

# Obelisk: Summoning Minions on an HPC Cluster

Dr. Peter Bui & Grant Wuerker

Department of Computer Science, UW – Eau Claire



## Purpose

In scientific research, having the ability to perform **rigorous** calculations in a **bearable** amount of time is an invaluable asset. Fortunately, the growing popularity of **distributed systems** at **universities** makes this a widely **accessible** resource. The purpose of **Obelisk** is to connect users and HPC systems through an **intuitive** web interface that **automates** the creation, submission, and retrieval of scientific computing tasks.



To someone that is familiar with Linux, parallel computing, and distributed systems, making a high performance computing (HPC) cluster do your bidding is relatively **straightforward**: you simply need to know the right commands! However, most people do not have this type of **specialized** knowledge, nor should they be expected to learn the seemingly **magical** incantations necessary to harness the power of multiple machines. Most people do not have the time or patience to learn these skills, indicating that the high performance computing (HPC) cluster has not been abstracted far enough.

## Design

An overview of **Obelisk** is shown in **Figure 1**. Rather than dealing with **arcane** Unix command shells and fighting daemons, Obelisk allows users to simply transcribe their computational goals and **summon minions** to perform their bidding via a responsive and easy-to-use web application.

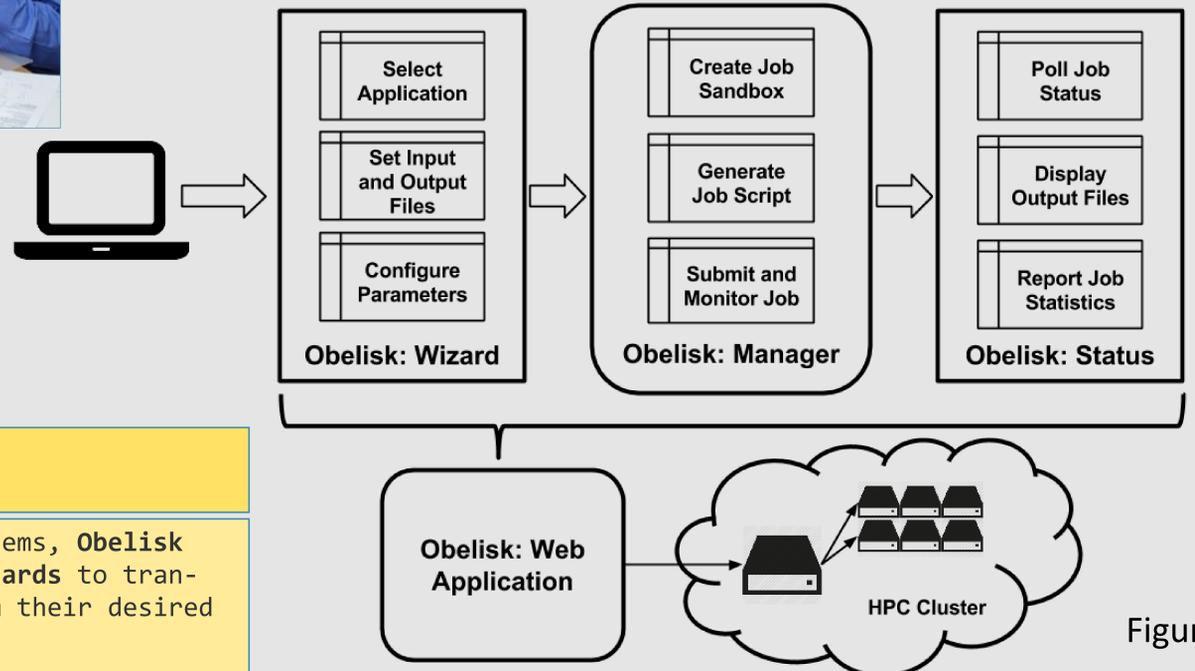
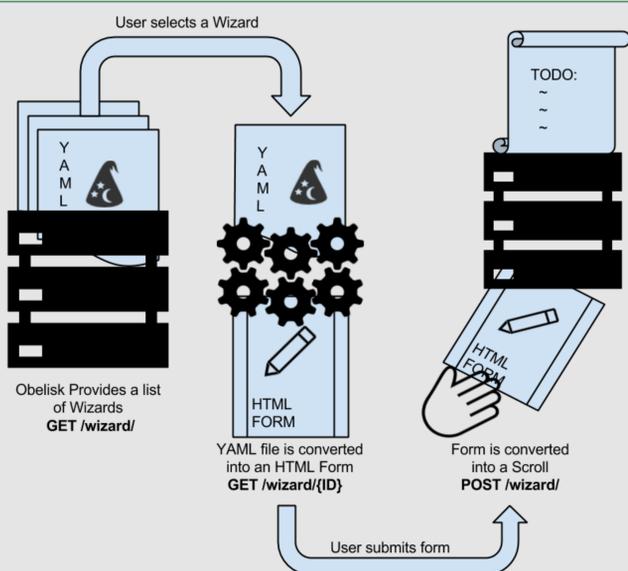


Figure 1.

## Design: Wizards, Scrolls, and Minions

In summary, in order to mask the **complexity** of traditional HPC systems, **Obelisk** provides an **intuitive** web interface that allows users to employ **Wizards** to transcribe **Scrolls**, which in turn are used to summon **Minions** to perform their desired computational tasks.

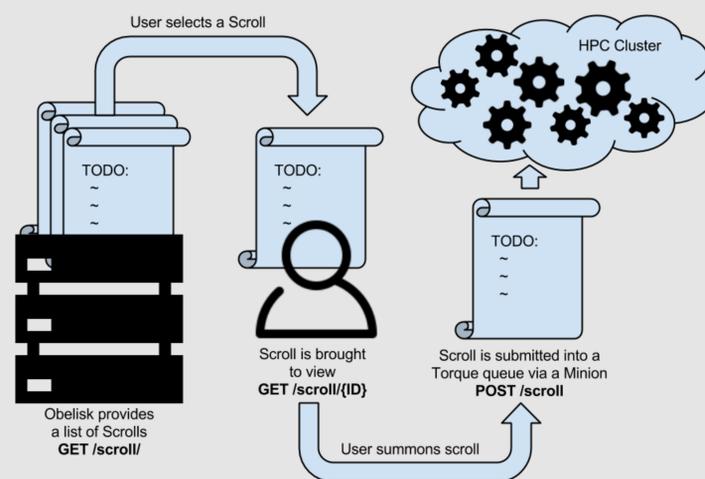
### Wizards



md5sum

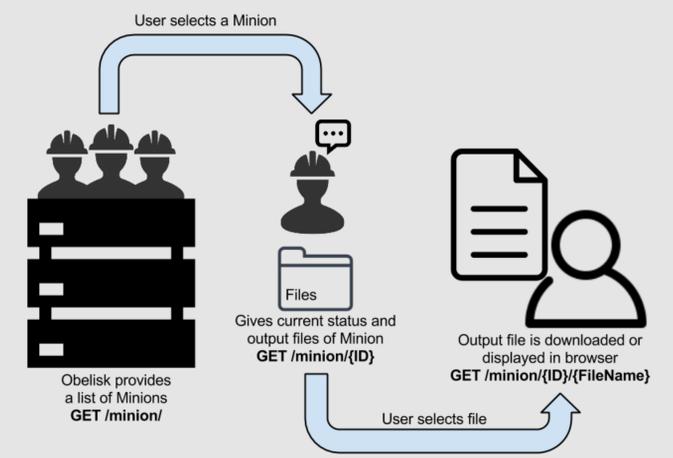
Name	md5sum
Command	md5sum
File	<input type="button" value="Choose File"/> No file chosen
StandardOutput	Output
StandardError	Error
Queue	batch
WorkingDirectory	
Modules	torque
Cores	1
Nodes	1

### Scrolls



```
My_Scroll
33 # Default working directory
34 #PBS -d
35
36 # Resource constraints
37 # nodes = Number of nodes
38 # gpus = Number of gpus per node
39 # walltime = Maximum run-time
40 #PBS -l nodes=1:ppn=1,walltime=24:00:00
41
42 # Standard output and Standard Error
43 #PBS -o Output
44 #PBS -e Error
45
46 # Modules
47
48 module load torque
49
50 # Debugging
51
52 env | sort > Environment
53
54 (echo "PBS Job: $PBS_JOBNAME" ;
55 echo "PBS Job ID: $PBS_JOBID" ;
56 echo "PBS Host: $PBS_NODEID" ;
57 echo "PBS Working Directory: $PBS_O_WORKDIR" ) > PBS
58
59 # Main Execution
60
61 md5sum sumofull.exe
62
```

### Minions



id	Job id	Path	Scroll	Status	Start time	Stop time
2	15791.head.bgsc.cluster	/inventory/minions/2	my_scroll	E	4 seconds ago	N/A

Files

Environment

Error

Output

PBS

```
scroll.sh
sumofull.exe
```

## Evaluation

We have tested the functionality of **Obelisk** and verified that it in fact works correctly. We will continue to add more features and fix bugs, but for now we will focus on getting **more Wizards** available for demonstration and use. In the future, we will work with domain scientists and have science faculty and students utilize our project to **summon minions** for class assignments and **research** experiments. In doing so, we will collect data that allows us to determine the effectiveness of **Obelisk** and find ways to improve the user experience. To test **Obelisk**, we took applications that are typically ran by faculty and students on our local HPC cluster, and converted them into **Obelisk Wizards**. By doing so, we were able to verify that **Obelisk** is applicable to their computational needs and works properly.