UM-Bridge models in the cloud and on classical HPC

UM-Bridge Workshop December 9, 2025

Chung Ming Loi[1]

[1] chung.m.loi@durham.ac.uk

Overview

Scaling up UM-Bridge Workloads

Goal

Automatically scale up UM-Bridge models for challenging UQ problems; keep UQ client simple

Scaling up UM-Bridge Workloads

Goal

Automatically scale up UM-Bridge models for challenging UQ problems; keep UQ client simple

Wishlist:

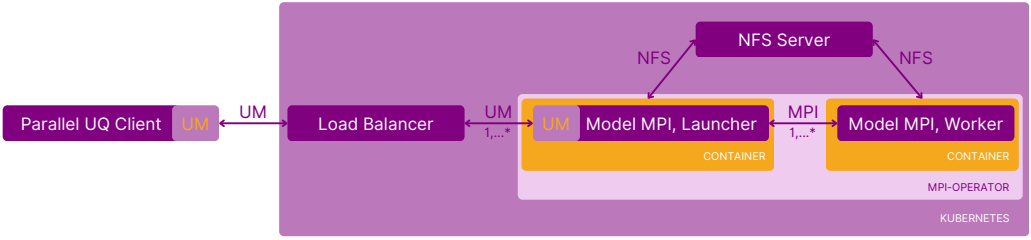
- Avoid machine specific model setup $\rightarrow$ containers
- Many instances of (in turn parallel) UM-Bridge models $\rightarrow$ HPC/cloud
- Automatic load balancing $\rightarrow$?

Kubernetes

kubernetes

- "Kubernetes, also known as K8s, is an open source system for automating deployment, scaling, and management of containerized applications."
- Setup defined through config files → reusable configuration for deploying UM-Bridge servers.

UM-Bridge servers in the Cloud



- Pre-built reference Kubernetes setup
- `mpioperator` base image needed for MPI support across containers (optional)

Scalability

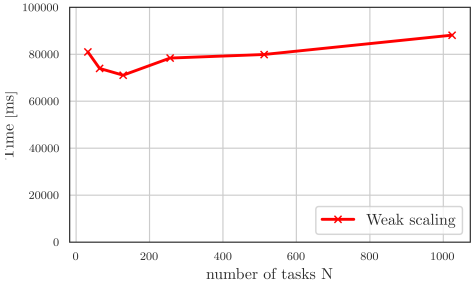


Figure: Weak scaling (more samples, more cores)

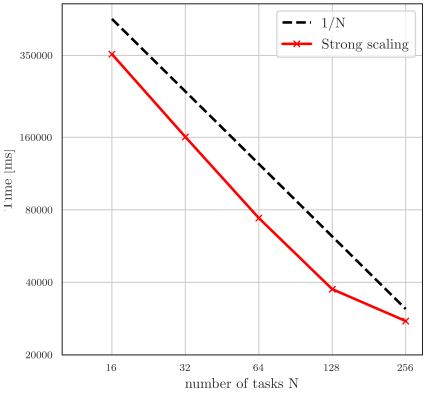


Figure: Strong scaling (constant number of samples, more cores)

Scaling up UM-Bridge Workloads

Goal

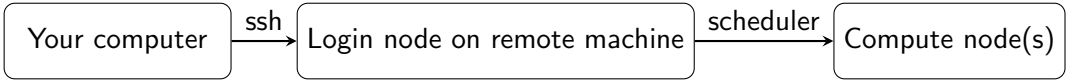
Automatically scale up UM-Bridge models for challenging UQ problems; keep UQ client simple

Wishlist:

- Avoid machine specific model setup → containers
- Many instances of (in turn parallel) UM-Bridge models → HPC/cloud
- Automatic load balancing → Kubernetes (for cloud)

But what about traditional HPC?

Remote access



- `ssh` provides secure connection to remote machine
  `ssh <username>@<server_ip>`
- `scp` to move files between host and remote machine
- Resources on HPC systems are allocated by schedulers → more later!

The `Module` system

- Almost impossible to get `sudo` rights as a normal user!
  - $\Rightarrow$ Use pre-installed packages by admin or build from source
- `module avail` lists all available modules
- `module load` initialises environment for various software/tools
- E.g. compilers, MPI, Python, Matlab etc...

The Module system

- Almost impossible to get sudo rights as a normal user!
  - ⇒ Use pre-installed packages by admin or build from source
- `module avail` lists all available modules
- `module load` initialises environment for various software/tools
- E.g. compilers, MPI, Python, Matlab etc...

What happens when you sudo...

```
(python3.9) mghw54@login2:~/nobackup$ sudo vi
[sudo] password for mghw54:
mghw54 is not in the sudoers file.  This incident will be reported.
```

HPC Schedulers

- Schedulers implements a batching system
- Submit compute intensive tasks to the scheduler
- Scheduler allocates requested compute resources to submitted jobs
- It also decides the priority of jobs based on several scheduling algorithms.
  $\Rightarrow$ Your jobs may not begin immediately!
- Several options SLURM, PBS, LSF etc. We focus on SLURM.

Example SLURM Script

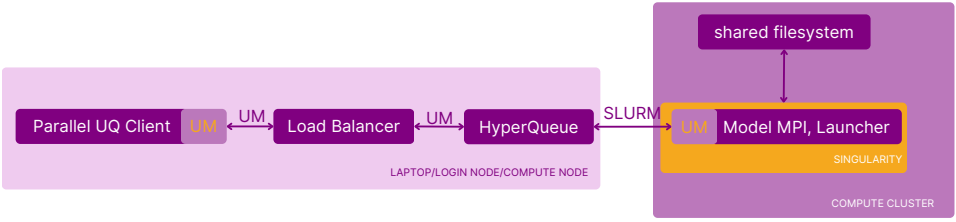
```
#!/bin/bash
#SBATCH -p test
#SBATCH -t 00:01:00
#SBATCH --nodes=1
#SBATCH --ntasks=1

module load gcc
gcc test.c -o myExecutable
./myExecutable
```

Submit to SLURM using `sbatch <job_script>`

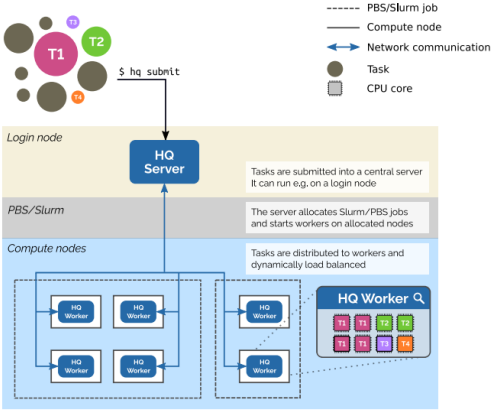
UM-Bridge: HPC Load-balancer

- Sends resources request to scheduler
- Distribute parallel client requests to model servers
- Client sees it as a server, server sees it as a client
- Can run on login node or compute node



HyperQueue

- Plugin scheduler to native HPC scheduler
- Designed to handle loosely coupled small jobs
- Does not require admin privilege
- Available as statically-linked binary



UM-Bridge: Load-balancer backends

Two load-balancer backends

- SLURM
  - Easier to setup
  - Higher job overhead
- HyperQueue
  - More difficult to setup
  - Lower job overhead
  - More flexibility per job

Compile load-balancer with `make` (Makefile included)
Execute load-balancer binary with

```
./loadbalancer --scheduler=<scheduler> --port=<port>
```

UM-Bridge: Load-balancer SLURM script

```
#! /bin/bash

#SBATCH --partition=devel
#SBATCH --ntasks=1
#SBATCH --time=00:05:00

# Launch model server, send back server URL and wait so that SLURM does not
# cancel the allocation.

[...]

# Assume that server sets the port according to the environment variable 'PORT'.

./testmodel & # CHANGE ME!
# or e.g.
python testmodel.py &

[...]

sleep infinity # keep the job occupied
```

UM-Bridge: Load-balancer HyperQueue scripts

Two separate scripts

HyperQueue script

```
#! /bin/bash

./hq alloc add slurm --time-limit 10m \
                --idle-timeout 3m \
                --backlog 1 \
                --workers-per-alloc 2 \
                --max-worker-count 3 \
                -- -p "devel"
# Any parameters after -- will be passed
# directly to sbatch
```

Job script (like previous slide)

```
#! /bin/bash

#HQ --time-request=1m
#HQ --time-limit=2m
#HQ --stdout none
#HQ --stderr none

[...]

./testmodel & # CHANGE ME!

[...]

sleep infinity # keep the job occupied
```

HPC vs. Cloud

HPC systems moving towards containers for portability and cloud systems moving towards HPC performance $\Rightarrow$ HPC and cloud are (somewhat) converging

HPC vs. Cloud

HPC systems moving towards containers for portability and cloud systems moving towards HPC performance $\Rightarrow$ HPC and cloud are (somewhat) converging

Cloud

Use kubernetes to orchestrate containers and use their provided loadbalancer.
UM-Bridge currently provides two kubernetes configurations:

- Parallel model instances, single node models
- multinode MPI parallel models

Traditional HPC

Instead of docker use Singularity containers
Start individual slurm jobs from the login node
Requires custom loadbalancer

Outlook

- UM-Bridge can help effectively connect intensive modelling software to UQ software
- It provides automatic support for parallelism with tested loadbalancers
- *Ongoing Work:* Improving error handling/reporting for current loadbalancers

Contribute

UM-Bridge the benchmark collection can be found on Github `https://github.com/UM-Bridge/`

The documentation is at `https://um-bridge-benchmarks.readthedocs.io/en/docs/`