University of
Southampton

Sustainable SLURM and greener HPC

Alister Boags
Senior Research Systems Engineer
University of Southampton – HPC - iSolutions

University of Southampton

Is HPC green?

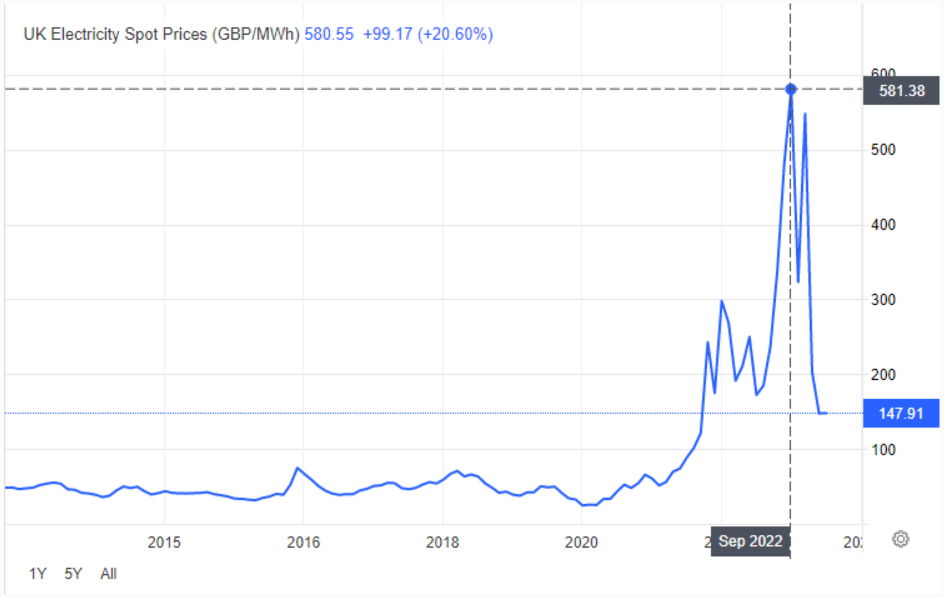
- HPC is generally a dirty business; we take large numbers of power-hungry servers that are expensive to make, cluster them together and attempt to keep them cool.
- These servers have an ingrained carbon cost to produce and maintain.
- Regardless of on-premises or cloud clusters, the machines are still *somewhere.*
- Is it improving?
- If we're measuring this, where are we truncating our measurements?

Energy costs for a cluster

- Electricity – naturally the power that we put in to run the machines.
- Cooling – liquid/air and the necessary adjustments for seasonal weather.
- Infrastructure – switches, lights, local machines and data center peripherals.
- Deliveries – building/delivering nodes or parts for repairs.
- Staff – how big is your team? how do they get to work, what's the carbon footprint of people, relative to the machines.

Electricity!

- Electricity prices have been volatile in recent months.
- This in turn causes institutions to double down on their sustainability agendas.
- Being greener is always a good thing, but in recent times there is a financial incentive.



Emissions

- Regarding emissions let’s consider carbon:

Build and install

Code optimisations

Running configuration

SLURM and Power

- SLURM has ways to report and manage the power usage of your cluster.
- Most important there are the SLURM power monitoring plugins and the SLURM Power functionality.
- Clusters/nodes can be configured to be powered off when not in use.
- Adds overheads to jobs – but saves power.

Slurm Workload Manager - Slurm Power Saving Guide (schedmd.com)

Power Monitoring and Management

- We are not limited to controlling the power state, but also monitoring/managing the power
- We have been pulling the power information into flat text files for a few months now
- This can be done with some scripting and usage of the RAPL plugin for power states
- # POWER MONITORING

```
AcctGatherEnergyType=acct_gather_energy/rapl
AcctGatherNodeFreq=30
```

[Slurm Workload Manager - Slurm Power Management Guide (schedmd.com)]

Power States

- When a node idles for a period, it is powered off.
- How this is accomplished is entirely within the control of the admins.
- Nodes are not presented as DRAIN/DOWN or unavailable for usage.
- Cloud HPC setups, such as Microsoft CycleCloud already use this functionality.
- Unwise to deploy to a whole estate without testing or sanity checking with the data center.

Installation and making it work

- Okay so you want SLURM power;
  - This isn't a default setting in a basic configure of SLURM
  - For us we had to recompile
  - If you attempt to active this feature without the recompile slurmctld will crash
  - As per usual you need to edit the slurm.conf
    - Which hardware benefits from this?
    - How do we define 'idle'?
    - How quick is your spin up?
    - What mechanism do you want to use?

- Build and configuration points to consider
- ./configure --build=x86_64-redhat-linux-gnu --host=x86_64-redhat-linux-gnu --disable-dependency-tracking --prefix=/local/software/slurm/22.05.2 --sysconfdir=/etc/slurm --localstatedir=/var --sharedstatedir=/var/lib --enable-pam --enable-multiple-slurmd **--with-json --with-yaml**
-
```
# POWER SAVING
SuspendTime=3600
SuspendRate=1
ResumeRate=1
SuspendProgram=/mainfs/home/slurm/power_save/suspendprogram.sh
ResumeProgram=/mainfs/home/slurm/power_save/resumeprogram.sh
SuspendExcParts=batch,largejobs,serial,scavenger,gpu,gtx1080,lyceum,amd,relgroup,worldpop,hydrology,veils,ngcm,ecsstaff,ecsall,ecsstudents,enginframe
BatchStartTimeout=1200
```

SLURM user settings

- SuspendProgram=suspendprogram.sh
  ResumeProgram=resumeprogram.sh
- SlurmUser=slurm
- Make sure you chmod +x these scripts!

```
#!/bin/bash
# Example SuspendProgram
source /mainfs/home/slurm/.bashrc
echo "`date` Resume invoked $0 $*"
>>/mainfs/home/slurm/power_save/power_saving.log
hosts=`scontrol show hostnames $1`
for host in $hosts
do
  rpower $host on
done
```

```
#!/bin/bash
# Example SuspendProgram
source /mainfs/home/slurm/.bashrc
echo "`date` Suspend invoked $0 $*"
>>/mainfs/home/slurm/power_save/power_saving.log
hosts=`scontrol show hostnames $1`
for host in $hosts
do
   rpower $host off
done
```

Considerations

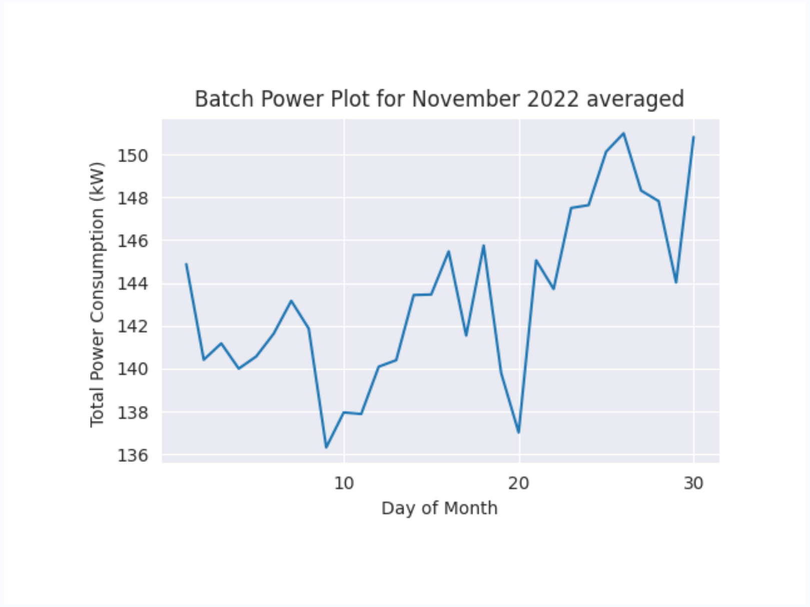
- We're currently excluding a lot of hardware from this setup.
- As we've piloted it, and it is successful, is it suitable for everyone?
  - What's your uptime like?
  - What's the power cost of node?
  - Is it feasible to power nodes on and off repeatedly?

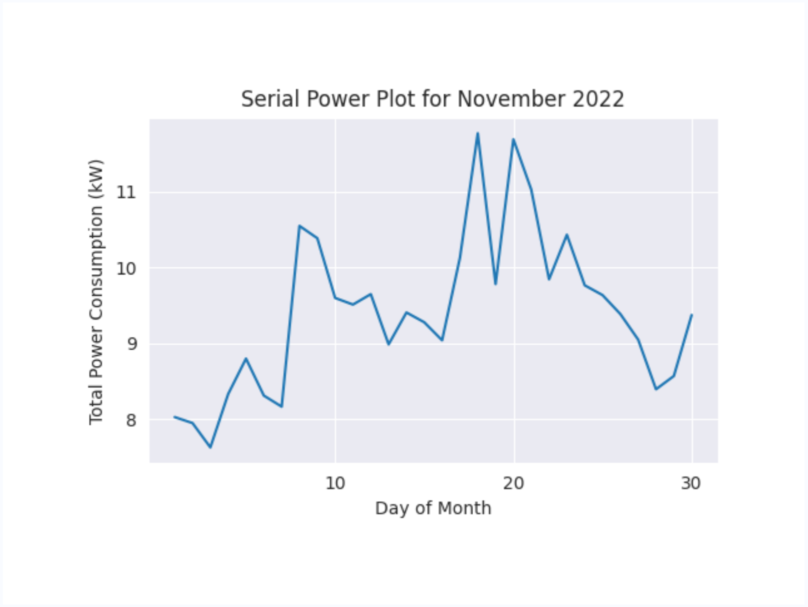
Partitions and their power usage

- If you've enabled power monitoring, then you can gather power stats from **scontrol show node**
- CurrentWatts=395 AveWatts=372
- As partitioning is set up to either section heterogenous hardware, or nodes that have different limits, this essentially means you can monitor a whole partition.
- If you have different setups the power consumption will vary and in certain scenarios leads to dual mode systems.
- What is traditional MPI enabled HPC isn't as big a deal as we think?

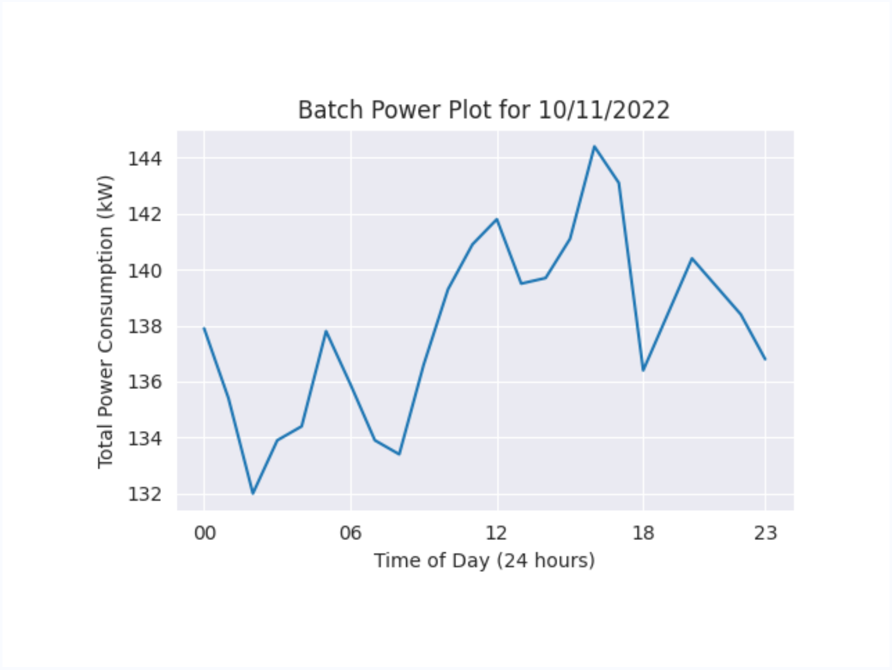
Serial vs Batch

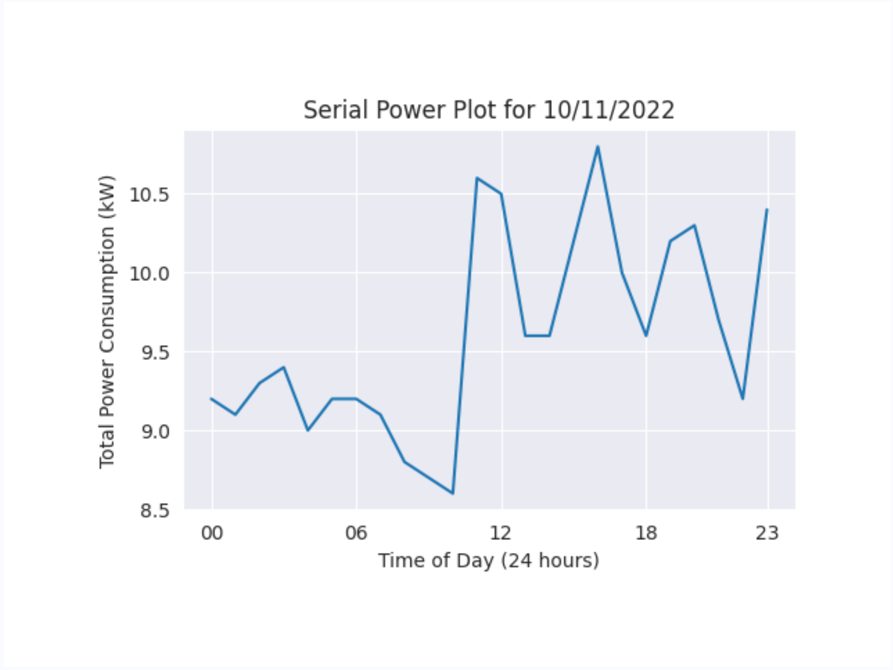
- If you have a heterogenous cluster, you are probably focusing on GPU vs Batch vs Serial
- Averages over a month:



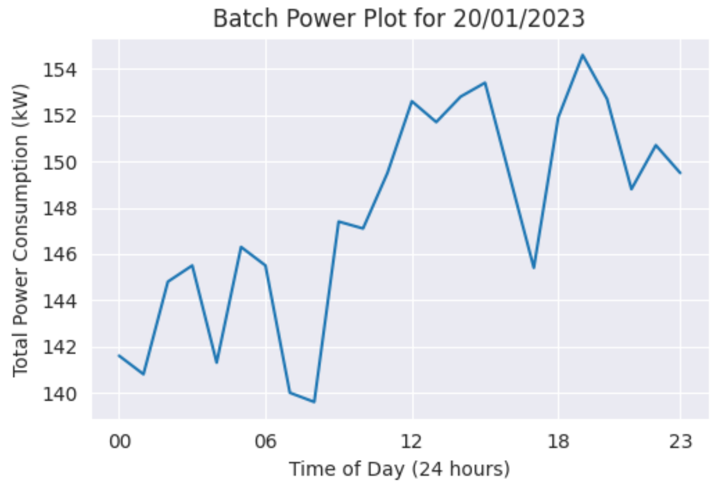


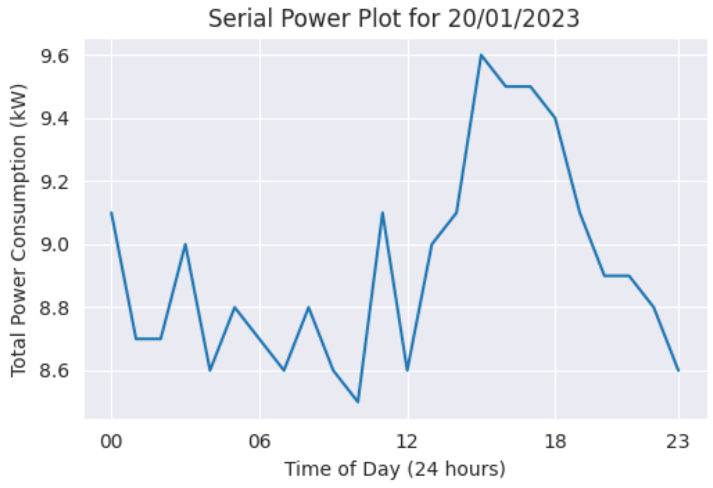
Serial vs Batch daily – November 2022





Serial vs Batch daily – January 2023





Why are we doing this?

- Does our system have a dual mode?
- If we're seeing power variations, does that inform future purchases and partitioning?
- We've got evidence for situations that are obvious – HPC is quieter at night
- We can obviously see that we are not at 100% uptime, in our batch partition we can see that power drops up to 10% regularly.
- Power is almost better than job stats, as it shows how much actual work is happening, as jobs that stall or fail don't consume power.

As an admin where is the pain?

- When we make changes like this, there are bound to be systemic effects.
- Constant rebooting leads to constant restarting of services and daemons that do not restart that often in production.
- Updating the nodes becomes difficult, particularly in the case of persistent builds.
- Education component, we need to explain it to users – we have already had tickets asking what the CF state is.
- It has not been plain sailing, issues with nodes in ALLOCATED+POWERED_DOWN state.

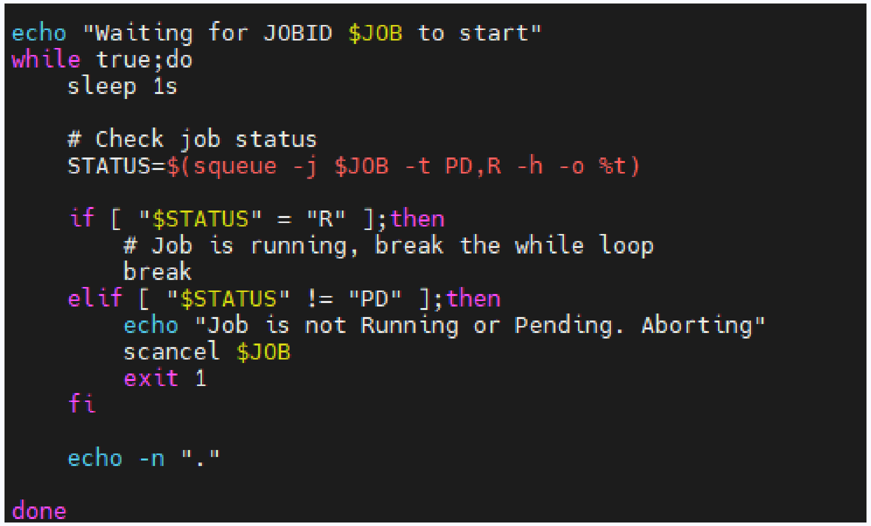
Diskless and diskful configurations

- Diskless configurations generally rely on configuration management like Salt and Puppet, as they need to in order to function.
- Diskful configurations rely on the changes that are made to persist, and this is an issue with SLURM power
- We’ve had to edit the SLURM service on the diskful builds to pull the SLURM.conf on every boot, or there will be discrepancies due to powered off nodes not receiving updates
- Rolling reboots for cleanliness?

University of Southampton

Interactive jobs

- A lot of sites have an interactive job mechanism, we use **sinteractive**



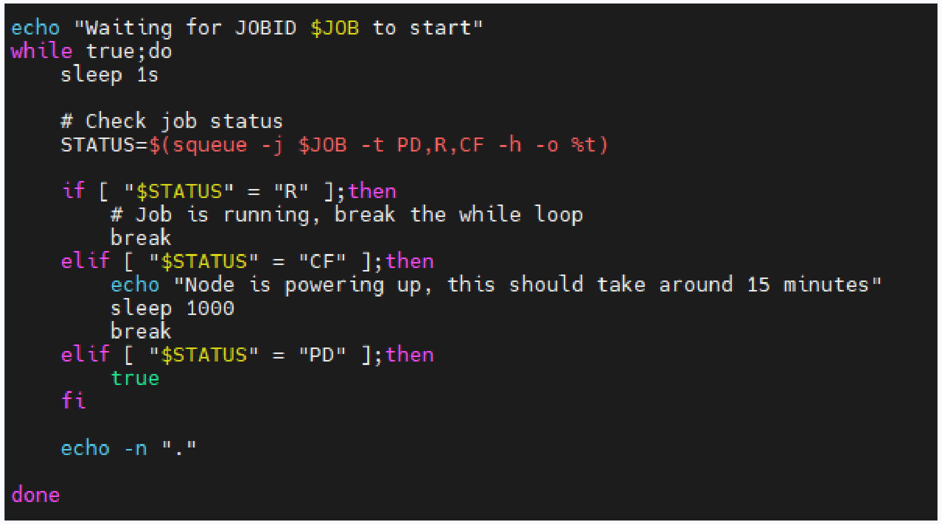
```
echo "Waiting for JOBID $JOB to start"
while true;do
    sleep 1s

    # Check job status
    STATUS=$(squeue -j $JOB -t PD,R -h -o %t)

    if [ "$STATUS" = "R" ];then
        # Job is running, break the while loop
        break
    elif [ "$STATUS" != "PD" ];then
        echo "Job is not Running or Pending. Aborting"
        scancel $JOB
        exit 1
    fi

    echo -n "."

done
```



```
echo "Waiting for JOBID $JOB to start"
while true;do
    sleep 1s

    # Check job status
    STATUS=$(squeue -j $JOB -t PD,R,CF -h -o %t)

    if [ "$STATUS" = "R" ];then
        # Job is running, break the while loop
        break
    elif [ "$STATUS" = "CF" ];then
        echo "Node is powering up, this should take around 15 minutes"
        sleep 1000
        break
    elif [ "$STATUS" = "PD" ];then
        true
    fi

    echo -n "."

done
```

University of Southampton

YOUR QUESTIONS