GSPDA Mini-Software Workshop (Part 2)

"Hands-on" SWIF2 Farm Submission Example

Sept. 6, 2024

Brad Sawatzky



Thomas Jefferson National Accelerator Facility



"Scientific Workflow Indefatigable Factotum"

- Or <u>SWIF</u>, to save time...
 - → Swif is a workflow manager designed at JLab
 - → Interfaces intelligently with the JLab tape system and slurm, the job scheduler
 - will efficiently pre-stage file dependencies
 - will efficiently ensure output files are placed on tape
 - \rightarrow Command line tool: <u>swif2</u>
 - \rightarrow Web monitoring

- SWIF == Workflow manager
 - → provides tools to let you to submit, run, and monitor groups of jobs
 - » ie. a "workflow"
 - → provides information on individual jobs as well as aggregate information
 - → tracks success and failures and allows you to modify and resubmit failed jobs
 - » ie. outlier jobs that need more time, RAM, disk, etc





Simple Linux Utility For Resource Management

- Or Slurm, to save time...
 - → Open source compute cluster resource manager and job scheduler
 - → <u>JLab Slurm Docs</u>
 - » sinfo
 - » srun
 - » Useful env-vars
- Jobs run under a particular
 - → <u>Slurm 'Account'</u> (ie. group)
 - » Ask helpdesk@jlab.org or <u>Compute Coordinator</u>
 - \rightarrow <u>Slurm 'Partition'</u>
 - » usually 'production' or 'priority'

- Cluster/Job Scheduling
 - → takes care of nitty-gritty of actually running jobs on a given compute node
 - → tracks and enforces resource usage and requirements (RAM, disk, time, etc)
 - enforces a fair balance of allocated farm time across our user base (Division, experiment, Hall, etc)
 - "Fairshare" allocations are managed by Compute Coordinators, SciComp and Physics Division



Thomas Jefferson National Accelerator Facility

workload manager







General Farm Access

- See "New users start here"
 - → At top of scicomp.jlab.org
 - → See this <u>excellent walkthrough</u> for BRIC by Cameron Clarke too!
- You will need (in general):
 - → JLab CUE account
 - → Get a 2-factor token (ssh access)
 - → Get access to Farm/ifarm
 - » talk to Project lead and/or Compute Coordinator
 - learn where to put your files (unix group access)
 - » get attached to a slurm group
 - → SciComp user certificate



Steps to getting started using the Physics Farm system at Jefferson Lab are listed below:

- 1. Obtain Farm/ifarm access.
- 2. Generate a SciComp user certificate for your account.
- 3. Please join the email list jlab-scicomp-briefs@jlab.org to get email about the status and updates of resources. This is highly recommended.
- 4. Considerable information about the status of the computing resources is available at the SciComp Portal: scicomp.jlab.org/scicomp/. On the entry page you can find information about the status of the various systems, as well as important announcements about new capabilities, current problems or planned outages. From the menu on the left you can get more detailed information about status and utilization, including reports by user or project for any arbitrary time interval.
- 5. For all other account questions please contact your Hall Compute Coordinator, the Physics Computing Coordinator Brad Sawatzky, or the Computer Center Help Desk



* Backed to tape. Excludes small files



< Experimental Physics User's Guide	up	Access
File system layout		
 Network certificate 		
 Start using JLab Jupyter Hub 		
 Login to SciComp GPUs 		
 Interactive login machines 		
 Create custom kernel for Jupyter Hub 		
 Access 		





Other Information Resources

- scicomp.jlab.org
 - \rightarrow SciComp web page
- scicomp-briefs
 - → mailing list for JLab Scientific Computing

	9 A					<u> </u>	sername G	etting Started	Support	Staff Members
Cluster Info				Jlab	Scientif	ic Comp	uting			
Farm Nodes	Welcome to the	Jefferson	Lab Scier	ntific compu	uting hom	e page. <u>Ne</u>	w users	start here.		
Slurm Jobs Swif2 Jobs Usages	Feb-27-24 Softwa CVMFS softwa compu The leg u/scraft	e Environme s now used s package av ng coordinat cy /site area h area has a	nt and Filesy to distribute se ailability, pleas or. has been remo	stem Changes oftware. It is roo se submit a Ser roved. The path ved.	The use of / oted under OA viceNow incid to Jasmine (apps is depreci SIS and can be dent. For hall-sp tape) and cach	ated and is no used with <u>m</u> secific softwa e tools is cha	ot available on far iodule <u>files</u> as befo ire distribution qu nged from /site/b	m AlmaLinus ore. For ques estions, cont in to /usr/loo	x 9 machines. tions about tact your cal/bin. The CUE /
File System ^	Feb-26-24 Farm C compo default constra introdu The int machin tempo	grade Sche ition with cha will change a nts. <u>This arti</u> tion and <u>SW</u> ractive (ifarm s that will ru ry disk space	dule and Worl ange from maj t a later step in <u>cle</u> provides d <u>F command li</u> n) nodes curre n AlmaLinux 9 a.	ker Node Select ority CentOS 7 in the conversion etails on feature <u>ne</u> reference for ntly run CentOS are on order. T	tion The fam to predomina h process. Us based node r details. 7. A new ma hey will repla	m is being upgr ntly AlmaLinux ers may curren selection. SWI chine, ifarm9.jl ce the existing	aded in a ser 9. At the tim- tly select whi F can pass fe ab.org is avai ifarm machir	ies of steps. Betw e of this writing, C ch nodes run thei atures through to lable for AlmaLin ies and include m	reen now and entOS 7 is th r jobs using : Slurm. See t ux 9 use now ore per-core	d June, the farm he default. This slurm features/ the SWIF v. Two new ifarm memory and
West	Plum Job (Outstan	(aba)		Charge Job (alahad laha)				
Usage History	Running Pene	ng Held	Other 5	Success	Failed	Cancelled	Timedout	OverMemeory 35	NodeFa	il
Tape Library][
Jobs Usage Data Mover Data Mover User's Guide (2) User Coverable File News Archive	Job Info Last 24 He	famile fa		Datamov 20 15 10 5 0	or Status	File Sys x x x 0	tem Status -	1 . .		

- Documentation links
 - \rightarrow <u>Getting Started</u>
 - \rightarrow SciComp Knowledge Base
 - \rightarrow <u>CST User Portal</u>
 - \rightarrow JLab Helpdesk
 - » helpdesk@jlab.org
 - » Incident Request





Thomas Jefferson National Accelerator Facility



The Plan (TM)

- Develop and test an example compute job
- Develop and test a 'steering script' that you will use to
 - \rightarrow setup your environment
 - \rightarrow stage your code/exe,
 - \rightarrow and run your code
 - as a non-interactive farm job



Backed to tape. Excludes small file:

 Submit a few example jobs to the Farm using SWIF2

- → monitor those jobs with some swif and slurm commands
- → modify and re-run any failed jobs







The Plan (TM)

- We are going to compute pi using a simple (but not super efficient) Monte Carlo program
 - → Pick random points in a 1x1 square area and count the number that land within a radius of 1 from the origin
 - \rightarrow Area of the square is 1x1
 - \rightarrow Area of the 1/4 circle is 1/4 π 1^2
 - \rightarrow So the ratio of the areas is 1/4 π
 - \rightarrow So lets throw N_{sample} points at the square and see what fraction of them are also in the circle



•
$$A_{circle} / A_{square} = \frac{1}{4} \pi$$

 $\rightarrow \pi = 4 \times \text{bits} + \frac{1}{4} / \text{bits}$

- $\rightarrow \pi = 4 \text{ x hits}_{\text{circle}} / \text{hits}_{\text{square}}$
- $\rightarrow \pi$ = 4 x hits_{circle} / N_{samples}
- More samples
 - \rightarrow better coverage
 - \rightarrow better accuracy
 - (but takes longer)



Thomas Jefferson National Accelerator Facility



Grab the Example Code

- ssh <you>@login.jlab.org
- ssh ifarm

```
brads@aether 1021% git clone https://code.jlab.org/brads/workshop-sept2024.git
Cloning into 'workshop-sept2024'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), 4.47 KiB | 4.47 MiB/s, done.
brads@aether 1022% cd workshop-sept2024/
brads@aether [git:master] 1023% ls -lF
total 16
-rwxr-xr-x 1 brads brads 4321 Sep 4 00:01 job-script.sh*
-rwxr-xr-x 1 brads brads 2186 Sep 4 00:01 piece-of.py*
-rwxr-xr-x 1 brads brads 2053 Sep 4 00:01 swif-submission.sh*
brads@aether [git:master] 1024%
```



Thomas Jefferson National Accelerator Facility JLab Software & Computing Workshop • Sept 6, 2024





Dev Process: Test your executable!

- Test your code on the ifarm
 - \rightarrow piece-of.py
 - » this does the actual calculation

	50.00
brads@ifarm2401 1006% cd workshop-sept2024/	6.18
brads@ifarm2401 1007% ls	
job-script.sh output output-work piece-of.py swif-submission.sh SWS-pitest-brads	2 2 2
brads@ifarm2401 1008% ./piece-of.py -v 100000000	6 6 0
Running 100.0 M samples	P Stander
10.0 M : 3.1418976	P67/600
	-D970580
	-6910
	57145
100.0 M : 3.14185776	

Number of samples: Number inside unit quarter-circle: 78546444 Pi is roughly: 3.141858 brads@ifarm2401 1009%





Thomas Jefferson National Accelerator Facility



10



Dev Process: Test your job-script!

- Create and test a 'job-script' wrapper on ifarm
 - → job-script.sh <arguments>
 - → this 'wrapper' will be invoked for every farm job
 - \rightarrow its purpose is to:
 - » definitively set-up your working environment
 - » load software dependencies
 - » check directories, etc
 - » configure and run your executable for each discrete job in your workflow
 - update run number
 - update config files
 - update event count
 - etc...
- Test this script in a clean 'working dir' to replicate farm node environment
 - → we will use /scratch/\$USER/workshop-test/

brads@ifarm2401 1009% cd /scratch/\$USER brads@ifarm2401 1010% mkdir test brads@ifarm2401 1011% cd test brads@ifarm2401 1012% ~/workshop-sept2024/job-script.sh 100000000

	Environment SHELL=/bin/bash LESS=-x4 -j4 -i -w -e -M -X -R -P%t?f%f :stdin .?pb%pb\%:?lbLine %lb:?bbByte %bb: HISTCONTROL=ignoreboth HOSTNAME=ifarm2401.jlab.org HISTSIZE=1000 PERLLIB=/home/brads/lib/perl BSTINPUTS=:/home/brads/LaTeX/Bibstyles/ DEBFULLNAME=Brad Sawatzky EDITOR=vim SIM HOME=/cymfs/oasis.opensciencegrid.org/ilab/geant4
	PWD=/cratch/brads/test
ľ	
	Number of samples: 100000000 (100 M) Number inside unit quarter-circle: 78534714 Pi is roughly: 3.141389
	Copy the file to /work/hallc/c-polhe3/brads/workshop-temp Nothing to do here. Let's handle this using swif2 '-output' option
	List Working Directory (END OF JOB) total 28
	drwx 3 brads a-phy 4096 Sep 3 22:36 ./
	drwx 5 brads a-phy 83 Sep 3 22:37/
	drwx 7 brads a-phy 155 Sep 2 20:11 .git/
	-rw 1 brads a-phy 160 Sep 2 19:59 .gitignore
	-IWXI-XI-X. I Drads a-phy 4435 Sep 3 20:13 job-script.sh*
	IIWXIWXIWX. I DIGUS G-DHY 40 SEP 5 19:00 OULDUL-WOIK -> /WOIK/HAILC/C-DOINE3/DIGUS/WOIKSNOD-TEMP/
	-rwxr-xr-x. 1 brads a-phy 1405 Sep 3 13:48 piece-of.pv*
	-rwxr-xr-x. 1 brads a-phy 2224 Sep 3 20:54 swif-submission.sh*



Thomas Jefferson National Accelerator Facility



Dev Process: Test your 'swif submission' script

Develop and test a script that can generate the 'swif2 addjob ...' command for each job in your workflow

- » swif-submission.sh
- » Update SLURM_ACCT
- → This can/will loop over runs / run number (from directory or file), etc and create a swif job submission for each case.
 - Declare (tape) -outputs
 - Declare (tape) -inputs
 - » NOTE: mss:// (tape) files will be copied to working dir!
 - Access them as ./<filename> in your analysis script <u>not</u> /cache/..../<filename>
- \rightarrow Also sets 'defaults' like:
 - » slurm partition/queue
 - » slurm 'account'
 - » RAM, disk space, CPU needs

#!/bin/bas<mark>h</mark>

- Use *your* slurm account
Find your name here: <u>https://scicomp.jlab.org/scicomp/slurmJob/slurmAccount</u>
SLURM_ACCI='hallc' # USE YOUR SLURM ACCOUNT/group here!

WORKFLOW="SWS-pitest-\$USER"

JOB_SCRIPT="*{SRCDIR}/job-script.sh" JOB_ARGS="5000000 50000000 500000000" **# 5M, 50M, 5000M**

ECH0="echo" ## Used to echo swif commands for testing
if [\${1:-unset} == "submit"]; then
 ECH0="" ## disables the 'echo' and actually runs the swif commands
fi

NOTES:

- The -input file is a semi-random file just used as an example of how to
- pull from tape and demonstrate the file lands in the working directory on the farm node. For example:
- input 'shms_all_09450.dat' 'mss:/mss/hallc/c-polhe3/raw/shms_all_09450.dat' \
- # For 'real' jobs, you would use the 'production' partition/queue:
- -partition 'production'
- The 'priority' partitin/queue has limited run time and resource restrictions
- but will schedule rapidly -- intended for quickly testing a short job
- or two before switching to 'production' and submitting a complete workflow

for N in \$JOB_ARGS; do \$ECHO \ swif2 add-job ∖ -create \ -workflow "≉WORKFLOW" \ -account "\$SLURM ACCT" \ -partition 'priority' \setminus -constraint 'e19' 🔨 '1' '100M' \ -time 5min \ -shell '/bin/bash' \ -output "\$OUTPUT_GLOB" "\$DESTDIR/" \ "\$JOB_SCRIPT" "\$N" done

Jefferson Lab





Dev Process: Test your 'swif add-job ...' script

Comments / Notes

-cores <N>

 \rightarrow know what your job can actually use

-ram <size>

 \rightarrow shoot for < 2GB/core if you can

-disk <size>

- $\rightarrow~$ disk space needed on the farm node itself
- \rightarrow If set, then <size> == size for both
 - » '-input' files, AND
 - » '-output' files on the local disk
- \rightarrow If unset, then Swif will use
 - » SizeOf(input files) * 1.1

-scratch_disk <size>

Coming Soon!

→ <size> only has to account for output files on local disk (space for input files is added to this automatically)

- "Gotchas" to remember:
 - → Do not modify your executable or job-script.sh until your workflow is fully completed
 - » no recompiling; no 'tweaks'
 - » (It will change what runs on the farm mid-workflow!)
 - → Remember many jobs will run at the exact same time!
 - » If you run your job from within a network mounted space (ie. /group, /work) then you *must* ensure input and output files have unique names, 'temp' files don't step on each other, etc.
 - → Access '-input mss://....' files from job's working directory (*not from /cache*)



Thomas Jefferson National Accelerator Facility





Submit your Jobs to SWIF

- Check lines from
 → swif-submission.sh
 - SWIT-SUDMISSION.SI
 - ~ Then ~
- Run 'swif-submission.sh submit'

brads@ifarm2401 id name	1084% ./swif-submission.sh submit = 32969382 = SWS-pietest-brads-0
id name	= 32969383 = SWS-pietest-brads-1
id name	= 32969384 = SWS-pietest-brads-2
Active SWIF (workflows

swif2 list

-- SWIF jobs for workflow SWS-pietest-brads -----swif2 status -workflow SWS-pietest-brads -jobs

NOTE: IF the workflow looks good, then tell SWIF to execute it with: swif2 run -workflow 'SWS-pietest-brads'



Thomas Jefferson National Accelerator Facility JLab Software & Computing Workshop • Sept 6, 2024





brads@ifarm2401 1021% ./swif-submission.sh swif2 add-job -create -workflow SWS-pitest-brads -account h allc -partition priority -constraint el9 -cores 1 -ram 100M -time 5min -shell /bin/bash -output match:pi-calc-output-* /work/hallc/c-polhe3/brads/workshop-temp/ /home/brads/work shop-sept2024/job-script.sh 500000 swif2 add-job -create -workflow SWS-pitest-brads -account h allc -partition priority -constraint el9 -cores 1 -ram 100M -time 5min -shell /bin/bash -output match:pi-calc-output-* /work/hallc/c-polhe3/brads/workshop-temp/ /home/brads/work shop-sept2024/job-script.sh 5000000 swif2 add-job -create -workflow SWS-pitest-brads -account h allc -partition priority -constraint el9 -cores 1 -ram 100M -time 5min -shell /bin/bash -output match:pi-calc-output-* /work/hallc/c-polhe3/brads/workshop-temp/ /home/brads/work shop-sept2024/job-script.sh 500000000 brads@ifarm2401 1022%

Don't forget this step!!

brads@ifarm2401 1020% cd ~/workshop-sept2024/

Useful SWIF2 Commands

- swif2 list
 - \rightarrow see all of your workflows
- swif2 status -workflow \$WORKFLOW
 - \rightarrow see summary details for \$WORKFLOW
- swif2 status -workflow "\$WORKFLOW" -jobs | less
 - \rightarrow see a list of jobs associated with \$WORKFLOW (can be big list!)
 - \rightarrow find SWIF job_ids, (*etc*)
- watch -d -n30 -- "swif2 status \$WORKFLOW"
 - \rightarrow if you're impatient like me...
- swif2 show-job -jid <swif_job_id>
 - \rightarrow get info on a particular SWIF2 job
 - » job_status, slurm_id, etc





Useful Slurm Commands

NOTE: Slurm commands take <u>slurm</u> job-ids, not SWIF job-ids!

- sacct
 - \rightarrow queries all the things (ie. slurm DB access)
 - \rightarrow https://slurm.schedmd.com/sacct.html
- sacct --long -j <slurm_id>
- sacct -- env-vars -j <slurm_id>
 - \rightarrow env vars set during a slurm job
 - \rightarrow SWIF_JOB_ID, SWIF_JOB_ATTEMPT_ID \leftarrow map back to SWIF job!
- seff <slurm_id>
 - \rightarrow wrapper around sacct that reports job efficiency numbers
 - \rightarrow RAM, CPU usage







Don't forget about /farm_out/\$USER/swif/...

- cd /farm_out/\$USER/swif
- Is
 - → Should see a list of stdout and stderr for jobs that have run (or are currently running!)
- less <job-name-id>.out
 - → shows everything you dumped in your job-script.sh and what your executable wrote to stdout
- less <job-name-id>.err
 - \rightarrow shows text written to stderr (usually 'errors') and/or debug info





Web-based SWIF/Slurm Reporting



- Most of the command-line info from SWIF and slurm can be found on <u>scicomp.jlab.org</u> web page
 - \rightarrow See Swif2 Jobs
 - \rightarrow See Slurm Jobs







Web-based SWIF2 Reporting

- SWIF2 Jobs
 - \rightarrow Workflow Summary
 - » aggregate job info
 - » failed jobs at bottom
 - \rightarrow Active (Dormant) Workflows
 - » What is running/waiting
 - Filter to find your workflow and select it for job list
 - click on job for job info

Cluster Info	^	Active Workflows								
Farm Nodes		Active Workflows Dormant Workflows Workflow Summary File Queue Globus Status								
Slurm Jobs		Go Back to Overview Table								
Swif2 Jobs		Filter brads	Expand to display job detail and input files							
Usages		SWS-pitest-brads (id: 129902)	500 52622137 1110							
		Job Id Job State Job Name Job Phase	Job 32822137 Run at jlab/enp							
File System	^	32822137 attempting SWS-pitest-brads-0	Command: ./job-script.sh 1000000 Batch Flags: null							
Lustre		32822972 attempting SWS-pitest-brads-1								
Cache		32824426 attempting SWS-pitest-brads-2	Input Files of Job '32822137' jab mss /mss/halic/o-polie3/raw/shms_all_09450.dat 16,613.376							
Guene		32826028 attempting SWS-pitest-brads-3								
Volatile		32826029 attempting SWS-pitest-brads-4								
Work		32826757 attempting SWS-pitest-brads-5	Filter Job Attempts for job '32822137'							
Usage History		32831170 attempting SWS-pitest-brads-6	Job Atternet Id Atternet Status Atternet Problem Details Atternet Time							
		32831279 attempting SWS-pitest-brads-7	44804190 problem SLIIBM FAILED Exited with code 127 Sep 3 2024 3:40:31 PM							
Tape Library	^	22072252 stampting SWS-nitaet.brade.9								





Thomas Jefferson National Accelerator Facility JLab Software & Computing Workshop • Sept 6, 2024



Web-based Slurm Reporting

Cluster Info	^	Outstanding (Pending/Active) Batch Farm Jobs							
Farm Nodes		Outstanding Jobs	Recent Jobs	Mem Efficiency	CPU Efficiency	Jobs query	Slurm Info	Slurm Account	Job Limit
Slurm Jobs		Filter							
Swif2 Jobs		User Name		Account	Pending	Running	Holding	Other Jobs	
Usages		aaustreg		halld	79	0	0	0	
		amgunsch		halla	0	30	0	0	
File System	^	barryp		jam	0	1	0	0	
		boyu		halld-pro	503	269	0	0	
Lustre		btumeo		clas12	3,182	172	0	0	
Cache		clas12		hallb-pro	398	0	0	0	
Valatila								_	

- Inefficient Job information
 - \rightarrow Mem / CPU efficiency tabs
 - \rightarrow Will get an email if you hit this list
- Jobs Query
 - \rightarrow look up slurm job information

- "Slurm info" tab
 - → Find 'feature' tags for use with job 'constraints'
 - \rightarrow See a list of partitions (job queues)
 - → Slurm Account
 - » Find your name; use the correct slurm account for your job!



Thomas Jefferson National Accelerator Facility





Fix and Resubmit failed job(s) with Swif2

- If we get this far and you are monitoring your jobs, you will find that 1 of the 3 jobs we submitted failed:
 - \rightarrow SLURM_TIMEOUT error
 - \rightarrow The 500M event job took longer than the 5min we declared in swif

- g Fix this and resubmit:
 - → swif2 modify-jobs \$WORKFLOW -time 20min
 - -problems SLURM_TIMEOUT

brads@ifarm2401 1117% swif2 modify-jobs \$WORKFLOW -time set 20min -problems SLURM_TIMEOUT Found 1 matching jobs Modified 1 job Resolving 1 problem job brads@ifarm2401 1118% []



Thomas Jefferson National Accelerator Facility JLab Software & Computing Workshop • Sept 6, 2024





Any Wisdom/Hints for Others?

- What issues have you run into?
 - → Raise them here so others can learn from your experience!
- Any quirks that need better documentation?
- Let me know so we have a shot at fixing the problems/issues?

- 'swif2 notify' (as is) not super useful
 - → only reports if all jobs in a workflow are tagged 'DONE'
 - → a single pending/broken will keep this from firing
- 'swif2 notify' is being tweaked to allow for email notification if there are no more jobs capable of running
 - \rightarrow could be successful or stalled
 - \rightarrow much more useful
 - » Coming Soon!





