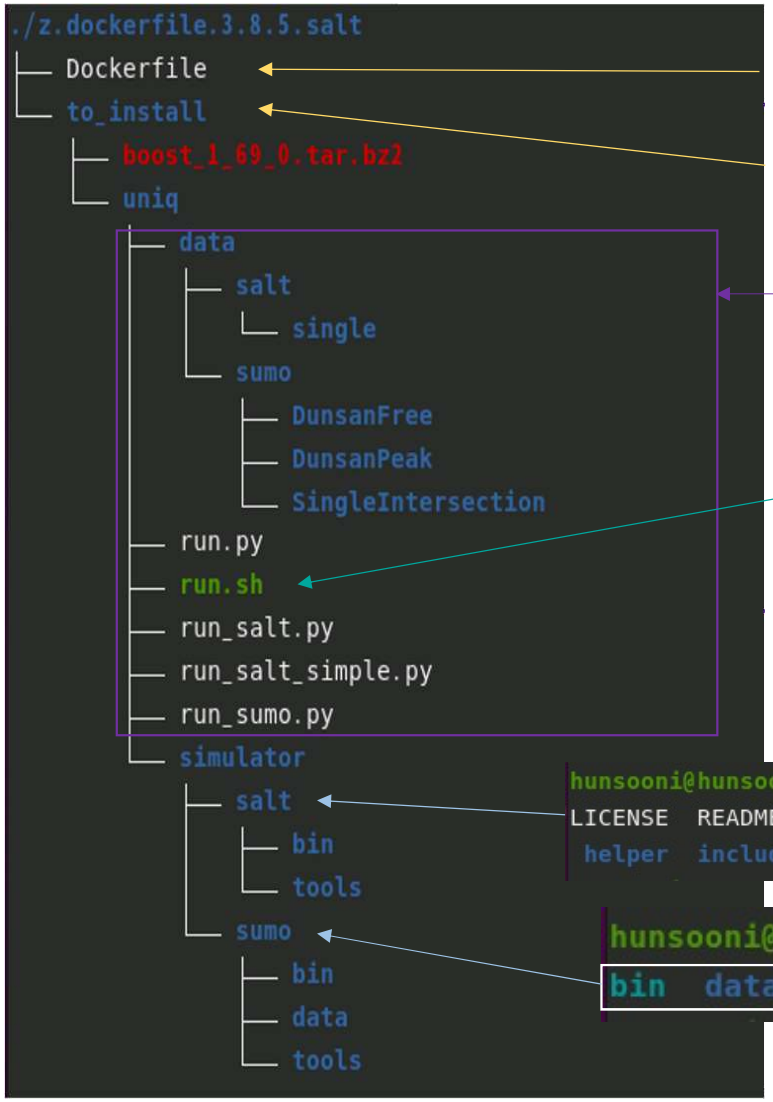
UNIQ 도커라이징

2021.05.11.

도거라이징을 위한 디렉토리 구성



Docker file : Simulator와 Optimizer가 동작되기 위해 필요한 환경 구성 및 실행 파일 복사

Docker file에서 도커라이징에 사용한 데이터 파일

최적화 서비스 관련 실행 파일/데이터

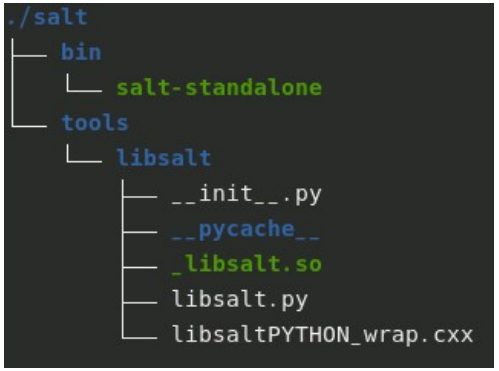
```
hunsooni@hunsooni-dev:~/z.tip/docker/z.dockerfile.3.8.5.salt/to_install/uniq$ cat run.sh
#!/bin/sh
default_exec_file="./run_salt_simple.py"

exec_file=${1:-$default_exec_file}

python $exec_file
hunsooni@hunsooni-dev:~/z.tip/docker/z.dockerfile.3.8.5.salt/to_install/uniq$
```

SALT 컴파일하여 실행 파일 복사

```
hunsooni@hunsooni-dev:~/z.uniq/simulator/traffic-simulator$ ls
LICENSE  README.md  analysis  bin  build  c.sh  conf  data  flow.txt
 helper  include  lib  src  test  tools
```

```
hunsooni@hunsooni-dev:/usr/share/sumo$ ls
bin  data  tools
```

SUMO 실행 파일 복사



Dockerfile

```
FROM ubuntu:20.04
RUN apt-get update && yes | apt-get update
RUN apt-get install -y build-essential

# 1. install libboost_1_69 for SALT simulator
WORKDIR /to_install
COPY ./to_install/boost_1_69_0.tar.bz2 /to_install
RUN tar xvfj boost_1_69_0.tar.bz2
WORKDIR /to_install/boost_1_69_0
RUN ./bootstrap.sh --prefix=/usr
RUN ./b2 cxxflags=-fPIC cflags=-fPIC link=static -a
RUN ./b2 install
WORKDIR /
RUN rm -rf /to_install

# 2. install python, pandas, tensorflow, keras, h5py for optimizer & simulator
RUN apt-get install -y python3.8 python3-pip
RUN apt-get install -y python3.8-dev
RUN ln -s /usr/bin/python3.8 /usr/bin/python

RUN python --version
RUN python -m pip install --upgrade pip
RUN python -m pip install pandas
RUN python -m pip install tensorflow==2.3.0
RUN python -m pip install keras==2.4.3
RUN python -m pip install h5py==2.10.0


# 3. copy uniq (optimizer & simulator)
WORKDIR /uniq
COPY ./to_install/uniq /uniq

# 4. set env variable for SALT
ENV SALT_HOME=/uniq/simulator/salt
ENV PYTHONPATH=$SALT_HOME/tools:$PYTHONPATH
```

SALT Simulator를 위해
Libboost+1_69_0 설치 파일을 이미지 쪽에 복사하고
컴파일 & 인스톨

이미지 쪽에 남아있는 설치 파일 제거

Libsalt 생성시 사용한 동일한 python 버전 설치
Optimizer 에서 사용하는 버전과 통일

Optimizer에서 사용하는 라이브러리 설치

```
#     libxerces, libproj, libfox for SUMO
ARG DEBIAN_FRONTEND=noninteractive
ENV TZ=Asia/Seoul
RUN apt-get install -y libxerces-c3.2
RUN apt-get install -y libproj-dev
RUN apt-get install -y libfox-1.6-0
```

SUMO에서 사용하는 라이브러리 설치

Docker 이미지에 Optimizer와 Simulator 복사(data 포함)
(공유 FS에 data 넣는 시각적 분석과 연동 고려해야)

Simulator(SALT) 관련 환경 변수 설정

※ libsalt 라이브러리 생성 환경, 최적화에서 사용한 라이브러리 등에 따라 설치해야 하는 라이브러리가 달라진다.

이미지 생성 / 실행

```
hunsooni@hunsooni-dev:~/z.tip/docker/z.dockerfile.3.8.5.salt$ sudo docker build -t my_uniq:0.1 .
Sending build context to Docker daemon    276MB
Step 1/26 : FROM ubuntu:20.04
 ---> 7e0aa2d69a15
Step 2/26 : RUN apt-get update && yes | apt-get update
 ---> Using cache
 ---> ef65d01fdaeb
Step 3/26 : RUN apt-get install -y build-essential
 ---> Using cache
 ---> c65dc1ea583d
Step 4/26 : WORKDIR /to_install
```

이미지 생성

```
Step 26/26 : ENTRYPOINT ["/bin/bash"]
 ---> Running in 717c3a2e9ad8
Removing intermediate container 717c3a2e9ad8
 ---> f2b6c2434431
Successfully built f2b6c2434431
Successfully tagged my_uniq:0.1
hunsooni@hunsooni-dev:~/z.tip/docker/z.dockerfile.3.8.5.salt$
```

```
hunsooni@hunsooni-dev:~/z.tip/docker/z.dockerfile.3.8.5.salt$ sudo docker images
REPOSITORY      TAG       IMAGE ID       CREATED              SIZE
my_uniq         0.1       f2b6c2434431   About a minute ago   3.48GB
```

이미지 생성 확인

```
hunsooni@hunsooni-dev:~/z.tip/docker/z.dockerfile.3.8.5.salt$ sudo docker run -it my_uniq:0.1 ./run.sh
##### start : test environment for SALT
/uniq/data/salt/single/2020-dj_sample.jso
scenarioFile: /uniq/data/salt/single/2020
[SALT Simulator 2.0]
--- Scenario File: /uniq/data/salt/single/
Loading Road Network ... done
Loading Vehicle Demand ... done
Loading Traffic Signal System ... done
[Simulation Output (Periodic)] >> output/
[Progress status] >> output/2020DJSampleT
--- Configuration Done ---
[Simulation Info]
```

```
hunsooni@hunsooni-dev:~/z.tip/docker/z.dockerfile.3.8.5.salt$ sudo docker run -it my_uniq:0.1 ./run.sh run_salt.py
##### start : test environment for SALT
/uniq/data/salt/single/2020-dj_sample.json
scenarioFile: /uniq/data/salt/single/2020-dj_sample.json
[SALT Simulator 2.0]
--- Scenario File: /uniq/data/salt/single/2020-dj_sample.json
Loading Road Network ... done
Loading Vehicle Demand ... done
Loading Traffic Signal System ... done
[Simulation Output (Periodic)] >> output/2020DJSampleTest/2020_dj_sample-PeriodicOutput.csv
[Progress status] >> output/2020DJSampleTest/progress.txt
```

Docker image 활용 실행

도커 이미지 공유

- 이미지를 생성하면 도커 리포지토리를 이용해서 공유 가능
- 도커 허브 : 대표적인 도커 리포지토리
  - https://hub.docker.com 가입 후 이미지 푸시
- 방법
  - 생성한 생성한 이미지에 추가로 태그를 붙인다.
    - 추가 태그에는 리포지토리 계정 이름을 사용
      - 도커 허브 계정이 hunsooni라면 hunsooni/리포지토리이름:태그 형태를 사용
    - $ sudo docker tag my_uniq:0.1.a hunsooni/my_uniq:0.1.a
  - Docker login 명령어 이용해서 도커허브에 로그인
    - $ sudo docker login
  - 로그인에 성공했다면 docker push로 이미지를 푸시
    - $ sudo docker push hunsooni/my_uniq:0.1.a
  - 푸시가 끝나면 도커 허브에서 이미지를 가져와 사용
    - $ sudo docker pull hunsooni/my_uniq:0.1.a
    - $ sudo docker run -it hunsooni/my_uniq:0.1.a run.sh

```
hunsooni@hunsooni-dev:~$ sudo docker tag my_uniq:0.1.a hunsooni/my_uniq:0.1.a
hunsooni@hunsooni-dev:~$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: hunsooni
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
hunsooni@hunsooni-dev:~$ sudo docker push hunsooni/my_uniq:0.1.a
The push refers to repository [docker.io/hunsooni/my_uniq]
81d64617d2c0: Pushed
```

```
ccdbb80308cc: Mounted from library/ubuntu

0.1.a: digest: sha256:b59ffd4617a0cc8f0e0be0f459f8374dfee2ecde0bb6c7b9b9e1dcc5c6
3f2307 size: 6196
hunsooni@hunsooni-dev:~$
```

```
root@SALT2:/home/hunsooni# sudo docker pull hunsoooni/my_uniq:0.1.a
Error response from daemon: pull access denied for hunsoooni/my_uniq, repository
 does not exist or may require 'docker login'
root@SALT2:/home/hunsooni# sudo docker pull hunsooni/my_uniq:0.1.a
0.1.a: Pulling from hunsooni/my_uniq
345e3491a907: Pull complete
57671312ef6f: Pull complete
```

```
Digest: sha256:b59ffd4617a0cc8f0e0be0f459f8374dfee2ecde0bb6c7b9b9e1dcc5c63f2307
Status: Downloaded newer image for hunsooni/my_uniq:0.1.a
root@SALT2:/home/hunsooni# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hunsooni/my_uniq    0.1.a               24866715912b        41 minutes ago      3.87
root@SALT2:/home/hunsooni# sudo docker run -it hunsooni/my_uniq:0.1.a run.sh
##### start : test environment for SALT
/uniq/data/salt/single/2020-dj_sample.json
scenarioFile: /uniq/data/salt/single/2020-dj_sample.json
[SALT Simulator 2.0]
--- Scenario File: /uniq/data/salt/single/2020-dj_sample.json
Loading Road Network ... done
Loading Vehicle Demand ... done
```

https://javacan.tistory.com/entry/docker-start-7-create-image-using-dockerfile

도커 이미지 태깅 / 저장소 등록

```
hunsooni@hunsooni-dev:~$ sudo docker tag my_uniq:0.1.a hunsooni/my_uniq:0.1.a
hunsooni@hunsooni-dev:~$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: hunsooni
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
hunsooni@hunsooni-dev:~$ sudo docker push hunsooni/my_uniq:0.1.a
The push refers to repository [docker.io/hunsooni/my_uniq]
81d64617d2c0: Pushed
```

태깅

저장소(도커허브) 로그인

저장소에 이미지 Push

```
ccdbb80308cc: Mounted from library/ubuntu

0.1.a: digest: sha256:b59ffd4617a0cc8f0e0be0f459f8374dfee2ecde0bb6c7b9b9e1dcc5c6
3f2307 size: 6196
hunsooni@hunsooni-dev:~$
```

공유 도커 이미지 다운로드 & 실행

```
root@SALT2:/home/hunsooni# sudo docker pull hunsooni/my_uniq:0.1.a
0.1.a: Pulling from hunsooni/my_uniq
345e3491a907: Pull complete
57671312ef6f: Pull complete
```

저장소에서 이미지 Pull

```
Digest: sha256:b59ffd4617a0cc8f0e0be0f459f8374dfee2ecde0bb6c7b9b9e1dcc5c63f2307
Status: Downloaded newer image for hunsooni/my_uniq:0.1.a
root@SALT2:/home/hunsooni# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hunsooni/my_uniq    0.1.a               24866715912b        41 minutes ago      3.87GB
root@SALT2:/home/hunsooni# sudo docker run -it hunsooni/my_uniq:0.1.a run.sh
##### start : test environment for SALT
/uniq/data/salt/single/2020-dj_sample.json
scenarioFile: /uniq/data/salt/single/2020-dj_sample.json
[SALT Simulator 2.0]
--- Scenario File: /uniq/data/salt/single/2020-dj_sample.json
Loading Road Network ... done
Loading Vehicle Demand ... done
Loading Traffic Signal System ... done
[Simulation Output (Periodic)] >> output/2020DJSampleTest/2020_dj_sample-PeriodicOutput
.csv
[Progress status] >> output/2020DJSampleTest/progress.txt
```

이미지 실행

시각적 분석에서 활용

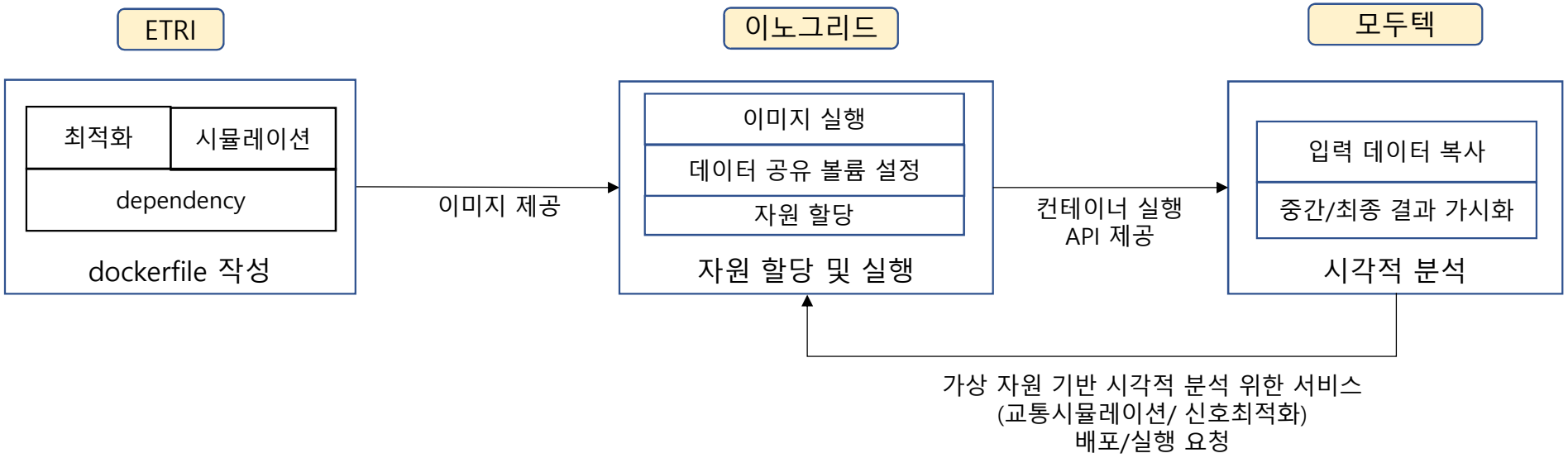
- 최적화/시뮬레이션 대상이 달라짐을 고려해야
- 이미지 실행시 대상에 대한 정보가 있는 위치 지정
  - $sudo docker run -it -v /path/to/our/workspace/uniq:/uniq --name uniq.v.0.1 hunsooni/my_uniq:0.1.a run.sh
    - -v /path/to/our/workspace/uniq:/uniq
      - /path/to/our/workspace/uniq 를 이미지상의 /uniq 로 메핑
- 이미지는 해당 위치를 인자로 받아서 수행하는 프로그램 탑재
  - run.sh
    - 해당 경로(/uniq) 로 이동 후 해야할 일 수행

```
hunsooni@hunsooni-dev:~$ sudo docker run -it -v /home/hunsooni/z.tip/docker/z.dockerfile.3.8.5.all/to_install/uniq:/uniq my_uniq:0.1.b /uniq/run.sh
##### start : test environment for SALT
/uniq/data/salt/single/2020-dj_sample.json
scenarioFile: /uniq/data/salt/single/2020-dj_sample.json
[SALT Simulator 2.0]
--- Scenario File: /uniq/data/salt/single/2020-dj_sample.json
Loading Road Network ... done
Loading Vehicle Demand ... done
Loading Traffic Signal System ... done
[Simulation Output (Periodic)] >> output/2020DJSampleTest/2020_dj_sample
[Progress status] >> output/2020DJSampleTest/progress.txt
--- Configuration Done ---
[Simulation Info]
Input
        -Simulation Name: 2020_dj_sample
        -Road Network: /uniq/data/salt/single/2020DJSampleTest/node.xml,
pleTest/connection.xml
        -Vehicle Demand: /uniq/data/salt/single/routes/dj_sample_mon.rou
        -Traffic Light System: /uniq/data/salt/single/2020DJSampleTest/tss.xml
        -Time Range: 25200~36000
```

```
hunsooni@hunsooni-dev:~/z.tip/docker/z.dockerfile.3.8.5.all/to_install/uniq$ cat run.sh
#!/bin/sh

cd /uniq
default_exec_file="./run_salt_simple.py"

exec_file=${1:-$default_exec_file}

python $exec_file
hunsooni@hunsooni-dev:~/z.tip/docker/z.dockerfile.3.8.5.all/to_install/uniq$
```

Backup slides

가상 자원 기반 시각적 분석 관련 협업(안)



※ 데이터 공유는 공유 FS 이용