Split JAliEn JobAgent

Maxim Storetvedt

Department of Computing, Mathematics, and Physics

March 15, 2018

Western Norway
University of
Applied Sciences

Goal

- Split the JAliEn JobAgent into two separate components
  - The first performs the job matching
  - The second instantiates the payload
  - Each executed on their own JVM

Goal

- Split the JAliEn JobAgent into two separate components
  - The first performs the job matching
  - The second instantiates the payload
  - Each executed on their own JVM

Goal

- Split the JAliEn JobAgent into two separate components
  - The first performs the job matching
  - The second instantiates the payload
  - Each executed on their own JVM

Goal

- Split the JAliEn JobAgent into two separate components
  - The first performs the job matching
  - The second instantiates the payload
  - Each executed on their own JVM

Why?

- Improves isolation
  - The JobAgent runs with token certificates for the JobAgent role
  - The payload will be executed with another token certificate specific for the job ID
    - Limits what a job can do based on its requirements
- Can be strengthened further by incorporating containers

Why?

- Improves isolation
  - The JobAgent runs with token certificates for the JobAgent role
  - The payload will be executed with another token certificate specific for the job ID
    - Limits what a job can do based on its requirements
- Can be strengthened further by incorporating containers

Why?

- Improves isolation
  - The JobAgent runs with token certificates for the JobAgent role
  - The payload will be executed with another token certificate specific for the job ID
    - Limits what a job can do based on its requirements
- Can be strengthened further by incorporating containers

Why?

- Improves isolation
  - The JobAgent runs with token certificates for the JobAgent role
  - The payload will be executed with another token certificate specific for the job ID
    - Limits what a job can do based on its requirements
- Can be strengthened further by incorporating containers

Why?

- Improves isolation
  - The JobAgent runs with token certificates for the JobAgent role
  - The payload will be executed with another token certificate specific for the job ID
    - Limits what a job can do based on its requirements
- Can be strengthened further by incorporating containers

Current progress

- JAliEn JobAgent Class split in two:
  - JobAgent
  - JobWrapper

Current progress

- JAliEn JobAgent Class split in two:
  - JobAgent
  - JobWrapper

Current progress

- JAliEn JobAgent Class split in two:
  - JobAgent
  - JobWrapper

JobAgent

- Has mainly two tasks:
  - Get matched job
  - Launch the JobWrapper
    - Generate launch command
    - Pipe over data
    - Provide monitoring

JobAgent

- Has mainly two tasks:
  - Get matched job
  - Launch the JobWrapper
    - Generate launch command
    - Pipe over data
    - Provide monitoring

JobAgent

- Has mainly two tasks:
  - Get matched job
  - Launch the JobWrapper
    - Generate launch command
    - Pipe over data
    - Provide monitoring

JobAgent

- Has mainly two tasks:
  - Get matched job
  - Launch the JobWrapper
    - Generate launch command
    - Pipe over data
    - Provide monitoring

JobAgent

- Has mainly two tasks:
  - Get matched job
  - Launch the JobWrapper
    - Generate launch command
    - Pipe over data
    - Provide monitoring

JobAgent

- Has mainly two tasks:
  - Get matched job
  - Launch the JobWrapper
    - Generate launch command
    - Pipe over data
    - Provide monitoring

JobAgent

- Has mainly two tasks:
  - Get matched job
  - Launch the JobWrapper
    - Generate launch command
    - ***Pipe over data***
    - Provide monitoring

JobAgent: Containers

- Initial support for launching the JobWrapper within either
  - Singularity
  - Docker
- Generate container-specific launch command when one of the following environment variables is set
  - USE_CONTAINERS=SINGULARITY
  - USE_CONTAINERS=DOCKER
- How to transfer data when JobWrapper runs in a container?

JobAgent: Containers

- Initial support for launching the JobWrapper within either
  - Singularity
  - Docker
- Generate container-specific launch command when one of the following environment variables is set
  - USE_CONTAINERS=SINGULARITY
  - USE_CONTAINERS=DOCKER
- How to transfer data when JobWrapper runs in a container?

JobAgent: Containers

- Initial support for launching the JobWrapper within either
  - Singularity
  - Docker
- Generate container-specific launch command when one of the following environment variables is set
  - USE_CONTAINERS=SINGULARITY
  - USE_CONTAINERS=DOCKER
- How to transfer data when JobWrapper runs in a container?

JobAgent: Containers

- Initial support for launching the JobWrapper within either
  - Singularity
  - Docker
- Generate container-specific launch command when one of the following environment variables is set
  - USE_CONTAINERS=SINGULARITY
  - USE_CONTAINERS=DOCKER
- How to transfer data when JobWrapper runs in a container?

JobAgent: Containers

- Initial support for launching the JobWrapper within either
  - Singularity
  - Docker
- Generate container-specific launch command when one of the following environment variables is set
  - USE_CONTAINERS=SINGULARITY
  - USE_CONTAINERS=DOCKER
- How to transfer data when JobWrapper runs in a container?

JobAgent: Containers

- Initial support for launching the JobWrapper within either
  - Singularity
  - Docker
- Generate container-specific launch command when one of the following environment variables is set
  - USE_CONTAINERS=SINGULARITY
  - USE_CONTAINERS=DOCKER
- How to transfer data when JobWrapper runs in a container?

JobAgent: Containers

- Initial support for launching the JobWrapper within either
  - Singularity
  - Docker
- Generate container-specific launch command when one of the following environment variables is set
  - USE_CONTAINERS=SINGULARITY
  - USE_CONTAINERS=DOCKER
- How to transfer data when JobWrapper runs in a container?

Creating a "pipe" (JobAgent)

```
final Process p;

OutputStream stdin;
OutputStream stdout;

ObjectOutputStream stdinObj;
ObjectOutputStream stdoutObj;

try {

  p = pBuilder.start();

  stdin = p.getOutputStream();
  stdinObj = new ObjectOutputStream(stdin);

  stdinObj.writeObject(jdl);
  ...
```

Creating a "pipe" (JobWrapper)

```
try {
      inputFromJobAgent = new ObjectInputStream(System.in);
      jdl = (JDL) inputFromJobAgent.readObject();
...
```

JobWrapper

- Responsible for job execution
  - Receive piped data from JobAgent
  - Set up execution environment
  - Run job
  - Save output
- In other words, most of the tasks previously handled by the JobAgent class

JobWrapper

- Responsible for job execution
  - Receive piped data from JobAgent
  - Set up execution environment
  - Run job
  - Save output
- In other words, most of the tasks previously handled by the JobAgent class

JobWrapper

- Responsible for job execution
  - Receive piped data from JobAgent
  - Set up execution environment
  - Run job
  - Save output
- In other words, most of the tasks previously handled by the JobAgent class

JobWrapper

- Responsible for job execution
  - Receive piped data from JobAgent
  - Set up execution environment
  - Run job
  - Save output
- In other words, most of the tasks previously handled by the JobAgent class

JobWrapper

- Responsible for job execution
  - Receive piped data from JobAgent
  - Set up execution environment
  - Run job
  - Save output
- In other words, most of the tasks previously handled by the JobAgent class

JobWrapper

- Responsible for job execution
  - Receive piped data from JobAgent
  - Set up execution environment
  - Run job
  - Save output
- In other words, most of the tasks previously handled by the JobAgent class

What remains...

- Identity handling in JobWrapper(**!**)
- Prepare for production
  - Merge with recent commits
  - Thorough testing
  - Proper launch commands for containers

What remains...

- Identity handling in JobWrapper**(!)**
- Prepare for production
  - Merge with recent commits
  - Thorough testing
  - Proper launch commands for containers

What remains...

- Identity handling in JobWrapper**(!)**
- Prepare for production
  - Merge with recent commits
  - Thorough testing
  - Proper launch commands for containers

What remains...

- Identity handling in JobWrapper**(!)**
- Prepare for production
  - Merge with recent commits
  - Thorough testing
  - Proper launch commands for containers

What remains...

- Identity handling in JobWrapper**(!)**
- Prepare for production
  - Merge with recent commits
  - Thorough testing
  - Proper launch commands for containers

End

Thanks!

[Questions, comments]?

E-mail: msto@hvl.no