An Alternative Way for Genome Analysis on Cloud

ETL, ExTL, and introduction of its engine: awsub

Hiromu OCHIAI - National Cancer Center Japan
Genome analysis on Cloud Resources

aws

Google Cloud Platform

Azure

DigitalOcean

IBM SOFTLAYER

openstack

and more
How to use "Cloud"?
"Building a Cluster on Cloud"

- Galaxy
- cf
- ElastiCluster
- Butler
- etc...

Galaxy
Pros and Cons of "Cluster on Cloud"

- Pros:
  - We are VERY used to cluster on HPC
    - Grid Engine, HTCondor, SLURM, etc...
    - e.g. qsub ./my-workflow.sh
Pros and Cons of "Cluster on Cloud"

• Pros:
  • We are VERY used to cluster on HPC
    - Grid Engine, HTCondor, SLURM, etc...
    - e.g. qsub ./my-workflow.sh

• Cons:
  • Persistent static resources
    - Scheduler Node, Queue Database, Filesystem
Suggestion: "On-Demand ETL"
ETL is

- Extract, Transform, Load
- Data processing model for general purpose
Do it on Cloud

Data Source → Extract → Computing Resources → Transform → Data Destination → Load
Do it on Cloud

Data Source -> Extract
Computing Resources -> Transform
Data Destination -> Load

Disposable!
Low costed than fs!
Do it with awsub!
If you have 4 Fastq samples

Your actual sample data

Common Data
e.g. Reference

List of data locations on the storage
Specify workflow script and samples

Your actual sample data

Common Data e.g. Reference

List of data locations on the storage

```
$ awsub \n   --tasks ./my-samples.csv \n   --script ./my-workflow.sh
```
Security Group

Your actual sample data

List of data locations on the storage

Common Data e.g. Reference
Instances for each sample

Your actual sample data

Common Data e.g. Reference

List of data locations on the storage
Fetch specific sample data according to the location

Your actual sample data

List of data locations on the storage

fastq A

fastq B

fastq C

fastq D

Common Data e.g. Reference
Fetch reference data from common data source

Your actual sample data ➔ Common Data e.g. Reference ➔ List of data locations on the storage
Execute your workflow for each

Your actual sample data

| List of data locations on the storage |

Common Data e.g. Reference
Push the result data back to the storage
Dispose all the computing resources no longer used
All you got is the result data!

Your actual sample data and result data

Common Data e.g. Reference

List of data locations on the storage
Overall

Your actual sample data

Common Data e.g. Reference

List of data locations on the storage
by using **awsub**

```bash
$ awsub \
  --tasks ./my-samples.csv \ 
  --script ./my-workflow.sh \ 
  --image otiai10/STAR-alignment  # any Docker image
```
Problems of ETL on Bioinformatics
Problems of ETL on Bioinformatics

- Common Reference Data is so huge
  - Copying huge reference data uses
    - inefficient traffic
    - inefficient instance time
    - inefficient storage area
  - e.g. Human Reference for STAR alignment: 30GB
Suggestion: *Extended ETL (ExTL)*
Suggestion: *Extended ETL (ExTL)*

- Create a **Shared Data Instance**
- Fetch external common data **once**
- Let computing instances **mount**

Diagram:
- Your actual sample data
- Common Data (e.g. Reference)
- List of data locations on the storage
Suggestion: Extended ETL (ExTL)

- Create a Shared Data Instance
- Fetch external common data once
- Let computing instances mount

Cost Saving!

- Network traffic, instance time, ...

![Diagram showing data flow and cost savings graph.](image)
ExTL by using `awsub`

```bash
$ awsub \
   --tasks ./my-samples.csv \
   --script ./my-workflow.sh \
   --image otiai10/STAR-alignment \
   + --shared REFERENCE=s3://bucket/huge/reference
```
Summary

- Another approach than "Cluster on Cloud"
  - "On-demand ETL on Cloud"
- Huge common data can be a problem of "ETL on Cloud"
- "Extended ETL" (ExTL)
- Working Example Implementation of ExTL: awsub
More on the poster

about

- How to **Get started**
- **Google Cloud**, Microsoft Azure, OpenStack and more
- Common Workflow Language (CWL)
- Execution **Protocol** and Security Groups / IAM Instance Profile
- **Go** implementation
- etc...

Come to poster **B29**, and any feedback is welcome!

https://github.com/otiai10/awsun