DIRAC Universal Pilots

Federico Stagni
Cinzia Luzzi
Andrew McNab
Wojciech Jan Krzemien

On behalf of the DIRAC consortium
Pilots: not a new concept

Agents form an overlay layer hiding the underlying diversity

We substituted the word “agent” with the word “pilot”
In the meantime...

It seems like the grid is not anymore “The Grid”

Heterogeneity is the norm

(WLCG, CREAM, ARC, HTCondor, HLT, HPC, Opportunistic, Volunteer, Virtual Machines, clouds, vac, BOINC, containers, Docker…)

Integration of non Grid resources in ATLAS is a big investment with a big return

Automation is the key

R. McPherson
127th LHCC
ATLAS report

20% of data processing happens today on non Grid resources

The grid

Heterogeneity is the norm

(WLCG, CREAM, ARC, HTCondor, HLT, HPC, Opportunistic, Volunteer, Virtual Machines, clouds, vac, BOINC, containers, Docker…)

Integration of non Grid resources in ATLAS is a big investment with a big return

Automation is the key
Pilots are the “federators”

Send it
as a “pilot job”

Or just Run it!
e.g. as part of the contextualization of a (V)M

OR

“Make a machine a pilot machine, and you are done”
10 years later, pilots are still in use, and keep being developed

Today’s pilots

F. Stagni
CHEP 2015
~2 years ago **DIRAC** introduced the so-called

**“Pilots 2.0”**

- A pilot 2.0 is a standalone python script
- Common to all communities using DIRAC
- Can be run on every computing resource
A toolbox of pilot capabilities ("pilot commands") is available for the pilot. Pilots can be configured to run a set of them.

Any configuration possible, any order possible.
PILOT
WRAPPER

#!/bin/bash

PILOT

Pilot

Check WN
Install DIRAC
Configure WN
Configure CPU&Me
Start JobAgent

Match special jobs

python

Anatomy
The bootstrap issue (pilot wrappers):
  ○ A pilot, on a VM, starts blind (in the vacuum). Need to supply info:
    ■ Where to get the pilot script(s)
    ■ Names: Site, CE, Queue
  ○ Solve the “difficult” case, generically

Simple pilot wrappers
2. Configure it

● Self-discovering WN capabilities
  ○ Including CPU power
    ■ Using DB12
    ■ And #processors
    ■ MJF supported
  ○ And memory

→ more details
Benchmarking worker nodes using LHCb productions jobs and comparing with HEP-Spec06

● Commands list configurable:
  With REST interface on top of DIRAC Configuration System
  ○ By type of “Grid”
    ■ i.e. some commands may be needed only for volunteer computing resources
  ○ By setup (e.g. production, test, …)
3. Get the jobs

- A DIRAC client is installed
- By default a “JobAgent” is used to match the capabilities of the WN with the requirements of the waiting jobs.
- Support for MultiProcessor jobs:
  - Pilots advertise multi-processor payload slots
  - Multi-processor payloads matched
  - No mix/backfilling allowed yet
4. Monitor it

● Pilot logging:
  ○ A list of messages like
    ■ "I've booted up" …
    ■ "I found the DIRAC pilot ok" …
    ■ "I'm about to shutdown"…
  ○ Uses MQ systems (stomp)

● Pilot self-upload their own logs before shutting down
  ○ Needs reliable/fast SE
● DIRAC Pilots are the real federator of “any” computing resource
● Used by all the DIRAC communities in every DIRAC installation
  ○ Single or multi-VO
● Already VERY flexible: highly configurable, easy to extend
  ○ E.g. LHCb uses extended the command to install LHCbDIRAC
● Actively developed
WMS overview

Pilot based Workload Management

- High user job efficiency
- Suitable for usage with heterogeneous resources
- Allowing application of community policies