

Table of Contents

Panda - Production and Distributed Analysis.....	1
More Details.....	2
Introduction.....	2
Architecture.....	2
Comments, Issues, Questions.....	2
List of Components.....	3

Panda - Production and Distributed Analysis

More Details

Introduction

We anticipate building a generic infrastructure for execution of many types of jobs and workflows, though the initial focus is on the simplest of ATLAS workflows: a) assume data is prestaged to the site, b) execute the ATLAS job, c) store the data products locally and register the output.

Architecture

The main components of the new system are shown in the main Panda page.

There are a few important principles the new architecture implements:

- Continued coherence with the common ATLAS prodsys software, including the supervisor and common executor components.
- Creation of a task queue, fed by Eowyn and the ATLAS production database, which is used as a highly performant local reservoir of jobs whose local state is tracked and monitored, and whose priorities can be dynamically modified. This task queue would be accompanied by infrastructure for job and resource matchmaking, and implementation of ATLAS policies.
- Creation of a grid pilot system which harvests grid resources on behalf of ATLAS, and communicates with the task queue to schedule ATLAS workloads via http mechanisms. Pilots advertise availability, site (data & software), and resource characteristics (CPU, memory, etc) to the central task queue.
- A separation between **grid** job management and **ATLAS** job management. Multiple ATLAS jobs can be processed by a single grid job, reducing grid latency and overheads.
- Optimization of job slots (wall time) by scheduling shorter jobs when wall time runs low.
- Recognition of multiple authorization and policy insertion points, both at the site and task queue levels.
- Delegation of data movement and cataloging tasks to the ATLAS DDM system, thereby building all fault tolerance and re-try procedures in one set of components, reducing redundancy and work involved for staging data and managing storage.
- Utilization of ATLAS managed storage elements which provide the infrastructure of a datagrid, and data placement tools and services for distributing data according to production and analysis requirements and policies.
- Creation of VO-specific, distributed monitoring framework which instruments components with monitoring sensors and agents which source event state and other logging information to a central collector.
- Provision of interfaces for multiple ATLAS job submission systems beyond prodsys: distributed analysis, user production, and other user analysis modes and workflows not yet designed.
- Provision for use of opportunistic resources via edge services, in later phases of the project.

Comments, Issues, Questions

- System will rely, eventually, on edge services, but initial steps should use by-hand methods with dedicated VO boxes.
- The system depends on a functional DDM system - a multi-site infrastructure and dedicated servers should be deployed and tested outside the system, demonstrating reliable file transfer, failure modes, "user" interfaces, and expose site and server specific issues.
- Site information service - use site local database for quick capture of everything ATLAS needs?
- Software distribution, installation, information and validation services have been missed.
- How do we interact with resource brokers - or use classAds in the Pull model?

List of Components

This is an partial list which is likely to change as the architecture becomes better understood. The development plan is to first create a **lightweight** infrastructure containing prototypes of the main components. Identify the initial set of interfaces and communication protocols, state transitions, assess functionality and expose design and architectural flaws and assumptions. This initial period would be followed by a settling in on the main components and engineering the production scale ingredients, and shaking out all the functionality and scalability problems. The third phase would involve deployment of services and system level testing at production scale (multiple ATLAS sites).

- PandaExecutorInterface - connect with the ATLAS common executor layer and Eowyn
- PandaTaskBuffer - the basic task queue
- PandaJobCreator - user interface for job definitions and task queue loading
- PandaBrokerage - matches job attributes with site and pilot attributes
- PandaJobDispatcher - communicates with pilots at sites
- PandaPolicyModule - defer initially
- PandaDataService - initially the placement service, optimize later
- PandaJobScheduler - the pilot submission and management system (condor g, schedd based, as usual)
- PandaGridMonitoring - this is a system: sensor, agent, collection, repository and user interface
- PandaSiteServices - information service for data, software, and other services

Major updates:

-- KaushikDe - 01 Aug 2005 -- RobertGardner - 10 Aug 2005

Responsible: KaushikDe

This topic: PanDA > PandaMoreDetails

Topic revision: r9 - 2008-01-17 - StephenHaywood



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback