Using the Farm
-and-
SWIF2

- A Hall C User’s Perspective

Author: Casey Morean
Date: 2/17/2022

Disclaimer:
The information in this presentation is static. Any future changes to the farm, slurm, swif2 may not be reflected here. This information is intended for a Hall C audience, but the information should be useful for everyone.
Overview

• General Farm Usage
• Quick Slurm Introduction
• Migration from SWIF1 to SWIF2
  • Differences in JSON variable names
• hcswif updates
• Filesystem
• Run an auger job in SWIF2
General Farm Usage – Really General

- Remote Access to JLab Computers
  - [https://cc.jlab.org/remoteaccess](https://cc.jlab.org/remoteaccess)
- Getting help on the farm
  - [ServiceNow](https://cc.jlab.org/remoteaccess) portal
    - Login with CUE ➔ Create Incident
      (Scientific Computing / Systems)
  - [helpdesk@jlab.org](mailto:helpdesk@jlab.org)
- Hall C Physics Computing Coordinator
  - Brad Sawatzky: [brads@jlab.org](mailto:brads@jlab.org)
- Review Getting Started Pages
  - [https://scicomp.jlab.org/docs/getting_started](https://scicomp.jlab.org/docs/getting_started)

⚠️ Some of this information is out of date and needs updated ⚠️

- Need to get a computer account?
- Need to become Slurm account member?
- Need access to the JupyterHub? (linux group)
- Need to learn about file system layout?
- Linux group memberships?
Farm Usage – **Slurm** / **SWIF** / **hcswif** / Filesystem

- Become a member of a **Slurm account** (Hall C)
- Start with a simple SLURM job (Generate `/farm_out/`)
  - `srun hostname`
- Auger to Slurm:
  - Project → Account (c-comm2017 → hallc)
  - Track → Partition (debug, test → priority, analysis → production)

```bash
#!/bin/bash
#SBATCH --partition=priority
#SBATCH --account=hallc
#SBATCH --mem-per-cpu=512
#SBATCH --output=/farm_out/%u/%x-%j-%N.out
#SBATCH --error=/farm_out/%u/%x-%j-%N.err
printenv; date;
```

**Slurm User Commands**

```
ifarm1802.jlab.org> sbatch --hold --account hallc --partition priority my_first_steps.sh
Submitted batch job 51219661

ifarm1802.jlab.org> squeue -u cmoren

  JOBID  PARTITION  NAME     USER  ST   TIME NODELIST(REASON)
      51219661  priority  my_first  cmoren  PD    0:00  1 (JobHeldUser)
          51146192  prodctio  External  cmoren  CG  16:00:18  1 farm160136

ifarm1802.jlab.org> scontrol release 51219661
```

```bash
ifarm1802.jlab.org> ls /farm_out/cmoren/my_first_steps.sh-51219661*
/farm_out/cmoren/my_first_steps.sh-51219661-farm13019.err
```

Thu Feb 10 13:29:49 EST 2022

---

**Slurm User Commands**

```
ifarm1802.jlab.org> tail -n 1 /farm_out/cmoren/my_first_steps.sh-51219661-farm13019.out
```
Farm Usage – Slurm / **SWIF** / hcswif / Filesystem

- Review the new documentation
  - Check out the commands
  - Pay special attention to JSON data structure for import and export commands (hcswif)
- Get Job-specific information from **swif2 show-job**
- Make use of the priority queue for debugging
  - Check out the staging areas for failed jobs!
  - Always define a stderr and stdout

**SWIF2 job-id can be hard to locate:**

- **swif2 list**
- **swif2 status <workflow_name> -jobs | grep -A 1 “job_id”**
- **swif2 show-job -jid <job_id>**

```bash
job_id = 856422
job name = trial hcswif xem2 hms all 01642
```

Copies working directory contents back to the staging area

- `$SWIF_DIR=/lustre/enp/swif2`
- `$SWIF_DIR/jobs/$USER/<JOB>`
  - Links to input files
- `$SWIF_DIR/input/<input_id>`

Figure out what went wrong!

Do Tag: pass1
Then do: pass2

Try using: antecedents
Tags
SWIF1 to SWIF2 changes:
• JSON data structure
  • No warnings for old JSON files!
• All Projects now just use slurm account: hallc
• Partition names have changed
• More strict control of resource overutilization
• Uses slurm instead of Auger: SWIF2 jobs show up in slurm jobs
• More job information on output

Check the JSON file parameters were actually applied:
swif2 import –file SOME_WORKFLOW.json
swif2 export | python –m json.tool > SOME_WORKFLOW_EXPORT.json
• Validate the parameters were set in swif2
Farm Usage – Slurm / SWIF / hcswif / Filesystem

• Update hcswif for SWIF2: [https://github.com/JeffersonLab/hcswif.git](https://github.com/JeffersonLab/hcswif.git)
  • Change JSON variable names
  • Update README.md with some more examples
  • Separate JSON from STDOUT and STDERR
  • Pretty JSON printing in output file
  • Update default soften to 2.4

Making personal Changes:

hcswif.py
• out_dir
• json_dir
• raw_dir

setup.sh
• hcana_dir
• hallc_replay_dir

setup.csh
• hcana_dir
• hallc_replay_dir

Running the command for a job in the JSON file on an ifarm computer can help debug problems locating the executables and directories
Review **filesystem layout** (figure 1)

- The `/group/` location is also backed up!

**Backups:**

- User and group is **backed up**, so DEVELOP CODE THERE!
  - `rm -rf /group/proj/hcswif/`
  - `cp /group/proj/.snapshot/group-daily-snapshot_YYYY-MM-DD_HH:MM/hcswif /group/proj/`

- Make use of tape for analysis!
  - Trim down the analysis trees for each ‘pass’ of the data and save the files.
  - Compress your analysis scripts, REPORT_FILES, results, etc to save to tape

**AVOID:** Why are the results from pass2 and pass3 so different from one another?
Farm Usage – Auger via SWIF2

“If jcache the input from tape library and/or use Auger Job Array (ForEach),…”

- Use SWIF2

Creating a SWIF2 job for AUGER submission

swif2 create –workflow <workflow_name>

swif2 add-jsub –script <script_name.xml>

- Track and Project still required in auger sub. file

If a member of multiple accounts, the slurm account chosen will be your default account.

The partition will default be to production

swif2 export | python –m json.tool > json_of_auger.xml

- Edit account and or partition to be correct

swif2 import –file <modified_json_of_auger.xml>

Farm Usage – Auger via SWIF2

“If jcache the input from tape library and/or use Auger Job Array (ForEach),…”

- Use SWIF2

Creating a SWIF2 job for AUGER submission

swif2 create –workflow <workflow_name>

swif2 add-jsub –script <script_name.xml>

- Track and Project still required in auger sub. file

If a member of multiple accounts, the slurm account chosen will be your default account.

The partition will default be to production

swif2 export | python –m json.tool > json_of_auger.xml

- Edit account and or partition to be correct

swif2 import –file <modified_json_of_auger.xml>
Summary

• Lots of links to documentation!
• Quick Slurm Introduction
  • Generate /farm_out/
• Migration from SWIF1 to SWIF2
  • Differences in JSON variable names
• hcswif updates
• Filesystem / jcache
• Run an auger job in SWIF2

How should we work together?
• Slack – Communicate / Help with issues and work through them, but no long-term solutions for all
• User based documentation – Becomes outdated. Who is in charge? No direct relationship with CS people