ilifu Online Training

Dr Jordan Collier

ilifu Support Astronomer, IDIA, Department of Astronomy, University of Cape Town
Adjunct Fellow, Western Sydney University

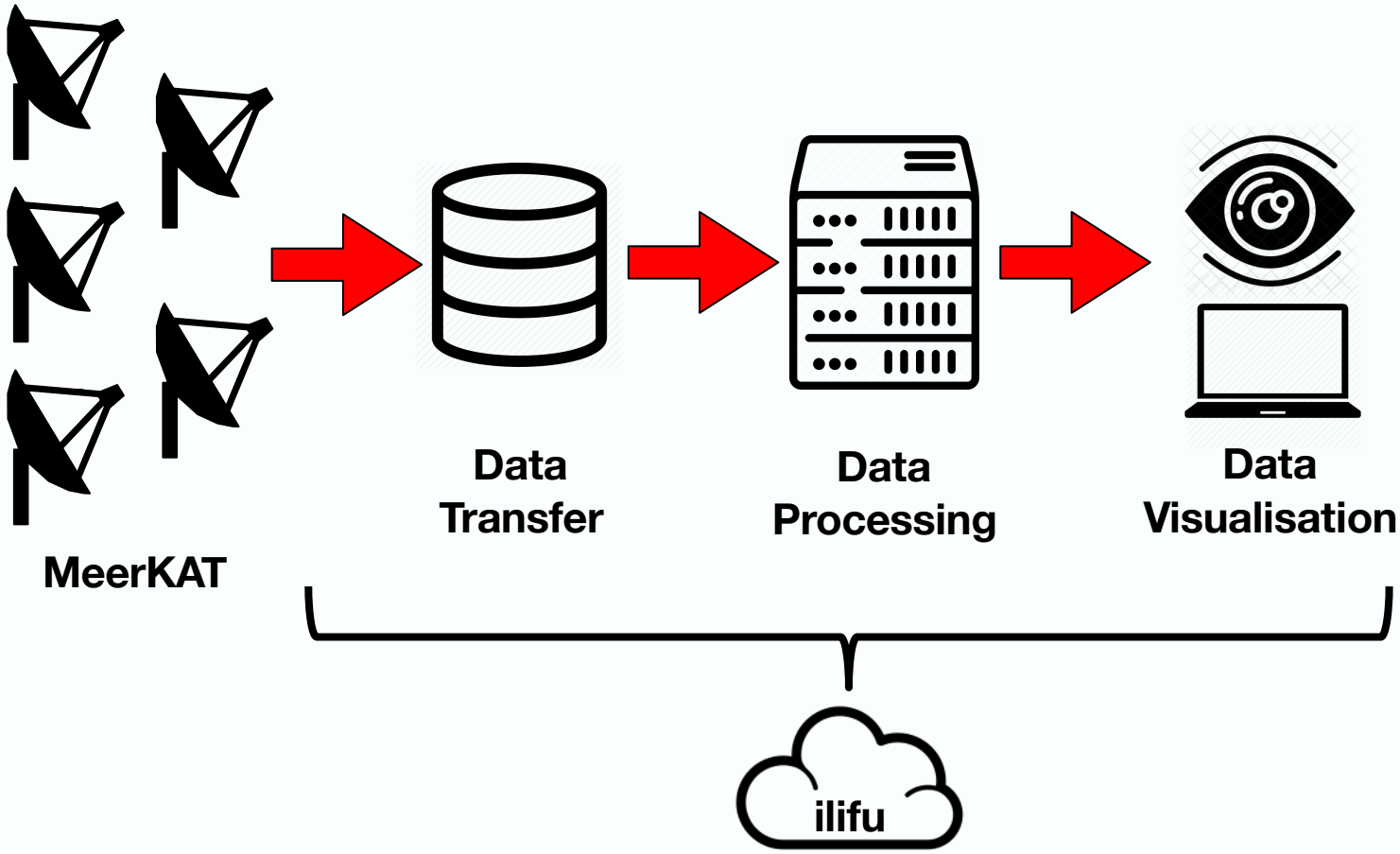
UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

ilifu

IDIA Inter-University Institute for Data Intensive Astronomy

WESTERN SYDNEY UNIVERSITY

ilifu (Biased) Dataflow



Data Transfer

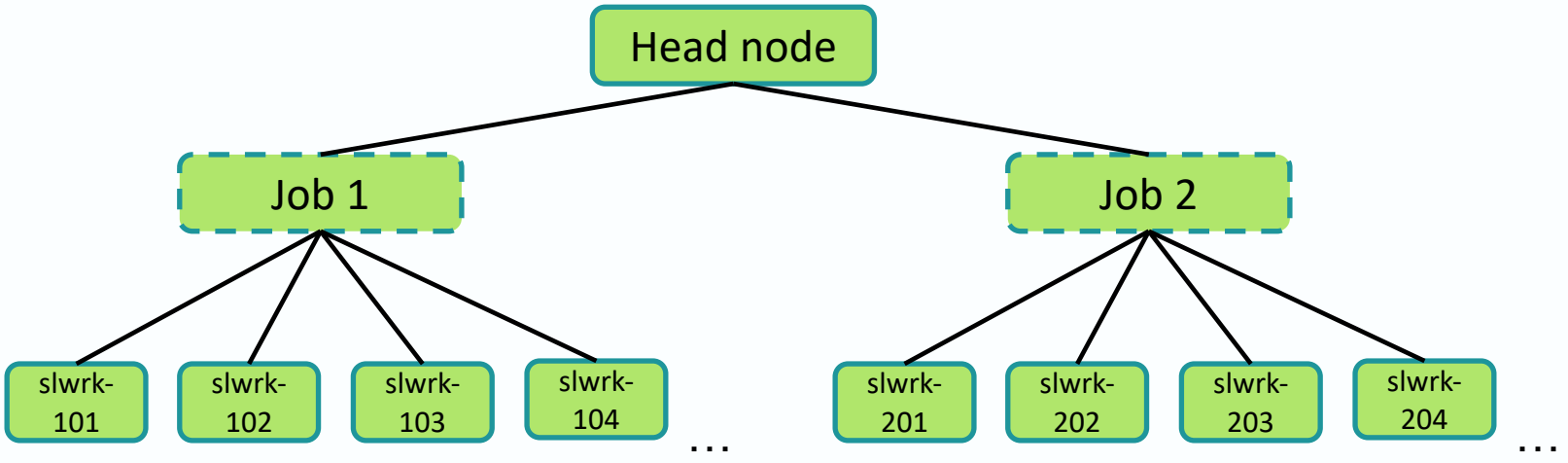
- http://docs.ilifu.ac.za/#/data/data_transfer
- Do not use head node!
- Don’t write large files to /users, only scripts/config files
- transfer.ilifu.ac.za
  - For scp and rsync
    - e.g. `$ scp /path/to/file/<filename> <username>@transfer.ilifu.ac.za:/idia/users/<username>/scripts/`
- Globus
  - Faster than scp and rsync

Data Transfer: Globus

- GridFTP transfer service setup between SARAO and IDIA
  - Currently extending to collaborators in UK, Netherlands, Italy, India, etc
- User-friendly globus connect built on top of gridFTP
  - GUI/web app or CLI
- Can connect any arbitrary end points
  - Server (DTN), desktop, etc for Mac OS X, Windows & Linux
  - Offers user-friendly but computationally efficient transfer service
- Demo time!

Parallelism

- Oxford definition for parallel processing
  - *a mode of operation in which a process is split into parts, which are executed simultaneously on different processors attached to the same computer.*
- The work is partitioned into smaller jobs, sometimes with a partition of the dataset



Parallelism

- Can be achieved on a single machine / node
  - Distributes work over many CPUs
  - Typically implemented using OpenMP

- Or over multiple machines / nodes
  - Distributes work over many tasks, over 1+ nodes
  - Each given amount of memory to use
  - Generally requires a cluster
  - Typically implemented using OpenMPI
  - Requires a message passing interface (MPI) wrapper
    - mpirun, aprun, srun (SLURM), mpicasa (CASA)

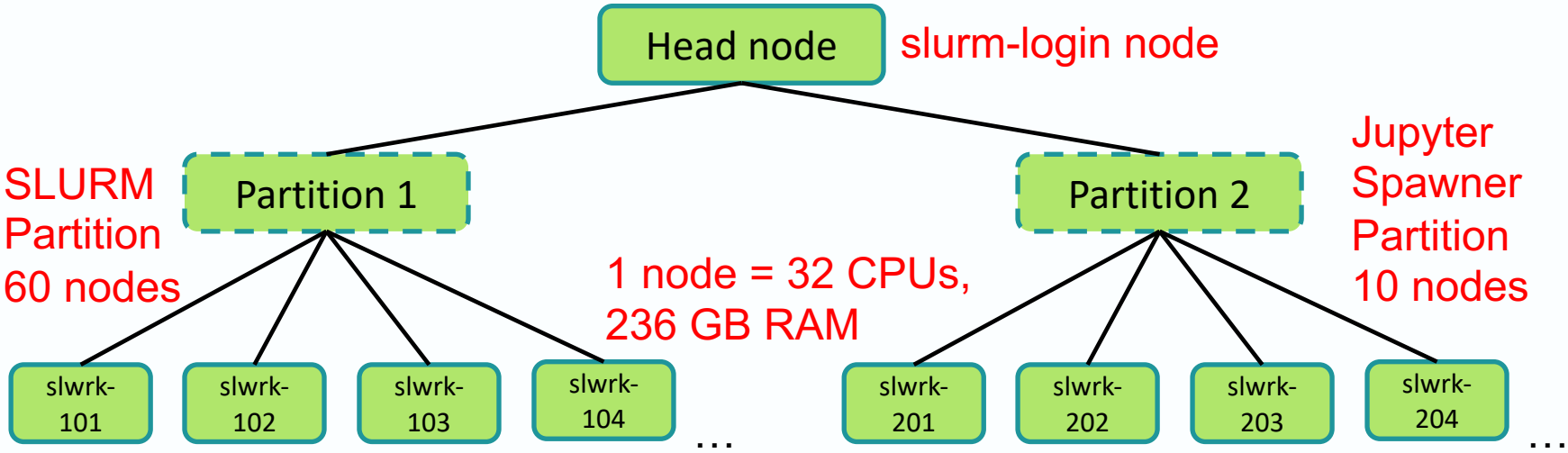
- Managed on ilifu by SLURM

Parallelism

- Implementing a normal job in SLURM
  - Will only use 1 CPU, 1 task, and 1 node
  - Default for many processes
- Implementing an OpenMP job in SLURM
  - Need to use >1 CPU, while nodes & tasks must be 1 (unless also using MPI)
    - `cpus-per-task`
    - May need to export `OMP_NUM_THREADS`
- Implementing an MPI job in SLURM
  - Need to use >1 task, while nodes and CPUs can be 1
    - `nodes`
    - `ntasks-per-node`
    - `cpus-per-task`
- See upcoming examples

SLURM

- http://docs.ilifu.ac.za/#/getting_started/submit_job_slurm
- Head node (job submission & management)
- Partitions
- Compute nodes

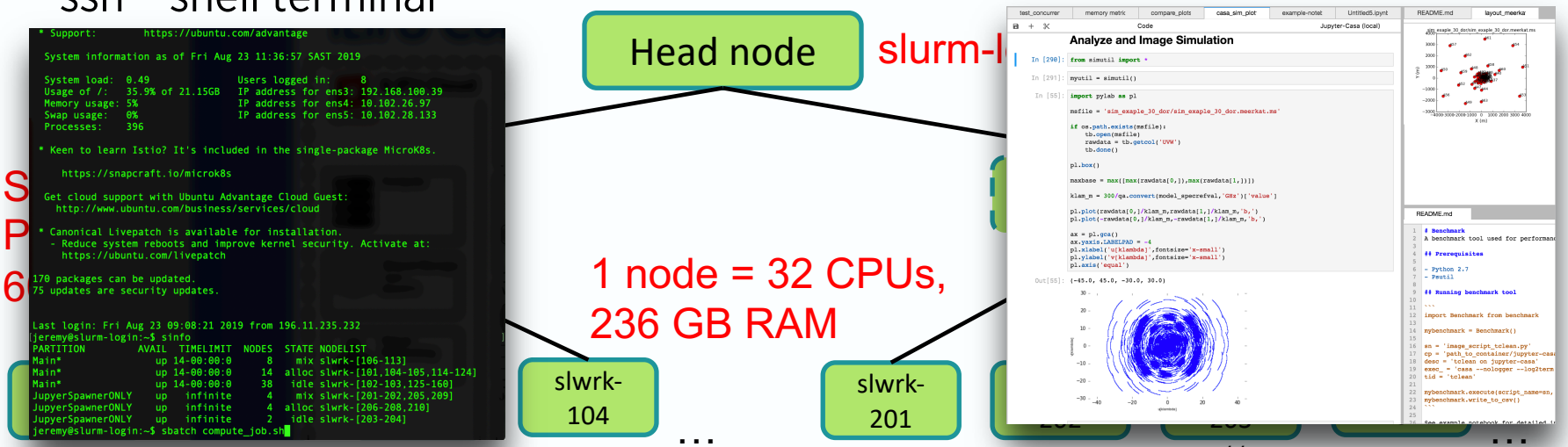


SLURM

- http://docs.ilifu.ac.za/#/getting_started/submit_job_slurm
- Head node (job submission & management)
- Partitions
- Compute nodes

ssh – shell terminal

JupyterHub



SLURM – user commands

- https://slurm.schedmd.com/

```
$ squeue #shows all jobs in slurm queue

$ squeue –u <username> #shows your jobs

$ sinfo #shows partitions and resources

$ sbatch slurm_job_script.sh #submit job to
slurm queue

$ sbatch –help #describes input parameters

$ scancel <jobid> #cancels job

$ scontrol show jobid <jobid> -dd #shows
useful information about your job
```

SLURM – running a job

```
$ cat slurm_job_script.sh
#!/bin/bash
#SBATCH --job-name='demo_job'
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --mem-per-cpu=4GB
#SBATCH --output=demo_job-%j-stdout.log
#SBATCH --error=demo-job-%j-stderr.log

echo "Submitting demo SLURM job"
singularity exec /idia/software/containers/SF-PY3-bionic.simg python myscript.py

$ sbatch slurm_job_script.sh #submit job to slurm queue
```

Describe job parameters/resources

container software script

what's being executed with above parameters

SLURM – running a job

```
$ cat slurm_job_script.sh
#!/bin/bash
#SBATCH --job-name='demo_job'
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --mem-per-cpu=4GB
#SBATCH --output=demo_job-%j-stdout.log
#SBATCH --error=demo-job-%j-stderr.log

echo "Submitting demo SLURM job"
singularity exec /idia/software/containers/casa-stable.img casa -c myscript.py

$ sbatch slurm_job_script.sh #submit job to slurm queue
```

Describe job parameters/resources

container software script

what's being executed with above parameters

SLURM – running an interactive job

- http://docs.ilifu.ac.za/#/getting_started/submit_job_slurm?id=interactive-slurm-session

```
$ salloc #opens bash shell session on compute node,
         #default 1 task, 1 CPU, 8GB RAM


$ salloc --cpus-per-task=4 --mem=64GB --pty singularity
exec /idia/software/containers/casa-stable.simg casa

         #opens interactive CASA session on compute node,
         #with 1 task, 4 CPUs, 64GB RAM,
         #using Singularity CASA container


$ salloc --help

         #view salloc help docs for input parameters
```

SLURM – running an interactive job

- With X11 support for viewing GUI

```
$ ssh-add –K / -k #this may need to be run before ssh, if you've copied keys from an old machine (-K for Mac OS X, -k for Linux)

$ ssh –YA <username>@slurm.ilifu.ac.za

#important to include the –YA parameters when logging into the SLURM login node

$ salloc --qos qos-interactive    #opens bash shell session on compute node with X11 support
                                  #default 1 task,1 CPU, 8GB RAM

$ salloc --cpus-per-task=4 --mem=64GB ––qos qos-interactive
```

SLURM – general advice

- Before running a large job, identify the available resources
  - Use sinfo
- Don't hog the cluster
  - Reduce your allocation if necessary / possible
- For interactive jobs, use salloc, or srun --pty bash
- Don't ssh onto the worker nodes!
  - This takes away from other peoples resources, even when they've reserved those resources with SLURM
  - We will kill your job if you ssh directly onto the node!
- Demo time!

IDIA MeerKAT Pipeline – A Good Framework

- Parallelised package for HPC processing (SLURM + cluster)
  - Uses multi-measurement sets (MMS) to parallelise across a cluster
- Outputs calibrated data (push of button!)
- HPC-friendly – dynamically uses resources & submits to queue
- Each job/script is a logical step that does / doesn't use MPI, and optionally uses a different container
- User can insert your scripts at start, middle or end
- https://idia-pipelines.github.io/docs/processMeerKAT
- Demo time!

CARTA

- Cube Analysis and Rendering Tool for Astronomy
  - IDIA (South Africa) – NRAO (US) – ASIAA (Taiwan)
  - Cloud-based Visual analytic of remote large image cubes
  - Supports many image formats: FITS, CASA, Miriad, and HDF5
- CARTA on ilifu
  - http://docs.ilifu.ac.za/#/astronomy/astronomy_software?id=carta
  - https://carta.idia.ac.za – login with same credentials as Jupyter
  - The only visible files are in /carta_share/
  - User should write IDIA HDF5 file to this filesystem
  - Even for continuum (single channel) image
    - `srun /carta_share/hdf_convert/run_hdf_converter image.fits /carta_share/users/<username>/image`
- Demo time!

THANK YOU

Dr Jordan Collier

ilifu Support Astronomer, IDIA
Department of Astronomy,
University of Cape Town

Adjunct Fellow, Western Sydney University
School of Computing, Engineering and Mathematics

Jordan@idia.ac.za
+27 664 343 953 (RSA Mobile)
+61 414 443 622 (AU Mobile / WhatsApp)