Job Orchestration at SDC-ES

BrownThrower (BT)

Pau Tallada

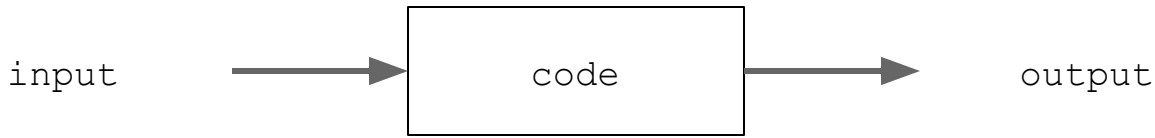
Contents

Preface

- IAL was not ready for mass-production
- BT already used in other projects
  - Re-use existing tools and knowledge at PIC
  - Very easy to integrate with OU-SIM pipeline
- But, this is a TEMPORARY solution until IAL is ready
- Brownthrower has some tightly coupled dependencies specific of PIC (gLite, among others)

Task

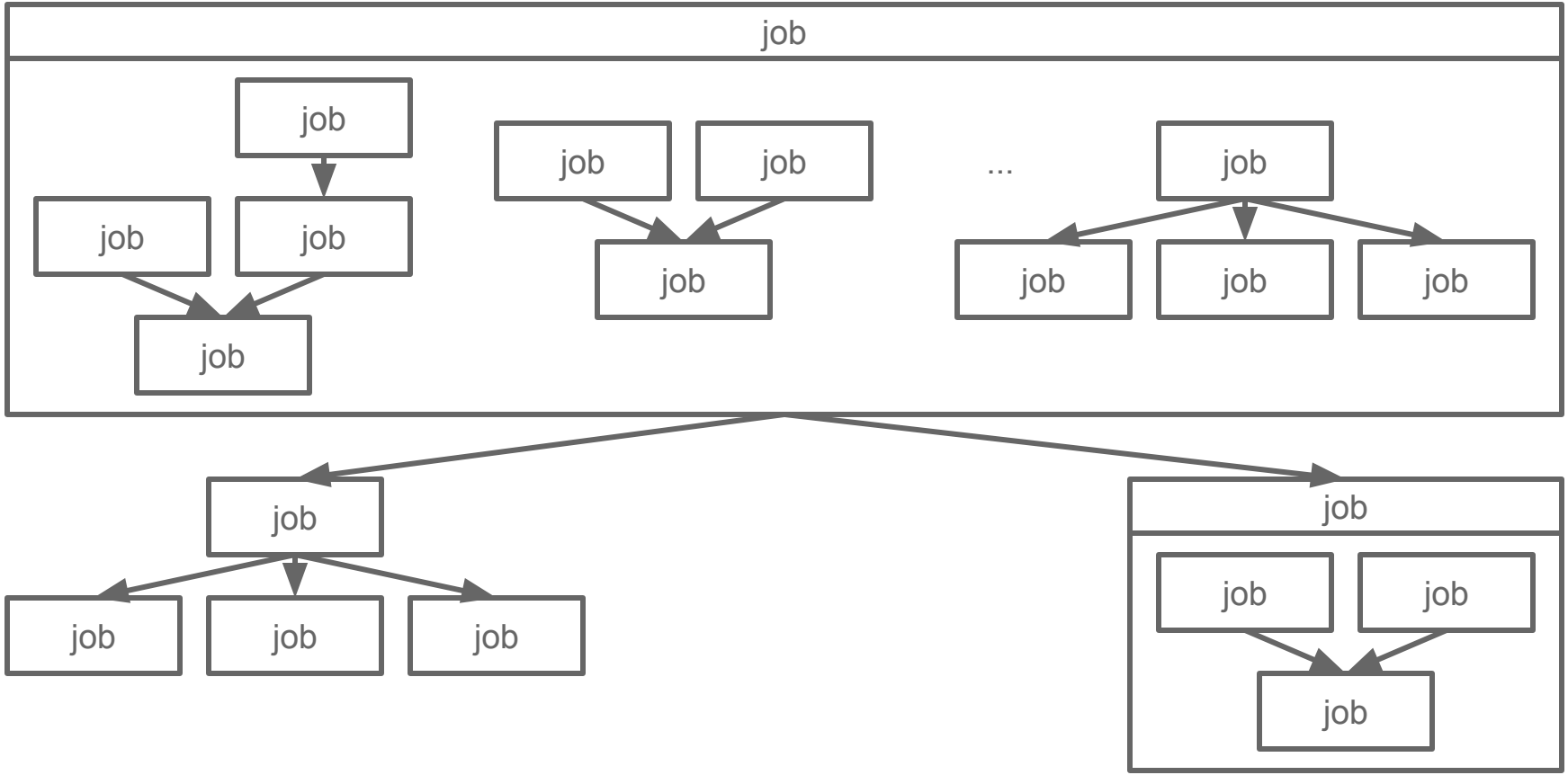
- A piece of code that produces something
- Receives an input in YAML format
- Produces an output in YAML format
- Similar to the "pipeline" concept in Euclid



Job

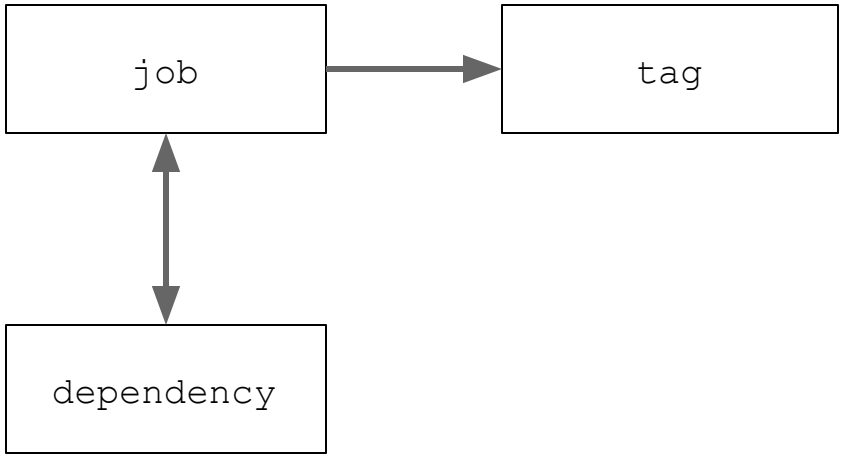
- A "instance" of a task
  - A task with a concrete input and output
- Can be run in a computing node
- May depend on an arbitrary number of other jobs
- May decompose itself in subjobs, unlimited nesting
- Has a status that describes its state (QUEUED, RUNNING, FAILED, among others)
- Can be monitored and debugged in real-time

Dependencies



Data Model

- Just 3 tables:
  - Job: input, config, status, ...
  - Tag: user-defined data attached to a Job (backtrace, logs, ...)
  - Dependency: parent and child jobs



Manager

- Main interface (CLI) for job management
  - Intuitive, easy to use
  - Allows a user to create, edit, submit, abort, remove and link jobs
  - Supports multiple and concurrent clients
  - Automatic transaction retry in case of conflict

```
[root@services ~]# /software/vo.paus.pic.es/ENV/scipic/bin/brownthrower -u postgresql://cosmohub:2hot2work@db03.pau.pic.es/scipic
(brownthrower): job list
  id    super_id  name            status     created              queued               started              ended
----  ----------  --------------  ---------  -------------------  -------------------  -------------------  -------------------
 375               catalog.query   DONE       2013-05-02 11:13:26  2013-05-02 11:13:26  2013-05-02 11:13:52  2013-05-02 11:21:47
 425               catalog.query   DONE       2013-05-10 17:36:22  2013-05-10 17:36:22  2013-05-10 17:38:20  2013-05-10 17:38:22
 426               catalog.query   DONE       2013-05-10 17:39:56  2013-05-10 17:39:56  2013-05-10 17:40:23  2013-05-10 17:40:23
 427               catalog.query   FAILED     2013-05-10 18:04:15  2013-05-10 18:04:15  2013-05-10 18:06:26  2013-05-10 18:06:26
 428               catalog.query   DONE       2013-05-10 18:08:59  2013-05-10 18:08:59  2013-05-10 18:09:26  2013-05-10 18:09:27
 429               catalog.query   DONE       2013-05-10 18:10:41  2013-05-10 18:10:41  2013-05-10 18:11:27  2013-05-10 18:11:56
 430               catalog.query   DONE       2013-05-11 11:15:53  2013-05-11 11:15:53  2013-05-11 11:16:08  2013-05-11 13:02:10
 431               catalog.query   DONE       2013-05-17 12:00:42  2013-05-17 12:00:42  2013-05-17 12:01:04  2013-05-18 00:53:33
 432               catalog.query   DONE       2013-05-17 12:14:24                       2013-05-17 12:34:15  2013-05-17 12:51:39
 433               catalog.query   DONE       2013-05-22 12:46:27  2013-05-22 12:46:27  2013-05-22 12:46:51  2013-05-22 12:46:55
 434               catalog.query   DONE       2013-05-22 12:49:10  2013-05-22 12:49:10  2013-05-22 12:49:55  2013-05-22 12:50:22
 435               catalog.query   DONE       2013-05-23 19:07:39  2013-05-23 19:07:39  2013-05-23 19:08:01  2013-05-23 20:51:49
 436               catalog.query   DONE       2013-05-24 11:30:17  2013-05-24 11:30:17  2013-05-24 12:45:46  2013-05-24 12:46:38
 437               catalog.query   DONE       2013-05-24 13:46:13  2013-05-24 13:47:17  2013-05-24 13:47:18  2013-05-24 13:47:48
 438               catalog.query   FAILED     2013-05-24 13:57:41  2013-05-24 14:04:12  2013-05-24 14:04:16  2013-05-24 14:04:16
 439               catalog.query   FAILED     2013-05-24 14:04:48  2013-05-24 14:05:16  2013-05-24 14:05:16  2013-05-24 14:05:16
 440               catalog.query   FAILED     2013-05-24 14:12:49  2013-05-24 14:13:29  2013-05-24 14:13:49  2013-05-24 14:13:50
 441               catalog.query   FAILED     2013-05-24 14:49:42  2013-05-24 14:49:46  2013-05-24 14:49:52  2013-05-24 14:49:53
 442               catalog.query   DONE       2013-05-24 15:59:07  2013-05-24 15:59:07  2013-05-24 15:59:24  2013-05-24 16:09:07
 443               catalog.query   DONE       2013-05-24 16:02:11  2013-05-24 16:02:11  2013-05-24 16:02:32  2013-05-24 16:11:18
 444               catalog.query   DONE       2013-05-24 16:05:48  2013-05-24 16:06:01  2013-05-24 16:06:59  2013-05-24 16:07:00
 445               catalog.query   DONE       2013-05-24 16:26:04  2013-05-24 16:26:04  2013-05-27 15:28:24  2013-05-27 15:28:25
 461               catalog.query   DONE       2013-06-04 09:42:49  2013-06-04 09:42:49  2013-06-04 09:43:23  2013-06-04 10:27:13
 462               catalog.query   DONE       2013-06-04 09:43:13  2013-06-04 09:43:13  2013-06-04 09:43:17  2013-06-04 10:27:09
```

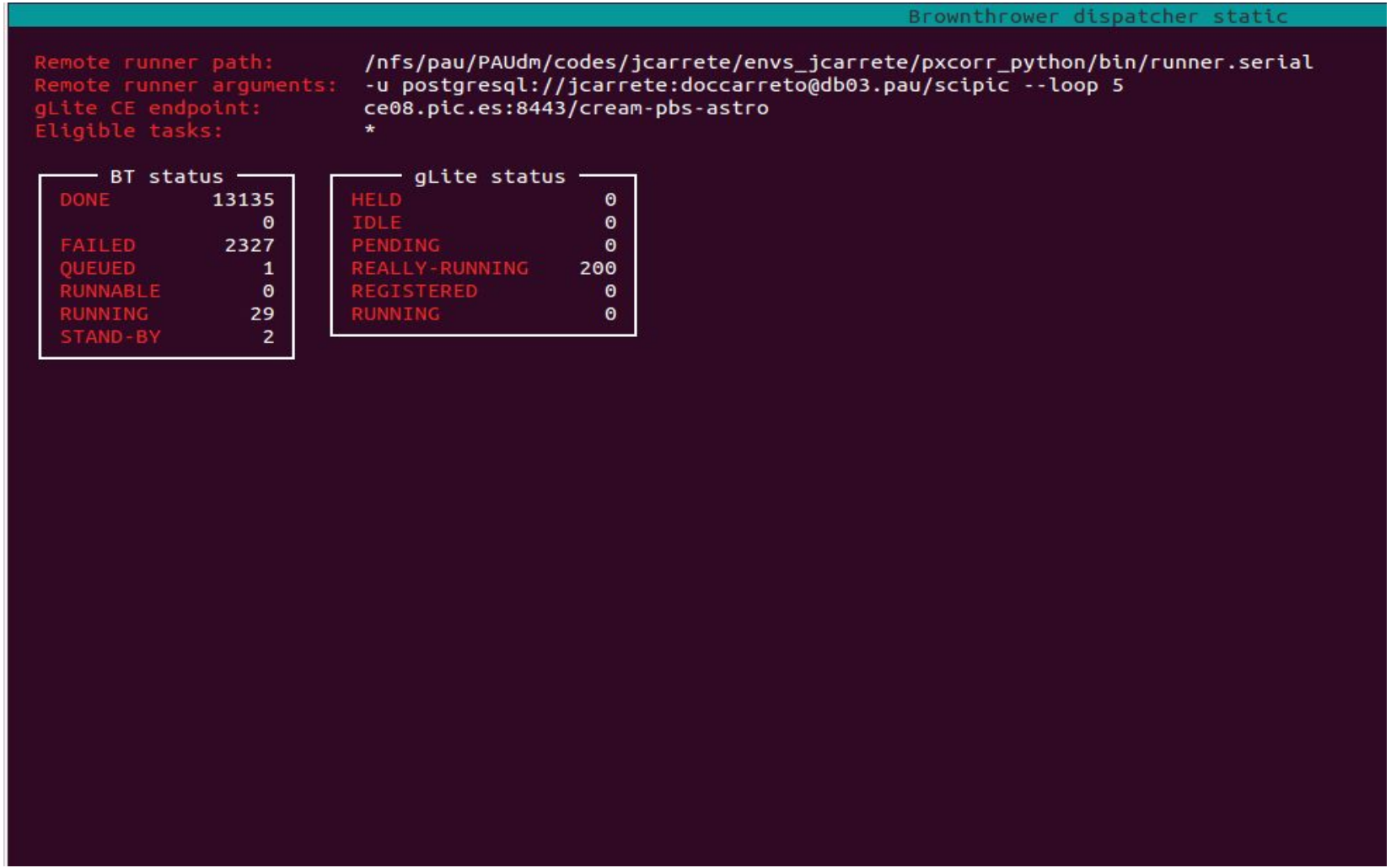
```
(brownthrower): job graph 137019
PARENT/CHILD JOBS:
kind    id      super_id  name          status    created              queued               started              ended
------  ------  --------  ------------  --------  -------------------  -------------------  -------------------  -------------------
#####   137019            ParentPxcorr  STAND-BY  2015-10-23 13:39:24  2015-10-23 13:40:57  2015-10-23 13:42:17  2015-10-23 13:42:17

SUPER/SUB JOBS:
kind    id      super_id  name                  status    created              queued               started              ended
------  ------  --------  --------------------  --------  -------------------  -------------------  -------------------  -------------------
#####   137019            ParentPxcorr          STAND-BY  2015-10-23 13:39:24  2015-10-23 13:40:57  2015-10-23 13:42:17  2015-10-23 13:42:17
SUB     137020    137019  ParentCorrelate       STAND-BY  2015-10-23 13:42:17  2015-10-23 13:42:17  2015-10-23 13:42:42  2015-10-23 13:42:58
SUB     137021    137019  ParentMakemap         DONE      2015-10-23 13:42:17  2015-10-23 13:42:17  2015-10-23 13:42:42  2015-10-23 13:42:42
SUB     137022    137019  CreateFinalPxcorrHdf5 QUEUED    2015-10-23 13:42:17  2015-10-23 13:42:17
```

Runner

- Very complex and multiple responsibilities:
  - Isolate and provide a consistent environment for jobs
  - Monitor the job and update its status in real-time
  - Capture and dump logs, stdout, stderr and subprocesses
  - Catch exceptions, store the backtrace
  - Allow in-place debugging and profiling
  - Job pre-emption (user can abort any job any moment)
- Uses 3 independent processes and PostgreSQL NOTIFY calls
- Some of these features are already present in the EuclidSIM wrappers

Dispatcher

- Submits and monitors a set of jobs to be run on the farm
- Allows static (fixed number of pilots) or on-demand job scheduling
- Usable in BT 2.x, but not fault-tolerant
- Updates job and queue status in real-time


Brownthrower dispatcher static
Remote runner path: /nfs/pau/PAUdm/codes/jcarrete/envs_jcarrete/pxcorr_python/bin/runner.serial
Remote runner arguments: -u postgresql://jcarrete:doccarreto@db03.pau/scipic --loop 5
gLite CE endpoint: ce08.pic.es:8443/cream-pbs-astro
Eligible tasks: *
BT status
DONE 13135
0
FAILED 2327
QUEUED 1
RUNNABLE 0
RUNNING 29
STAND-BY 2
gLite status
HELD 0
IDLE 0
PENDING 0
REALLY-RUNNING 200
REGISTERED 0
RUNNING 0

Results

- Most projects in our department are using it for production
  - MICE photoz, galaxy shapes, ...
  - PAU data management
  - DES galaxy clustering
  - Euclid OU-SIM
- More than 3 years since BT v1.0, 20 months running v2.x
- Lots of lessons, experience and knowledge
- More than 550.000 jobs successfully executed
- More than 20.000 hours (wall time) of work done