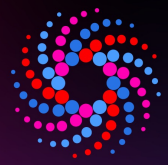


SC22
Dallas, TX | hpc accelerates.

CloudQ:
A Secure AI/HPC Cloud Bursting System

Shinichiro Takizawa[1], Masaaki Shimizu[2], **Hidemoto Nakada**[1],
Hiroya Matsuba[2], Ryousei Takano[1]

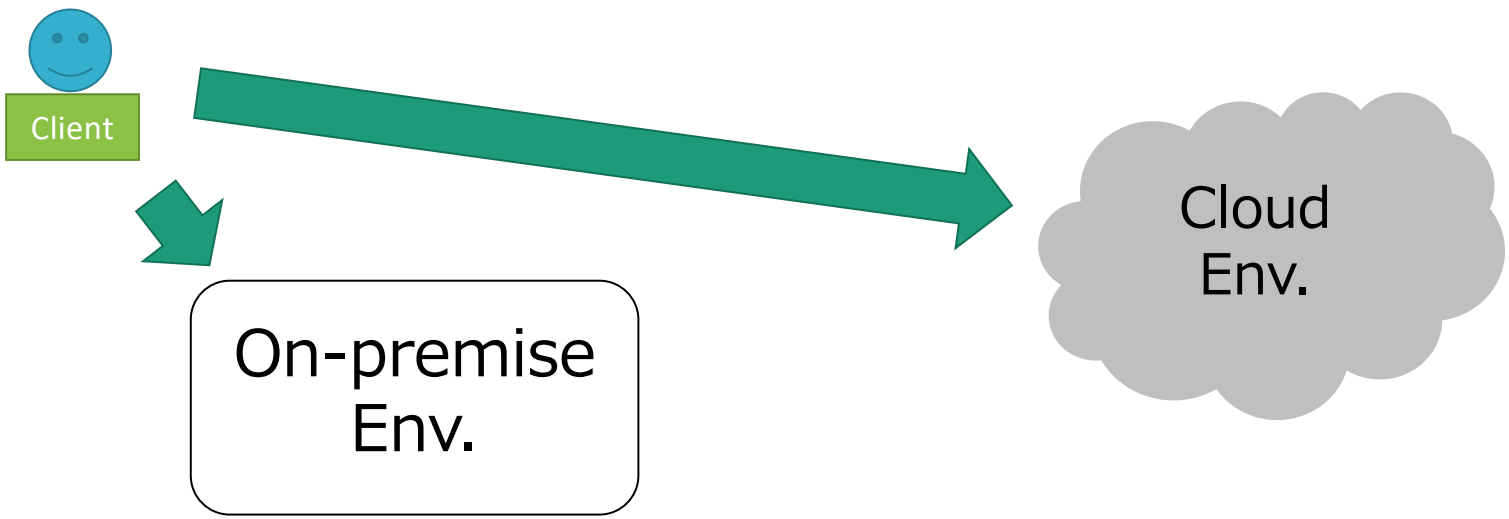
1: National Institute of Advanced Science and Technology
2: Hitach Ltd.

Background: Cloud Bursting

On-demand use of cloud environment

Seamlessly utilize both of on-premise and cloud env.

- On-premise
  - Process highly sensitive data
  - Provide base-line resouce
- Cloud
  - Process non-sensitive data
  - Allocated on-demand fashion when on-premise resources are not enough



Requirements

Secure Communication

- Between client and the cloud environment
- **ssh** is not enough, it is too powerful

Abstract Execution Environment

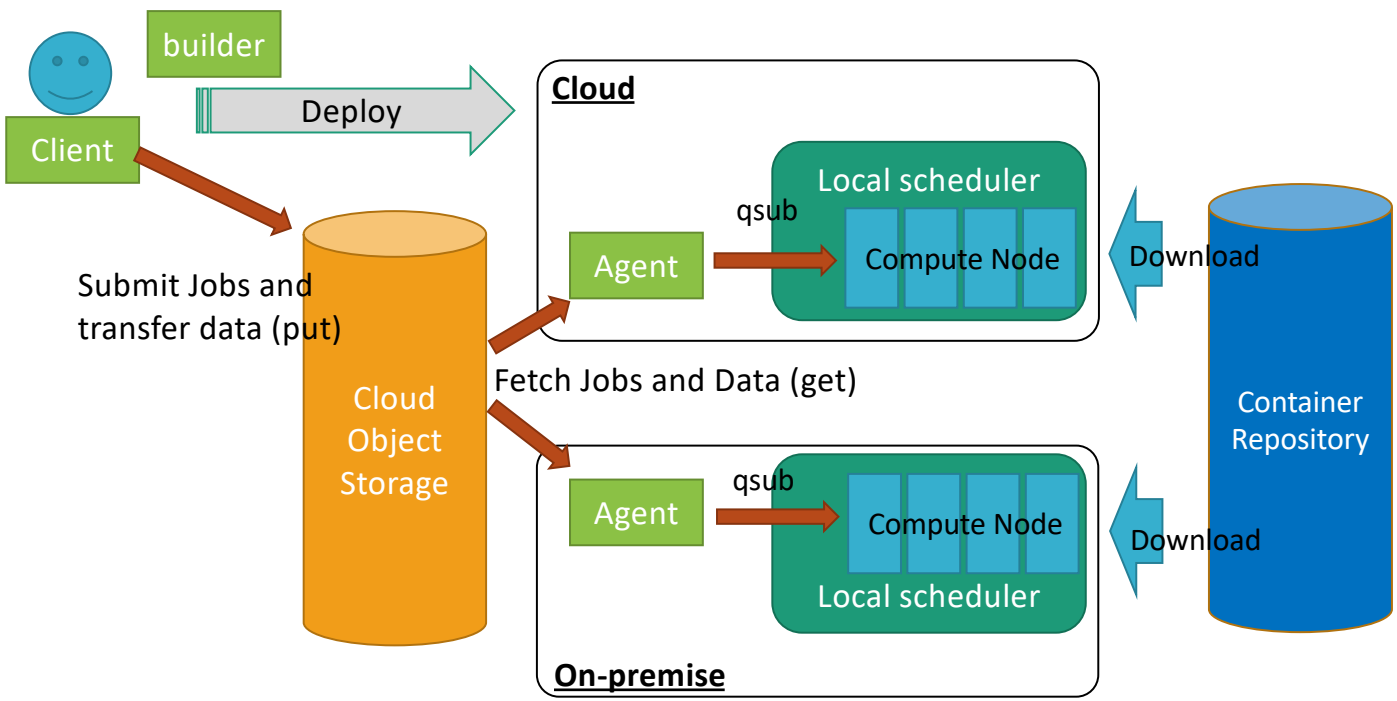
- The execution environments should look the same, even though the on-premise and cloud environments are not the same.

Execution Environment Deployment

- Minimize monetary / operational costs
- Manage user name space

CloudQ

- Cloud Storage based communication
- Containerised job execution environment
- Abstract job description
- Automatic deployment of the cloud environment



Requirements

Secure Communication

- Between client and the cloud environment
- **ssh** is not enough, it is too powerful

Abstract Execution Environment

- The execution environments should look the same, even though the on-premise and cloud environments are not the same.

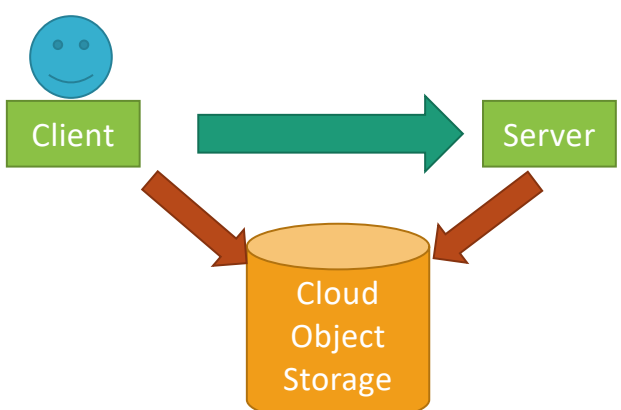
Execution Environment Deployment

- Minimize monetary / operational costs
- Manage user name space

Job submission via Cloud storage

- Communicate with shared storage
  - Both of the client and server periodically poll the pre specified storage are.

- Advantage
  - No incoming port required
  - No SSH access required
    - SSH provides too much privileges to the user.
  - No in-house server implementation required
    - Server implementation is easy, while ‘Secure’ one is really difficult
  - Take advantage of Cloud Storage authentication / authorization mechanism
    - Token with automatic expiration
    - Easy to manage for administration



Requirements

Secure Communication

- Between client and the cloud environment
- **ssh** is not enough, it is too powerful

Abstract Execution Environment

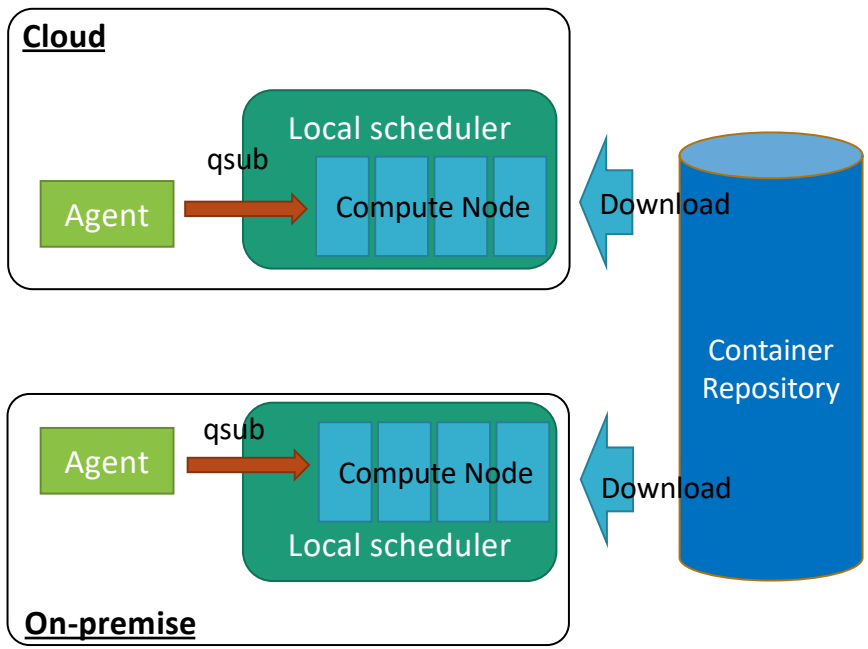
- The execution environments should look the same, even though the on-premise and cloud environments are not the same.

Execution Environment Deployment

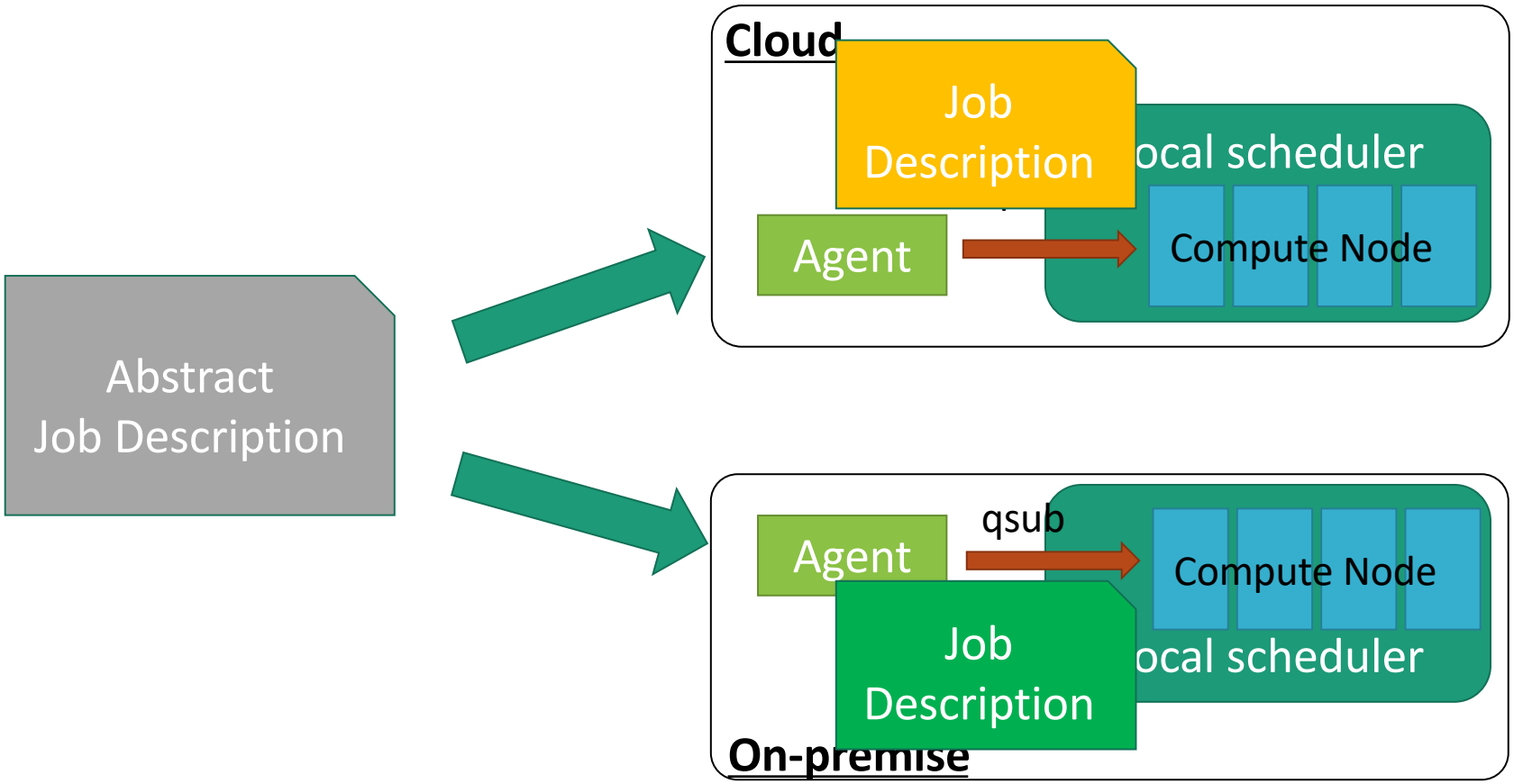
- Minimize monetary / operational costs
- Manage user name space

Container based environment

- Execution environments are encapsulated as container images
- Local Scheduler downloads and launches the images and run the jobs in containers



Abstract Job Description



Abstract Job Description

```
#$ run_on:ANY
#$ project:project01
#$ resource:type1
#$ n_resource:1
#$ walltime:1:00:00
#$ shell:bash
#$ container_image: img0=\
   docker://nvcr.io/nvidia/tensorflow:19.07-py3

wget https://script.is/here train.py
cloudq_cs_cp s3://myobjs/data ./data
cloudq_container_run $IMG0 python ./train.py data.ssh/
```

Abstract job script

Concrete script on ABCI

```
#!/bin/bash
#$ -l rt_F=1
#$ -l h_rt=1:00:00
#$ -cwd

source /etc/profile
source /etc/profile.d/modules.sh
module load aws-cli/2.0 singularitypro/3.5

IMG0=_IMG0\
singularity pull $IMG0 \
    docker://nvcr.io/nvidia/tensorflow:19.07-py3

wget https://script.is/here train.py
aws --endpoint-url https://s3.abci.ai \
    s3 cp --quiet s3://myobjs/data ./data
singularity exec --nv $IMG0 python train.py

rm $IMG0
```

Requirements

Secure Communication

- Between client and the cloud environment
- **ssh** is not enough, it is too powerful

Abstract Execution Environment

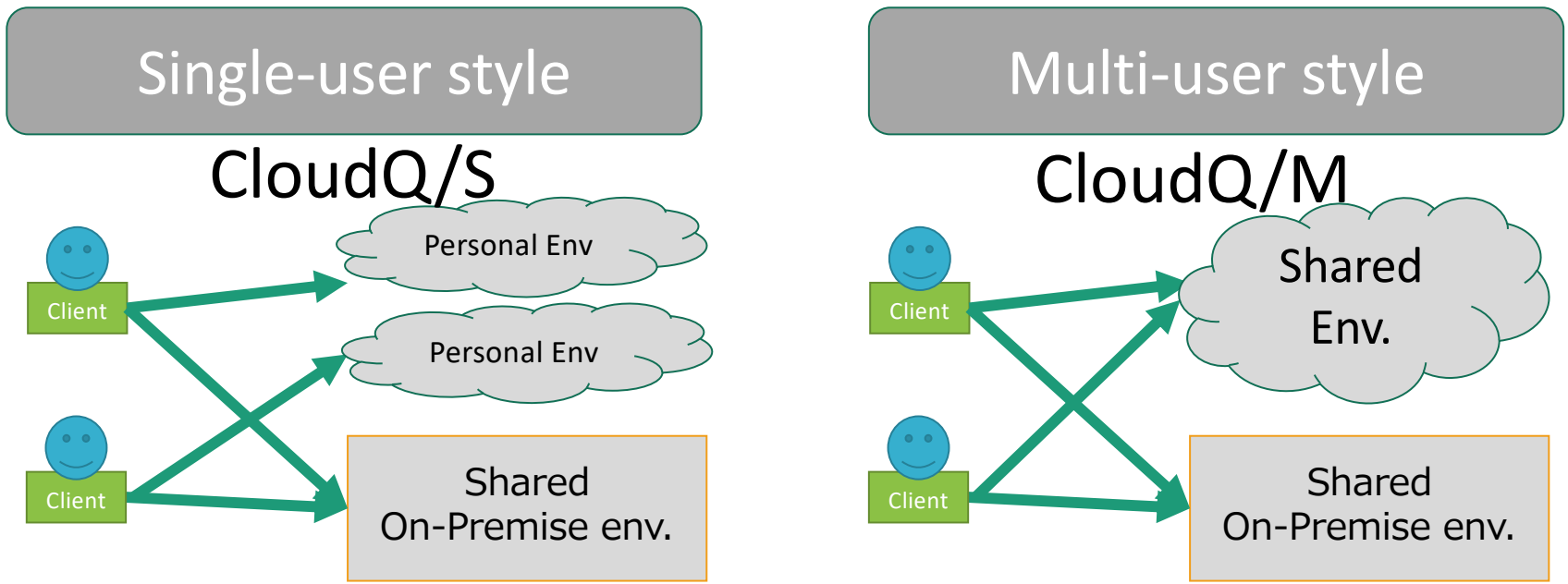
- The execution environments should look the same, even though the on-premise and cloud environments are not the same.

Execution Environment Deployment

- Minimize monetary / operational costs
- Manage user name space

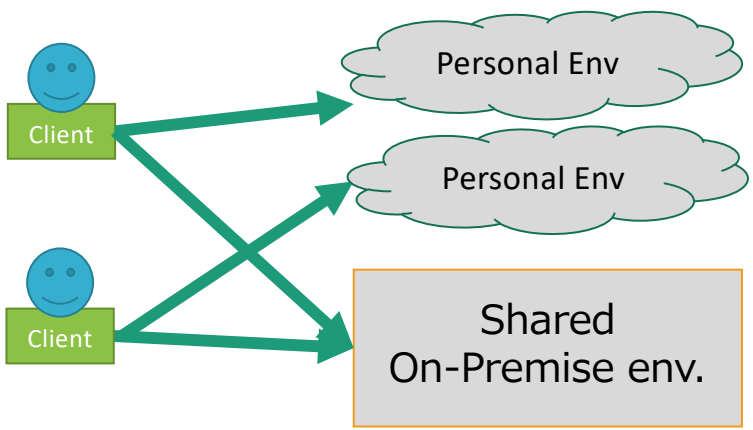
Execution Environment Deployment on the Cloud

- Minimize cost
  - Monetary cost – launch nodes only when they are actually required
  - Operational cost – account tracking / management
- User-name space management



Single-user style

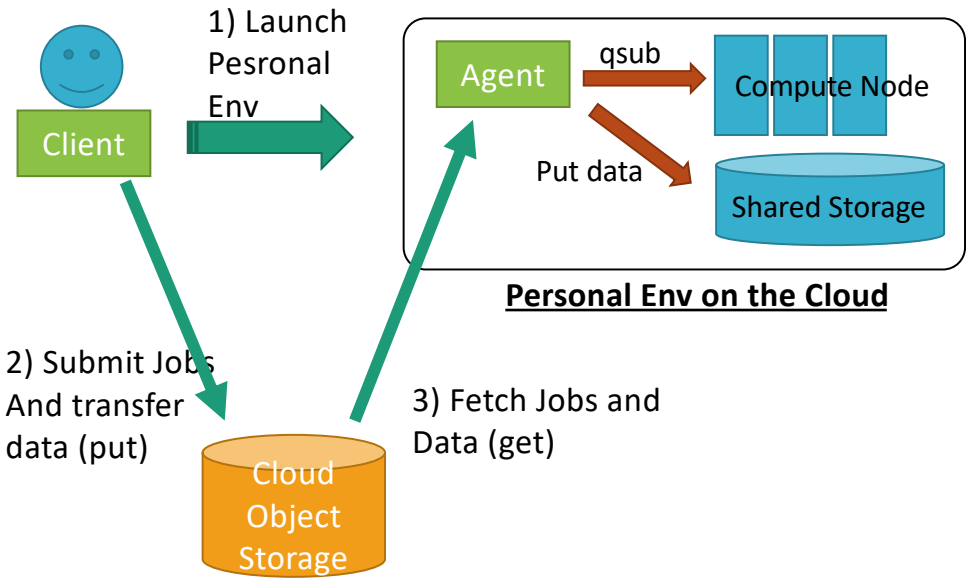
- Each user launches his/her own cloud environment in on-demand fashion.
- No need to manage ‘use space’
- Bills from cloud vender are enough for accounting management



Overview of CloudQ/S – single-user style

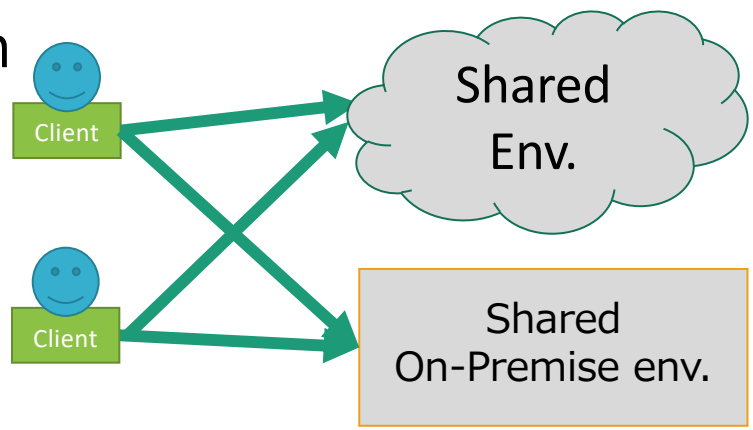
- Based on AWS parallel Cluster
  - Automatic Scaling
- Logs are stored CloudWatch Logs
  - No login required at all
- Each user is allocated dedicated AWS account

- Easy to implement
- Easy to track accounting



Multi-user style

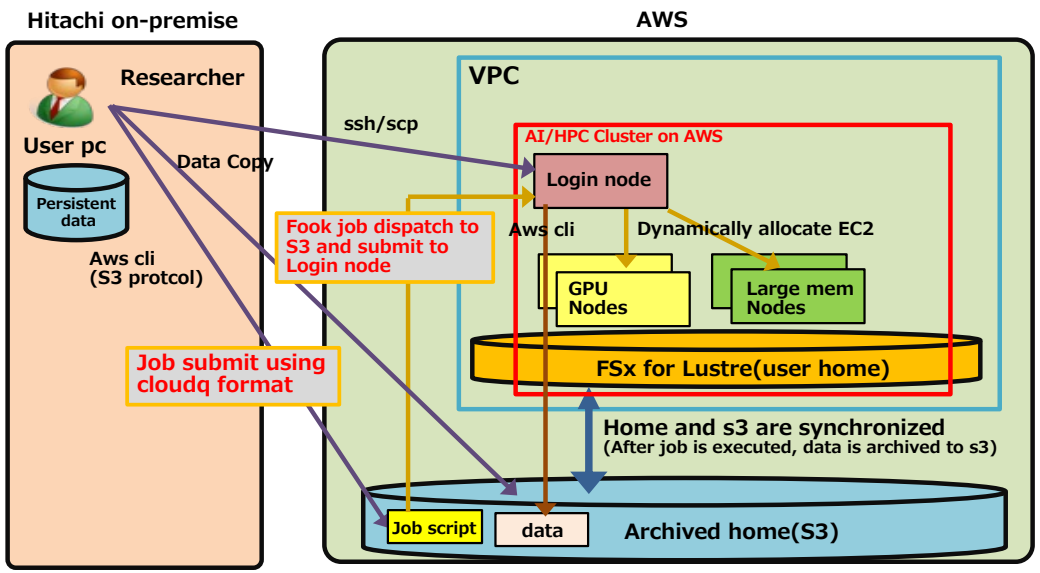
- Establish an environment that mimics the on-premise environment
  - The environment is persistent
    - Computation nodes are dynamically launched/stopped
  - User namespace is synchronized with the on-premise environment
- For accounting some engineering effort is required get the information form the batch queueing system log



Overview of CloudQ/M – Multi-user style

- Based on AWS CloudFormation
- Runs on single AWS account
- Use FSx for Luster with HSM enabled
  - S3 area for job submission can be monitored through Luster file system

- Implementation is not easy, especially accounting.
- Strict management of the users is possible



Single-user vs. Multi-user

Single-user

- Easy to implement
  - Thanks to the ParallelCluster
- Easy to operate
  - No extra-effort is required for accounting
  - Disposable environment. Zero-cost for zero-job
    - Except for Cloud strage and logwatch cost
- Easy to make it secure
  - Login capability can be entirely banned
    - No globally reachable network interface, no ssh daemon

Multi-user

- No start-up overhead
- Can provides users with similar environment to the on-premise
- From administrator perspective, this is preferable because
  - Easy to monitor each user’s activity

Summary

- CloudQ
  - Cloud storage-based communication
  - Containerized environment
  - Abstract job description
  - Automatic environment deployment
- CloudQ/S is available
  - https://github.com/aistairc/cloudq
  - Can be installed via PyPI
- CloudQ/M is in operation in Hitach Ltd.

Thank you