WORK OVERVIEW

Paul Nilsson
Introduction

• Who am I?
  • Physicist working for BNL since 5 years, stationed at CERN, born in Sweden, living in France, married to a Colombian, one child
  • Background: PhD in relativistic heavy ion physics, Lund, Sweden
  • Work history: EMU-01, WA98, PHENIX, ALICE, ROOT, ATLAS
  • LinkedIn: https://www.linkedin.com/in/paulnilsson/
  • Job task: project lead for PanDA Pilot
Current Work

• What does the PanDA Pilot do?
  • Short version: Execute and monitor payload on a resource
  • Not quite as simple as that may sound
    • ~140 grid sites & HPC centers & Harvester & PanDA server & aCT & AGIS information system & DDM & wrappers & proxies & production jobs & user jobs & containers & special payloads & error recognition & event service & remote/direct file access & monitoring & .. = lots of details
    • ~10 developers over the past 5 years (although only ~2 FTE)

• Original PanDA Pilot used by ATLAS and others for well over a decade

• Code has now been rewritten from scratch, adopting a more flexible design -> Pilot 2 project
Pilot 2 Continued

• How does the Pilot fit into the PanDA hierarchy?
  • Runs on the worker nodes on local resources, on grids and clouds, on HPCs and on volunteer computers via BOINC
  • Interacts with the PanDA server either directly, via a local instance of the ARC Control Tower (a job management framework used on Nordugrid) or with the resource-facing Harvester service

• Pilot Code
  • Component based, with each component being responsible for different tasks
    • The main tasks are sorted into controller components, such as Job Control, Payload Control and Data Control
    • Essential features can be accessed via simplified APIs (e.g. Harvester is using Data API for file transfers)
  • “Flexible” code design relies on plug-ins (e.g. “ATLAS”, HPC-resources), multi-threaded, queue-based (job objects passed around in Python Queues)
  • Python 2.7 (slow migration to Python 3 -> Pilot 3 project)
Pilot 2 Continued

• Workflows
  • In the **standard workflow**, the Pilot performs payload download; setup; stage-in; execution; stage-out, along with various verifications, monitoring and server job updates
  • The **HPC Pilot workflow** refers to a dedicated workflow used on HPCs
    • When this is selected the normal workflow of the Pilot is skipped in favour of a streamlined workflow that is relevant for HPCs
    • Resource specific code, such as environmental setup, is kept in plugins
  • The **stage-in workflow** means that Pilot will only stage-in input files and leave for later processing
    • Can e.g. be useful for pre-populating a cache
    • To be done..
  • The **payload + stage-out workflow** can be used with pre-filled caches
    • To be done..
Pilot 2 Status

• Main development stage (i.e. of main features) finished late last year
• Development of additional features (especially new features/requests) continue, bug fixes, adaptation of existing code to an ever changing system..
• Commissioning (replacing Pilot 1 on production and user analysis sites) now in rapid progression