PBSWeb: A Web-based Interface to the Portable Batch System

George Ma and Paul Lu

Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
{george|paullu}@cs.ualberta.ca

http://www.cs.ualberta.ca/~paullu

With support from: C3, Multimedia Advanced Computational Infrastructure (MACI), Canada Foundation for Innovation (CFI), Natural Sciences and Engineering Research Council of Canada (NSERC)

Overview

1. Introduction to PBSWeb
2. PBSWeb Architecture
3. PBSWeb Implementation
4. Concluding Remarks and Future Work

PBSWeb

On the surface, PBSWeb is a Web-based interface to the Portable Batch System (PBS).

Below the surface, PBSWeb adds (or will add) support for:

1. Greater ease-of-use
2. Remote access of resources: metacomputing, grid computing
3. Workflow management

NOTE: We are **not** affiliated with PBS or Veridian Systems.

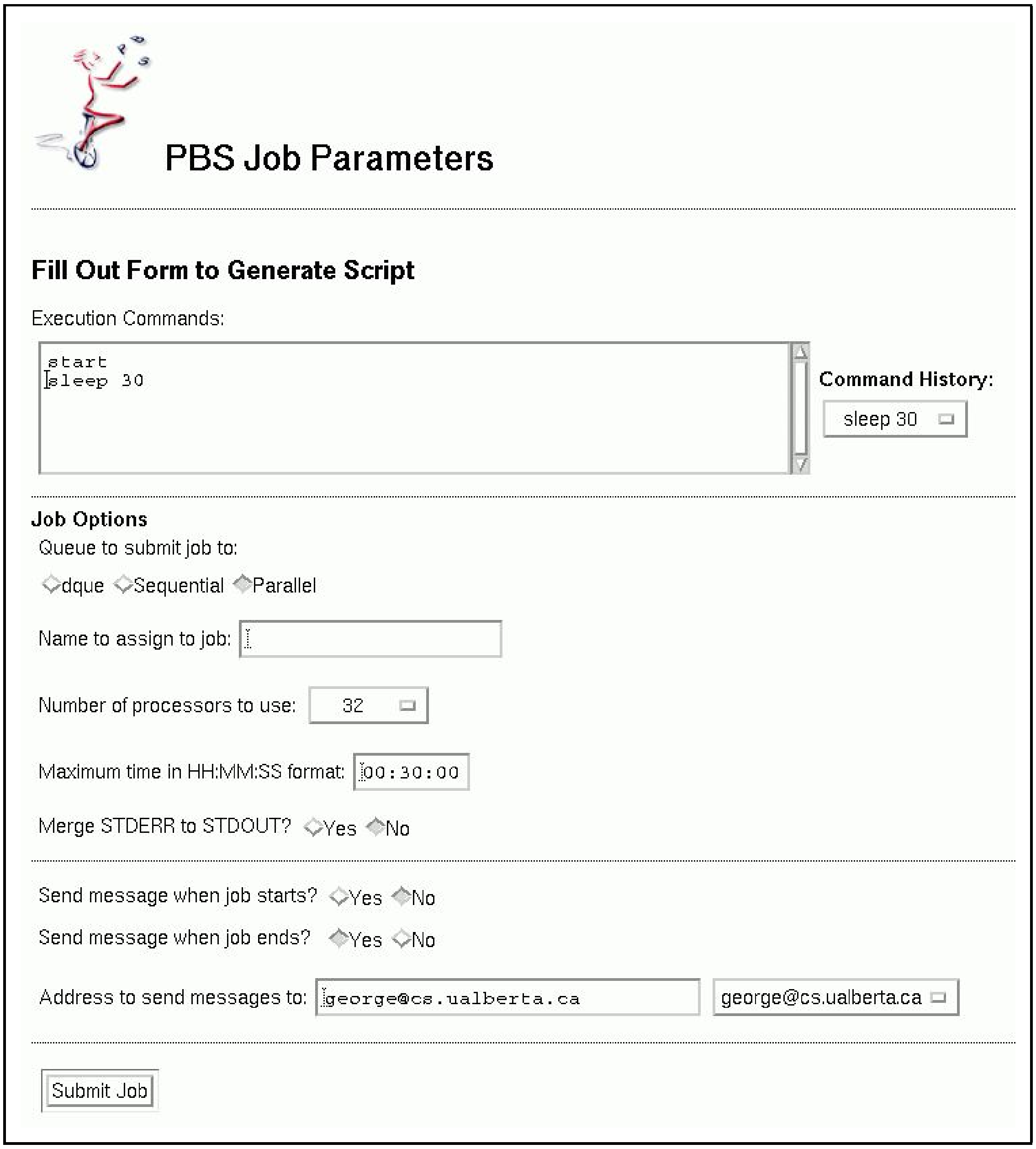
PBS Job Scripts

Batch schedulers can be difficult to use.

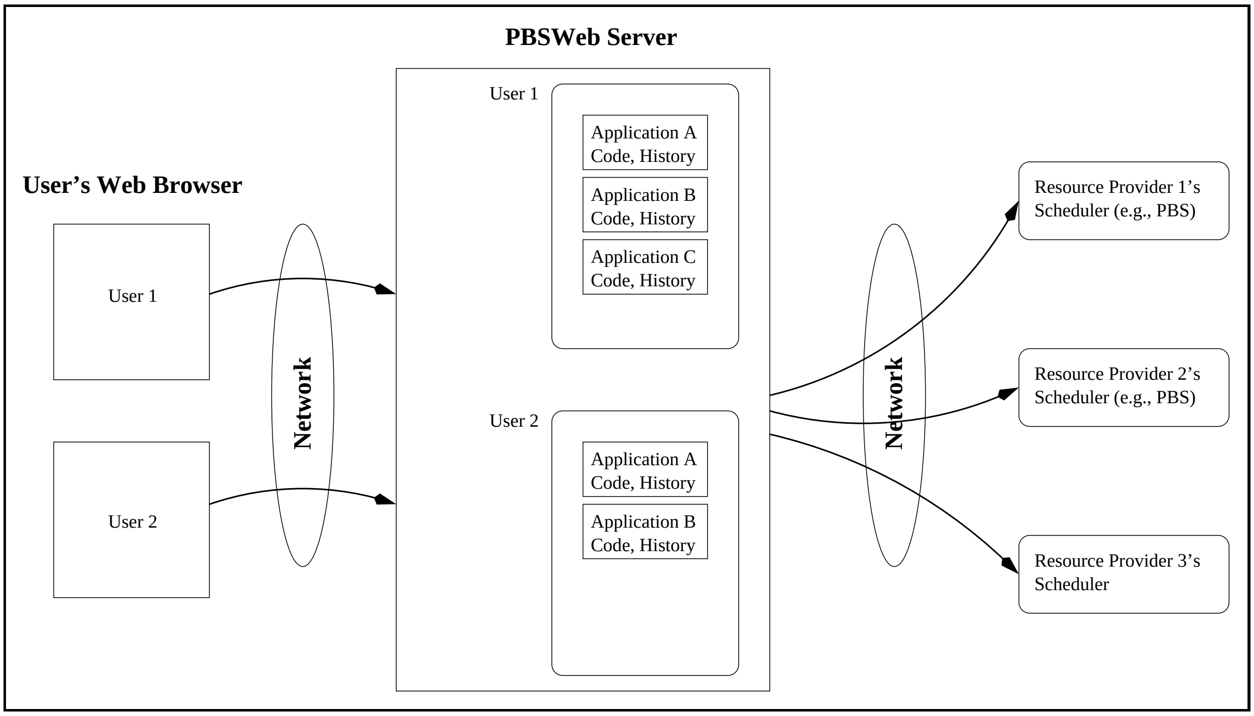
```
# Do all this under tcsh
#PBS -S /bin/tcsh

# Use default queue
#PBS -q dque

# How many CPUs?
#PBS -l ncpus=4

# Name of job
#PBS -N TestCharron

# CPU time limits
#PBS -l walltime=00:20:00
#PBS -l cput=02:40:00

# Mail is to be sent for abort, begin, end conditions
#PBS -m abe
#PBS -M someemail@cs.ualberta.ca

mpirun -np 4 parallelsort.psrs
```

Various commands: `qsub`, `qstat`, `qdel`, `qhold`, `qalter`

A GUI for PBS

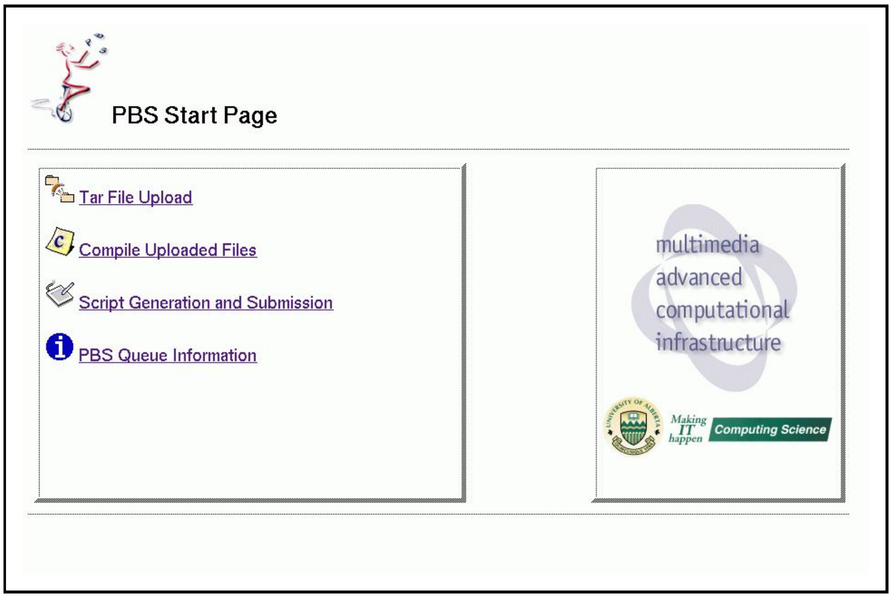

PBS Job Parameters
Fill Out Form to Generate Script
Execution Commands:
start
sleep 30
Command History:
sleep 30
Job Options
Queue to submit job to:
dque Sequential Parallel
Name to assign to job:
Number of processors to use:
32
Maximum time in HH:MM:SS format:
00:30:00
Merge STDERR to STDOUT? Yes No
Send message when job starts? Yes No
Send message when job ends? Yes No
Address to send messages to:
george@cs.ualberta.ca
george@cs.ualberta.ca
Submit Job

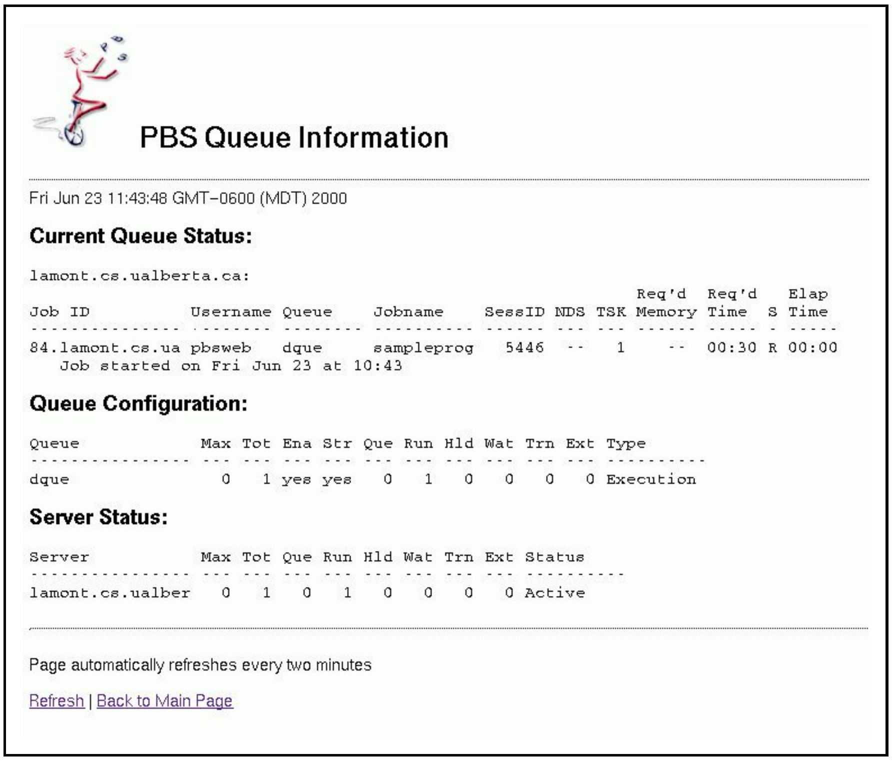
PBSWeb Architecture



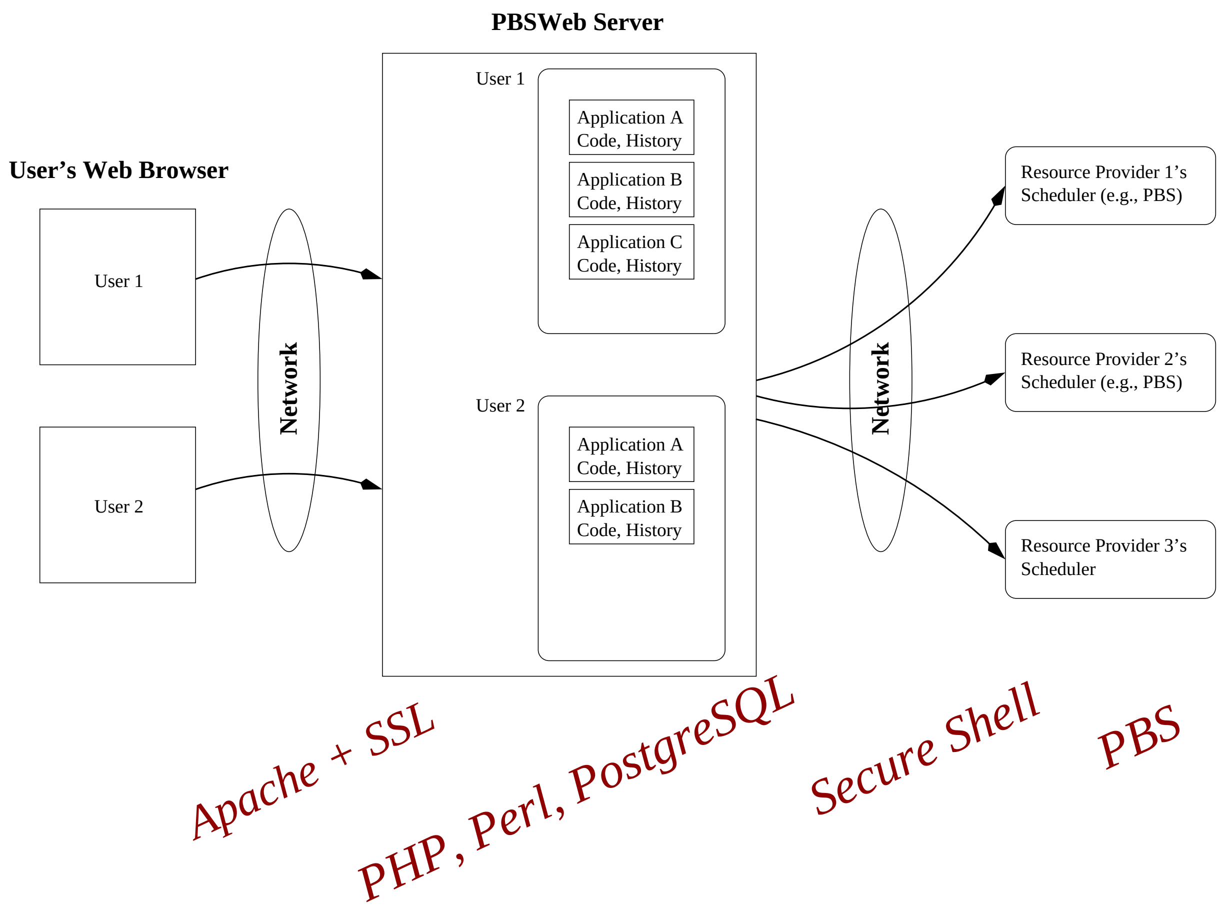
1. GUI is Web-based
2. PBSWeb Server maintains a history and source code repository
3. Layered on top of resource scheduler (e.g., PBS, LSF)

PBSWeb: Screenshots





Implementation



Concluding Remarks

Our approach:

1. Web-based GUI
   - improved ease-of-use
   - easy remote access, platform independence
2. History Database
   - per-user, per-application
   - easier to manage experiments and workflow
3. Source Code Repository
   - easier to share among experimenters
   - metacomputing, grid computing

PBSWeb is an on-going project. Contact us for the software.