Historical Database for DynaMIT2.0

Meng Yue

Department of Automation,Tsinghua University, China

SMART FM-IRG, August 6, 2016

Outline

1. Background
2. Implement
3. Test Result
4. Summary
5. Future work

Outline

1. Background
2. Implement
3. Test Result
4. Summary
5. Future work

Starting point

The aim of the on–line calibration is to use the off–line calibrated parameter values as starting points and perform a local optimization step towards the unobserved true values[1].

Precise historical OD flow $=>$ Accurate estimated OD-flow

[1] Constantinos,A(2004) On–line Calibration for Dynamic Traffic Assignment

OD-Flow Analysis

Altered by many factors:

- rush hour
- weather
- holiday
- ...

Needs to be stratified under several tags

Insights

- Set up database for storage
- Update historical data with estimated data
- Provide best-fit historical flow

Goal

To design a program that can automatically **save** results from the DynaMIT simulation ,**update** the historical OD-flow and **render** proper demand input for the real-time DynaMIT simulation.

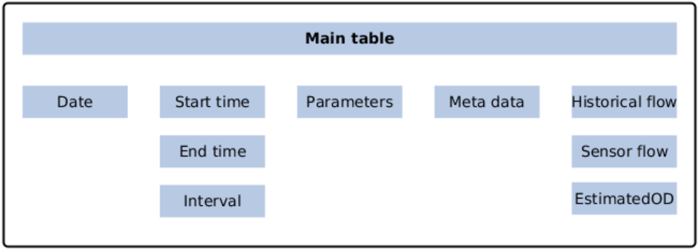
Outline

1. Background
2. Implement
3. Test Result
4. Summary
5. Future work

Functions

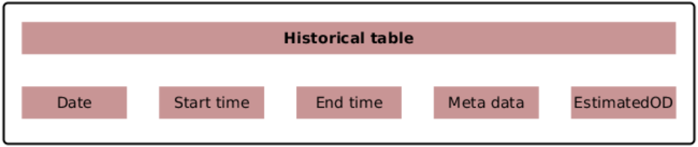
- Save DynaMIT input and output files to database
- Update the exist records in database
- Render best-fit historical data given by the input parameters of real-time DynaMIT simulation
- Auto-check and backup

Table definition



Main table

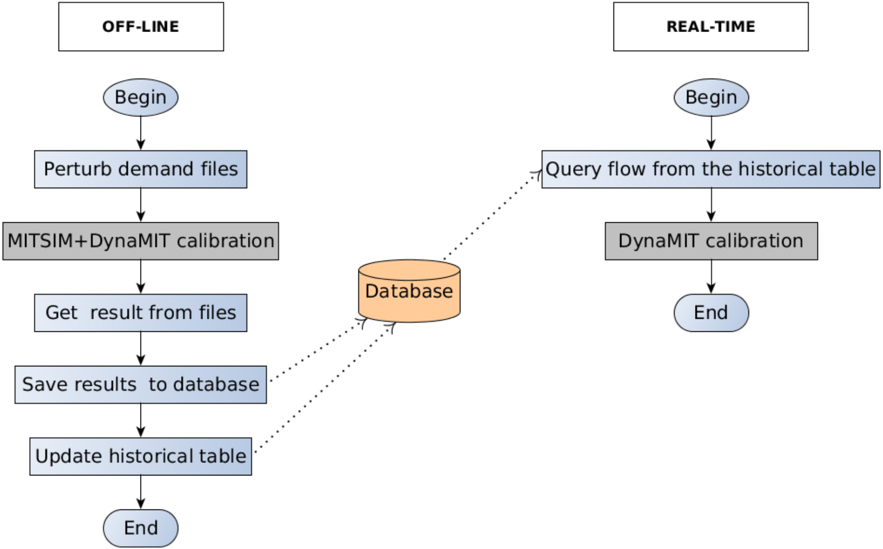
- Date
- Start time
- End time
- Interval
- Parameters
- Meta data
- Historical flow
- Sensor flow
- EstimatedOD



Historical table

- Date
- Start time
- End time
- Meta data
- EstimatedOD

Flow Diagram

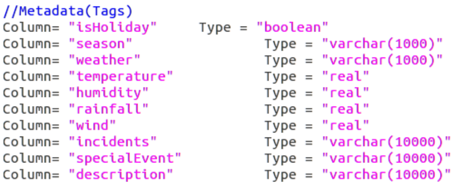


Project description

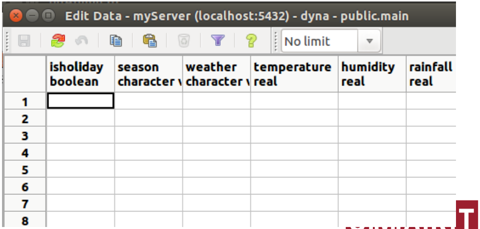
- Database: PostgreSQL
- Language:
  - Python (file operation)
  - Java (database I/D/U/Q)
  - Shell (whole process)

Setup process

- CREATE TABLE: database.config
- Framework parameter: params.config & init.sh
- Generate demands: demand_perturb.py



```
//Metadata(Tags)
Column= "isHoliday"      Type = "boolean"
Column= "season"                 Type = "varchar(1000)"
Column= "weather"                Type = "varchar(1000)"
Column= "temperature"            Type = "real"
Column= "humidity"               Type = "real"
Column= "rainfall"               Type = "real"
Column= "wind"                   Type = "real"
Column= "incidents"              Type = "varchar(10000)"
Column= "specialEvent"           Type = "varchar(10000)"
Column= "description"            Type = "varchar(10000)"
```



Edit Data - myServer (localhost:5432) - dyna - public.main

	isholiday boolean	season character v	weather character v	temperature real	humidity real	rainfall real
1						
2						
3						
4						
5						
6						
7						
8						

Insert process

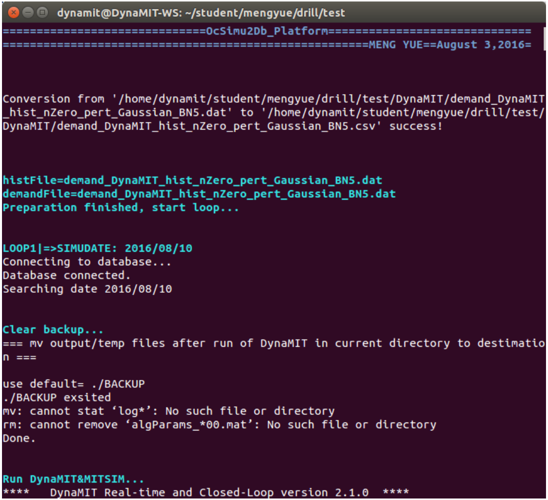
- dtaparam.dat
- behavior.dat
- supplyparam.dat
- sensor.out
- demand.dat
- estimatedOD*
- EOD.txt
- sen_flw_*
- sen_spd_*
- ...

Update process

Generate historical data process

SELECT estimatedOD FROM hod WHERE ...

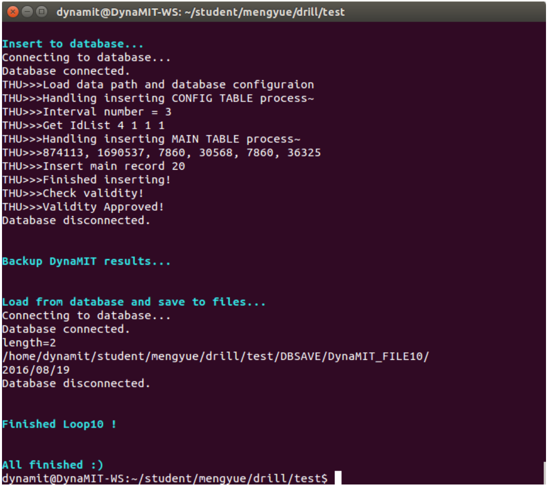
Screen-shot(Start)



```
==============================OcSimu2Db_Platform==============================
======================================================MENG YUE==August 3,2016=



Conversion from '/home/dynamit/student/mengyue/drill/test/DynaMIT/demand_DynaMIT
_hist_nZero_pert_Gaussian_BN5.dat' to '/home/dynamit/student/mengyue/drill/test/
DynaMIT/demand_DynaMIT_hist_nZero_pert_Gaussian_BN5.csv' success!



histFile=demand_DynaMIT_hist_nZero_pert_Gaussian_BN5.dat
demandFile=demand_DynaMIT_hist_nZero_pert_Gaussian_BN5.dat
Preparation finished, start loop...


LOOP1|=>SIMUDATE: 2016/08/10
Connecting to database...
Database connected.
Searching date 2016/08/10


Clear backup...
=== mv output/temp files after run of DynaMIT in current directory to destinatio
n ===

use default= ./BACKUP
./BACKUP exsited
mv: cannot stat ‘log*’: No such file or directory
rm: cannot remove ‘algParams_*00.mat’: No such file or directory
Done.


Run DynaMIT&MITSIM...
****   DynaMIT Real-time and Closed-Loop version 2.1.0   ****
```

SMART
Singapore-MIT Alliance for Research and Technology
FUTURE URBAN MOBILITY

Screen-shot(End)

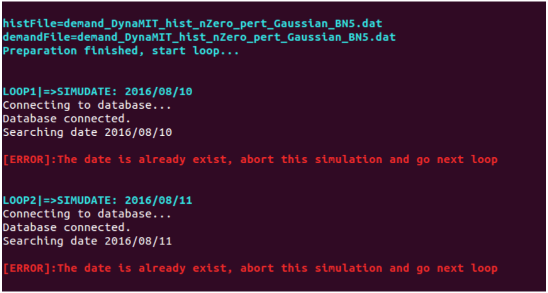


dynamit@DynaMIT-WS: ~/student/mengyue/drill/test

```
Insert to database...
Connecting to database...
Database connected.
THU>>>Load data path and database configuraion
THU>>>Handling inserting CONFIG TABLE process~
THU>>>Interval number = 3
THU>>>Get IdList 4 1 1 1
THU>>>Handling inserting MAIN TABLE process~
THU>>>874113, 1690537, 7860, 30568, 7860, 36325
THU>>>Insert main record 20
THU>>>Finished inserting!
THU>>>Check validity!
THU>>>Validity Approved!
Database disconnected.


Backup DynaMIT results...


Load from database and save to files...
Connecting to database...
Database connected.
length=2
/home/dynamit/student/mengyue/drill/test/DBSAVE/DynaMIT_FILE10/
2016/08/19
Database disconnected.


Finished Loop10 !


All finished :)
dynamit@DynaMIT-WS:~/student/mengyue/drill/test$
```

Screen-shot(Error)

Check for conflict primary key DATE



```
histFile=demand_DynaMIT_hist_nZero_pert_Gaussian_BN5.dat
demandFile=demand_DynaMIT_hist_nZero_pert_Gaussian_BN5.dat
Preparation finished, start loop...


LOOP1|=>SIMUDATE: 2016/08/10
Connecting to database...
Database connected.
Searching date 2016/08/10

[ERROR]:The date is already exist, abort this simulation and go next loop


LOOP2|=>SIMUDATE: 2016/08/11
Connecting to database...
Database connected.
Searching date 2016/08/11

[ERROR]:The date is already exist, abort this simulation and go next loop
```

Outline

1. Background
2. Implement
3. Test Result
4. Summary
5. Future work

Test

Result

Outline

1. Background
2. Implement
3. Test Result
4. Summary
5. Future work

Conclusion

- Run simulation and recording automatically
- Easy to debug and add new tags for database
- Potential value

Outline

1. Background
2. Implement
3. Test Result
4. Summary
5. Future work

Future work

- Find source for the metadata
- Use XXX algorithm for ”update process”
- Use XXX algorithm for ”render historical data”
- Refactoring & Documentation

Question

Any questions?

Thank you!