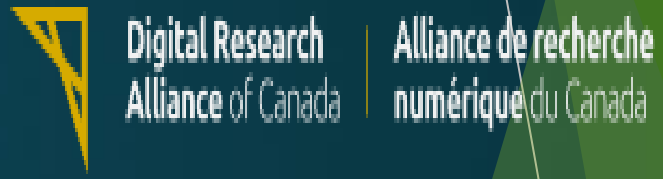
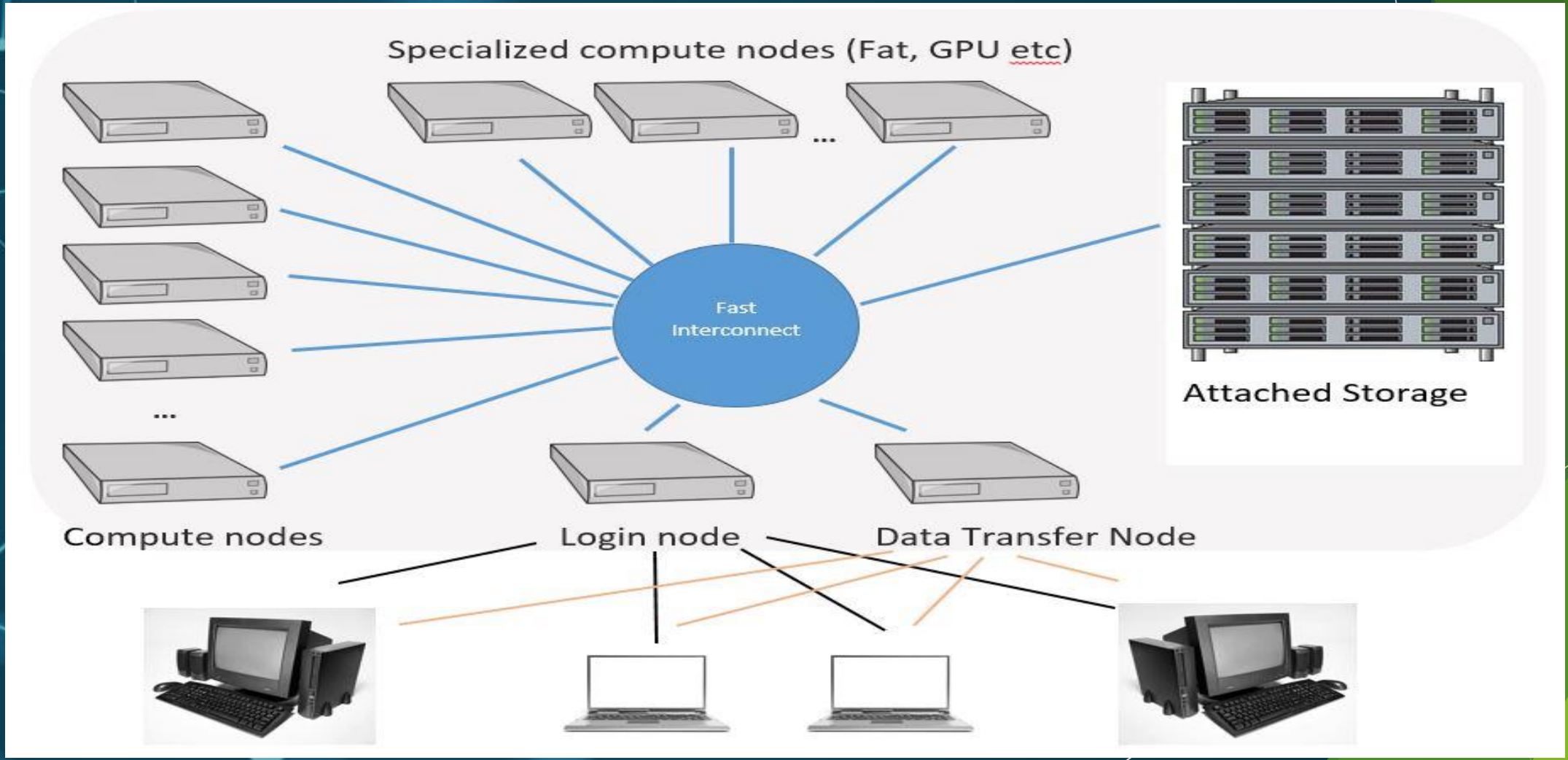
Best practices in HPC/HTC environments

Roman Baranowski UBC ARC

A bit of history and current/modern HPC/HTC architecture



HPC/HTC architecture



What do You as the user see ?

- Login nodes
- Data transfer nodes
- Compute nodes
- Special nodes (interactive, gpu, ’big’ mem, pre- and post processing [interactive], etc.)
- Data communication network

What You as the user DO NOT see ?

- Admin + Management nodes
- System/Services nodes
- File System serving nodes
- Special nodes (monitoring, provisioning, etc)
- Hardware servicing the network

All of that is shared !!!!

The performance and stability of a system affects YOU & US

What is HPC ?

tightly coupled parallel jobs requesting multiple nodes using MPI layer

What is HTC ?

independent, mostly sequential jobs (single node) that can be individually scheduled on many different computing resources

Login Nodes + DTN Nodes

- Interactive sessions (editing compiling etc.)
- Short testing (use batch for longer runs)
- Use Data Transfer nodes for data movement
- Globus !!!! (globus connect on your WS)
- Clean up when you are done……

Interacting with the global File Systems

Name	Disk space quota	Inode/Number of files quota
/home	50GB	0.5M
/project	1TB	0.5M
/nearline	2TB	0.5M
/scratch	20TB	1.0M

I see the performance issue magically resolved around midnight Saturday. We don't really know exact cause, but we suspect jobs from one user may have something to do with this. We will see if we hit the issue again.

How to interact with the FS ?

- Checkpoint your jobs
- Avoid excessive logging
- Do not create many small files (use local on the node disk i.e. $SLURM_TMPDIR)
- Some systems offer special FS for that
- If you are not sure talk to us PLEASE !!!

Software

- Use “module” utility
- List your modules in the job script
- Be consistent
- Always start fresh (slurm job inherits your current environment)

Scheduler

- Querying frequency of the scheduler
- Number of the jobs in the queue
- Walltime, resources requested
- Job Arrays
- Job dependencies

HTC job script

```
#!/bin/bash

#SBATCH --time=HH:MM:SS
#SBATCH --nodes=1
#SBATCH --mem-per-cpu=MMGb
#SBATCH --cpus-per-task=4          #  How many threads to start,  the best is to use all cores on the node
#SBATCH -- “All other options such as account, mail, output/error files etc”


module load “all you need”
module list

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

echo `hostname`

./Your_executable

exit
```

HPC job script

```
#!/bin/bash

#SBATCH --time=HH:MM:SS
#SBATCH --nodes=NN
#SBATCH --ntasks-per-node=NN    #  use as many MPI processes as there are cores on the node
#SBATCH --mem-per-cpu=MMGb
#SBATCH -- “All other options such as account, mail, output/error files etc”

module load “all you need”
module list

echo `hostname`

mpiexec  ./Your_executable


exit
```

Hybrid (MPI/threading) job script

```
#!/bin/bash

#SBATCH --time=HH:MM:SS
#SBATCH --nodes=NN
#SBATCH --ntasks-per-node=1    # How many MPI processes  to start per node
#SBATCH --cpus-per-task=4      #  How many threads to start,  the best is to use all cores on the node
#SBATCH --mem-per-cpu=MMGb
#SBATCH -- “All other options such as account, mail, output/error files etc”

module load “all you need”
module list

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

echo `hostname`

mpiexec   ./Your_executable   -pass $OMP_NUM_THREADS

exit
```

We appreciate Your Input !!!
Contact us

support@tech.alliancecan.ca

Contact Your Local Support Team

Let’s stay in touch and let’s talk

Thank You !!!

roman.baranowski@ubc.ca