

Balsam Tutorial

DOE Joint-Facility Workflows Training
April 12, 2023



Christine Simpson (ALCF), Tom Uram (ALCF), Andrew Naylor (NERSC), Nicholas Tyler (NERSC)



Session Overview

Balsam at the DOE

- Overview of the Balsam model, terms, and basic functionality
- Worked example: Running LAMMPS with Balsam
- Lets do some setup now, so we're ready for the hands-on portion



Balsam Environment Setup

- If you haven't already, clone the repository on the file system at your facility:
`git clone git@github.com:CrossFacilityWorkflows/DOE-HPC-workflow-training.git`

- Go to the Balsam folder: `DOE-HPC-workflow-training/Balsam`

- Either run the `setup.sh` script or load pre-made conda module:

```
source setup.sh --site ALCF
```

or

```
module load conda  
conda activate /grand/projects/WALSforAll/conda_environments/balsam
```

Balsam Workflow Management Tool



A unified platform to manage high-throughput workflows across the HPC landscape

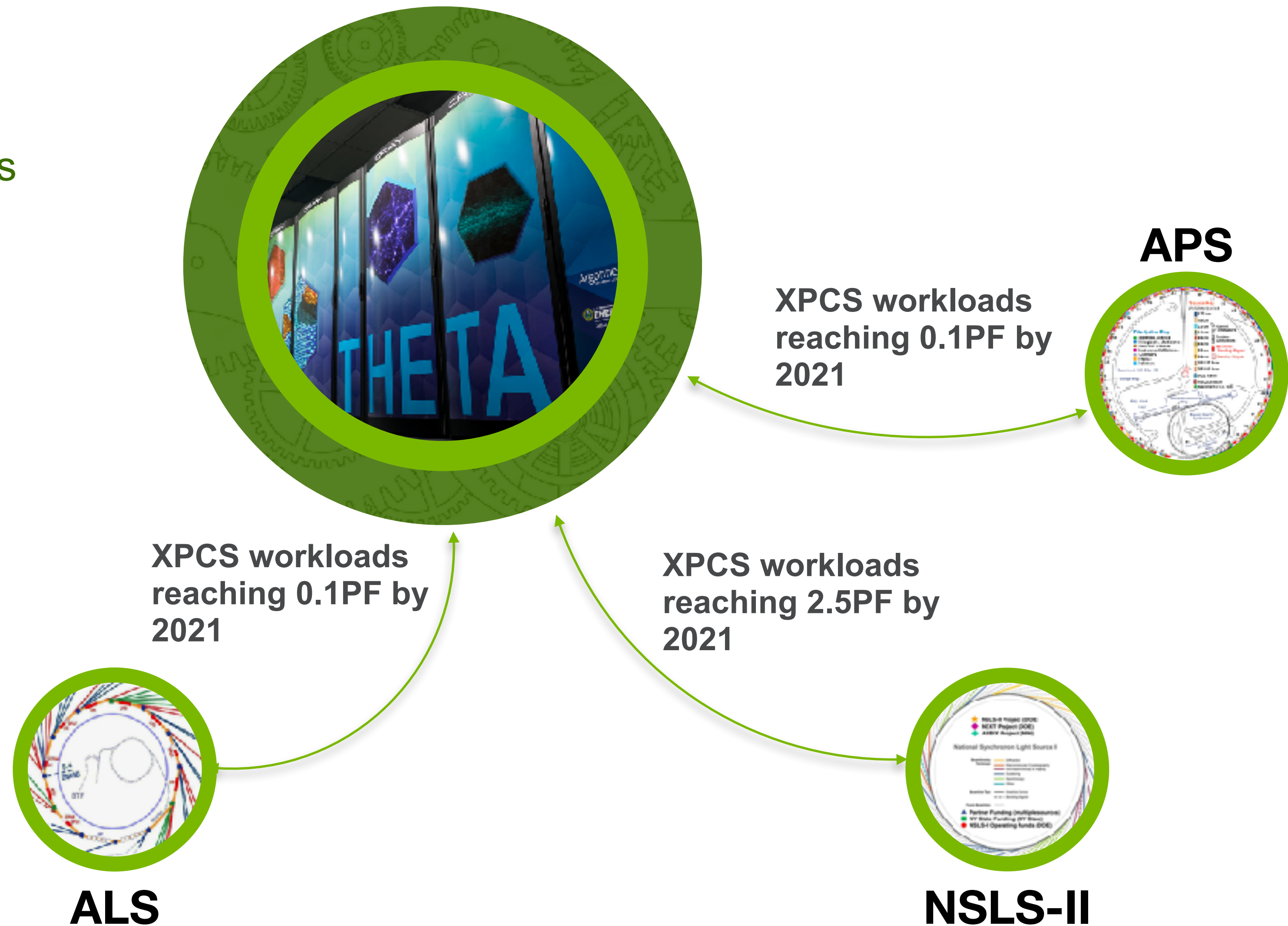
- Balsam was developed at ALCF and is used for deploying workflows on DOE HPC machines
- Applications:
 - Large Ensembles of Calculations in Material Sciences, Chemistry, Biology
 - Computing for Experiments, e.g. Light Sources, Neutrino Detection
 - ML and AI workflows - integration with DeepHyper and Colmena (in development)
 - Multi-site workflows - (i.e. different applications are run at different facilities)

BALSAM SUPPORT OF LIGHT SOURCE COMPUTING

Balsam enables **transparent access** to remote ALCF resources

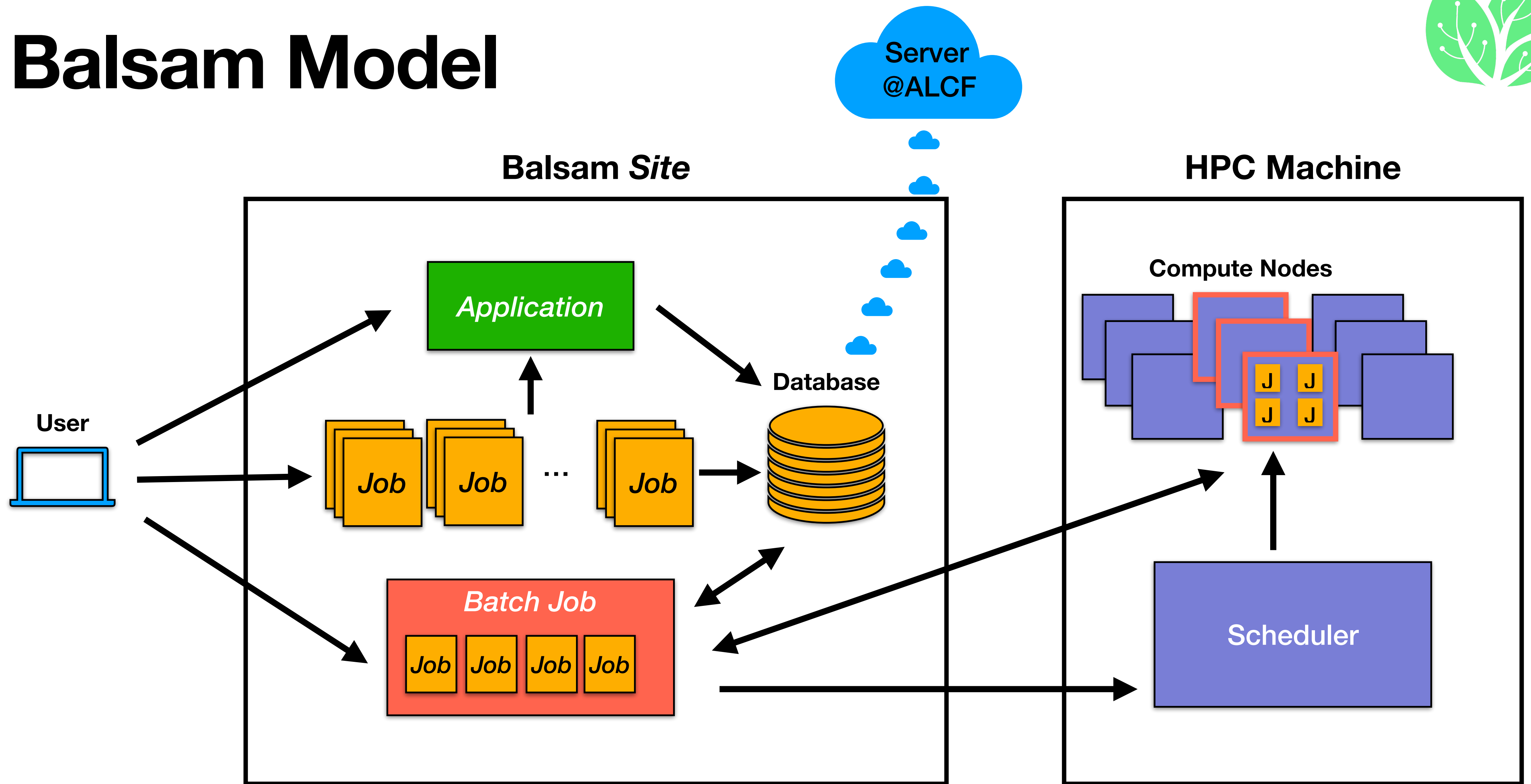
- identity management
- simplified scheduler API
- high-speed transfers via Globus

ALCF Theta (11.7 PetaFLOPs)



M. Salim, T. Uram, J. T. Childers, V. Vishwanath and M. Papka,
Balsam: Near Real-Time Experimental Data Analysis on
Supercomputers, SC2019

Balsam Model



Balsam Vocabulary



- **Application**

- Declaration of how to run an application expressed in Python. Persistent in the Site.

- **Job**

- A description of how to run one instance of an Application on specified compute resources.

- **Batch Job**

- A job on the scheduler. It will provision compute resources on the machine to run Jobs. It will automatically pull available Jobs from the database and run them according to their resource needs.

- **Site**

- Composed of two elements:
 - A workspace on the machine file system containing data, logs & settings
 - A background process running on the machine that communicates with the machine scheduler and the database on the server

A Simple Example

Running a compiled executable



```
from balsam.api import ApplicationDefinition, Job, BatchJob
```

```
class Lammps(ApplicationDefinition):  
    site = "polaris_tutorial"  
  
    def shell_preamble(self):  
        return f'source /path/to/envs.sh'  
  
    command_template = 'lmp -in /path/to/input.in -var tinit {{tinit}}'  
  
Lammps.sync()
```

```
initial_temps = [0.7, 1.0, 1.5]  
  
jobs = [Lammps.submit(workdir="LJ/1", parameters={"tinit": 0.7}),  
        Lammps.submit(workdir="LJ/2", parameters={"tinit": 1.0}),]  
  
jobs = Job.objects.bulk_create(jobs)
```

```
site = Site.objects.get("polaris_tutorial")  
BatchJob.objects.create(  
    site_id=site.id,  
    num_nodes=1,  
    wall_time_min=10,  
    job_mode="mpi",  
    project="WALSforAll",  
    queue="R476170",)
```

Application

- Includes executable
- Can set environment variables, load modules

Jobs

- Each instance of running an application
- Can vary inputs, resources for each job

Batch Job

- A Job on the scheduler that runs the Balsam 'jobs' declared above

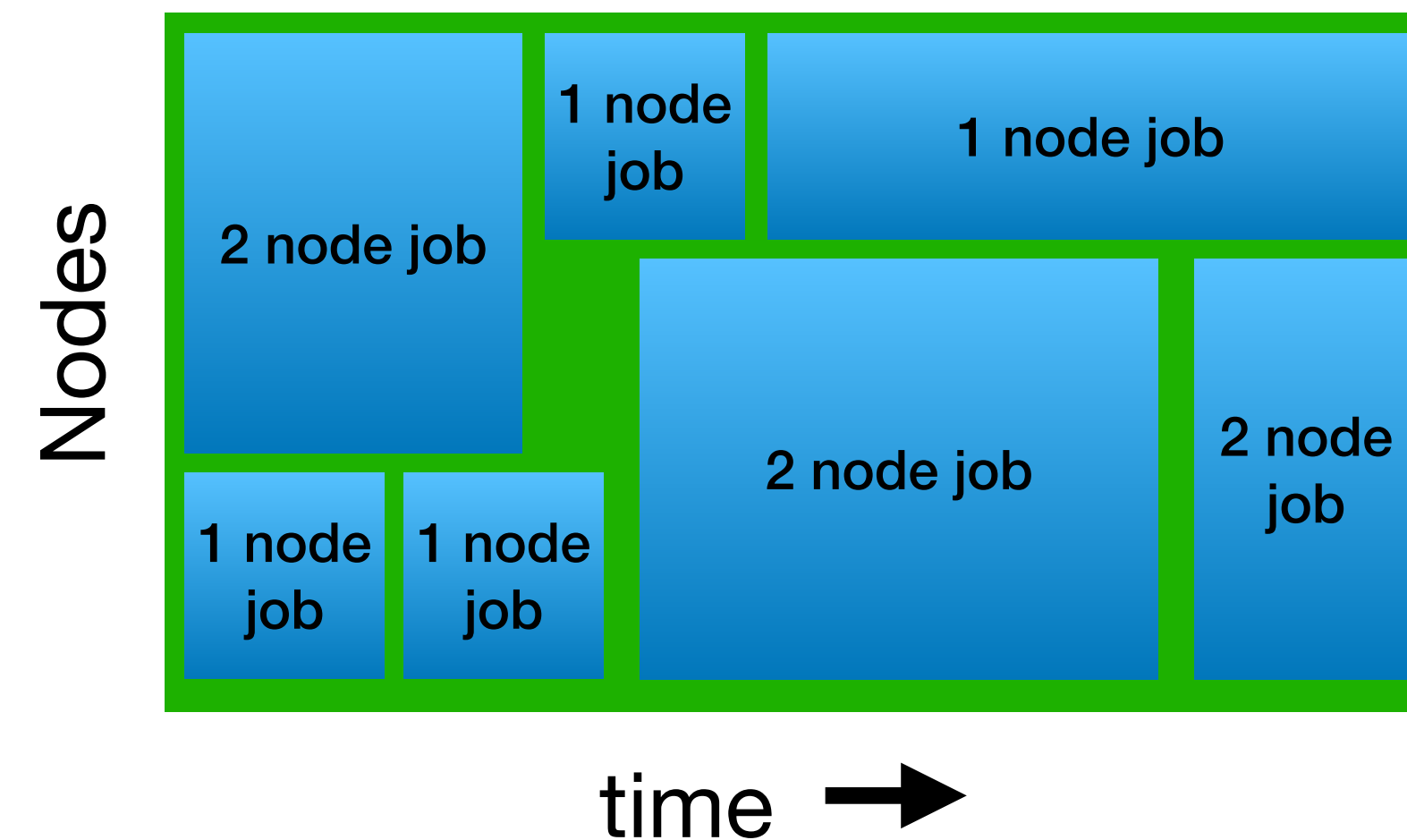
Balsam Features



- Able to run multi-node MPI jobs
- Can run multiple jobs that have diverse resource needs asynchronously in one batch job
- Allows for dependencies between jobs on different machines, i.e. multi-site workflows
- Command line interface and python API
- Pre-configured for DOE machines
- Support for data transfers via Globus online
- Automated elastic queuing option available
- Workflow stored in central server



Batch Job asynchronously
running Jobs





Tutorial Materials and References

- Tutorial repository is here:
<https://github.com/CrossFacilityWorkflows/DOE-HPC-workflow-training>
- Balsam materials are located in the folder:
DOE-HPC-workflow-training/Balsam
- Navigate to the materials for your facility and machine
- Balsam documentation can be found at:
<https://balsam.readthedocs.io/en/latest/>

Hands-on Section