

Qmio Cheat Sheet

by Álvaro Caride (acaride) via cesga.es

Access	
connect	ssh USER@qmio.cesga.es

Slurm	
sinfo	Check node status
squeue	Information about your queue
sacct	Info about your accounting
scontrol	Deeper slurm info/operation
sbatch	Submit to queueing system
batchlim	Check queueing limits
myquota	Check storage space user quota
To request memory and a time limit is a must.	

Modules	
module av	Module availability
module list	List loaded modules
module load/unload	Load/unload module
module spider	Complete modules info and search tool
module purge	Clean all modules (but sticky)

A64 nodes	
48	cores
32GB	memory
sbatch -p a64	Submit to ARM partition
source /etc/profile.d/lmod.sh	Swaps to correct module tree for ARM
module load qulacs-hpcx	Qulacs module

Optimization	
OMP_PROC_BIND=True	Process binding
numactl -N 0-3	Include this before your command to increase performance
To get a interactive session type: salloc -p a64 -mem-per-cpu=600M -t 5:0:0 -c 48 srun -c 48 -pty --preserve-env /bin/bash -i	

x86 nodes	
64	cores
1TB	memory
compute	interactive session
module -r spider '.*qulacs.*'	Qulacs module info
ilk is the default partition.	

QPU	
sbatch -p qpu	Submit jobs to qpu partition
module load qmio-run	Load qmio module
pip install qmio	Install qmio module in your venv
from qmio import QmioRuntimeService	Python import
bk=QmioRuntimeService.backends(name)	name=qpu
bk.run(circuit, shots)	run command
Recommended use: with bk as backend: backend.run(circuit, shots)	

MPI-Qulacs	
QULACS_NUM_CORES	Overwrites OMP_NUM_CORES
MPIRUN_OPTIONS=-map-by <>	Increases performance
from mpi4py import MPI	Import python mpi library
QuantumState,QuantumCircuit	Import qulacs methods
QuantumState(nqubits, 1)	Initializa a distributed state vector
QuantumCircuit(nqubits)	Initialize quantum circuit object
circuit.add_H_gate(index)	Add h gate to circuit in qubit index
opt=QuantumCircuitOptimizer	Optimizer object
opt.optimize_light(circuit, swap_level)	swap_level 0:off, 1:add swaps or 2:add swaps and reorder gates
circuit.update_quantum_state(state)	Update state vector
state.sample(shots)	Sampling method