Using the CICS computing clusters

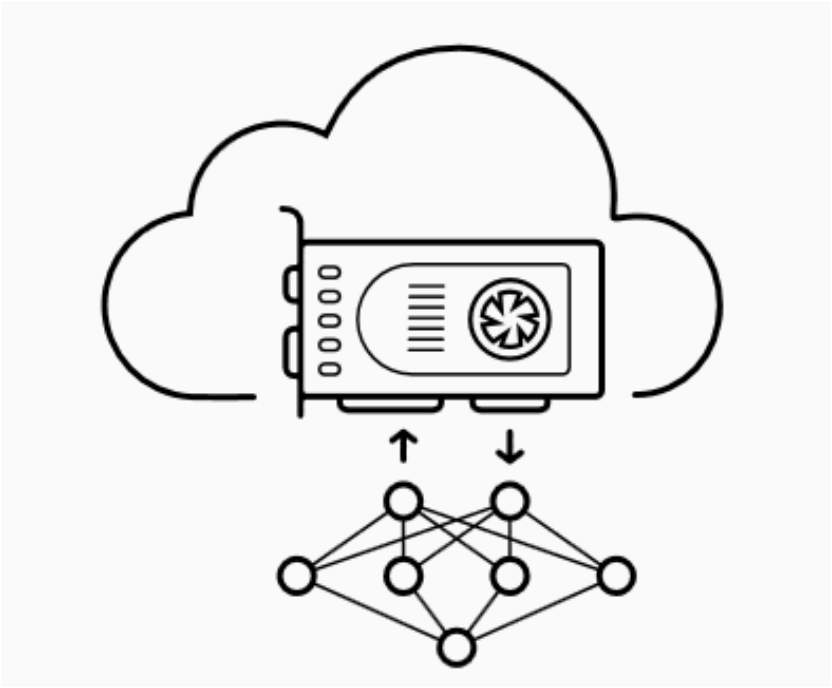


Image by Oleksandr Panasovskyi

Evangelos Kalogerakis

Our clusters

- The “Gypsum” Cluster: used for **GPU** computing
  MUST READ: https://gypsum-docs.cs.umass.edu/

- The “Swarm” Cluster: used for **CPU** computing
  MUST READ: https://people.cs.umass.edu/~swarm/

➢ **Mainly for research!**

CPU vs GPU computing

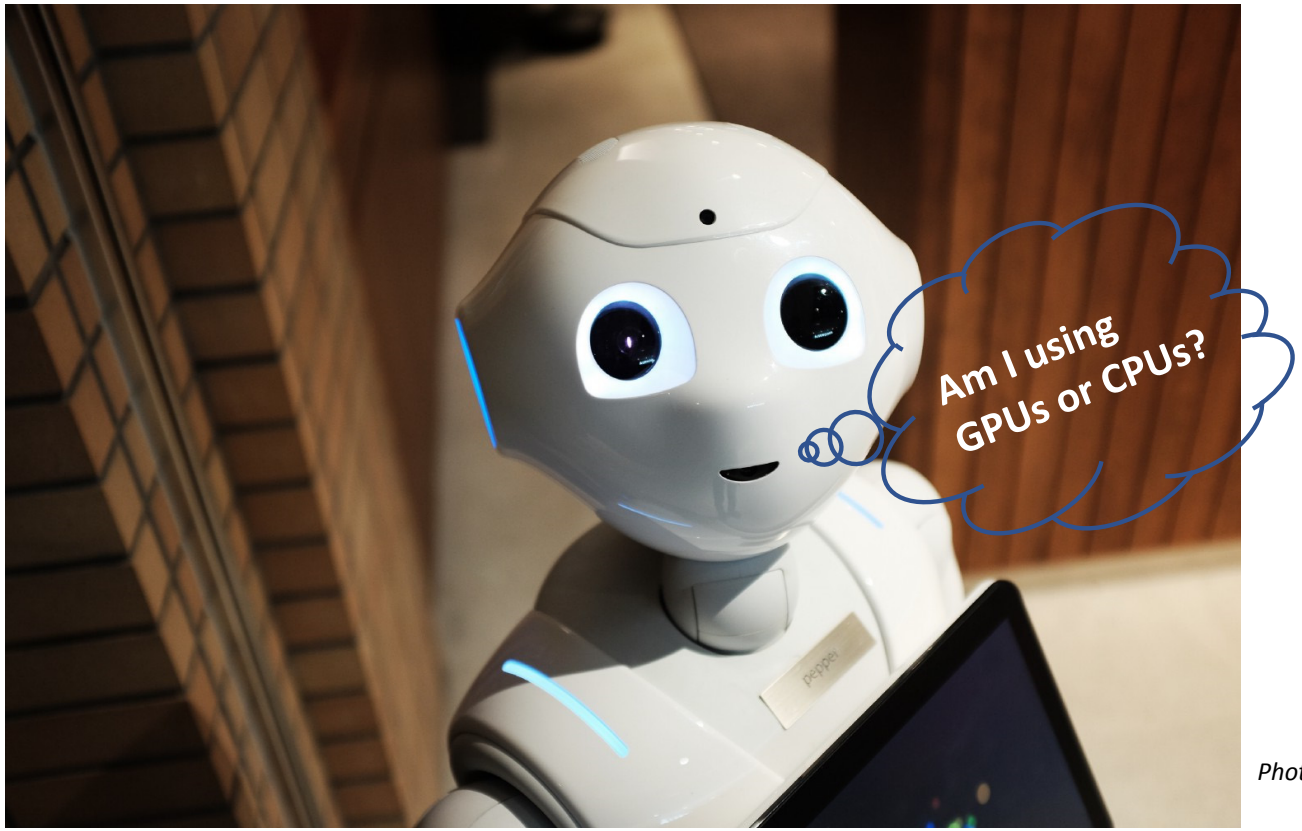
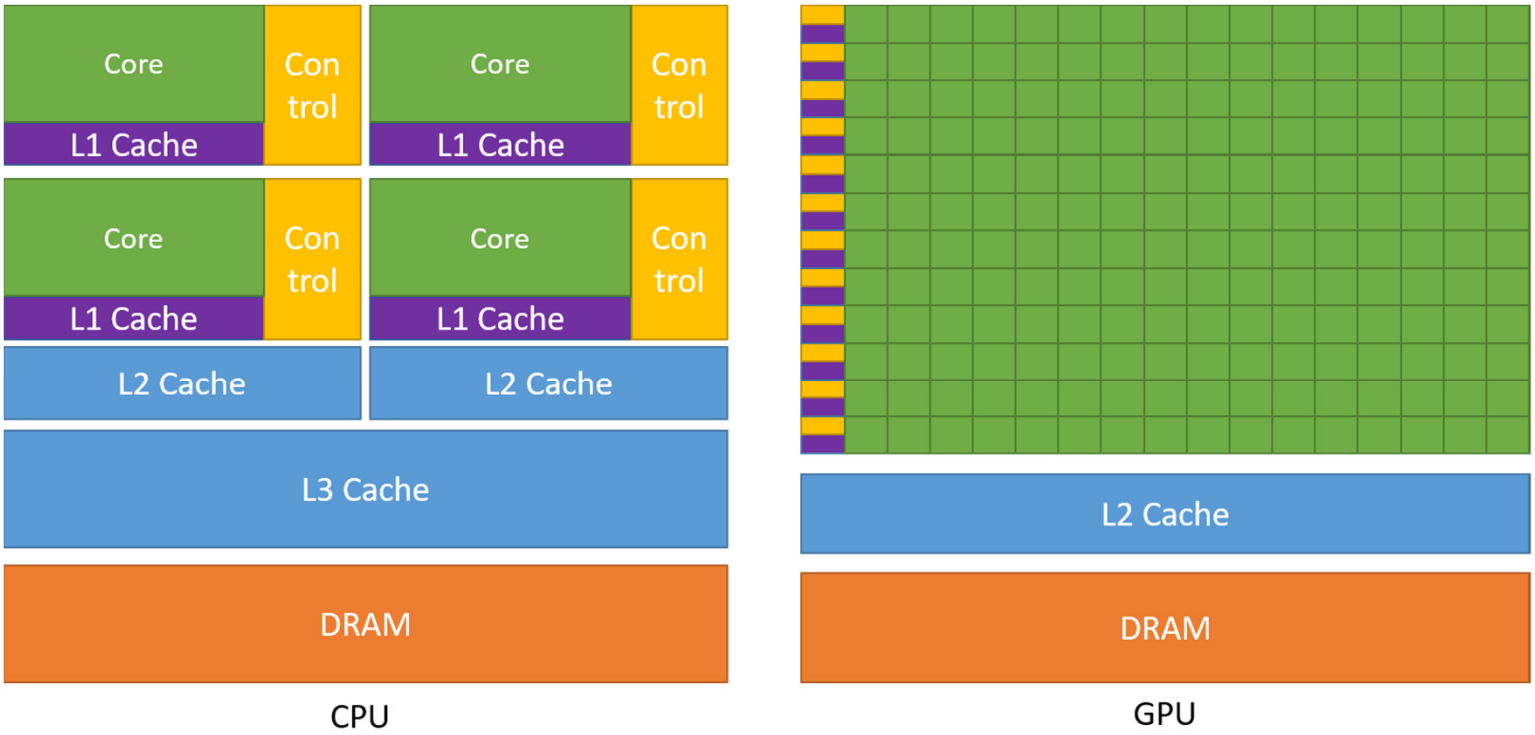


Photo by Alex Knight on Unsplash

CPU vs GPU computing

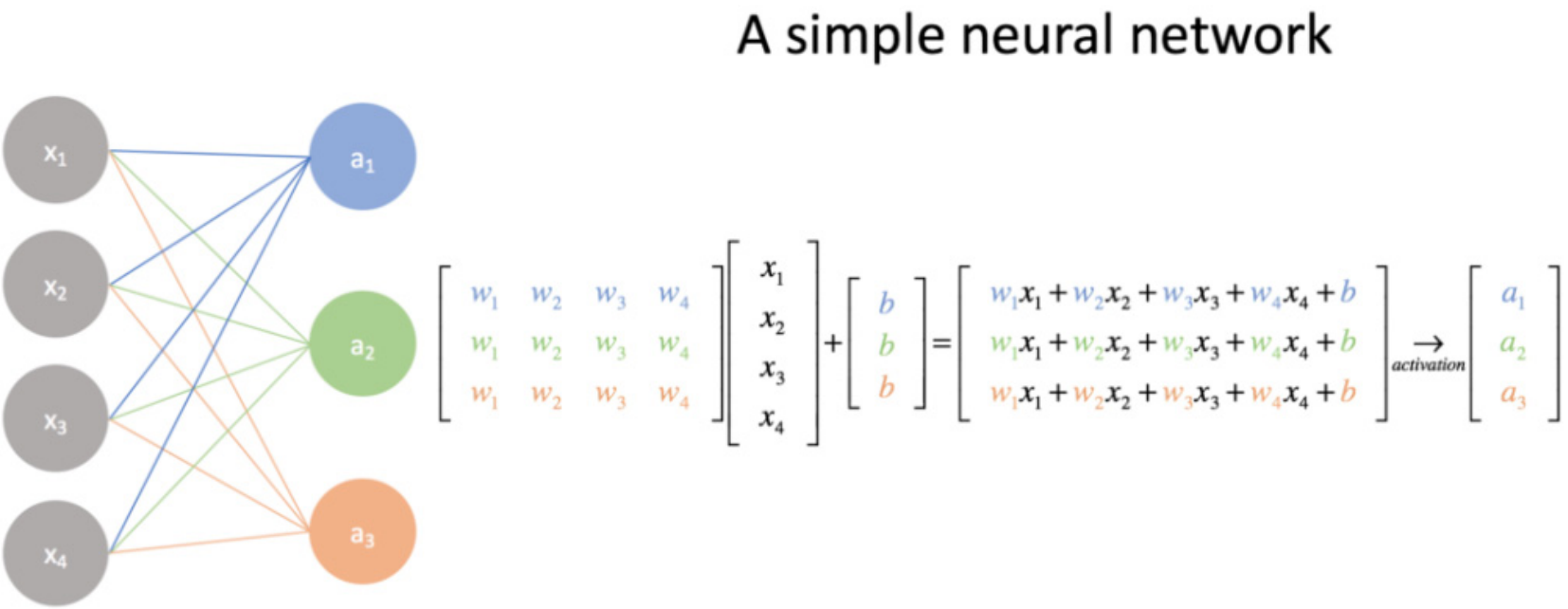


https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

CPU vs GPU computing

CPU	GPU
I have a **few complex cores**	I have **thousands of cores**
I can fetch data from my mem **quickly (latency-optimized)**	I can fetch **large chunks of data** from my mem **(bandwidth-optimized)**
I am good for **serial processing** with **complex control logic**	I am good for **parallel processing** with lots of **matrix calculations**
I often have **lots of mem available** (system RAM) (128GB+ in Swarm2)	I have **limited mem available** (dedicated video RAM) (<=24GB).

GPUs are particularly good for deep learning



Our inventory in more detail

The "Gypsum" GPU Cluster:

25 compute nodes with **4 NVIDIA Tesla M40 GPUs** each [*24GB dedicated mem*]

75 compute nodes with **4 NVIDIA TITAN X GPUs** each [12*GB dedicated mem*]

53 compute nodes with **8 NVIDIA GTX 1080 Ti GPUs** each [11*GB dedicated mem*]

25 compute nodes with **8 NVIDIA RTX 2080 Ti GPUs** each [11*GB dedicated mem*]

Upcoming:

+**5 compute nodes** with **8 NVIDIA RTX 2080 Ti GPUs** each [11*GB dedicated mem*]

+8 compute nodes with **8 Quadro RTX 8000 GPUs** each [48*GB dedicated mem*]

Our inventory in more detail

The "Gypsum" GPU Cluster:

25 compute nodes with **4 NVIDIA Tesla M40 GPUs** each [6.8 TFLOPS (FP32)]

75 compute nodes with **4 NVIDIA TITAN X GPUs** each [11 TFLOPS (FP32)]

53 compute nodes with **8 NVIDIA GTX 1080 Ti GPUs** each [11.3 TFLOPS (FP32)]

25 compute nodes with **8 NVIDIA RTX 2080 Ti GPUs** each [13.4 TFLOPS *(FP32)*]

Upcoming:

+**5 compute nodes** with **8 NVIDIA RTX 2080 Ti GPUs** each [13.4 TFLOPS (FP32)]

+8 compute nodes with **8 Quadro RTX 8000 GPUs** each [16.3 TFLOPS (FP32)]

Note (after talk): regarding half/mixed precision (FP16) performance also related to deep learning, you may check:
https://www.microway.com/knowledge-center-articles/comparison-of-nvidia-geforce-gpus-and-nvidia-tesla-gpus/

Our inventory in more detail

The "Swarm2" CPU Cluster:

50 compute nodes with 2x Xeon E5-2680 v4 CPU [28 phys. cores, 128GB RAM]

5 compute nodes with 2x Xeon E5-2680 v4 CPU [28 phys. cores, 256GB RAM]

50 compute nodes with 2x Xeon Gold 6240 CPU [36 phys. cores, 192GB RAM]

... check benchmarks e.g. PassMark for the performance of the above CPUs
(the Xeon Gold 6240 is a bit faster)

Our inventory in more detail

The "Swarm2" CPU Cluster:

50 compute nodes with 2x Xeon E5-2680 v4 CPU [56 logical cores (thr.), 128GB RAM]

5 compute nodes with 2x Xeon E5-2680 v4 CPU [56 logical cores (thr.), 256GB RAM]

50 compute nodes with 2x Xeon Gold 6240 CPU [72 logical cores (thr.), 192GB RAM]

... check benchmarks e.g. PassMark for the performance of the above CPUs
(the Xeon Gold 6240 is a bit faster)

Interested in using the **GPU** cluster?

- CICS faculty members:
  - You send an email to gypsum-admin@cs.umass.edu
  - Each researcher will be assigned with a group (gid). Resources (e.g., disk space) are allocated according to groups.

- CICS students / collaborators:
  - Your advisor sends an email to gypsum-admin@cs.umass.edu OR you send an email to the above address, and CC your CICS advisor **who needs to approve your request**
  - You are added to your advisor's group. Be mindful of the shared disk space with other people in your group.

Interested in using the **CPU** cluster?

- CICS faculty members:
  - You send an email to system@cs.umass.edu
  - Each researcher will be assigned with a group (gid). Resources (e.g., disk space) are allocated according to groups.

- CICS students / collaborators:
  - Your advisor sends an email to system@cs.umass.edu OR you send an email to the above address, and CC your CICS advisor **who needs to approve your request**
  - You are added to your advisor's group. Be mindful of the shared disk space with other people in your group.

Disk Space in our GPU cluster

Three key file systems on Gypsum:

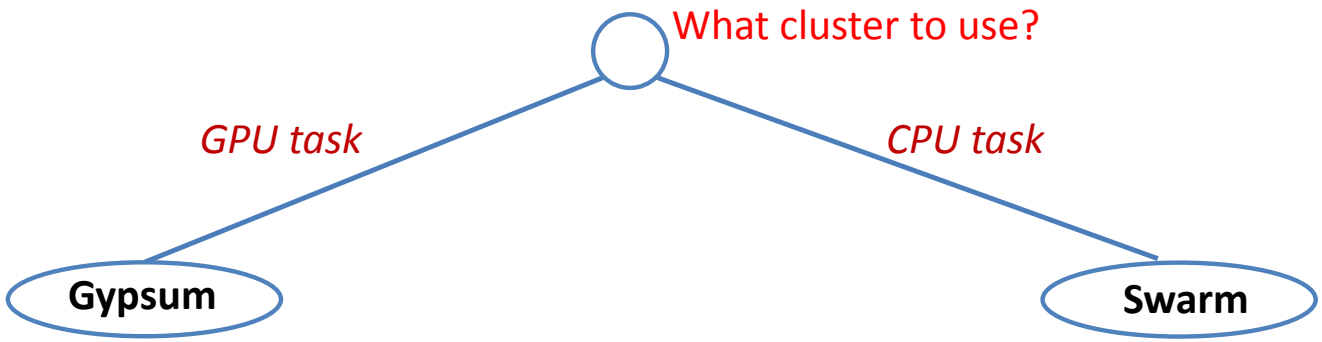
- /home/<user> [per user]
  - Backed up nightly
  - Uncompressed
  - User Quota: 100 GB
- /mnt/nfs/work1/<faculty>/<user> [shared within the group]
  - Backed up nightly
  - Compressed
  - Group Quota: 5TB
- /mnt/nfs/scratch1/<user> [per user]
  - NOT backed up
  - Compressed
  - User Quota: 1TB

Disk Space in our CPU cluster

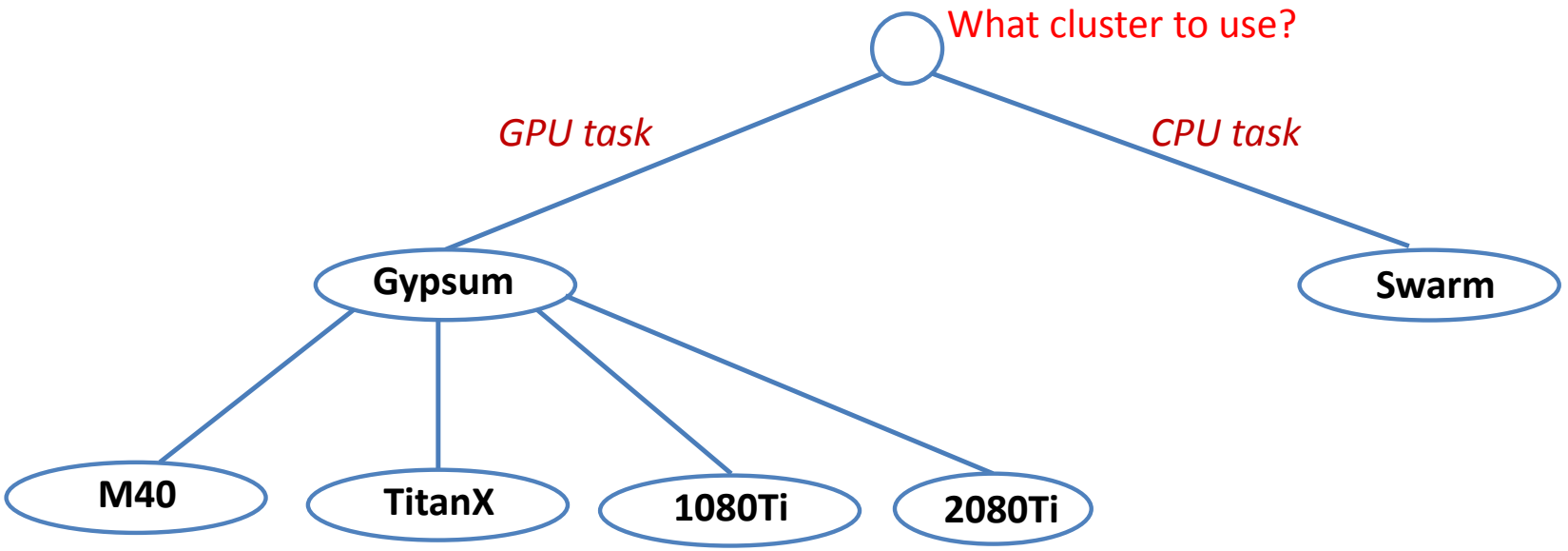
Three key file systems on Swarm2:

- /home/<user> [per user]
  - Backed up nightly
  - Uncompressed
  - User Quota: 10 GB
- /mnt/nfs/work1/<faculty>/<user> [shared within the group]
  - Backed up nightly
  - Compressed
  - Group Quota: 2TB
- /mnt/nfs/scratch1/<user> [per user]
  - NOT backed up
  - Compressed
  - No quota [yet, files should not stay there for more than 2 weeks]

The hardware decision tree

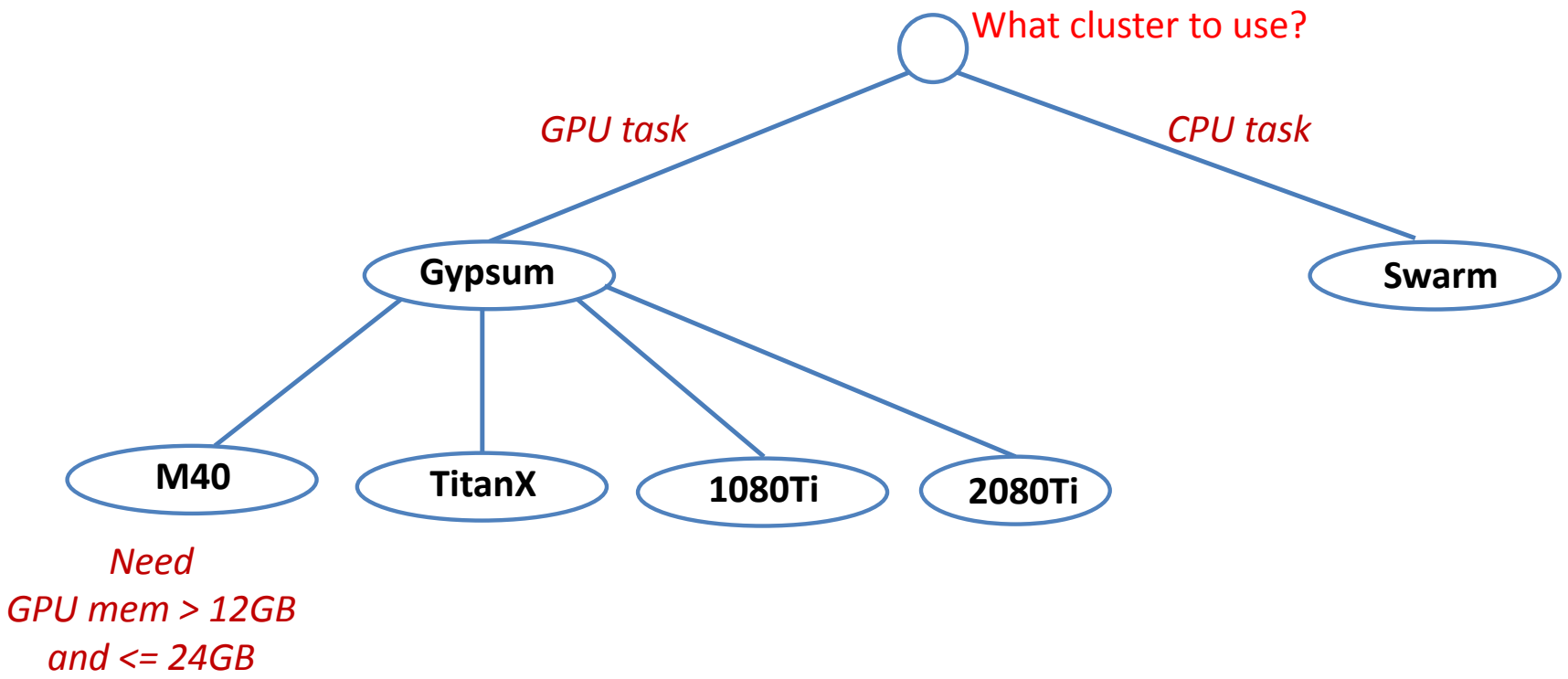


The hardware decision tree

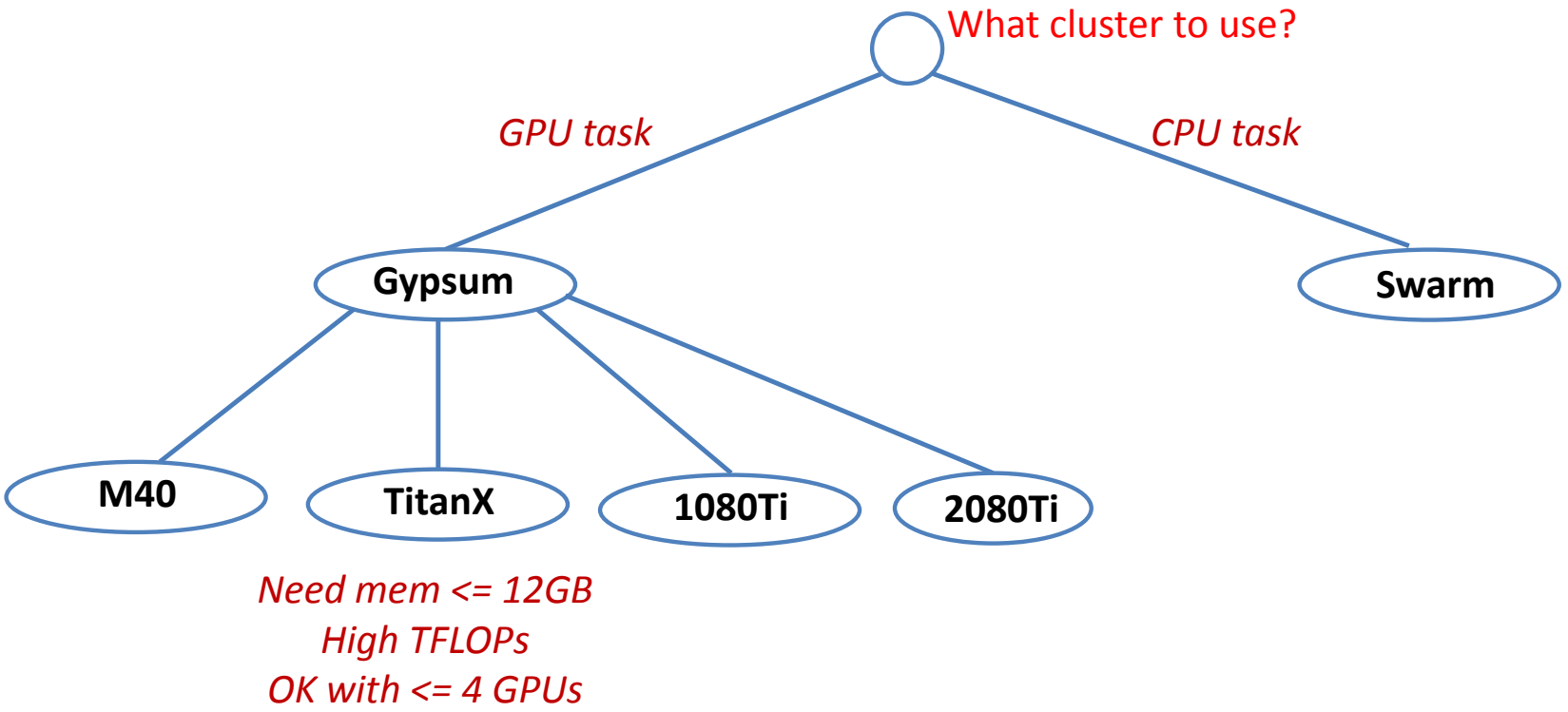


(a) Do I need 11GB/12GB/24GB GPU mem?
(b) Do I need lots of TLFLOPs?
(c) Do I need 4 or 8 GPUs? **Or less**?
(d) What GPUs are available?

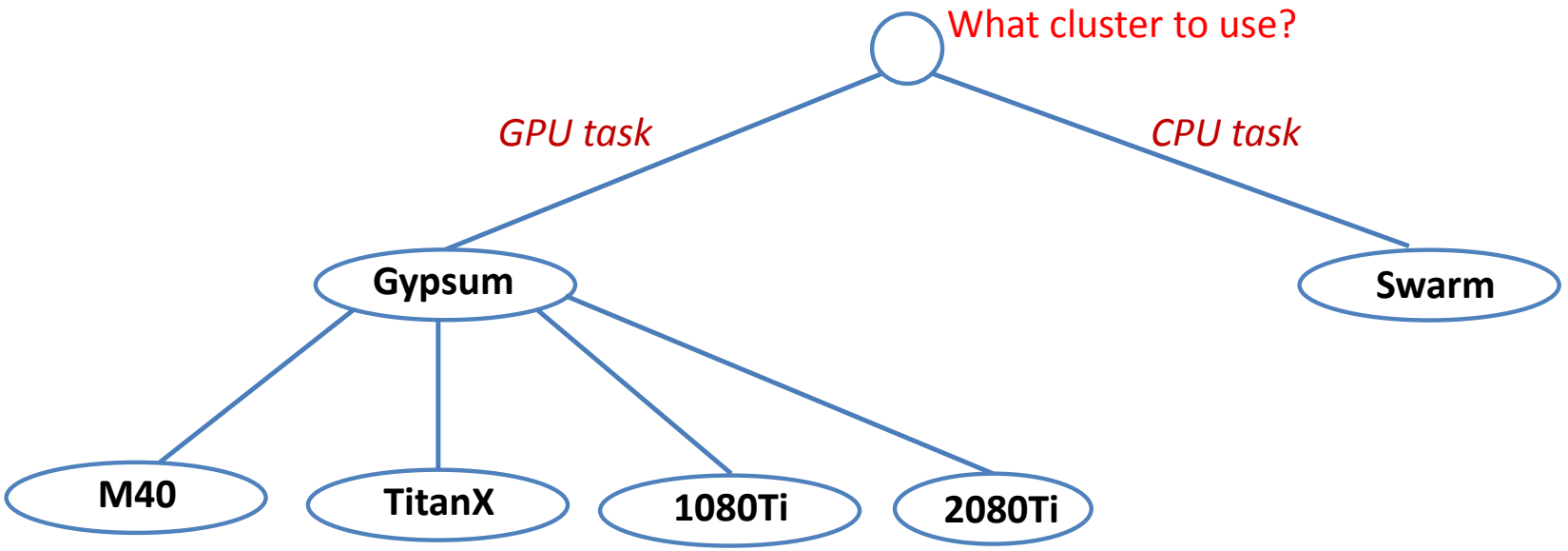
The hardware decision tree



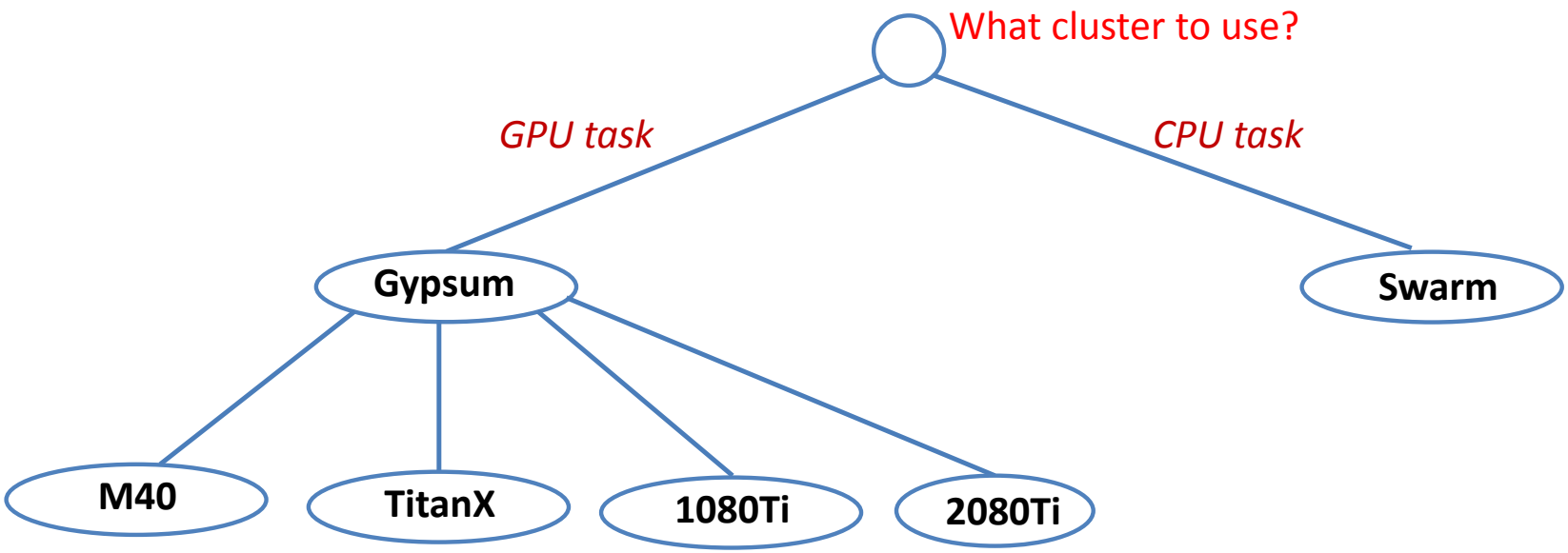
The hardware decision tree



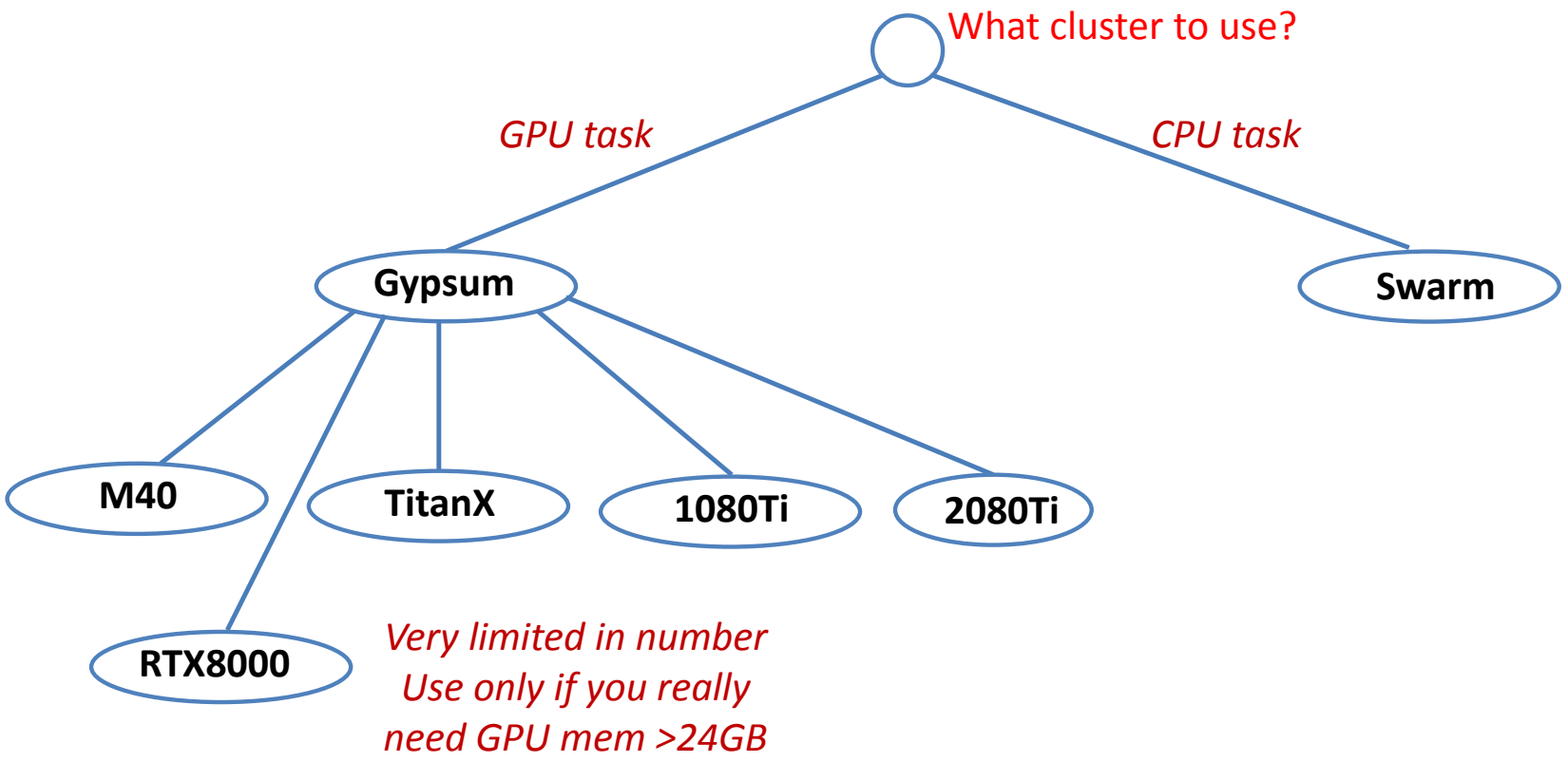
The hardware decision tree



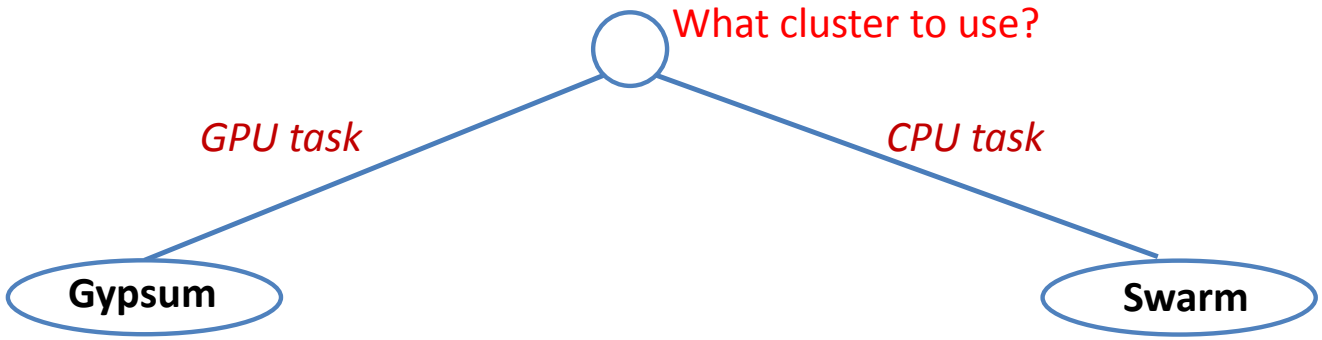
The hardware decision tree



The hardware decision tree



The hardware decision tree



(a) Do I need 128GB/192GB/256GB CPU mem?
(b) Do I need 28/36 cores
(Xeon Gold 6240 vs Xeon E5-2680)
(c) What CPUs are available?

...but how to examine which GPU/CPUs/nodes are available?

Resources and job scheduling are managed by **slurm**

MUST READ: https://slurm.schedmd.com/quickstart.html

There are **queues:** when a job is launched, it might run right away or be placed in a queue for some time depending on:

- the queue you asked for (another decision tree!)
- the resources you asked for your job (mem, #threads, #GPUs...)
- the resources your group and you have used so far

Time to connect to the clusters!

- Connect to Gypsum (GPU cluster):

```
ssh   gypsum.cs.umass.edu // within umass.edu OR use VPN
ssh   gypsum-gateway.cs.umass.edu   // accessible outside cs
                          // yet you need SSH keys for the gateway
                  // read: https://gypsum-docs.cs.umass.edu/ssh.html
```

- Connect to Swarm2 (CPU cluster):

```
ssh   swarm2.cs.umass.edu // also accessible outside cs
                          // [may change]
```

Now that you are connected, learn this:

You are connected to the **HEAD node** (both clusters)

The HEAD node is useful for establishing a connection, navigating to your directory, upload/download files, script editing.

Now that you are connected, learn this:

You are connected to the **HEAD node** (both clusters)

The HEAD node is useful for establishing a connection, navigating to your directory, upload/download files, script editing.

DO NOT RUN JOBS IN THE HEAD NODE

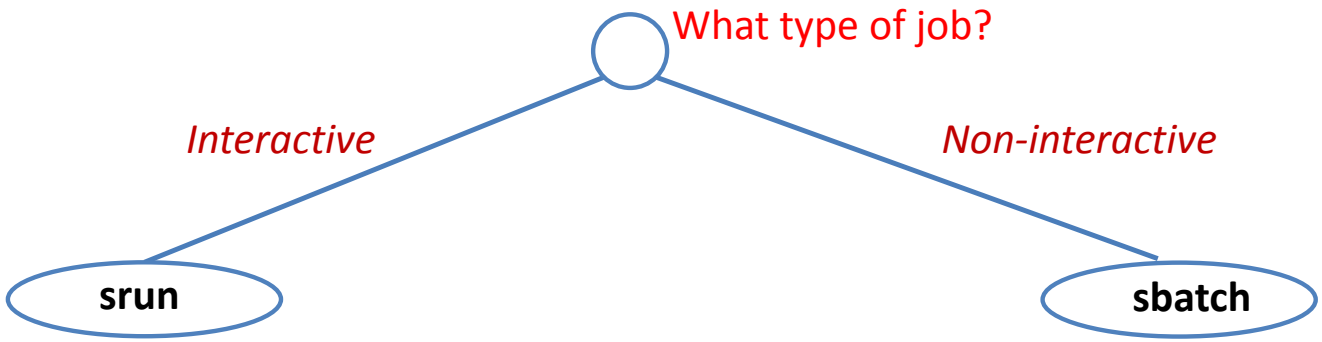
Now that you are connected, learn this:

You are connected to the **HEAD node** (both clusters)

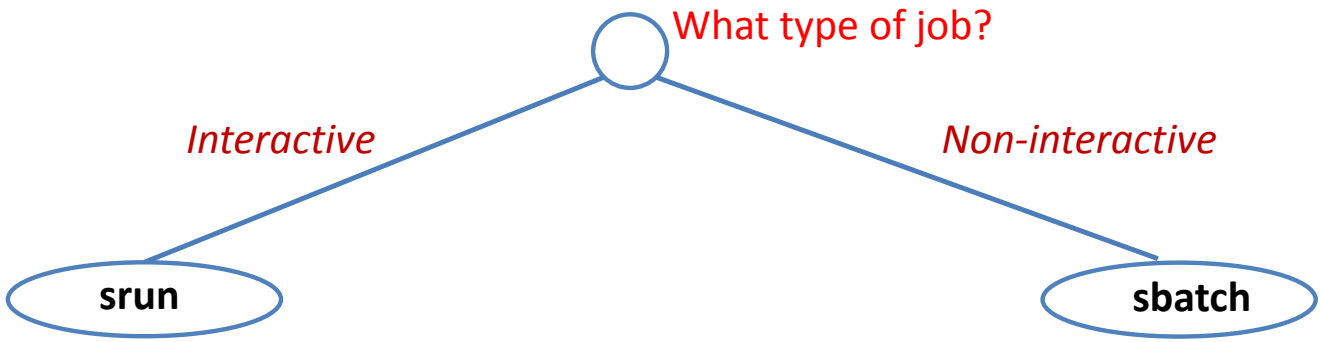
The HEAD node is useful for establishing a connection, navigating to your directory, upload/download files, script editing.

If you type: ./run-my-job-taking-few-min in the HEAD node, it is a no-no (you have to use slurm to launch jobs)

So want to **schedule** a job? (both clusters)



So want to **schedule** a job? (both clusters)



Interactive: useful for debugging (gdb), compiling, sequence of small tasks...

Non-interactive: useful for training a net, long tasks without user interaction

Interactive jobs (both clusters)

While in the head node, type:

```
srun --pty /bin/bash
```

```
srun --mem=10M python -c 'import os; print("hi
from", os.uname())'
```

Interactive jobs (both clusters)

While in the head node, type:

```
srun --pty /bin/bash
```

```
srun --mem=10M python -c 'import os; print("hi
from", os.uname())'
```

These are flags that you should know about. MUST CHECK:
https://slurm.schedmd.com/srun.html

... let's see a demo

Useful Flags for Gypsum

Resource	Flag Syntax	Description	Notes
partition	--partition=titanx-short	Partition is a queue for jobs	Default is titanx-short
time	--time=02-01:00:00	Time limit for the job	2 days and 1 hour; default is MaxTime for partition
#GPUs	--gres-gpu:2	Number of GPUs for the job	Default is 1. Code must also handle the multi-GPU setting
memory	--mem=2400	Memory limit per compute node for the job. Do not use with mem-per-cpu flag	memory in MB; default limit is 4096MB per logical core
memory	--mem-per-cpu=4000	Per logical core (logical 'CPU') memory limit. Do not use the mem flag	memory in MB; default limit is 4096MB per logical core
output file	--output=test.out	Name of file for stdout	default is the JobID

This is not enough for your code to use multiple GPUs. By default, DL libraries run on a single GPU!
MUST READ tensorflow: https://www.tensorflow.org/guide/distributed_training
MUST READ pytorch: https://pytorch.org/tutorials/beginner/blitz/data_parallel_tutorial.html

Resource	Flag Syntax	Description	Notes
partition	--partition=titanx-short	Partition is a queue for jobs	Default is titanx-short
time	--time=02-01:00:00	Time limit for the job	2 days and 1 hour; default is MaxTime for partition
#GPUs	--gres-gpu:2	Number of GPUs for the job	Default is 1. Code must also handle the multi-GPU setting
memory	--mem=2400	Memory limit per compute node for the job. Do not use with mem-per-cpu flag	memory in MB; default limit is 4096MB per logical core
memory	--mem-per-cpu=4000	Per logical core (logical 'CPU') memory limit. Do not use the mem flag	memory in MB; default limit is 4096MB per logical core
output file	--output=test.out	Name of file for stdout	default is the JobID

This is not enough for your code to use multiple GPUs. By default, DL libraries run on a single GPU!
MUST READ tensorflow: https://www.tensorflow.org/guide/distributed_training
MUST READ pytorch: https://pytorch.org/tutorials/beginner/blitz/data_parallel_tutorial.html

Resource	Flag Syntax	Description	Notes
partition	--partition=titanx-short	Partition is a queue for jobs	Default is titanx-short
time	--time=02-01:00:00	Time limit for the job	2 days and 1 hour; default is MaxTime for partition
#GPUs	--gres-gpu:2	Number of GPUs for the job	Default is 1. Code must also handle the multi-GPU setting
memory	--mem=2400	Per node memory limit. Do not use with mem-per-cpu flag	memory in MB; default limit is 4096MB per logical core
memory	--mem-per-cpu=4000	Per logical core (logical 'CPU') memory limit. Do not use the mem flag	memory in MB; default limit is 4096MB per logical core
output file	--output=test.out	Name of file for stdout	default is the JobID

DO NOT REQUEST MULTIPLE GPUs UNTIL YOU CHANGED YOUR CODE TO HANDLE MULTIPLE GPUs. DON'T WASTE GPUs.

This is system (main) memory – not dedicated GPU memory. Understand how much memory your code needs. Before you launch your job, run a profiler on your machine e.g., https://pypi.org/project/memory-profiler/ , or other monitoring tools (e.g., htop in linux)

Resource	Flag Syntax	Description	Notes
partition	--partition=titanx-short	Partition is a queue for jobs	Default is titanx-short
time	--time=02-01:00:00	Time limit for the job	2 days and 1 hour; default is MaxTime for partition
#GPUs	--gres-gpu:2	Number of GPUs for the job	Default is 1. Code must also handle the multi-GPU setting
memory	--mem=2400	Memory limit per compute node for the job. Do not use with mem-per-cpu flag	memory in MB; default limit is 4096MB per logical core
memory	--mem-per-cpu=4000	Per logical core (logical 'CPU') memory limit. Do not use the mem flag	memory in MB; default limit is 4096MB per logical core
output file	--output=test.out	Name of file for stdout	default is the JobID

This is system (main) memory – not dedicated GPU memory. Understand how much memory your code needs. Before you launch your job, run a profiler on your machine e.g., https://pypi.org/project/memory-profiler/ , or other monitoring tools (e.g., htop in linux)

Resource	Flag Syntax	Description	Notes
partition	--partition=titanx-short	Partition is a queue for jobs	Default is titanx-short
time	--time=02-01:00:00	Time limit for the job	2 days and 1 hour; default is MaxTime for partition
#			Default is 1. multi-GPU setting
memory	--mem=2400	Memory limit per compute node for the job. Do not use with mem-per-cpu flag	memory in MB; default limit is 4096MB per logical core
memory	--mem-per-cpu=4000	Per logical core (logical 'CPU') memory limit. Do not use the mem flag	memory in MB; default limit is 4096MB per logical core
output file	--output=test.out	Name of file for stdout	default is the JobID

DO NOT ASK FOR MORE SYSTEM MEMORY THAN WHAT YOU NEED!

Useful Flags for Swarm

Resource	Flag Syntax	Description	Notes
partition	--partition=defq	Partition is a queue for jobs	Default is defq
time	--time=02-01:00:00	Time limit for the job	2 days and 1 hour; default is MaxTime for partition
#nodes	--nodes=2	Number of compute nodes needed (distributed job)	Default is 1
#threads	--cpus-per-task=8	Number of logical cores (threads) needed (multi-threaded job)	Default is 1
memory	--mem=2400	Memory limit per compute node for the job. Do not use with mem-per-cpu flag	memory in MB; default limit is 4096MB per logical core
memory	--mem-per-cpu=4000	Per logical core (logical 'CPU') memory limit. Do not use the mem flag	memory in MB; default limit is 4096MB per logical core
output file	--output=test.out	Name of file for stdout	default is the JobID

Your code does not automatically use multiple nodes or threads. Read Justin's guidelines for multi-threaded and multi-core jobs:
https://people.cs.umass.edu/~swarm/index.php?n=Main.NewSwarmDoc

Resource	Flag Syntax	Description	Notes
partition	--partition=defq	Partition is a queue for jobs	Default is defq
time	--time=02-01:00:00	Time limit for the job	2 days and 1 hour; default is MaxTime for partition
#nodes	--nodes=2	Number of compute nodes needed (distributed job)	Default is 1
#threads	--cpus-per-task=8	Number of logical cores (threads) needed (multi-threaded job)	Default is 1
memory	--mem=2400	Memory limit per compute node for the job. Do not use with mem-per-cpu flag	memory in MB; default limit is 4096MB per logical core
memory	--mem-per-cpu=4000	Per logical core (logical 'CPU') memory limit. Do not use the mem flag	memory in MB; default limit is 4096MB per logical core
output file	--output=test.out	Name of file for stdout	default is the JobID

Your code does not automatically use multiple nodes or threads. Read Justin's guidelines for multi-threaded and multi-core jobs: https://people.cs.umass.edu/~swarm/index.php?n=Main.NewSwarmDoc

Resource	Flag Syntax	Description	Notes
partition	--partition=defq	Partition is a queue for jobs	Default is defq
time	--time=02-01:00:00	Time limit for the job	2 days and 1 hour; default is MaxTime for partition
#nodes	--nodes=2	Number of compute nodes needed (distributed job)	Default is 1
#threads	--cpus-per-task=8	Number of logical cores (threads) needed (multi-threaded job)	Default is 1
memory	--mem=2400	Memory in MB per node for the job. Do not use with mem-per-cpu flag	Default limit is 4096MB per logical core
memory per cpu	--mem-per-cpu=2400	Per logical core (logical 'CPU') memory limit. Do not use the mem flag	memory in MB; default limit is 4096MB per logical core
output file	--output=test.out	Name of file for stdout	default is the JobID

PLEASE DO NOT ASK FOR MULTIPLE CORES IF YOUR CODE USES ONE CORE/THREAD. CHANGE YOUR CODE TO HANDLE MULTI-THREADING OR DISTRIBUTED EXECUTION.

Your code does not automatically use multiple nodes or threads. Read Justin's guidelines for multi-threaded and multi-core jobs:
https://people.cs.umass.edu/~swarm/index.php?n=Main.NewSwarmDoc

Resource	Flag Syntax	Description	Notes
partition	--partition=defq	Partition is a queue for jobs	Default is defq
time	--time=02-01:00:00	Time limit for the job	2 days and 1 hour; default is MaxTime for partition
#nodes	--nodes=2	Number of compute nodes needed (distributed job)	Default is 1
#threads	--cpus-per-task=8	Number of logical cores (threads) needed (multi-threaded job)	Default is 1
memory	--mem=2400	Do not use with mem-per-cpu flag	memory in MB; default limit is 4096MB per logical core
		Per logical core (thread) memory limit. Do not use the mem flag	memory in MB; default limit is 4096MB per logical core
output file	--output=test.out	Name of file for stdout	default is the JobID

PLEASE DO NOT ASK FOR MULTIPLE CORES IF YOUR CODE USES ONE CORE/THREAD. CHANGE YOUR CODE TO HANDLE MULTI-THREADING OR DISTRIBUTED EXECUTION.

********* THIS ALSO HOLDS FOR GYPSUM!!! *********

Non interactive jobs

While in the head node, type:

```
sbatch -o my_output.txt --gres=gpu:2 my_experiment.sh
```

Similar flags. MUST CHECK:
https://slurm.schedmd.com/sbatch.html

Non interactive jobs

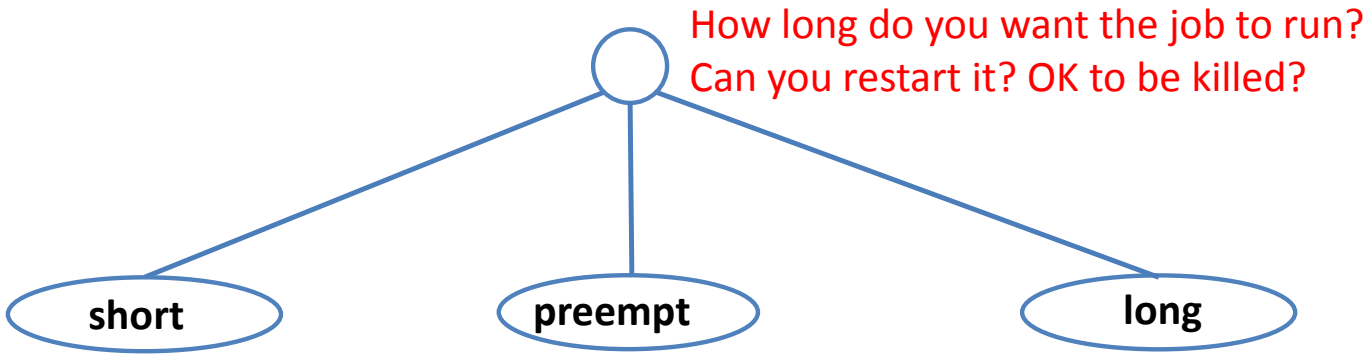
While in the head node, type:

```
sbatch my_experiment.sh
```

```
#!/bin/sh
#SBATCH -o my_output.txt
#SBATCH -e my_errors.txt
#SBATCH --time=12:00:00
#SBATCH --partition=1080ti-long
#SBATCH --gres=gpu:1
#SBATCH --mem=20GB
#SBATCH --cpus-per-task=2
./mynet/train.bin
```

... let's see a demo

The queue decision tree (Gypsum)

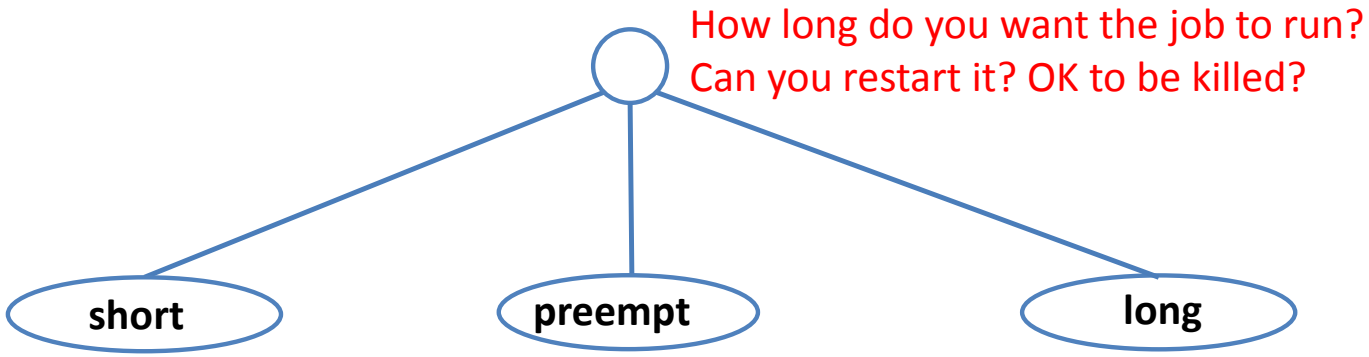


Short: Restricted to <=4hrs. **Less waiting. GPUs are faster to grab.**

Long: Restricted to <=7 days. **Use if you cannot restart your job** after 4hrs.

Preempt: Low priority jobs. Must be able to restart. Can run up to 28 days, but will be **killed instantly by pending jobs in other queues**!

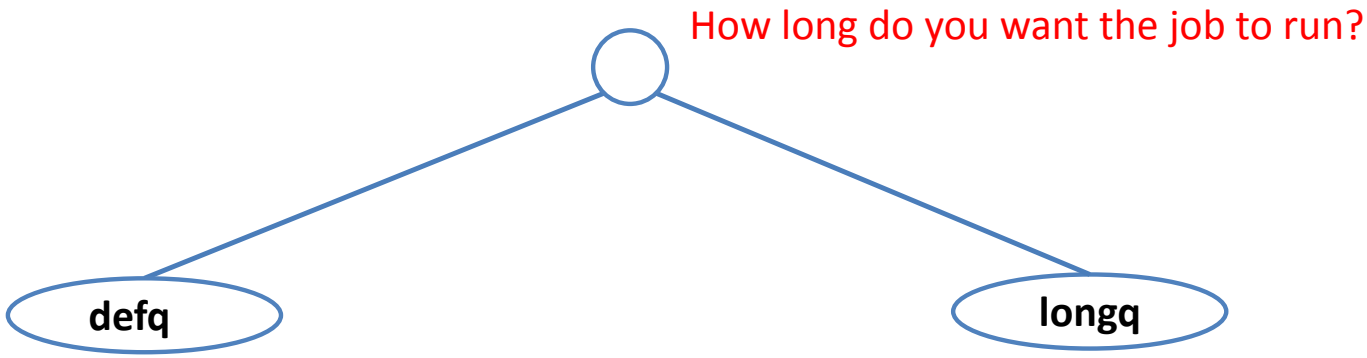
The queue decision tree (Gypsum)



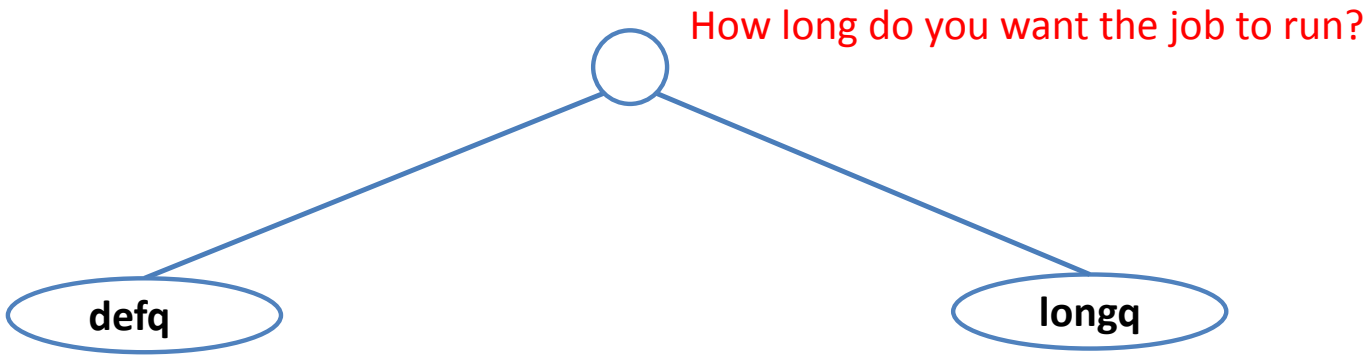
Selecting a queue involves (a) selecting desired GPU type (b) short/long/preempt kind:

titanx-long (40 jobs max), titanx-short (80 jobs max) [subject to change, PhD students/faculty]
m40-long (12 jobs max), m40-short (20 jobs max)
1080ti-long (40 jobs max), 1080ti-short (88 jobs max)
2080ti-long (40 jobs max), 2080ti-short (88 jobs max) [i think]
preempt (500 jobs max)
matlab (for matlab lovers!)

The queue decision tree (Swarm)



The queue decision tree (Swarm)



Defq: Restricted to <=12hrs. Default 2hrs. **Less waiting.**

Longq: Restricted to <=21 days. Default 2 days. **Use if you cannot restart your job**.

Queue names are defq, longq.

Saving a snapshot/checkpoint

In general, it is better to use short queues. You can re-launch your job every N hours, where N is the short queue restriction.

When your job is relaunched, it should automatically re-start the execution from a **saved state**. Your job should save the program state frequently.

PyTorch: https://pytorch.org/tutorials/beginner/saving_loading_models.html

Tensorflow: https://www.tensorflow.org/tutorials/keras/save_and_load

C++ Serialization: https://www.boost.org/doc/libs/1_74_0/libs/serialization/doc/tutorial.html

Saving a snapshot/checkpoint

In general, it is better to use short queues. You can re-launch your job every N hours, where N is the short queue restriction.

When your job is relaunched, it should automatically re-start the execution from a **saved state**. Your job should save the program state frequently.

Remember: group's disk space is not unlimited. Check disk usage. Save every 15min-1h.

PyTorch: https://pytorch.org/tutorials/beginner/saving_loading_models.html

Tensorflow: https://www.tensorflow.org/tutorials/keras/save_and_load

C++ Serialization: https://www.boost.org/doc/libs/1_74_0/libs/serialization/doc/tutorial.html

Useful slurm commands: sinfo

https://slurm.schedmd.com/sinfo.html

Information about nodes and partitions e.g., show idle nodes:

```
sinfo --state=idle

PARTITION     AVAIL  TIMELIMIT  NODES  STATE NODELIST
m40-short        up    4:00:00      8   idle node[013,019-025]
m40-long         up 7-00:00:00      8   idle node[013,019-025]
titanx-long      up 7-00:00:00      1   idle node060
titanx-short     up    4:00:00      1   idle node060
…
```

Useful slurm commands: sinfo

https://slurm.schedmd.com/sinfo.html

Information about nodes and partitions e.g., more information on CPU/GPUs/mem specs of nodes

```
sinfo -o  "%25N %10c %10m %5G" -e
```

```
NODELIST                  CPUS       MEMORY     GRES
node[001-046,048-100] 24         257864     gpu:4
node107                   48         385614     gpu:8
node133                   48         385599     gpu:8
...
```

Useful slurm commands: squeue

https://slurm.schedmd.com/squeue.html

Information about current jobs in queues e.g., all running jobs in a queue:

```
squeue --partition=titanx-short --states=R -o "%.18i
%.9P %.8j %.8u %.2t %.10M %.6D %R %5b %5c %10m"
```

```
JOBID   PARTITION   NAME     USER   ST        TIME  NODES NODELIST GRES MIN_CPU
6992679 titanx-sh  [redacted]  R       1:11      1 node043  gpu:1  1
6992679 titanx-sh  [redacted]  R       1:13      1 node066  gpu:1  1
6992679 titanx-sh  [redacted]  R       2:51      1 node041  gpu:1  1
6992679 titanx-sh  [redacted]  R       3:24      1 node066  gpu:2  5
```

Useful slurm commands: squeue

https://slurm.schedmd.com/squeue.html

Information about current jobs in queues e.g., all current jobs launched by a user:

```
squeue -u kalo -o "%.18i %.9P %.8j %.8u %.2t %.10M %.6D
%R %5b %5c %10m"
```

```
JOBID   PARTITION   NAME      USER   ST        TIME  NODES NODELIST GRES MIN_CPU
6992679 titanx-sh  [redacted]  R       1:11      1 node043  gpu:1  1
6992679 titanx-sh  [redacted]  R       1:13      1 node066  gpu:1  1
6992679 titanx-sh  [redacted]  R       2:51      1 node041  gpu:1  1
6992679 titanx-sh  [redacted]  R       3:24      1 node066  gpu:2  5
```

Useful slurm commands: squeue

https://slurm.schedmd.com/squeue.html

Information about current jobs in queues e.g., all current jobs of your group

```
squeue -o "%.18i %.9P %.8j %.8u %.2t %.10M %.6D %R %5b
%5c %10m %20g" | grep kalo
```

```
JOBID   PARTITION   NAME     USER   ST        TIME  NODES NODELIST GRES MIN_CPU
6992679 titanx-sh  [redacted]  R       1:11      1 node043  gpu:1  1
6992679 titanx-sh  [redacted]  R       1:13      1 node066  gpu:1  1
6992679 titanx-sh  [redacted]  R       2:51      1 node041  gpu:1  1
6992679 titanx-sh  [redacted]  R       3:24      1 node066  gpu:2  5
```

Useful slurm commands: squeue

https://slurm.schedmd.com/squeue.html

A courtesy from Mohit's group based on squeue:

`/mnt/nfs/work1/miyyer/scripts/slurm/gpu_counts.sh`

```
************
1080ti
************
COMPLETING=1
PENDING=167
RUNNING=278
************
2080ti
************
COMPLETING=5
PENDING=32
RUNNING=84
************
titanx
************
PENDING=156
RUNNING=225
************
m40
************
PENDING=3
RUNNING=52
```

Useful slurm commands: scontrol

https://slurm.schedmd.com/scontrol.html

Information about job status and configuration

```
scontrol show jobid | grep -B 1 -A 24 UserId=kalo
```

```
UserId=[redacted]GroupId=[redacted]MCS_label=N/A
   Priority=1 Nice=0 Account=[redacted] QOS=normal
   JobState=PENDING Reason=AssocMaxJobsLimit Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:00 TimeLimit=04:00:00 TimeMin=N/A
   SubmitTime=2020-09-24T20:26:19 EligibleTime=2020-09-24T20:26:19
   StartTime=Unknown EndTime=Unknown Deadline=N/A
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=titanx-short AllocNode:Sid=node057:8271
...
```

Useful slurm commands: sacct

https://slurm.schedmd.com/sacct.html

Accounting information for jobs invoked by your group

```
sacct  --allusers --gid kalo
--starttime 2020-09-24
--format=User,JobID,Jobname,partition,state,time,start,
end,elapsed,MaxRss,MaxVMSize,nnodes,ncpus,nodelist
```

User	JobID	JobName	Partition	State	Timelimit	Start	End	Elapsed	MaxRSS	MaxVMSize	NNodes	NCPUS	NodeList
kalo	6992016	bash	titanx-sh+	CANCELLED+	04:00:00	2020-09-24T13:49:32	2020-09-24T13:49:57	00:00:25			1	2	node041
kalo	6992064	bash	titanx-sh+	COMPLETED	04:00:00	2020-09-24T14:08:46	2020-09-24T14:08:48	00:00:02	1084K	297240K	1	2	node041
kalo	6992065	python3	titanx-sh+	FAILED	04:00:00	2020-09-24T14:08:54	2020-09-24T14:08:54	00:00:00	1068K	230676K	1	2	node041
kalo	6992073	python	titanx-sh+	FAILED	04:00:00	2020-09-24T14:09:23	2020-09-24T14:09:24	00:00:01	1068K	230676K	1	2	node041
kalo	6992074	python	titanx-sh+	COMPLETED	04:00:00	2020-09-24T14:09:39	2020-09-24T14:09:40	00:00:01	1068K	230676K	1	2	node041
kalo	6992241	test.sh	2080ti-sh+	COMPLETED	04:00:00	2020-09-24T15:15:54	2020-09-24T15:15:55	00:00:01			1	2	node157
	6992241.bat+	batch		COMPLETED		2020-09-24T15:15:54	2020-09-24T15:15:55	00:00:01	1268K	164504K	1	2	node157

Useful slurm commands: scancel

https://slurm.schedmd.com/scancel.html

Regretting a job? Please use scancel.
Don’t let useless jobs running!!!

scancel 6992927 `// cancel job with this id, find jobids with scontrol/squeue`

scancel -u `kalo // cancel all my jobs`

Modules

The Gypsum cluster uses Environment Modules: maintain multiple versions of compilers, libraries and applications for different users on the cluster.

Use the following commands to adjust your environment:

- module avail `// show available modules`
- module list `// show modules currently loaded`
- module add <module> `// adds a module to your env. for current session`
- module initadd <module> `// configure module to be loaded at every login`

MUST READ: https://gypsum-docs.cs.umass.edu/user.html

THE 10 COMMANDMENTS

- DO NOT RUN JOBS IN THE HEAD NODE
- DO NOT REQUEST MORE RESOURCES THAT WHAT YOU ACTUALLY NEED
- DO NOT GRAB MULTIPLE GPUS IN A NODE UNTIL YOU CHANGED YOUR CODE TO HANDLE MULTIPLE GPU PARALLELISM
- DO NOT GRAB MULTIPLE THREADS/CORES UNTIL YOUR CHANGED YOUR CODE TO SUPPORT MULTI-THREADED EXECUTION / DISTRIBUTED PARALLELISM
- DO NOT ASK FOR MORE SYSTEM MEMORY THAN WHAT YOU NEED
- DO NOT USE TESLA/QUADRO GPUs IF YOU NEED <=12GB OF GPU MEMORY.
- DO NOT RUN INTERACTIVE/SHORT JOBS IN LONG QUEUES
- DO NOT LET USELESS FILES STAY IN YOUR GROUP'S SPACE
- DO NOT PERFORM TOO FREQUENT I/O (SLOW!). PREFETCH DATA IN BIG CHUNKS.
- DO NOT KEEP USELESS JOBS RUNNING

Questions?

Note after talk: Nader Akoury & Kalpesh Krishna also pointed to these useful slides for you to check:

https://docs.google.com/presentation/d/17KqhMpe1NfZVsu1WVO9QB3yAsrhCcVxe10yqoK2D3Bg/edit#slide=id.p