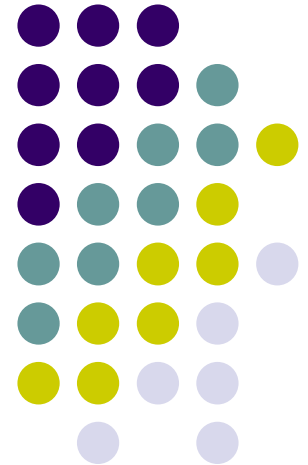


# Practical Parallel Computing (実践的並列コンピューティング)

## Appendix: Using Job Scheduler on TSUBAME

Toshio Endo  
endo@scrc.iir.isct.ac.jp





# Using Larger Resource

This material describes usage of larger TSUBAME resource  
(larger than 24 cores + 0.5GPU)

You can skip learning of this, since you can get credits in this lecture  
with “interactive usage” as usual

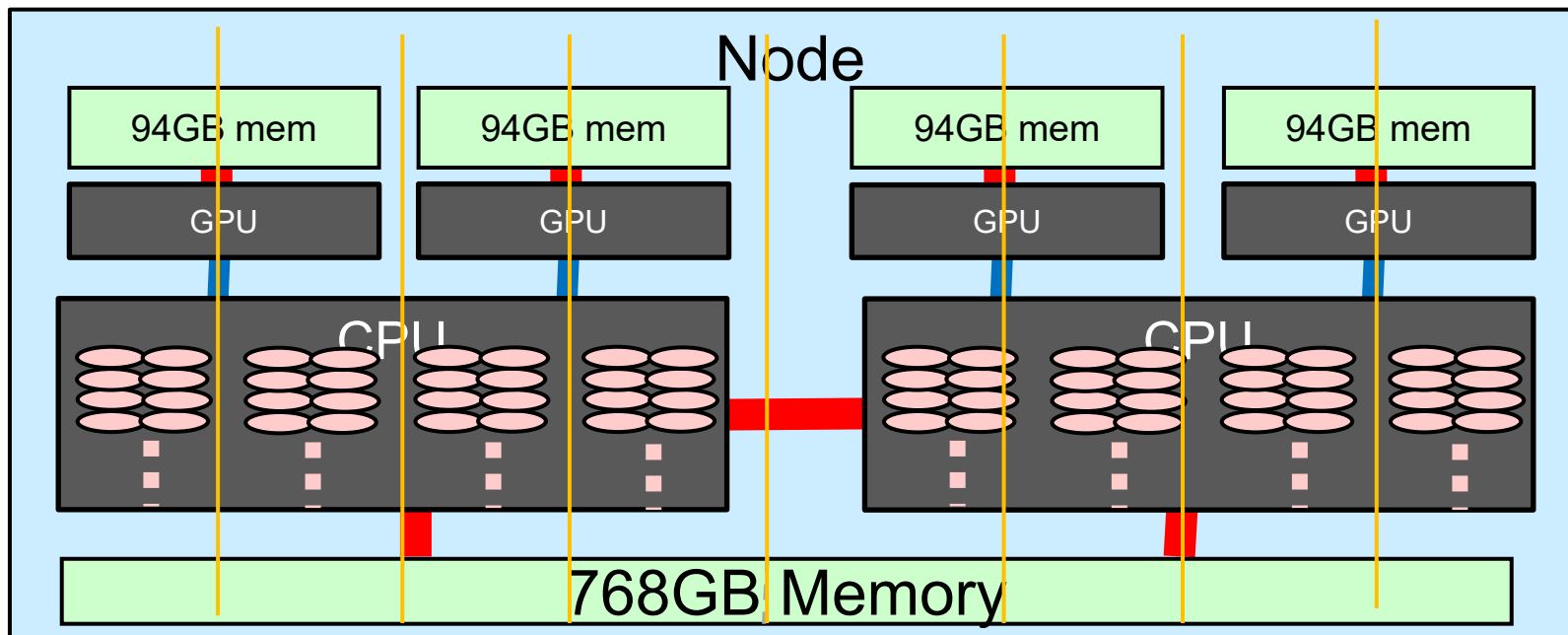
この資料ではTSUBAMEをより大規模に使う説明をする  
(24コア+0.5GPUを超える場合)

単位取得には普段の「インタラクティブ利用」で十分なので、この部分を省  
いてもよい

# Interactive Nodes on TSUBAME

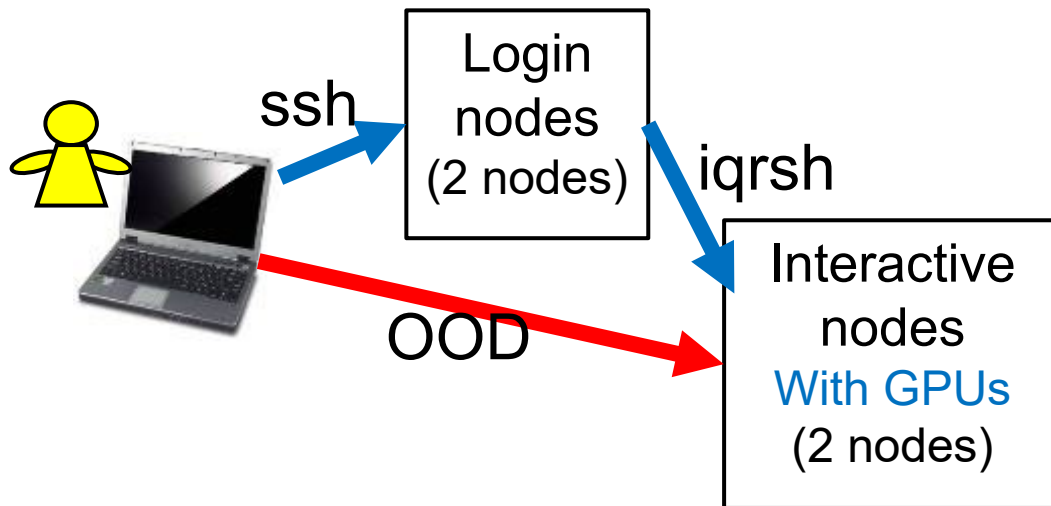


- In this lecture, “interactive nodes” are mainly used
  - 24 cores (48 hyper threads)+ 0.5 GPU
  - may be shared by several users





# Using Interactive Nodes

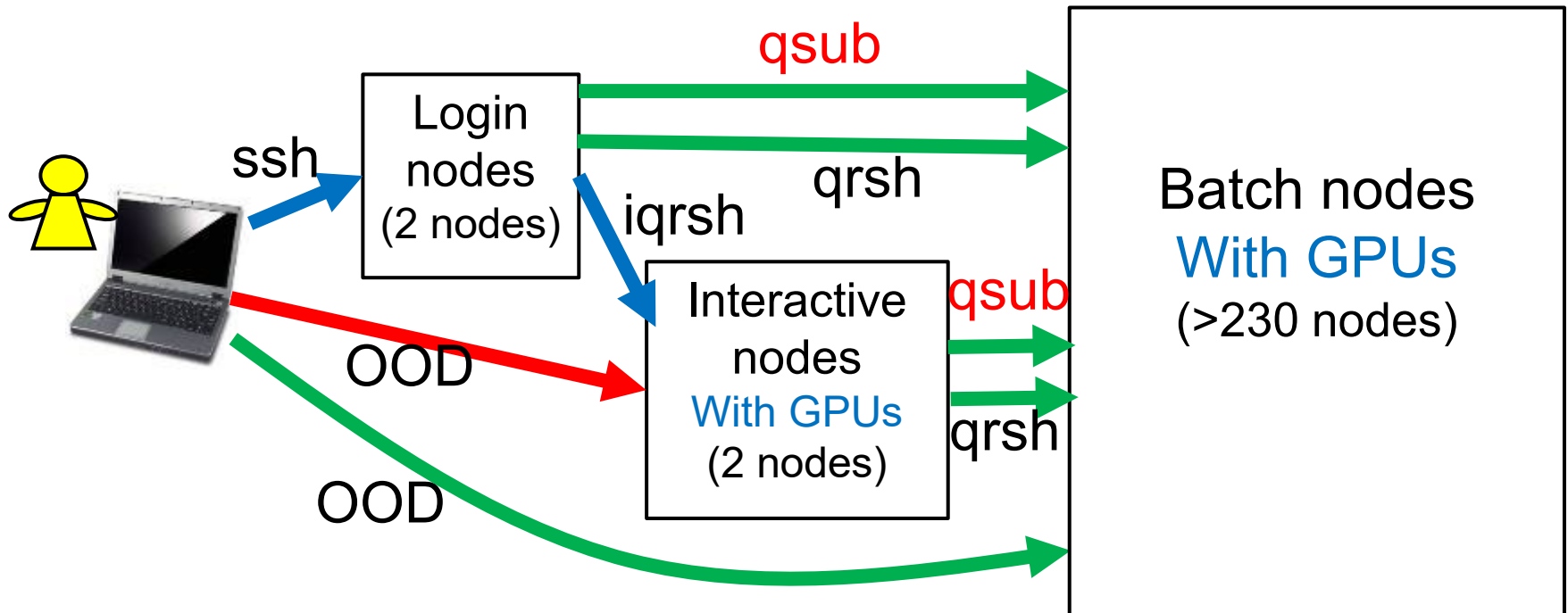


OOD = Open on Demand by web browser



# Using Batch Nodes

- Major parts of TSUBAME4 are “Batch nodes”
- There are several methods to use them



These slides focus on `qsub` command

# Using Batch Nodes via Job Scheduler



The **job scheduler** is a “gate” to use batch nodes in TSUBAME

With job scheduler,

😊 We can use more and dedicated cores

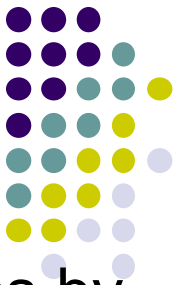
- With OpenMP, we can use up to 192 cores
- With MPI, we can use several nodes
- Cores are not shared with other users

😞 It is not “real-time” (with qsub)

😞 Take care of charge! (TSUBAME point)

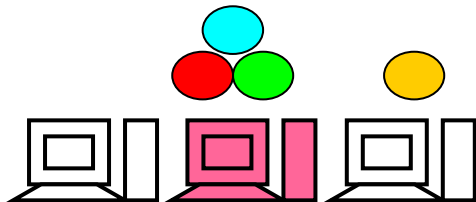
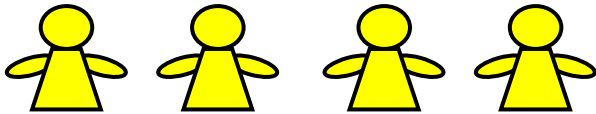
- TSUBAME points are limited shared among group members
- If one member uses up, other members cannot use scheduler 😞

# What is Job Scheduler?



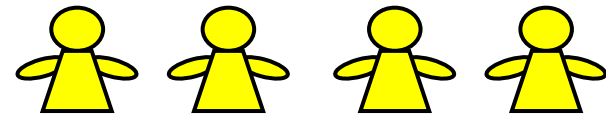
- The job scheduler does “traffic control” of many programs by many users
  - TSUBAME4.0 uses “Altair Grid Engine”

Without scheduler

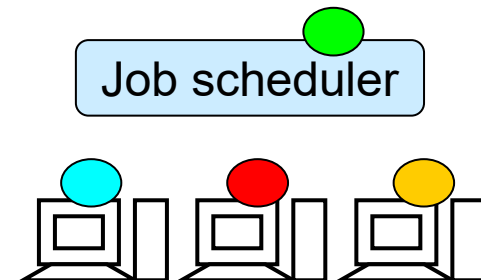


If users execute programs without control, there will be congestions

With scheduler



Job scheduler



Scheduler determines nodes for each job.  
Some program executions may be “queued”

# Overview of Job Submission

(Section 5 in TSUBAME4.0 User's Guide  
at [www.t4.gsic.titech.ac.jp](http://www.t4.gsic.titech.ac.jp))



Do the following on login nodes or interactive nodes

- (1) Prepare programs to be executed
- (2) Prepare a text file called **job script**, which includes
  - how the program is executed
  - resource (nodes/CPU) amounts required
- (3) Submit the job to the job scheduler with **qsub** command  
(and wait patiently)
- (4) Check the output of the job





# Prepare a Job Script

- In the case of `mm-omp` example
  - `ppcomp-ex/omp/mm`
- `job.sh` is a sample job script
  - Different file name is ok, but with “.sh”

job.sh

```
#!/bin/sh
#$ -cwd
#$ -l cpu_16=1
#$ -l h_rt=00:10:00

export OMP_NUM_THREADS=16
./mm 2000 2000 2000
```

Resource type and count:  
How many processor cores/  
nodes are allocated

Maximum run time

What are done on the  
allocated node



# Resource Types on TSUBAME4.0

- Choose one of resource types
  - It is like “instance types” in cloud systems
  - Please specify “proper” one
  - For example, if your program use 1 core, node\_f (192 cores) is too large (and expensive)

Resource type	Physical CPU cores	Memory (GB)	GPUs
node_f	192	768	4
node_h	96	384	2
node_q	48	192	1
node_o	24	96	0.5
gpu_1	8	96	1
gpu_h	4	48	0.5

In OMP/ACC/CUDA part, specify “1”

↓  
`#$ -l [resource_type] = [count]`

`#$ -l node_q=1`

↑ Allocates 48 cores and 1 GPU

`#$ -l node_f=1`

↑ Allocates 192 cores and 4 GPUs



# Resource Types (Full List)

Resource type	Physical CPU cores	Memory (GB)	GPUs
node_f	192	768	4
node_h	96	384	2
node_q	48	192	1
node_o	24	96	0.5
gpu_1	8	96	1
gpu_h	4	48	0.5
cpu_160	160	368	0
cpu_80	80	184	0
cpu_40	40	92	0
cpu_16	16	36.8	0
cpu_8	8	18.4	0
cpu_4	4	9.2	0

← largest

← same size with  
a partition of  
interactive node

← smallest



# Job Submission

- Job submission command

```
qsub job.sh
```

File name of the job script

- No charge (無料)
- But this works only when  $h\_rt \leq 0:10:00$  (10 minutes) and the number of resources must be  $\leq 2$

```
qsub -g [group-name] job.sh
```

- This works for longer jobs, but **Charged!** (有料)

Job ID

- You will see output like:

*Your job 123456 ("job.sh") has been submitted*

- If a job execution takes longer time, you have to specify a "TSUBAME group" name

# Notes in This Lecture



- First, please consider usage of interactive node (web usage/iqrsh)
- まずはインタラクティブノードの利用を検討してください (web usage/iqrsh)
- If necessary for reports, you can use up to 5.0 points in total per student. For more, please ask Endo
- 本講義のレポートの作成の目的で、一人あたり合計で5.0ポイントまで利用を認めます。より必要な場合は遠藤へ相談を
  - 5.0 points  $\doteq$  node\_f x 5 hours
  - You can check point consumption on TSUBAME portal
- The TSUBAME group name is [tga-ppcomp](#)



# Check Job's Outputs

- Where “mm” s outputs go to?
- When the job is executed successfully, two text files are generated automatically
  - File names look like
    - “job.sh.o123456” ← “stdout” outputs are stored
    - “job.sh.e123456” ← “stderr” outputs are stored

# Other Commands for Job Management



- `qstat`: To see the status of jobs under submission

```
qstat
```

job-ID	prior	name	user	state	submit/start at	...
123456	0.00000	job. sh	ua02023	qw	04/02/2024 09:19:30	

↖ Job ID

↖ **qw**: not started yet  
**r**: running now

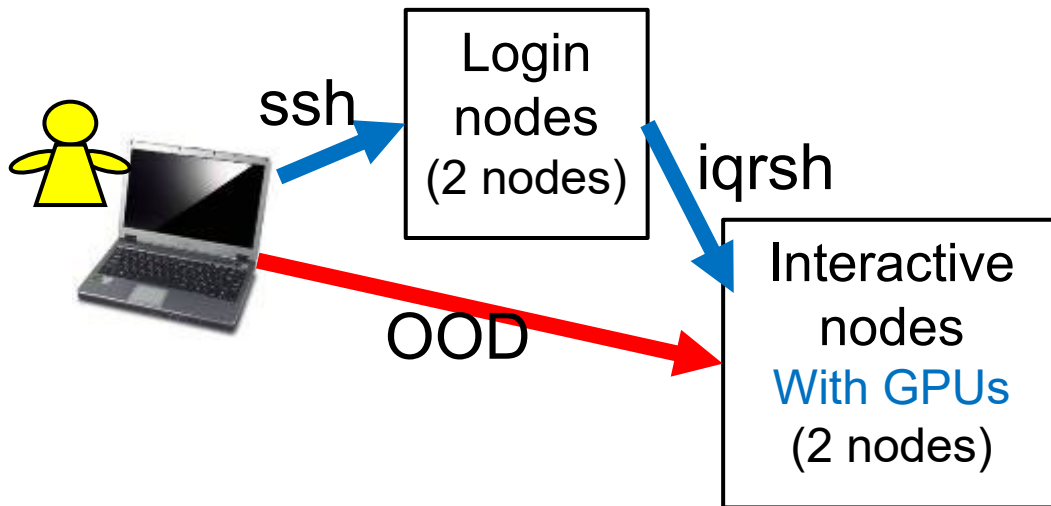
- `qdel`: To delete a job before its termination

```
qdel 123456
```

← Job ID

For interactive sessions, you can use `iqstat`, `iqdel` commands

# FYI: Information on Interactive Sessions



- Each user can use **only 1 interactive session** at same time
  - including session on OOD or iqrsh
  - If you try to invoke “second session”, it would cause an error, or you will wait very long





# Managing Interactive Sessions

- You can see and manage your interactive sessions (including OOD)
  - `iqstat`, `iqdel` can be used instead of `qstat`, `qdel`
  - They may be useful for trouble shooting

```
iqstat
```

job-ID	prior	name	user	state	submit/start at	...
13579	0.55500	QRLOGIN	ua02023	r	04/21/2025 06:57:30	

```
iqdel 13579
```

← Job ID



The followings are for MPI Part



# Sample Job Script for MPI job

- In the case of `mpi/mm` example (with Intel MPI)
  - `ppcomp-ex/mpi/mm`
  - `job.sh` is a sample job script

`job.sh`

```
#!/bin/sh
#$ -cwd
#$ -l cpu_16=2
#$ -l h_rt=00:10:00

module load intel-mpi

mpiexec -n 32 -ppn 16 ./mm 2000 2000 2000
```

Resource type and count:  
How many processor cores/  
nodes are allocated

Maximum run time

What are done on the  
allocated node

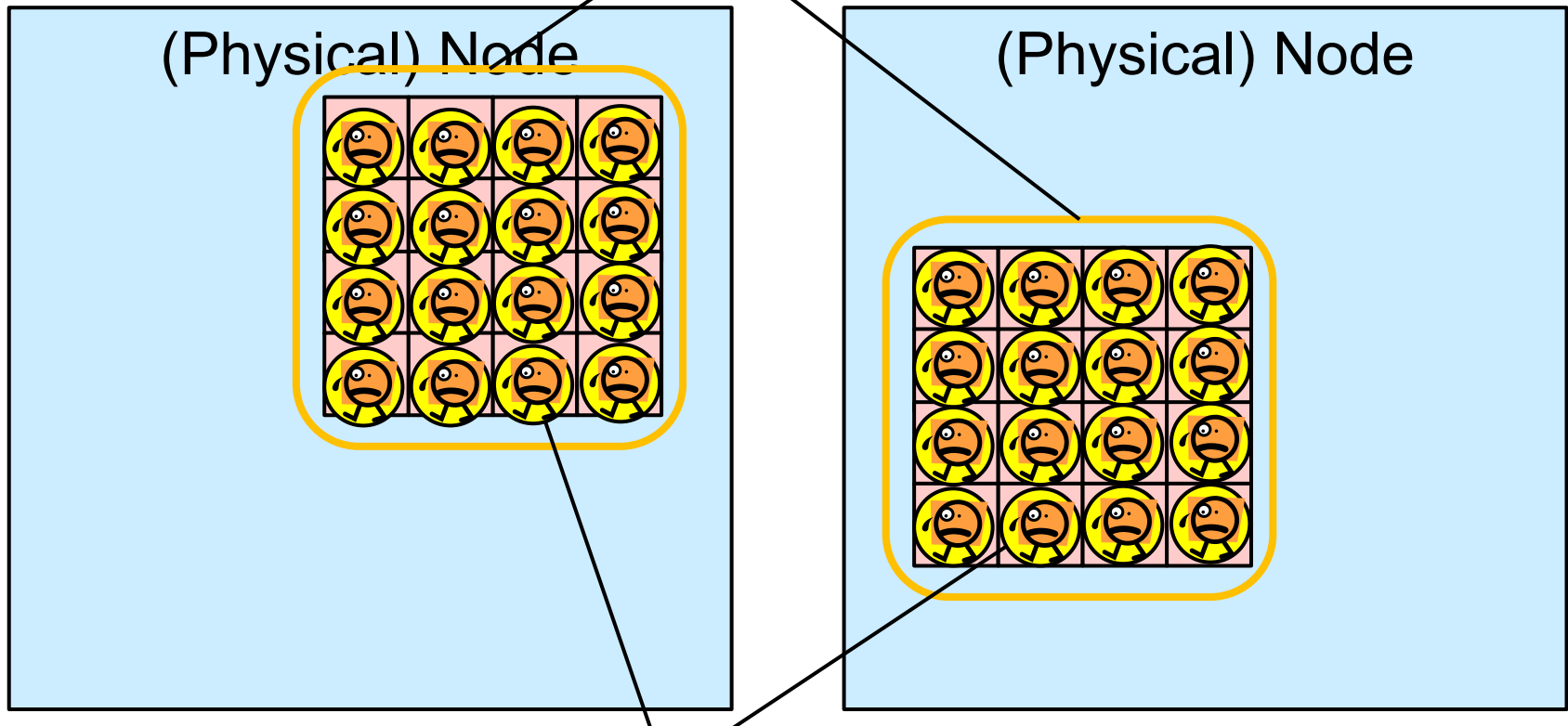
Number of  
processes

Number of  
processes  
per node

# What Happens on TSUBAME with mm-mpi/job.sh



(1) `-l cpu_16=2` → Job scheduler allocates 2 node partitions. Each has type `cpu_16` (with 16 CPU cores)



(2) `-n 32 -ppn 16` → `mpiexec` invokes 32 process, 16 processes per node (partition)