SLURM pipeline

Processing 115 billion NGS reads

Terry Jones
Dept. of Zoology
University of Cambridge
tcj25@cam.ac.uk

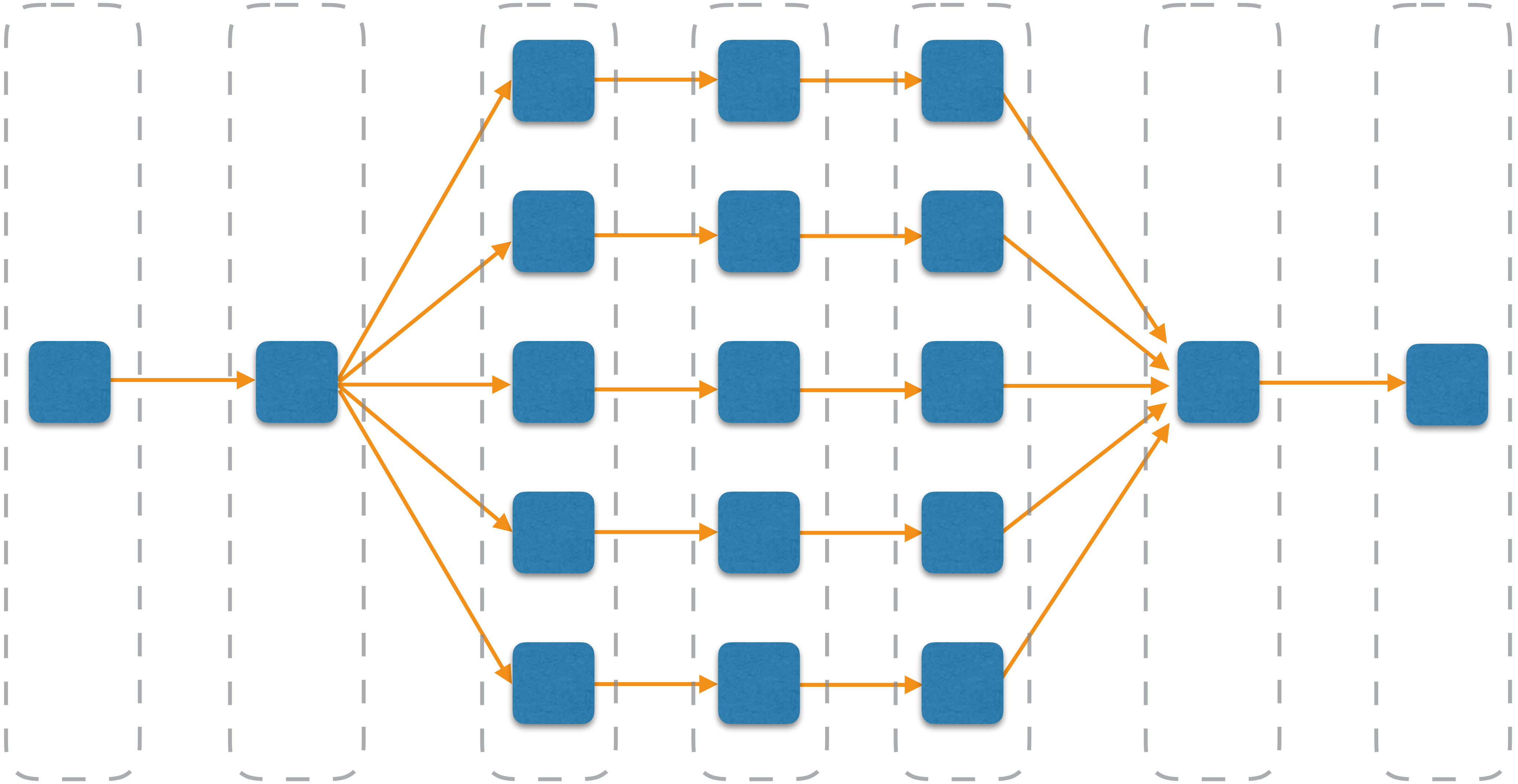
Ceci n'est pas une pipeline.
Magritte
Lunpi

slurm

workload manager

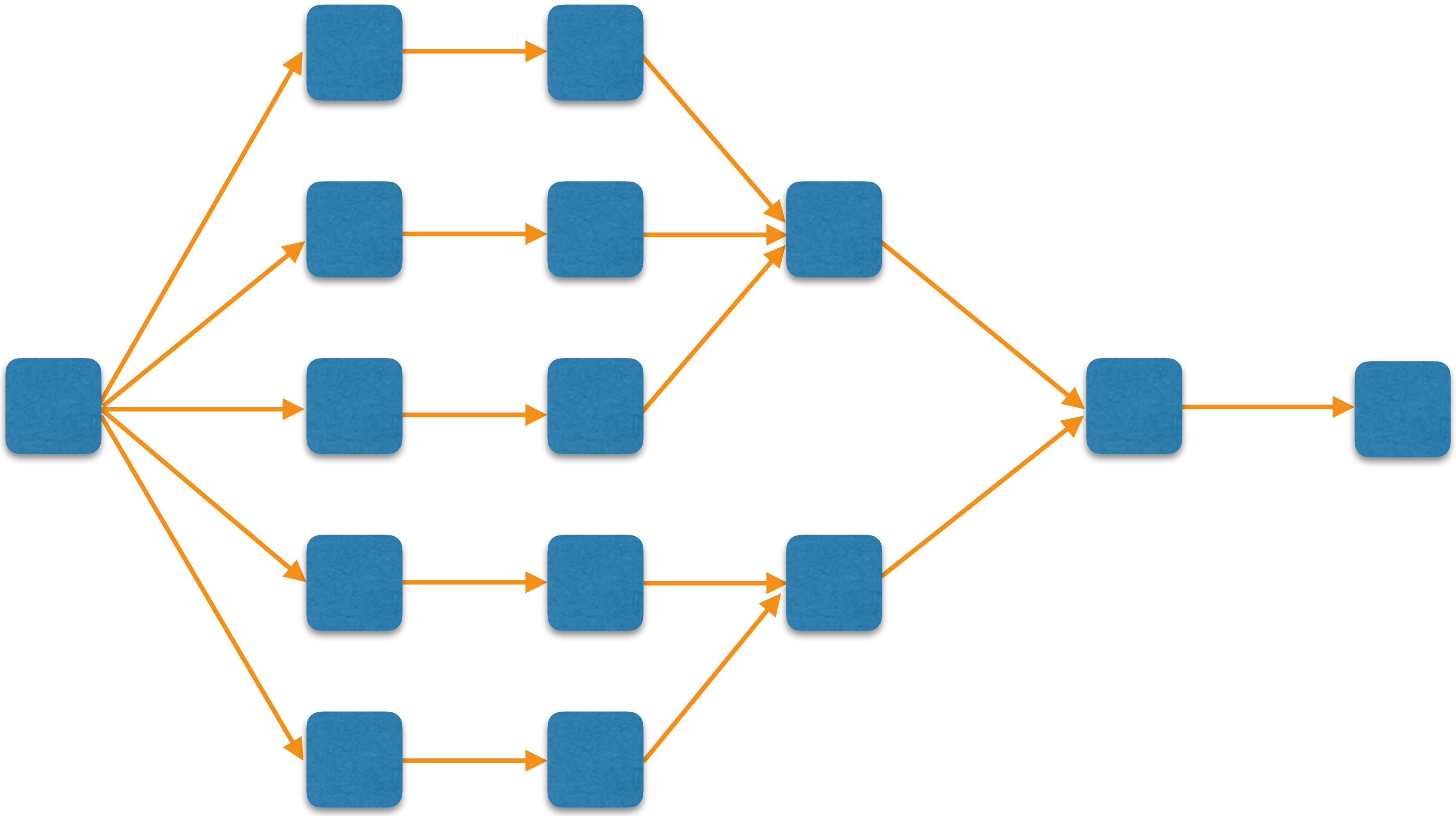
- Simple Linux Utility for Resource Management
- Resource allocation, job launch, manage queues
- Used on ~60% of the TOP500 supercomputers
- Open source (https://slurm.schedmd.com/)

Typical NGS data flow

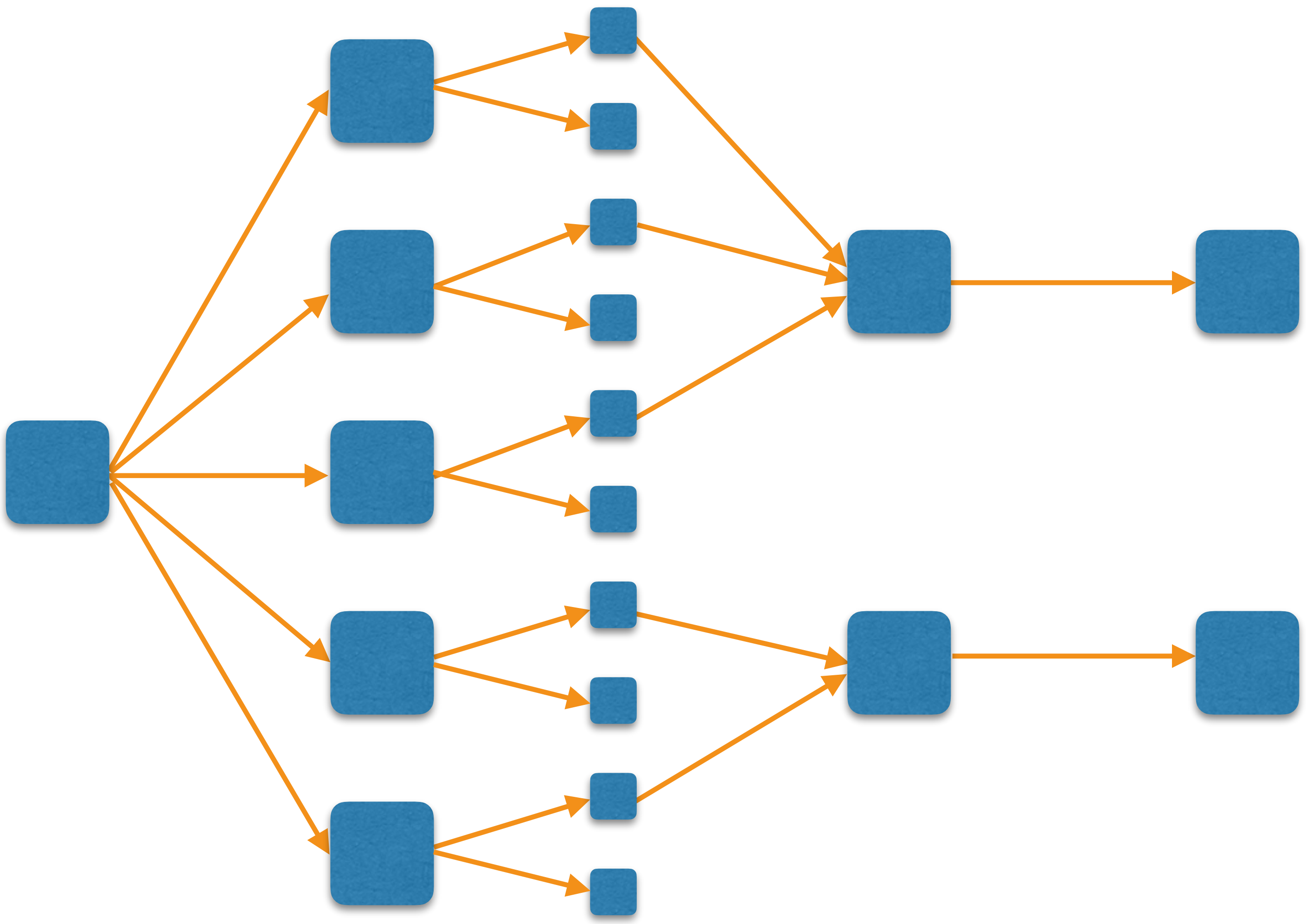


Start Split Map Contigs Match Summary Stop

Alternate data flow



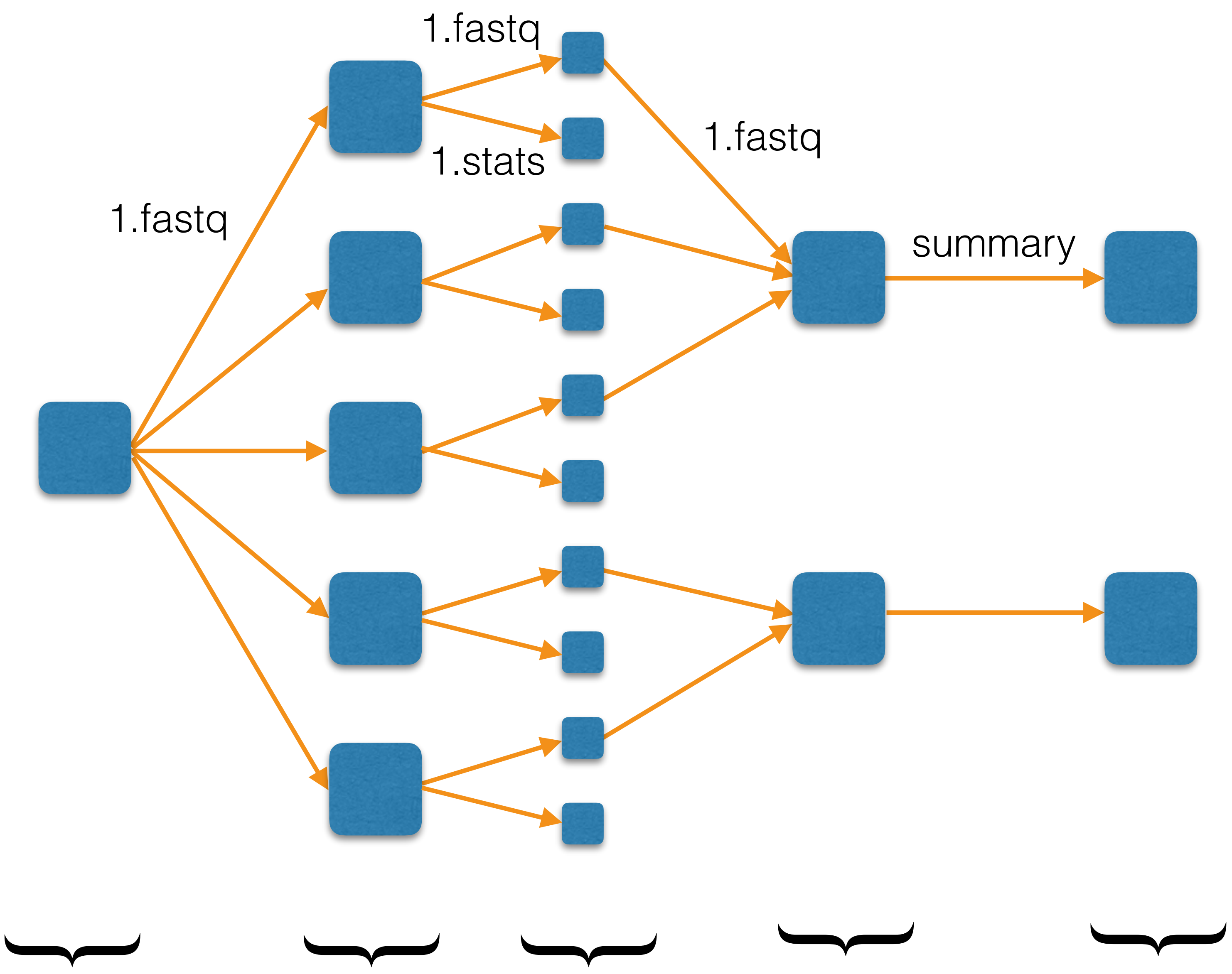
Yet another data flow



Aims

- Allow for these kinds of workflows
- Make specification as simple as possible
- Make no assumptions about underlying operations
- A formal framework for our pipelines

Steps and tasks



Steps are conceptual, tasks are concrete

Steps:

Tasks: → named tasks are emitted by steps

Pipeline specification (JSON)

The specification gives an ordered list of steps & their dependencies

```
{
    "steps": [
        {
            "name": "start-log",
            "script": "start-log.sh"
        },
        {
            "name": "split",
            "script": "split-fasta.sh"
        },
        {
            "dependencies": ["split"],
            "name": "blast",
            "script": "blast.sh"
        },
        {
            "collect": true,
            "dependencies": ["blast"],
            "name": "summarize",
            "script": "summarize.sh"
        },
        {
            "dependencies": ["summarize"],
            "name": "end",
            "script": "summarize.sh"
        }
    ]
}
```

A "collect" step runs after all the *tasks* emitted by its dependent steps are finished

Extras

- Start / stop at arbitrary pipeline steps
- Allow simulation and step skipping
- Add tools to inspect, cancel, start after jobs

Open source

- Written in Python
- https://github.com/acorg/slurm-pipeline
- Documentation, tests, examples
- We built multiple pipelines, to process 115 billion NGS reads in various ways