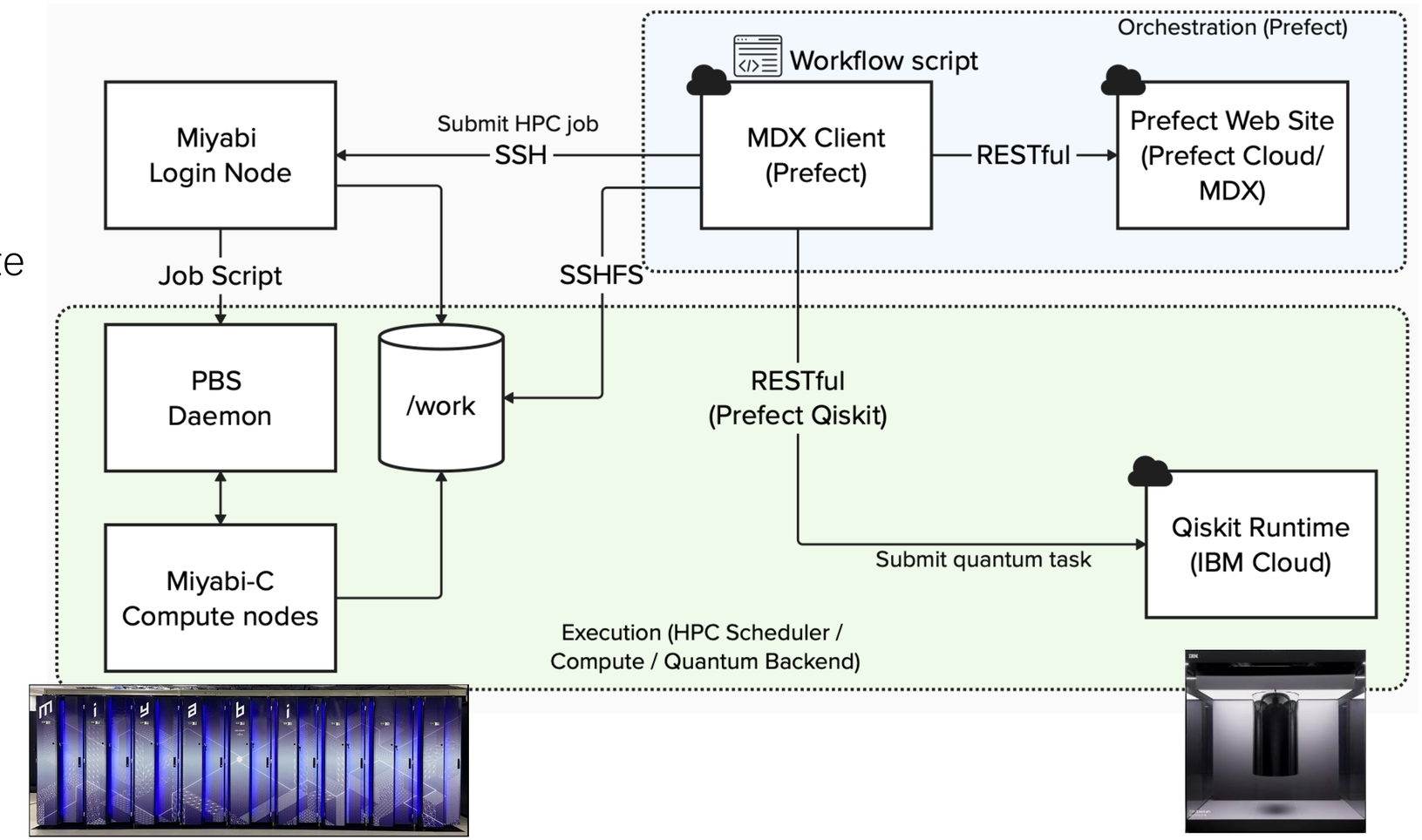
Prefect Tutorial for Miyabi

Hitomi Takahashi
IBM Quantum

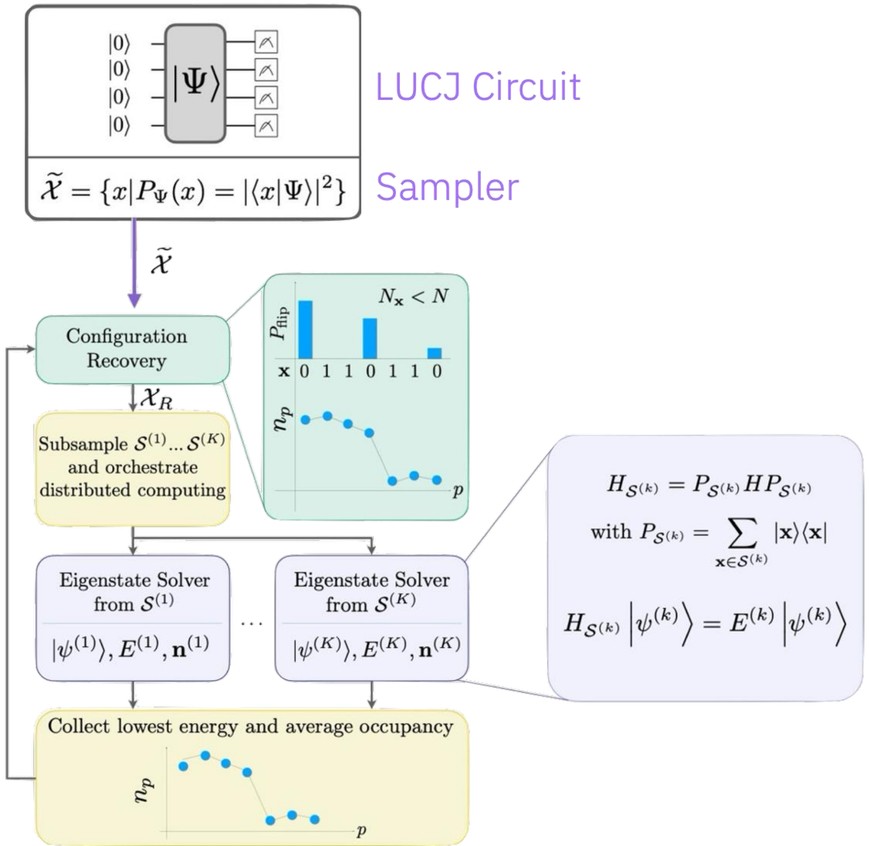
What Is Prefect?

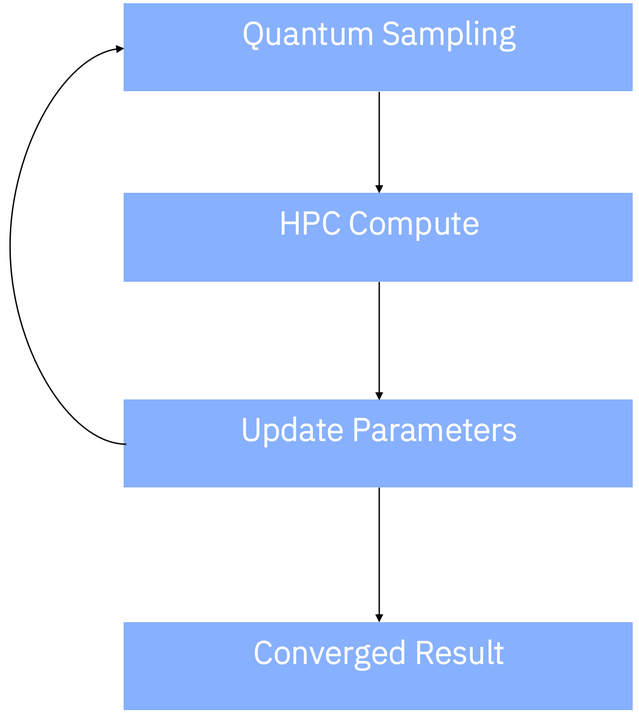
- Prefect is a Python workflow orchestration framework.
- You define Flow and Task as code
- It provides scheduling, retries, state tracking, and logs.
- Prefect orchestrates; HPC schedulers (PBS) execute.



Why QCSC Needs an Iterative Workflow?

- Quantum sampling is probabilistic; one run is not enough
- HPC compute evaluates and updates from sampled data
- The loop repeats until convergence





Why Scripts Are Not Enough for QCSC Workflow?

The problems for typical script-based approach

Execution history is not systematically recorded

It becomes unclear which parameters were used and when jobs were executed.

Recovery from failures is difficult

A failure in the middle often requires restarting the entire workflow.

Quantum and classical steps are fragmented

Quantum executions, HPC jobs, and post-processing are managed separately.

Reproducibility and comparison are hard

It is difficult to trace why a particular result was obtained.

How Prefect Changes This

Applying a Workflow Engine to QCSC with Prefect

Visible execution and failures

Each step’s status and logs are visible, making it clear where and why a run failed.

Efficient recovery from failures

Only failed tasks need to be re-executed, avoiding unnecessary re-runs of expensive computations.

Unified quantum–classical workflows

Quantum executions and HPC jobs are handled within a single, coherent workflow.

Reproducible experimental runs

Results are automatically linked with parameters and execution context, enabling reliable comparison.

Prefect Core Concepts

Flow (end-to-end workflow)

- The Flow represents the full QCSC iteration

Tasks (individual execution steps)

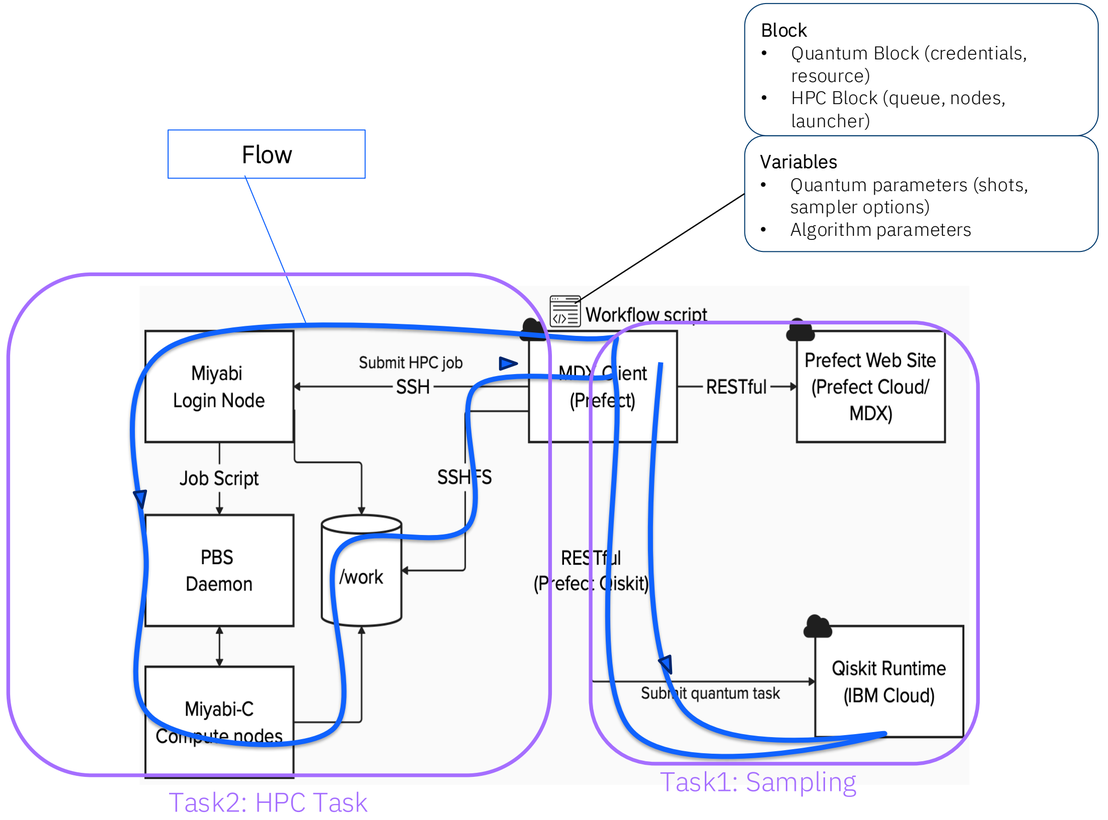
- Sampling Task: run the Sampler.
- HPC Task: submit a PBS job to run MPI programs.

Blocks (reusable configuration + credentials)

- Quantum Blocks: IBM Quantum credentials / runtime configuration.
- HPC Blocks: PBS job configuration.

Variables (run-time parameters)

- Quantum parameters: shots, circuit or sampler options.
- Algorithm parameters: iteration used by the workflow logic.



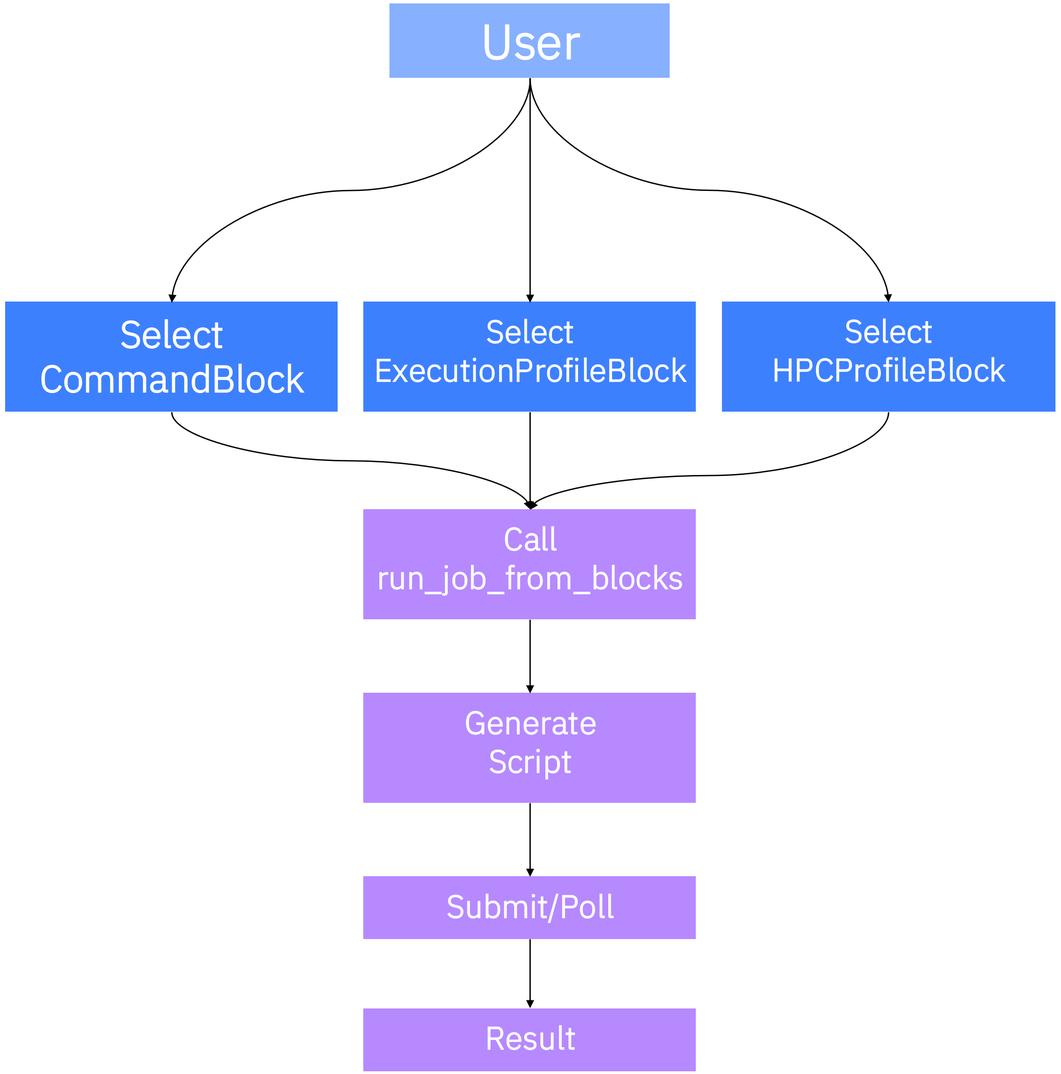
How Prefect + HPC-Prefect Simplifies Operations

- One API call triggers HPC execution
- Validation, script rendering, submit, and polling are automated
- Failures and retries are visible in Prefect

```
await run_job_from_blocks(
    command_block_name="cmd-diag",
    execution_profile_block_name="exec-diag-mpi",
    hpc_profile_block_name="hpc-miyabi",
    work_dir="./work",
    script_filename=”diag",
)
```

3-Step User Operation

1. **Select CommandBlock**: command identity (WHAT)
2. **Select ExecutionProfileBlock**: runtime shape (HOW)
3. **Select HPCProfileBlock**: target environment (WHERE)



Switch HPC by Changing 1-2 Lines

- Workflow logic stays unchanged.
- Change **hpc_profile_block_name**

```
await run_job_from_blocks(
    command_block_name="cmd-diag",
    execution_profile_block_name="exec-diag-mpi",
    hpc_profile_block_name="hpc-miyabi" or "hpc-fugaku",
    work_dir="./work",
    script_filename="diag",
)
```