : 294

Implementing Finer Grained Authorization on the Open Science Grid

Securely authorizing incoming users with appropriate privileges on distributed grid computing resources is a
difficult problem. In this paper we present the work of the Open Science Grid Privilege Project which is a
collaboration of developers from universities and national labs to develop an authorization infrastructure to
provide finer grained authorization consistently to all grid services on a site or domain. The project supports
the utilization of extended proxy certificates generated with identity, group and role information from the
European Data Grid (EDG) Virtual Organization Management System (VOMS). These proxies are parsed at the
grid interface and an authorization request is sent a central Grid User Mapping Service (GUMS). The GUMS
service will return the appropriate mapping based on the identity, role or group. This allows the user to
propagate information about affiliation and activity in the credentials and allows the site to make decisions on
authorization, privilege, and priority based on this information. The Privilege components have been
packaged and deployed on OSG sites. The infrastructure has been used to support sites with multiple
computing elements and storage elements. We will present the motivation and architecture for finer grained
authorization as well as the deployment and operations experience.

Primary authors : CARCASSI, Gabriele (Brookhaven National Laboratory) ; GARZOGLIO, Gabriele (FERMILAB) ; LORCH, Markus (IBM) ; PERELMUTOV, Timur (FERMILAB) ; SINGH RANA, Abhishek (UCSD) ; WEIGAND, John (ProApps) ; WUERTHWEIN, Frank (UCSD) ; SILL, Alan (Texas Tech University)

Co-authors :

Presenter : SINGH RANA, Abhishek (UCSD)

Track classification : Grid middleware and e-Infrastructure operation

Contribution type : --not specified--

Submitted by : Dr. FISK, Ian

Submitted on Friday 11 November 2005

Last modified on : Friday 11 November 2005

Comments :

Abstract ID : 340

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 : RATNIKOVA, Natalia (FERMILAB) ; EVANS, David (FERMILAB) ; WILDISH, Tony (Princeton) ; DARMENOV, Nickolay (CERN)

Co-authors :

Presenter : RATNIKOVA, Natalia (FERMILAB)

Track classification : Software Tools and Information Systems

Contribution type : poster

Submitted by : Dr. FISK, Ian

Submitted on Monday 14 November 2005

Last modified on : Monday 14 November 2005

Comments :