



Development of Compute Cluster Simulator

Author: Artem Makogon

Supervisor: Oleg Sukhoroslov

May 19, 2024

Introduction

Testing of cluster scheduling algorithms



- Compute clusters are widely used for complex calculations.
- Scheduling algorithms are crucial for their performance.
- These algorithms are the subject of active research.



- Compute clusters are widely used for complex calculations.
- Scheduling algorithms are crucial for their performance.
- These algorithms are the subject of active research.



- > Researches need a tool for testing their hypotheses.
- > Using real clusters is expensive and time-consuming.

- Compute clusters are widely used for complex calculations.
- Scheduling algorithms are crucial for their performance.
- These algorithms are the subject of active research.



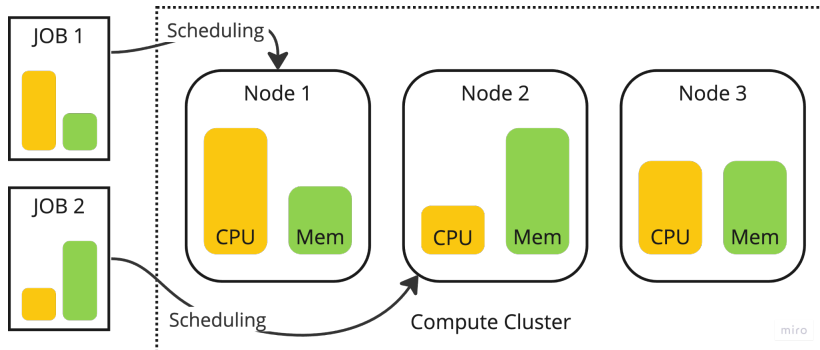
- > Researches need a tool for testing their hypotheses.
- > Using real clusters is expensive and time-consuming.



- ✓ **Simulators** are used for effective development.

Introduction

Cluster architecture



Simple model of cluster architecture

Introduction

Two approaches for simulation



Standard Workload Format (SWF)

- CPU/memory resources

Custom workloads

- CPU/memory/disk/network resources

Introduction

Two approaches for simulation



Standard Workload Format (SWF)

- CPU/memory resources
- Given execution time and resources
- Workload is calculated

Custom workloads

- CPU/memory/disk/network resources
- Given workload and resources
- Execution time is calculated

Introduction

Two approaches for simulation

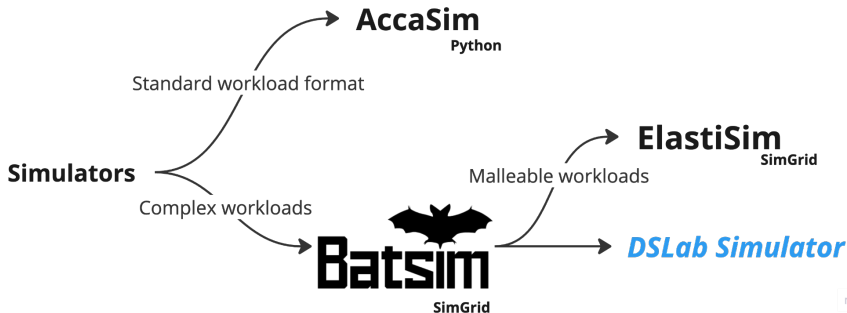


Standard Workload Format (SWF)

- CPU/memory resources
- Given execution time and resources
- Workload is calculated
- Used in famous cluster traces (e.g. Google traces)

Custom workloads

- CPU/memory/disk/network resources
- Given workload and resources
- Execution time is calculated
- NDA



Existing cluster simulators

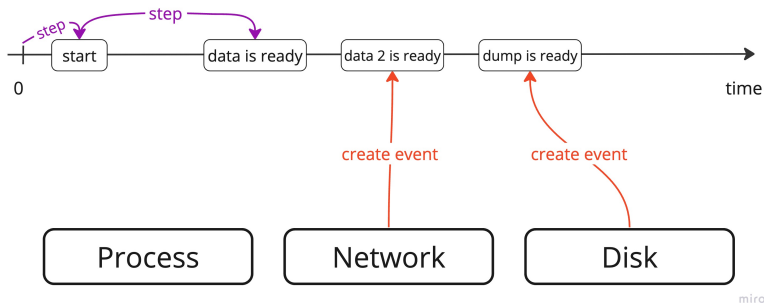


- Based on SimGrid simulator platform.
- Supports SWF and custom workloads defined as JSON profile.
- Supports connecting scheduling algorithms using inter-process communication.



```
"jobs": [  
  {"id": "job1",    ...  "res": 4, "profile": "sequence"},  
],  
  
"profiles": {  
  "homogeneous": {  
    "type": "parallel_homogeneous",  
    "cpu": 10e6,  
    "com": 1e6  
  },  
  "sequence": {  
    "type": "composed",  
    "repeat" : 4,  
    "seq": ["simple", "homogeneous", "simple"]  
  },  
}
```

- Fast & scalable simulation platform
- Written in Rust
- Provides models of compute/network/storage



Discrete-event modeling in DSLab

Methods



Asynchronous event management. Rust futures combinators

```
async fn process_task(&self, req: TaskRequest) {  
    let mut task = TaskInfo {req};  
  
    self.download_data(&task).await;  
    self.read_data(&task).await;  
    self.run_task(&task).await;  
    self.write_data(&task).await;  
    self.upload_result(&task).await;  
}
```

Example of sequent task execution

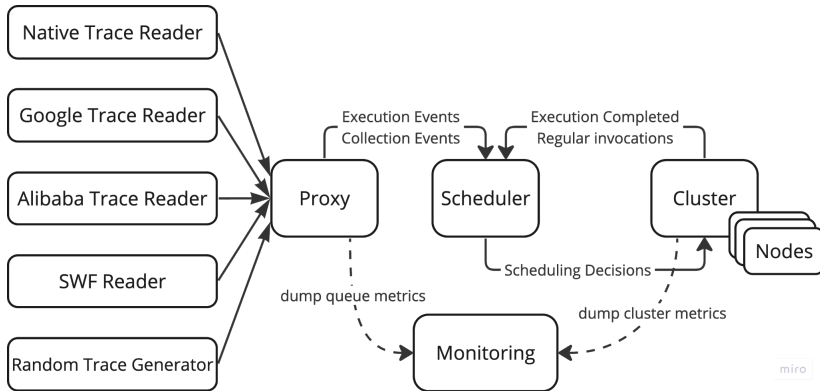
```
async fn run(&self, args: JobArgs) {  
    futures::join!(  
        self.download_data(args.node_1),  
        self.download_data(args.node_2)  
    )  
}
```

Example of parallel tasks execution



```
pub trait JobProfile {  
    async fn run(self: Box<Self>, ctx: JobContext)  
}
```

Job trait description



Simulator architecture

Expected Results

Fast and scalable compute cluster simulator



- ✓ SWF-based simulation
 - Custom workload simulation
 - Support reading workload traces from popular sources (Google, Alibaba)
 - Support for collecting metrics during the simulation and writing them to a file with the results
 - High performance, support for cluster modeling from 1-10K servers



1. The standard workload format specification. <https://www.cs.huji.ac.il/labs/parallel/workload/swf.html>
2. Dslab repository.
<https://github.com/osukhoroslov/dslab>
3. BatSim docs.
<https://batsim.readthedocs.io/en/latest/>