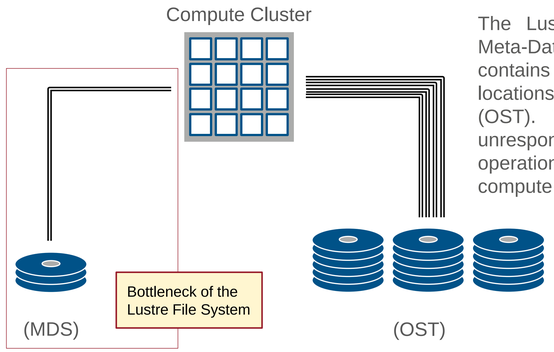
Leveraging Neural Networks to Predict Job IO in HPC Systems

Student: Michael Wyatt[1,2]
Advisors: Todd Gamblin[2], Adam Moody[2], Michela Taufer[1]
[1]University of Delaware, [2]Lawrence Livermore National Laboratory

Challenges in HPC

- Application performance is reliant on the cluster, other running jobs, and availability of *shared resources*
- The Lustre File System is a shared resource which may become unresponsive if several IO-intensive jobs are running simultaneously
- Knowing the number of *metadata operations* an application will use during its runtime allows for smarter scheduling



The Lustre File System contains a Meta-Data Server (MDS) which contains information about object locations on the Object Storage Targets (OST). The single MDS can become unresponsive when too many IO operations are requested by the compute servers.

Goal: Leverage Neural Networks to extract features and patterns from job submissions and predict runtime and metadata operations.

Limits of Prior Techniques

- Standard machine learning algorithms use simple features extracted from job scripts with regular expressions
  - Fail to utilize features embedded in irregular patterns
- A Neural Network is able to detect features in irregular patterns

Regular patterns can be parsed by Regular Expressions

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -l partition=cab
#PBS -W x=partition:cab
#PBS -N EX_nu
##
#PBS -q pbatch
#PBS -e run.err

#PBS -A mixapp
#PBS -l nodes=1:ppn=8

#PBS -l walltime=02:55:00
```

```
#!/bin/bash
#PSUB -r R2D2-C3PO-BB8
#PSUB -eo
#PSUB -o psub.out
#PSUB -ro
#PSUB -d 0
#PSUB -tM 9h
#PSUB -ln 12
#PSUB -b gsmisc
#PSUB -c cab
#PSUB -me   # send email when job completes
#PSUB -x    # pass-through environment variables
#PSUB  # no more embedded options
```

Irregular patterns cannot be parsed by Regular Expressions

```
###pdebug

PBS_PWD="$(pwd)"
cd "${PBS_O_WORKDIR}"

#run_mpirun()
#{

. /usr/local/tools/dotkit/init.sh
use openmpi-intel
mpirun -np 8 /k/k22/frodo1/bin/mpi_misc < big_bird.in >>
cookiemonster.out

#}
```

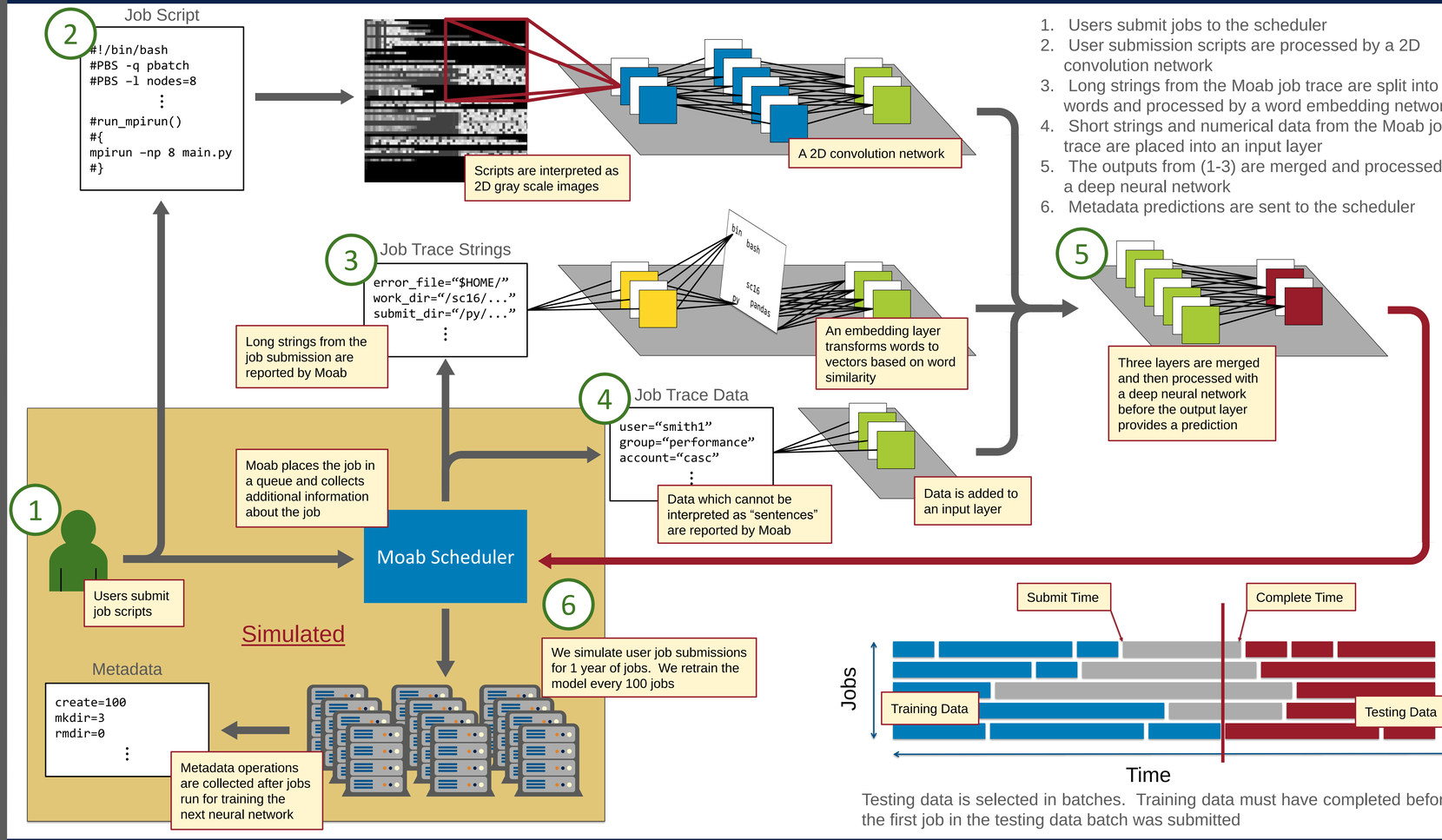
```
echo job ID = $PSUB_JOBID
date
echo
set echo

#cd /p/lscratche/boots99/onering/Eyes_Wide/shut+0
cp INCAR1 INCAR
srun -N12 -n192 /usr/tools/space/2001
cp INCAR2 INCAR
/usr/tools/drinks/martini -d #-check
srun -N12 -n192 /usr/tools/drinks/mix --shake --no-stir

wait
exit
```

Examples of two job submission scripts. There exist regular (top half) and irregular (lower half) patterns in the scripts. The irregular patterns cannot be reliably parsed with regular expressions.

Neural Network Workflow for IO Prediction



1. Users submit jobs to the scheduler
2. User submission scripts are processed by a 2D convolution network
3. Long strings from the Moab job trace are split into words and processed by a word embedding network
4. Short strings and numerical data from the Moab job trace are placed into an input layer
5. The outputs from (1-3) are merged and processed by a deep neural network
6. Metadata predictions are sent to the scheduler

Testing data is selected in batches. Training data must have completed before the first job in the testing data batch was submitted

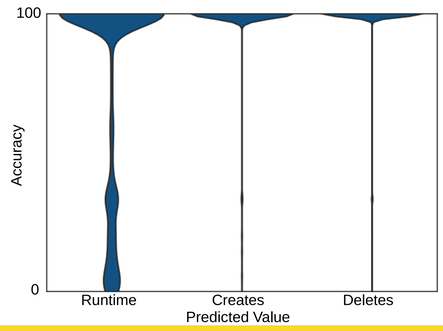
Predictions and Observed Accuracy

- Accuracy Metric:

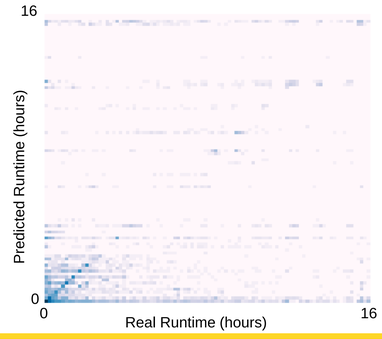
$$1 - \frac{|true - pred|}{\max(true,\ pred)}$$

- Batch Parameters:
  - 100 Testing Samples
  - 500 Training Samples
- Job Feature Accuracy:
  - Runtime: 75%
  - Creates: 96%
  - Deletes: 96%

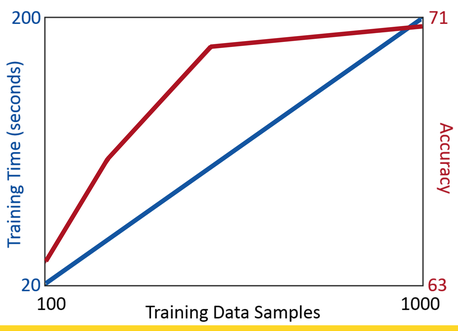
Accuracy distributions for predicted features



Predicted runtime versus real runtime



Neural network performance over training size



This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.
IM#: LLNL-POST-698218