(GENOMICS) WORKLOAD ORCHESTRATION WITH NEXTFLOW

Paolo Di Tommaso, CRG

22 June 2017, ISC HPC, Frankfurt
WHO IS THIS CHAP?

@PaoloDiTommaso
Research software engineer
Comparative Bioinformatics,
Notredame Lab
Center for Genomic Regulation (CRG)
GENOMIC WORKFLOWS

- Data analysis application to extract information from genomic datasets
- Mash-up of many different tools and scripts
- Embarrassingly parallelisation, can spawn 100-100k jobs over distributed cluster
- Complex dependency trees and configuration → very fragile ecosystem
* Companion parasite genome annotation pipeline, Steinbiss et al., DOI: 10.1093/nar/gkw292
To reproduce the result of a typical computational biology paper requires 280 hours.

≈ 1.7 months!
CONTAINERS

Containers are emerging as a solution to the problem of reproducibility of scientific workflows

- 100~ scientific publications mentioning Docker
- Large adoption in bioinformatics:
  - DockStore
  - BioContainers
  - BioShadock
  - Bioboxes
CONTAINER ISOLATION

- Allows you to create a ready-to-run package with all software dependencies
- Just one dependency instead of dozens
- Consistent results over time
HOW TO MANAGE A CONTAINERISED WORKLOAD AT SCALE?
CONTAINERISED WORKLOADS

Host

Container image
- Workflow scripts
- Workflow runtime
- Environment
- Third party deps
- Binary tools
- Libraries

Batch scheduler

Workflow app

Orchestrator

HPC cluster
ARE THE RIGHT TOOL FOR
SCIENTIFIC WORKLOADS?

kubernetes

OPENSHIFT

MESOS

Istio
SERVICES ORCHESTRATION ≠ JOBS SCHEDULING
CHALLENGES

• Isolate each task execution in its own container
• Manage jobs scheduling and dependencies
• Allow user to use any existing tools and scripts
• Automatic errors failover & execution checkpoints
• Enable portability across platforms (HPC and cloud)
#!/bin/bash
blastp -query sampla.fasta -outfmt 6 \
  | head -n 10 \n  | cut -f 2 \n  | blastdbcmd -entry - > sequences.txt
process foo {

input:
file 'sample.fasta' from fasta_files

output:
file 'sequences.txt' into result_files

script:
""
#!/bin/bash
blastp -query sampla.fasta -outfmt 6 \
| head -n 10 \
| cut -f 2 \
| blastdbcmd -entry -> sequences.txt
""
REACTIVE NETWORK

- Declarative computational model for parallel process executions
- Processes wait for data, when an input set is ready the process is executed
- They communicate by using dataflow variables i.e. async FIFO queues called channels
- Parallelisation and tasks dependencies are implicitly defined by process in/out declarations
PORTABILITY

Config

Nextflow

Local executor

Grid executor

Local OS

Batch scheduler

NFS

Platform Computing

UNIVA

PBS Works

HTCondor

kubernetes

Amazon Web Services
process {
  executor = 'sge'
  queue = 'cn-el6'
  memory = '10GB'
  cpus = 8
  time = '2h'
  container = 'ncbi/blast:3.2'
}
BENEFITS

• Dead easy deployment

• Precise control on the execution runtime

• Portable across different execution platforms

• Decouple application logic from infra/configuration

• Enable reproducibility across systems and over time
WHO IS USING NEXTFLOW?
ACKNOWLEDGMENT

Notredame Lab, CRG

Evan Floden
Emilio Palumbo
Cedric Notredame

http://nextflow.io