

MICHIGAN STATE

UNIVERSITY

Project Plan

Simplifying High Performance Computing

The Capstone Experience

Team Michigan State University HPCC

Christian Luedtke

Zach Roush

Krista Solem

Zhihan Wang

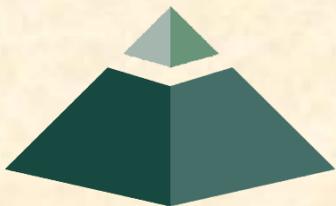
Matt Williams

Caleb Winner

Department of Computer Science and Engineering

Michigan State University

Spring 2019



*From Students...
...to Professionals*

Functional Specifications

- HPCC uses SSH/terminal and bash scripts to submit jobs.
- Researchers may not have strong technical background and find this difficult.
- Purpose is to create a Web app to view job statistics and create bash scripts for them
- This will make HPCC more accessible and friendlier to researchers.



Design Specifications

- Webpage User Interface
- UI to view job statistics
- Bash Script Generator
- Group/Admin Tools (stretch goal)
- Job Test Environment (stretch goal)



Screen Mockup: Job Overview

The screenshot displays a web application interface for job management. At the top, there is a navigation bar with tabs for 'Jobs', 'Playground', and 'Resource', and a 'Create' button. Below the navigation bar is a search bar and a list of jobs. The job list has columns for 'ID #' and 'Name'. The third job, 'test 3' with ID '02', is selected and highlighted in blue. To the right of the job list, a detailed view for 'Job Overview: #02. test 3' is shown, with a status of 'RUNNING'. This view includes two tabs: 'Details' and 'Unix Script'. The 'Unix Script' tab is active, displaying a shell script with various SLURM directives and module load commands.

ID #	Name
8885	test 1
1	test 2
02	test 3

Job Overview: #02. test 3 Status: **RUNNING**

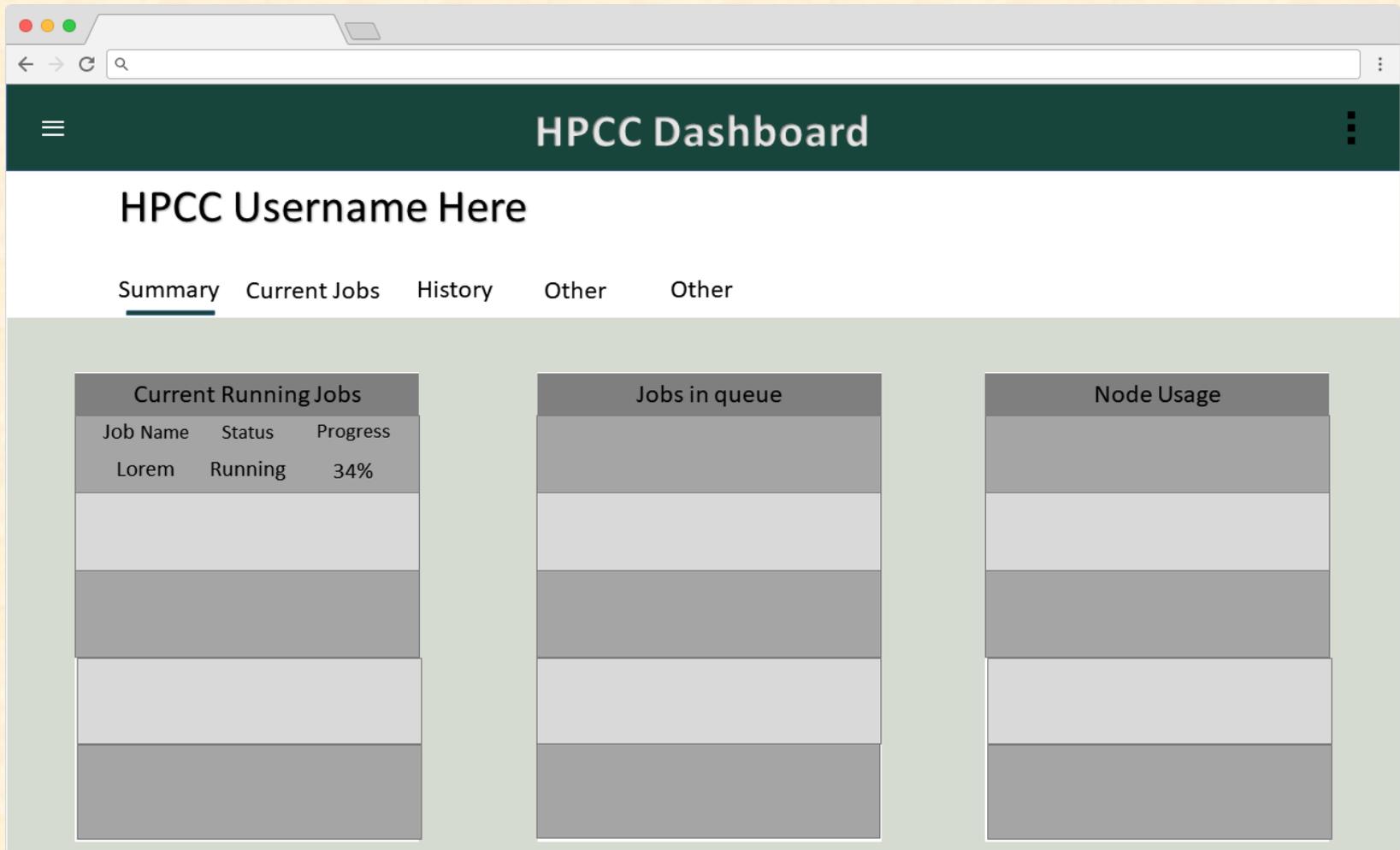
```
#!/bin/bash

# Job name:
#SBATCH --job-name=mothur_test
#
# Number of MPI tasks needed for use case:
#SBATCH --ntasks=8
#
# Processors per task:
#SBATCH --cpus-per-task=1
#
# Memory:
#SBATCH --mem-per-cpu=100M
#
# Wall clock limit:#SBATCH --time=30
#
# Standard out and error:#SBATCH --output=%x-%j.SLURMout

module purge
module load icc/2017.1.132-GCC-6.3.0-2.27 impi/2017.1.132 Mothur/1.40.3-Python-2.7.13 # Mothur MPI version
cd /mnt/research/common-data/Examples/mothur/ # note: regular users do not have write permission to this dir
mpirun -n $SLURM_NTASKS mothur batch.m
```



Screen Mockup: Administrator



Screen Mockup: Script Generator

The screenshot shows a web browser window displaying the HPC Dashboard. The dashboard has a dark green header with the text "HPC Dashboard" and a hamburger menu icon on the left and a vertical ellipsis on the right. Below the header, the main content area is titled "SLURM Script Generator". There are three tabs: "Resources" (which is underlined), "Constraints", and "Advanced".

The "Resources" tab contains several input fields with green question mark icons to their right:

- Wall time: ?
- Job name: ?
- Script path: ?
- Call Executable: ?

Below these fields, there are two rows of input fields:

- Nodes: Range Memory
- Cores: Parallel ?

To the right of the input fields is a large white box with the text "Preview of script will populate here". At the bottom of the form, there are two buttons: a teal "Preview" button and a green "Submit" button.



Screen Mockup: Playground

The screenshot displays a web browser window with the HPCC Dashboard. The dashboard has a dark green header with a hamburger menu on the left and the title 'HPCC Dashboard' on the right. Below the header, the user is logged in as 'HPCC Username Here'. A navigation bar contains links for 'Summary', 'Current Jobs', 'History', 'Test Ground' (which is underlined and active), and 'Other'. The main content area is light green and features a 'Select Job' dropdown menu currently set to 'Test_Job.sh'. Below this is a 'Begin Test' button. Two side-by-side panels compare 'Requested Resources' and 'Resources Utilized'. The 'Requested Resources' panel shows: Walltime: 000:00:10, Nodes: 4, Processors per Node: 1, and Memory: 4gb. The 'Resources Utilized' panel shows: Walltime: 000:00:10, Nodes: 8, Processors per Node: 4, and Memory: 16gb.

Requested Resources	Resources Utilized
Walltime: 000:00:10	Walltime: 000:00:10
Nodes: 4	Nodes: 8
Processors per Node: 1	Processors per Node: 4
Memory: 4gb	Memory: 16gb

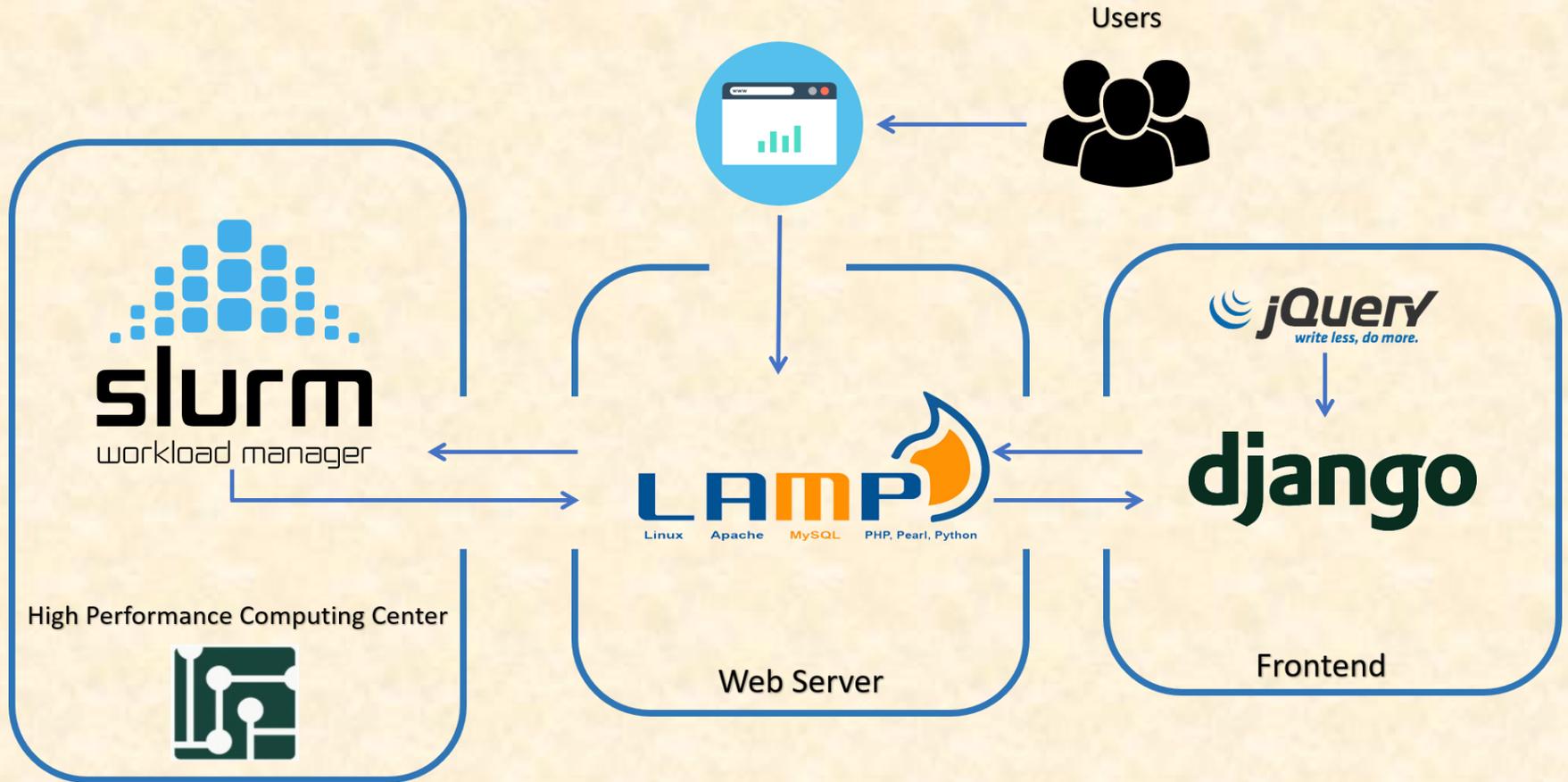


Technical Specifications

- Cluster Job Scheduler – Slurm
- Web Server – CentOS, Apache2, MySQL, Python
- Web full stack framework – Django
- Frontend UX - JavaScript with jQuery



System Architecture



System Components

- Hardware Platforms
 - The HPCC
 - Dell Servers
- Software Platforms / Technologies
 - OpenHPC
 - JQuery
 - Django
 - SLURM
 - MySQL Database



Risks

- **Simulating the HPCC**
 - Cluster setup has stalled due to hardware limitations.
 - Working with system admin on alternative cluster setup. Also looking into different workflows which will support script testing without cluster.
- **Connecting cluster to website**
 - Connect website to scheduler without a cluster
 - Prototype simple echo application to test/debug Slurm to Django interface.
- **Unsure how to implement admin privileges**
 - HPCC has various levels of administrator privilege for controlling user actions
 - Discuss with HPCC admins the hierarchy and organization of privileges
- **Don't know how to implement an isolated playground cluster**
 - Need a testing environment to see if a job utilizes appropriate resources
 - Examine similar testing environments. Stretch goal that isn't required.



Questions?

?

?

?

?

?

?

?

?

?

