NERSC Job Script Generator

Robin Shao[1)], Thorsten Kruth[2)], Zhengji Zhao[3)]

[1)] University of California, Berkeley

[2)] NVIDIA, Inc

[3)]National Energy Research Scientific Computing Center (NERSC),
Lawrence Berkeley National Laboratory

HUST-22: 9th International Workshop on HPC User Support Tools
Held in Conjunction with SC22, Dallas, Texas
November 14, 2022

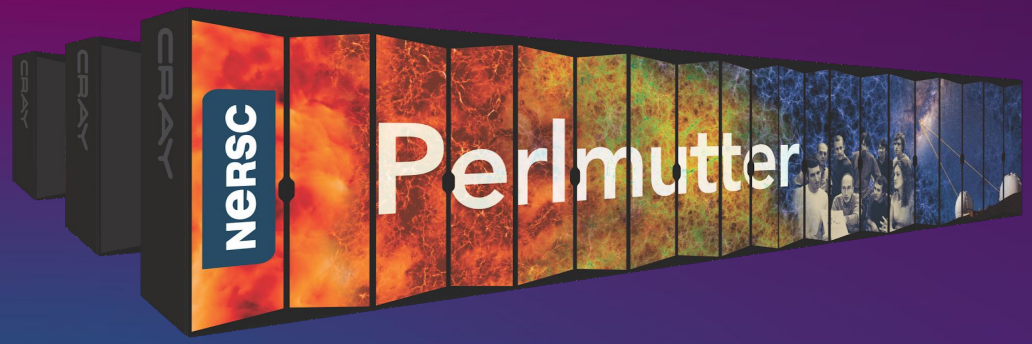
Overview

1. Motivation
2. Implementations
3. Features Supported
4. Job Script Examples
5. Future Work
6. Demo

NERSC Systems

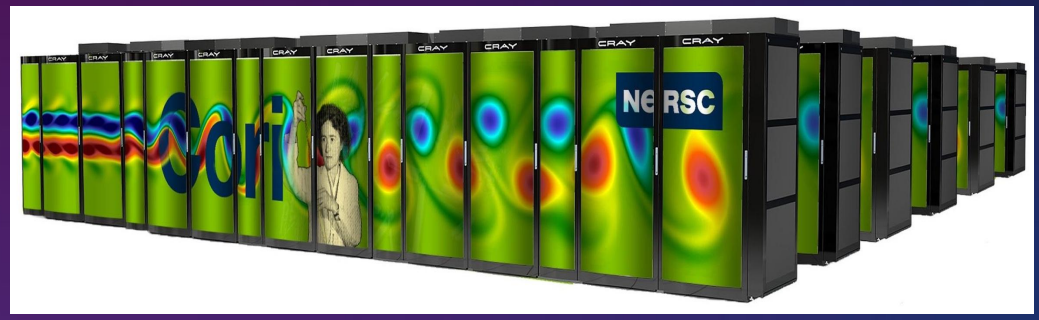
Perlmutter

1,500 NVIDIA A100x4 accelerated nodes
3,000 AMD dual-socket “Milan” CPU nodes
384 TB (CPU) + 240 TB (GPU) memory
HPE Cray Slingshot high speed interconnect
140 PF Peak
Batch system is **Slurm**



Cori

9,600 Intel Xeon Phi “KNL” manycore nodes
2,000 Intel Xeon “Haswell” nodes
700,000 processor cores, 1.2 PB memory
Cray XC40 / Aries Dragonfly interconnect
30 PF Peak
Batch system is **Slurm**



Motivation

- NERSC jobs are **complicated**:

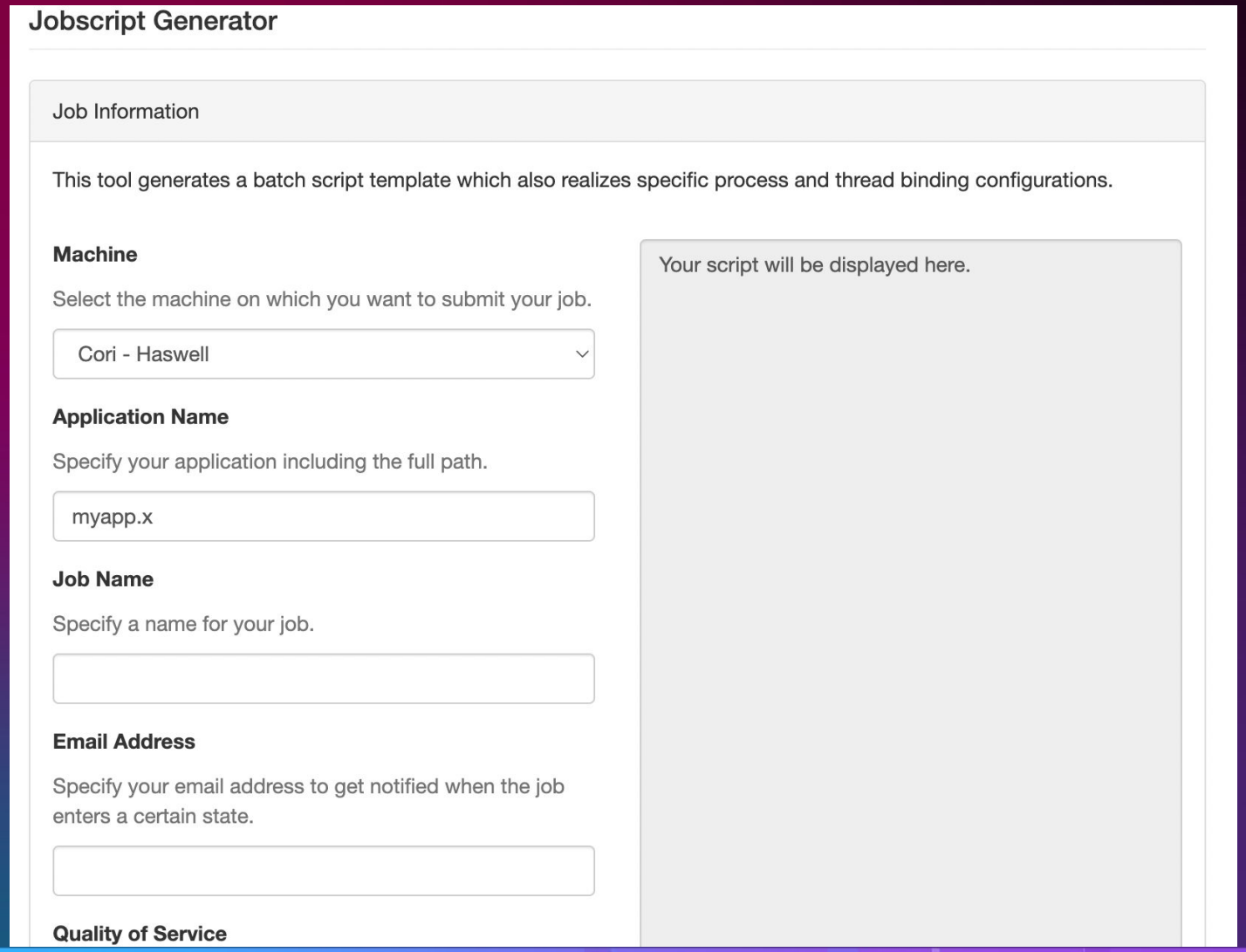
  NERSC supports diverse production workloads

  (complicated queue structure with various resource limits and priorities)

  => challenging for users to generate proper job scripts to optimally use the systems

- We developed Job script generator, a web tool, to help users
  - generate proper job scripts
  - learn how the batch system works @ NERSC

Job Script Generator



https://my.nersc.gov/script_generator.php

https://bit.ly/3De087S (development version)

Implementation Overview

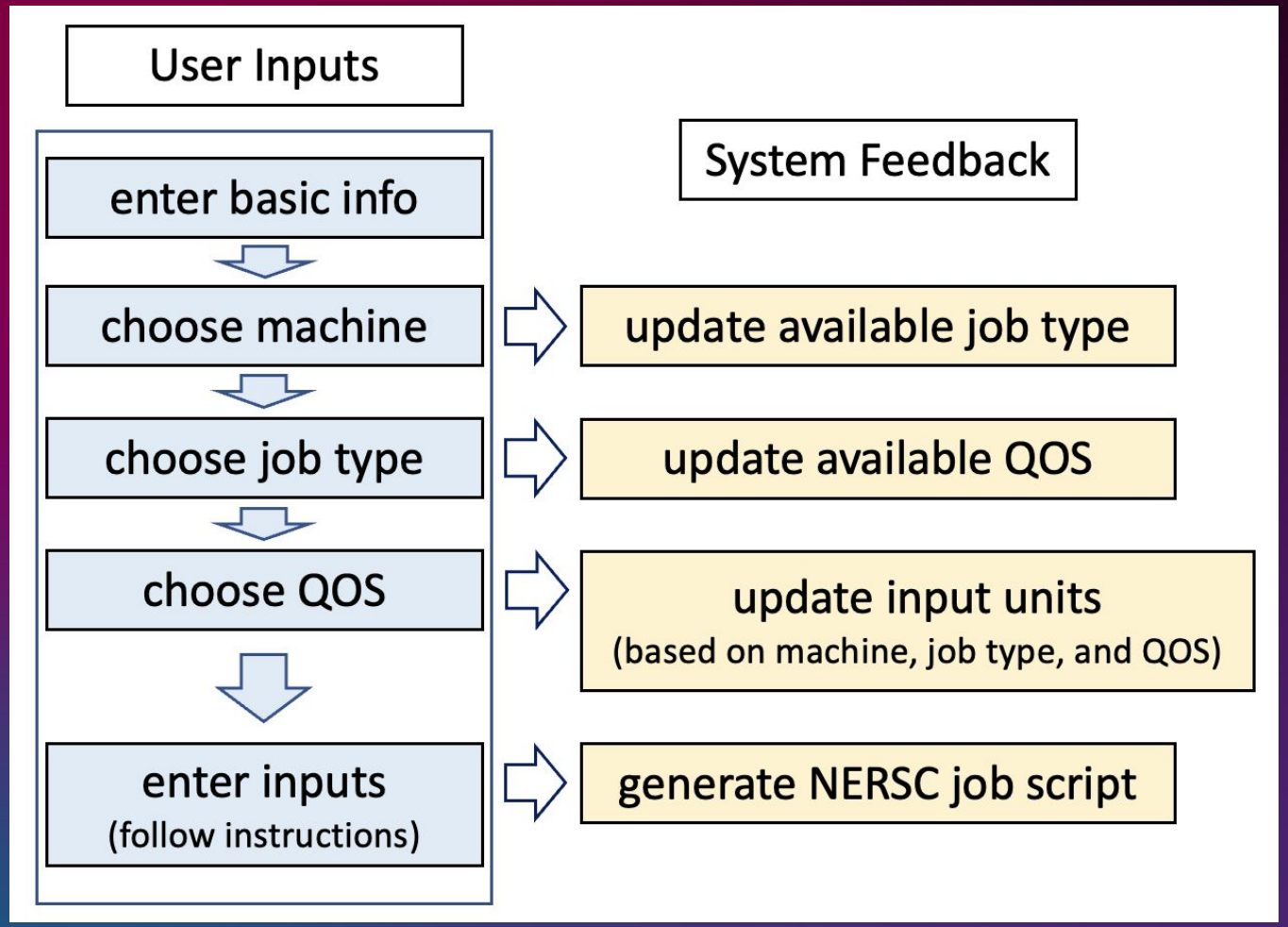
language: **JavaScript, PHP**

under the framework of the MyNERSC website

two source files (~ 1500 lines of code):

- **script_generator.php:**
  - a web user interface
- **script_generator.js:**
  - process the user inputs
  - dynamically update the user interface
  - generate the job script
  - display the generated job script on the web

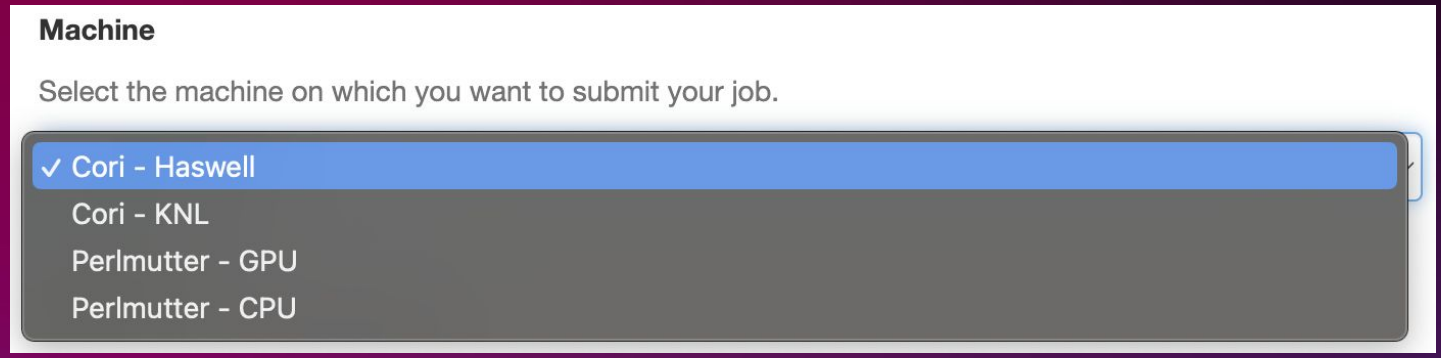
Flow Chart


User Inputs
System Feedback
enter basic info
choose machine
update available job type
choose job type
update available QOS
choose QOS
update input units
(based on machine, job type, and QOS)
enter inputs
(follow instructions)
generate NERSC job script

Features Supported: Systems and Job Types

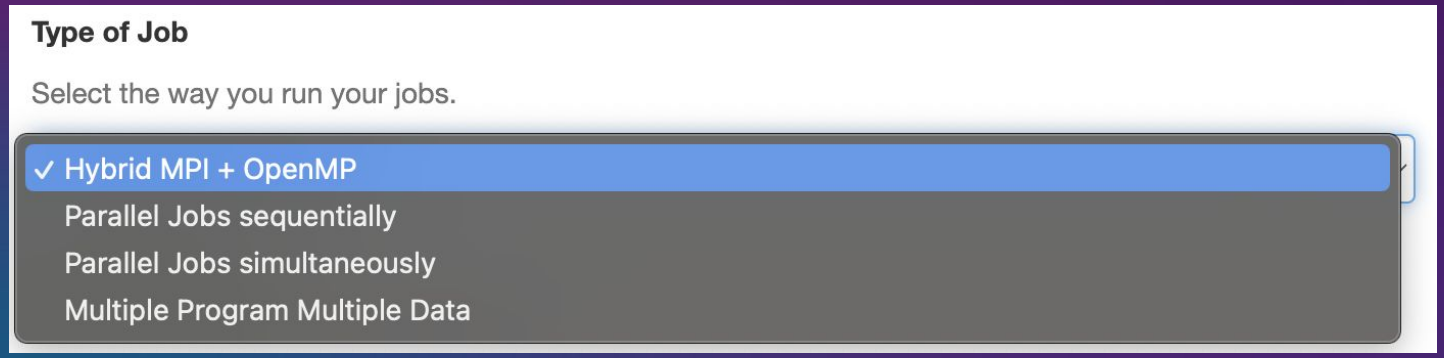
Machine:

Cori Haswell, Cori KNL, Perlmutter CPU, Perlmutter GPU



Type of Jobs:

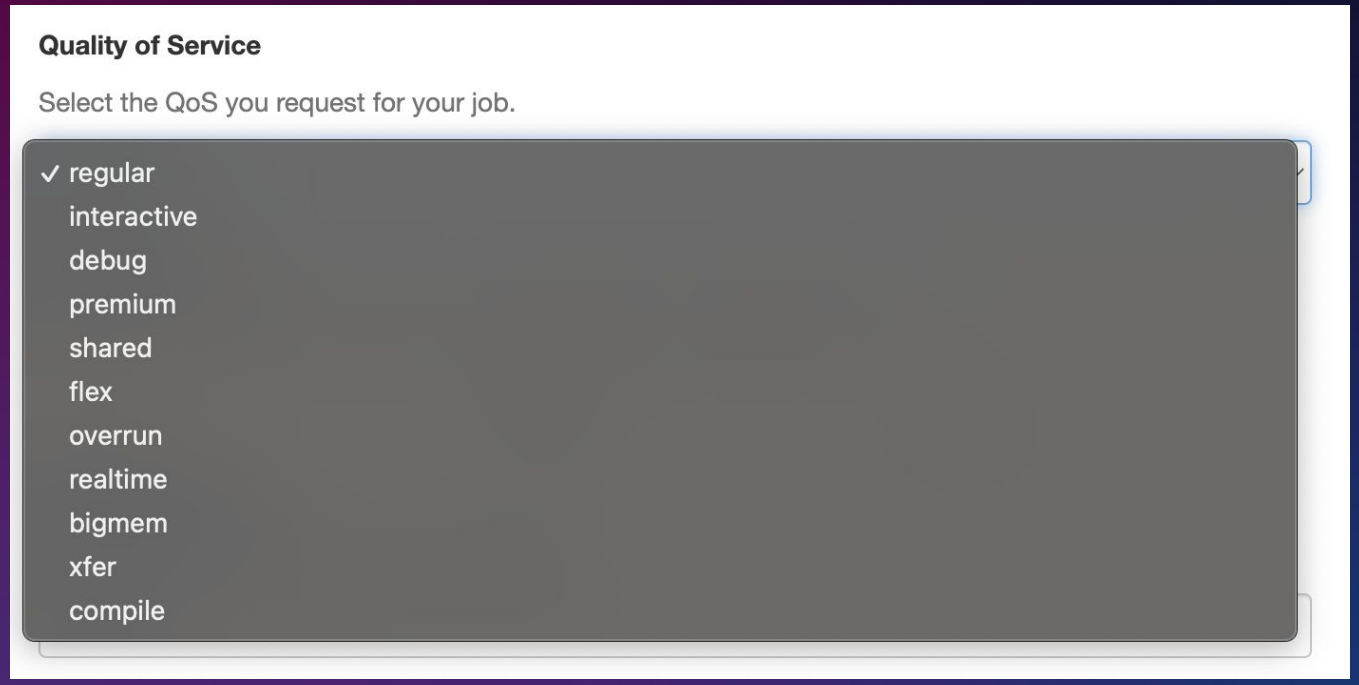
Hybrid MPI + OpenMP, Parallel Job sequentially & simultaneously, Multiple Program Multiple Data



Features Supported: QOS

When a QOS is selected, the instruction sections for the input units will be updated with the associated QOS resource limits, e.g., max walltime, max nodes, etc

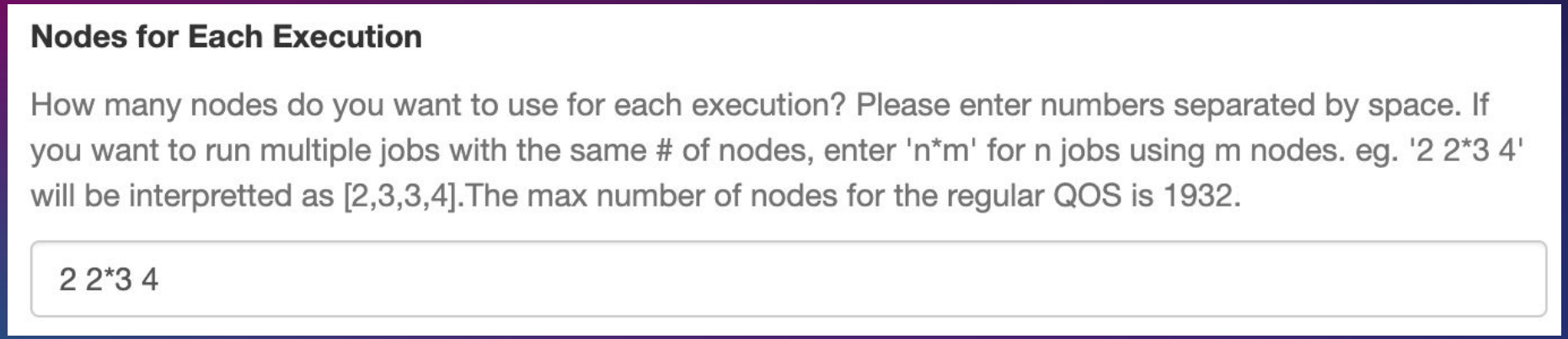
All the selections needed are done after choosing the QOS. The rest of the webpage will be updated based on the machine, job type, and QOS selected.



Features Supported: User Experience

- all the input units are instantly changed to match users' choices
- instructions for all the input units guiding users to make the right specifications for the jobs
- carefully design the input units
  - set default values
  - allow shortcuts for multiple input of same value

    (for parallel jobs, users may want to use the same spec for multiple jobs)



Job Script Example 1: MPI + OpenMP

- machine: Perlmutter CPU
- type: Hybrid MPI + openMP
- QOS: regular
- number of node: 4
- basic thread binding
- process per node: 16
- threads per process: 8

(default value for other fields)

```
#!/bin/bash
#SBATCH -N 4
#SBATCH -C cpu
#SBATCH -q regular
#SBATCH -J Example1
#SBATCH --mail-user=eric@lbl.gov
#SBATCH --mail-type=ALL
#SBATCH -t 00:30:00

#OpenMP settings:
export OMP_NUM_THREADS=8
export OMP_PLACES=threads
export OMP_PROC_BIND=true

#run the application1:
srun -n 64 -c 16 --cpu_bind=cores myapp.x
```

Job Script Example 2: Running Multiple Parallel Jobs Simultaneously

- machine: Perlmutter GPU
- type: Parallel Jobs simultaneously
- QOS: regular
- Nodes for each Execution: 2*4
- Processes per Node: 1 2
- threads per process: 1 2
- project: m9999_g
  (default value for other fields)

```
#!/bin/bash
#SBATCH -N 8
#SBATCH -C gpu
#SBATCH -G 32
#SBATCH -q regular
#SBATCH -J Example1
#SBATCH --mail-user=eric@nersc.gov
#SBATCH --mail-type=ALL
#SBATCH -t 00:30:00
#SBATCH -A m9999_g

#OpenMP settings:
export OMP_NUM_THREADS=1

#run the application1:
srun -N 4 -n 4 -c 128 --cpu_bind=cores -G 4 --gpu-bind=single:1
myapp1.x &

#OpenMP settings:
export OMP_NUM_THREADS=2

#run the application2:
srun -N 4 -n 8 -c 64 --cpu_bind=cores -G 8 --gpu-bind=single:1
myapp2.x &

wait
```

Job Script Example 3: Checkpoint/Restart with Flex QOS

User inputs:

- machine: Cori Haswell
- QOS: flex
- desired walltime: 96 hours
- tab: For Automatic Job Resubmission
- checkpoint command: ckpt_mana

Variable-time-job script:

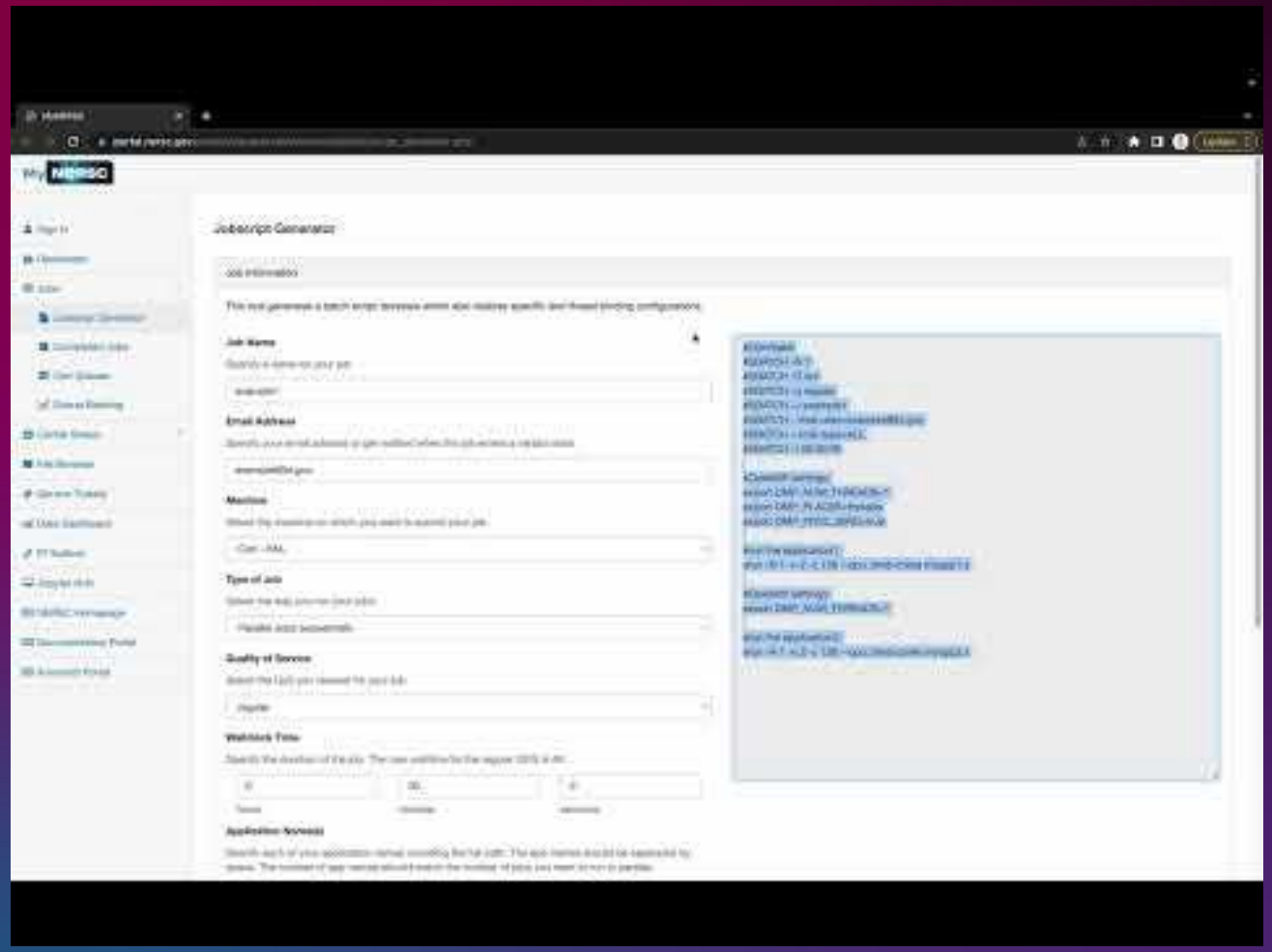
- Splits a long running jobs into multiple shorter ones
- Automatically resubmits the pre-terminated jobs until the accumulated runtime reaches the desired runtime (can be of any length!) or the job completes earlier.
- The flex QOS is designed to promote C/R adoption among NERSC users with a large charging discount (75%). The required minimum time is 2 hours.

```
#!/bin/bash
#SBATCH -N 2
#SBATCH -C haswell
#SBATCH -q flex
#SBATCH -t 48:00:00
#SBATCH --time-min=02:00:00
#SBATCH --comment=96:00:00
#SBATCH --signal=B:USR1@600
#SBATCH --requeue
#SBATCH --open-mode=append

#checkpointing once every hour
module load mana nersc_cr
mana_coordinator -i 3600

#checkpoint/restart jobs
if [[ $(restart_count) == 0 ]]; then
   export OMP_NUM_THREADS=1
   #other OMP settings are omitted

   srun -n 64 -c2 --cpu_bind=cores mana_launch myapp.x &
elif [[ $(restart_count) > 0 ]] && [[ -e dmtcp_restart_script.sh ]]; then
   srun -n 128 -c 4 --cpu_bind=cores mana_restart &
else
   echo "Failed to restart the job, exit"; exit
fi

# requeueing the job if remaining time >0
ckpt_command=ckpt_mana    #additional checkpointing before pre-emption
requeue_job func_trap USR1

wait
```

Future Work

- test the new extension thoroughly and deploy it in production
- plan to add support for more job types
- add job submission feature to the web using the Superfacility APIs
- ongoing efforts to generate “smart” job scripts by utilizing the parallel scaling data for top applications

Q&A



Acknowledgement

- This work was supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231, and used the resources of National Energy Scientific Computing Center (NERSC) at the Lawrence Berkeley National Laboratory.
- Shao thanks the Computing Research Summer Student Program at NERSC/LBNL
- Richard Gerber, Kevin Got, Helen He, and other NERSC staff for valuable feedback

Thank you for your time!

Job Information

This tool generates a batch script template which also realizes specific and thread binding configurations.

Job Name

Specify a name for your job.

Email Address

Specify your email address to get notified when the job enters a certain state.

Machine

Select the machine on which you want to submit your job.

Cori - Haswell

Cori - Haswell
Cori - KNL
Perlmutter - GPU
Perlmutter - CPU

Type of Job

Select the way you run your jobs.

Hybrid MPI + OpenMP

Hybrid MPI + OpenMP
Parallel Jobs Sequentially
Parallel Jobs Simultaneously
Multiple Program Multiple Data

Quality of Service

Select the QoS you request for your job.

regular

Wallclock Time

Specify the duration of the job. The max walltime for the regular QOS

0 hours | 30 minutes | 0 seconds

regular
interactive
debug
premium
shared
flex
overrun
realtime
bigmem
xfer
compile

Application Name

Specify your application including the full path.

myapp.x

Number of Nodes

How many nodes are used? The max number of nodes for the regular QOS is 1932.

1

Basic Thread Binding | Advanced Thread Binding

Processes per Node

How many processes, e.g. MPI ranks per node, do you want to use? There are 32 cores per Cori Haswell node; 2 hardware threads per core.

1

Threads per Process

How many threads per process do you want to use? Note that SLURM currently only supports jobs which use the same number of threads per process.

1

Generate Script

NERSC Background

- **National Energy Research Scientific Computing (NERSC)**
  - provides **High Performance Computing and Storage facilities** and support for research sponsored by the U.S. Department of Energy (DOE)
  - NERSC currently using are **Cori** and the relatively new **Perlmutter** system
    - **Cori:** NERSC-8 system containing Haswell nodes and KNL nodes
    - **Perlmutter:** the latest NERSC-9 system made phase 1 delivery in late 2020

Contribution

As work assistance tool:

help users to generate proper and efficient job scripts

As educational tool:

help new users to learn how the batch system works @ NERSC

NERSC Job Script Background

- Slurm: the resource management and scheduling system to submit job scripts or start interactive jobs @ NERSC
- NERSC's environment is configured to support diverse workload including high-throughput serial tasks, full system capability simulations and complex workflows
- Use job Script Generator to generate the correct directives, arguments, and process affinity settings