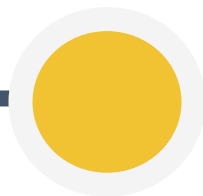


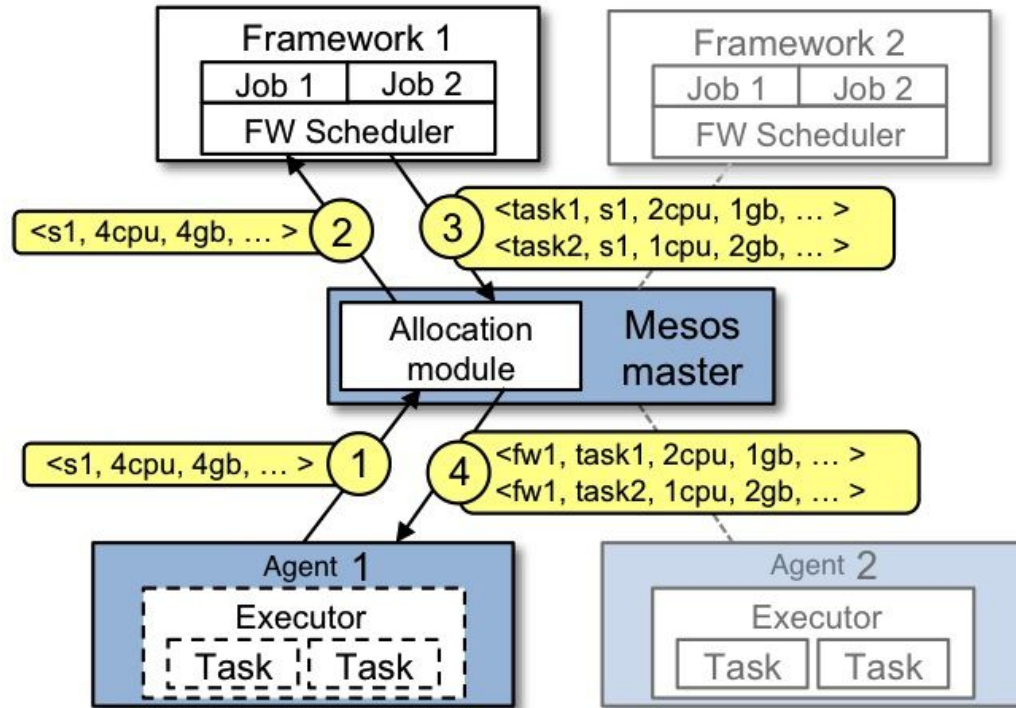
Connecting Workflows To Mesos



Charles Zheng
czheng2@gmail.com



Mesos architecture

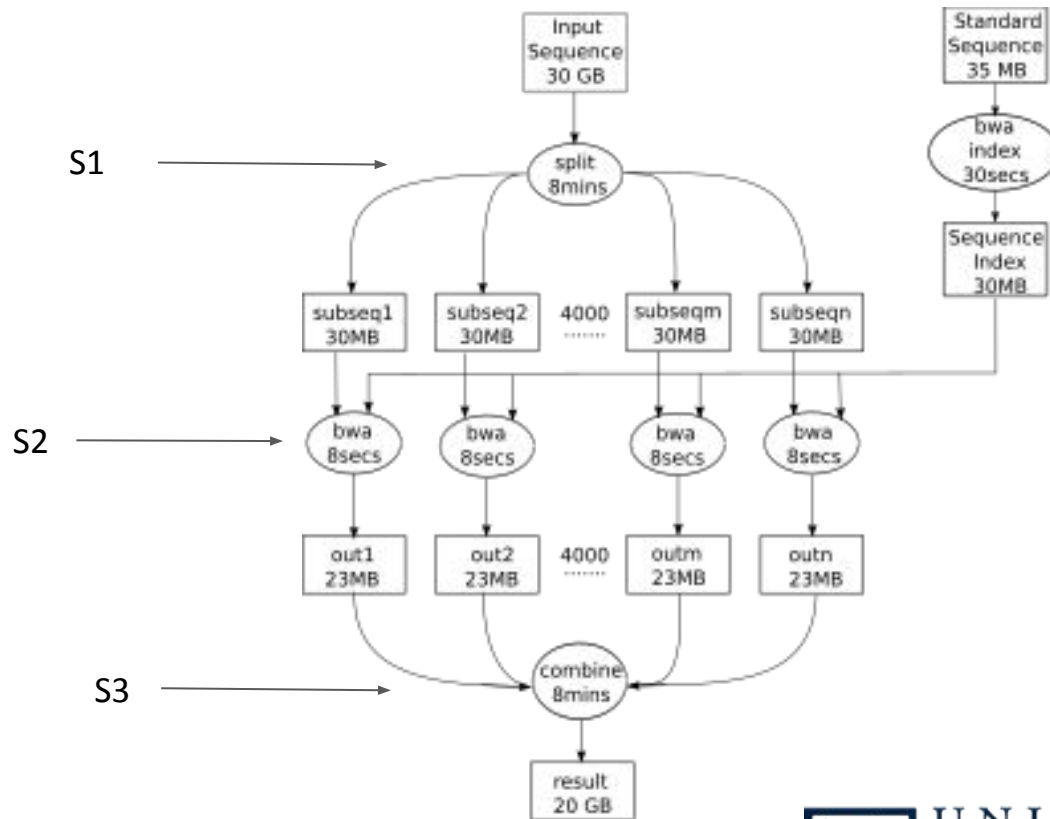


Why Mesos

- Mesos uses container technology for resource management that has lower overhead comparing to system using virtual machine
- Support Containers (Mesos container and Docker)
- The two level architecture is extensible that allow different users to build their own applications on the same platform
- It has been deployed on many commercial clouds



A typical HPC workflow

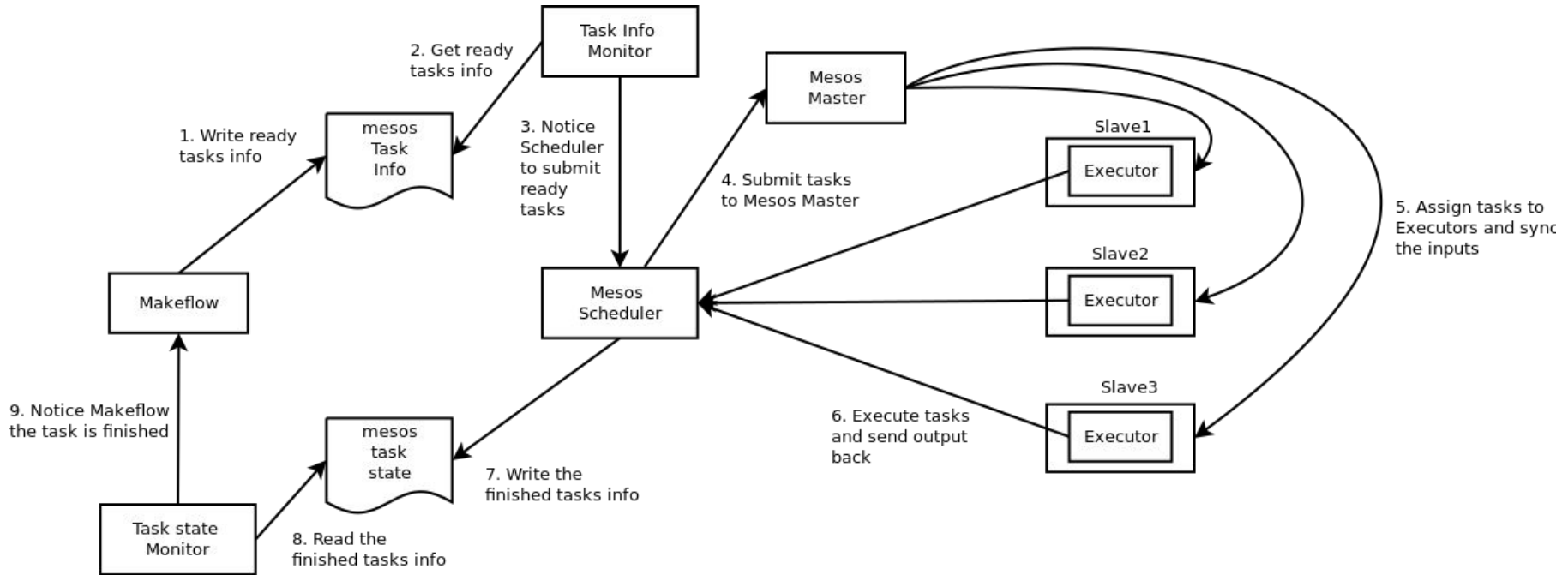


Design challenges

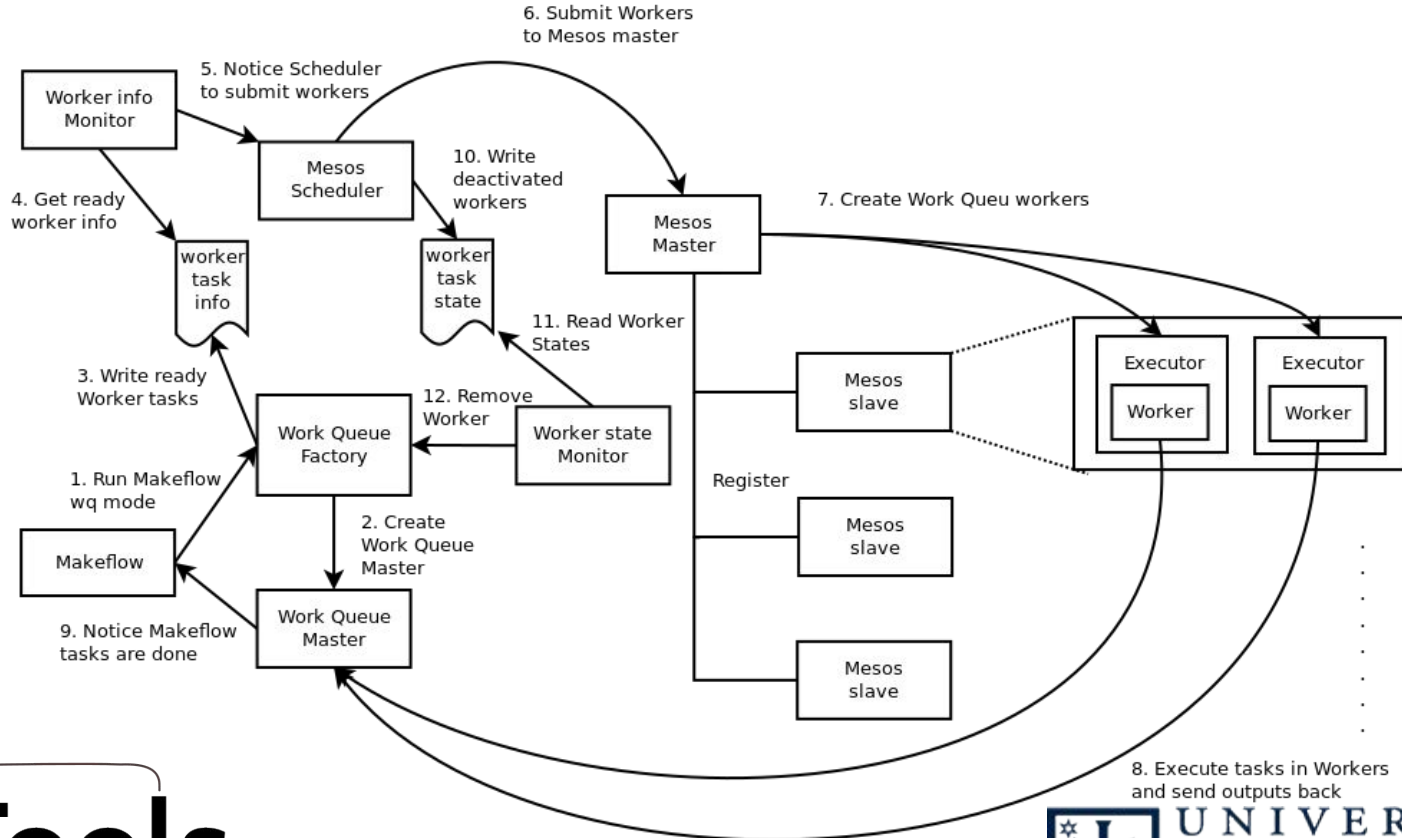
- Synchronize workflow status between workflow engine and Mesos
- Avoid starvation caused by the default resource allocation policy
- Executors do not transfer results back to user
- Garbage collection policy used by Mesos will cause the disk be filled up quickly
- Tasks resource requirements are not known in advance



Makeflow and Mesos



Makeflow, Work Queue and Mesos



How to use

- Makeflow and Mesos

- `makeflow -T mesos --mesos-master=<ip:port> --mesos-path=</path/to/mesos/python/site-packages>`
Makeflow

- Makeflow, Work Queue and Mesos

- `makeflow -T wq -N <project_name>` Makeflow
- `work_queue_factory -T mesos -M <project_name> --mesos-master=<ip:port>`
`--mesos-path=</path/to/mesos/python/site-packages>`



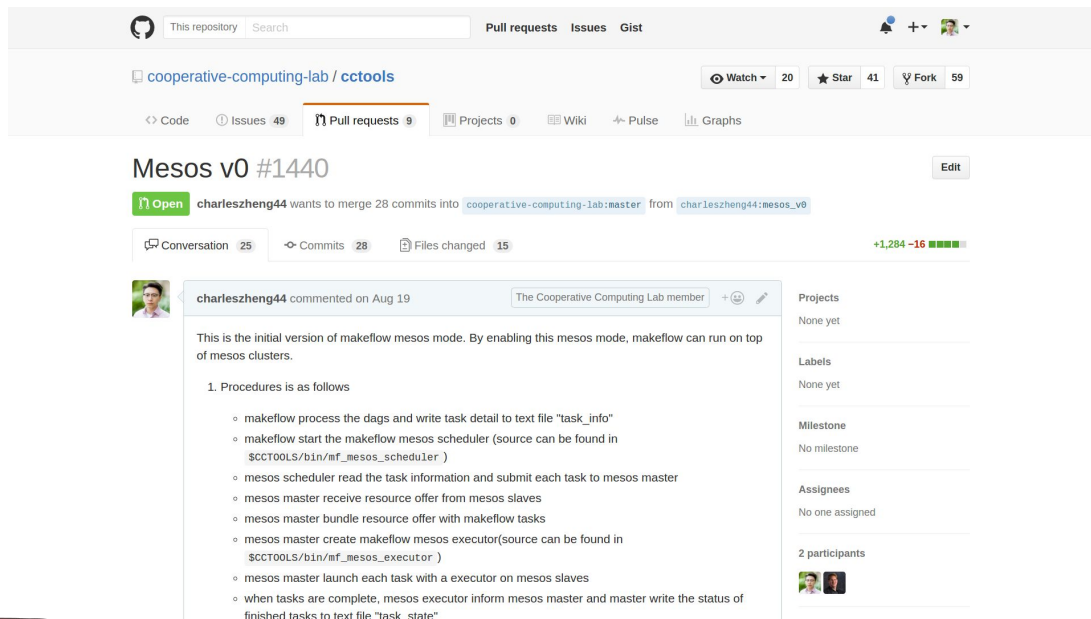
Lessons learned

- A workflow monitor is required to communicate between workflow engine and Mesos
- Uncarefully task arrangement of workflows can cause starvation
- Remember to delete results !
- An intermediate task scheduling framework (Work Queue) is preferred



Where to get it

A pull request is currently available.



The screenshot shows a GitHub pull request interface. At the top, it displays the repository name 'cooperative-computing-lab / cctools' with 20 watches, 41 stars, and 59 forks. The pull request title is 'Mesos v0 #1440' and it is currently open. The pull request description states: 'charleszheng44 wants to merge 28 commits into cooperative-computing-lab:master from charleszheng44:mesos_v0'. Below the title, it shows 25 conversations, 28 commits, and 15 files changed, with a net change of +1,284 lines and -16 lines. A comment from charleszheng44, dated August 19, explains that this is the initial version of 'makeflow mesos mode' and lists the following procedures:

- Procedures is as follows
 - makeflow process the dags and write task detail to text file "task_info"
 - makeflow start the makeflow mesos scheduler (source can be found in `$CCTOOLS/bin/mf_mesos_scheduler`)
 - mesos scheduler read the task information and submit each task to mesos master
 - mesos master receive resource offer from mesos slaves
 - mesos master bundle resource offer with makeflow tasks
 - mesos master create makeflow mesos executor(source can be found in `$CCTOOLS/bin/mf_mesos_executor`)
 - mesos master launch each task with a executor on mesos slaves
 - when tasks are complete, mesos executor inform mesos master and master write the status of finished tasks to text file "task_state"

The right sidebar shows project, label, milestone, assignee, and participant information.

