UL HPC School 2015

PS 6C: Advanced workflows on parametric job management

H. Cartiaux

University of Luxembourg, Luxembourg

Latest versions available on ulhpc-tutorials.readthedoc.org:

UL HPC tutorials:	https://github.com/ULHPC/tutorials
UL HPC School:	http://hpc.uni.lu/hpc-school/
PS 6Ctutorial sources:	http://bit.ly/1Gwjgba

Summary

Introduction

Objectives

- Discover the advanced features of OAR
- Show how it can improve your workflows
- Use the advanced launcher scripts

Follow the tutorial on readthedocs:

- http://bit.ly/1Gwq3BL

Summary

Container jobs

- special job type
- a container is a pool of resources
- you can submit subjobs in a container

```
(frontend)$> oarsub -t container -l nodes=2 "sleep 1800"
(frontend)$> oarsub -I -t inner=<container id>
```

Array jobs

- submit N jobs in one oarsub command
- split your workload according to the index
- other possibility: "–array-param-file"

```
(frontend)$> oarsub -array <N> -l /core=1
/path/to/prog.sh
```

Summary

Best effort

- low priority
- overcome the limits (50 jobs in the default queue, 1000 in the besteffort queue)
- can be killed if the resources are required by the default queue
- killed jobs can be resubmitted automatically (idempotent)
- use a short walltime and resubmit the jobs until completion

```
(frontend)$> oarsub -t besteffort /path/to/prog
(frontend)$> oarsub -t besteffort -t idempotent
/path/to/prog
```

Summary

Checkpointing

- save the state of an application
- be able to restart it from the saved state
- overcome the limits (walltime)
- bonus: fault tolerance
- case by base custom implementation, or generic with BLCR

```
(frontend)$> oarsub -checkpoint 30 -signal 12 -l
walltime=00:02:00 -t besteffort -t idempotent
/path/to/prog
```

Questions?

1. Exercise 1: Advanced OAR features: container, array jobs
2. Exercise 2: Best effort jobs
3. Exercise 3: Checkpoint restart