Example: How to Use the Parallel BerlinMOD Benchmark

Jiamin Lu, April 2013

SECONDO was built by the SECONDO team

1 Motivation

BerlinMOD is a benchmark prepared for evaluating moving objects databases. It generates a set of spatio-temporal data simulating the movement of vehicles running in Berlin, Germany, and uses a set of example queries to measure the performance of various databases.

BerlinMOD also provides a set of scripts to generate the data sets and process the example queries in a SECONDO database. However, SECONDO can only be installed on a single computer and thus its capability is restricted by the underlying hardware. Since the data generation and certain example queries in the BerlinMOD are I/O- and CPU- intensive, it is difficult to generate and process data sets with large scale factors. Regarding this issue, Parallel BerlinMOD is proposed. It revised the existing scripts so as to make them run in Parallel SECONDO with a cluster of computers, and has the following advantages:

1. The size of BerlinMOD data sets is decided by a scale factor S, while the number of simulated vehicles is $2000 * \sqrt{S}$, and the observation period is $28 * \sqrt{S}$ days. In the past, generating a data set with the scale factor 1.0 costs several hours. At present, a parallel generation is prepared in Parallel BerlinMOD, with which we can generate a data set with the scale factor of 30 on a cluster consisting of 110 AWS (Amazon Web Services) EC2 large-type instances in five hours. The generated data can either be evaluated in the single-computer databases, or be used to appraise other parallel database systems.

2. All BerlinMOD OBA queries are revised in order to be processed in Parallel SECONDO. It not only proves that Parallel SECONDO has a good compatibility with the single-computer SECONDO database, but also improves the efficiency of all complex queries in BerlinMOD with a factor increased by the number of the computers of the cluster, making Parallel SECONDO competitive.

In order to help the user to quickly get familiar with Parallel BerlinMOD, this document is prepared to introduce its usage. All benchmark scripts can only be run in Parallel SECONDO[1], which can either be installed on the user's own physical computer cluster, or on a virtual cluster consisting of AWS EC2 instances. Parallel BerlinMOD is mainly divided to two parts, the first contains all programs and scripts for generating the data set in parallel, and the second prepares all sequential and parallel OBA queries to run in Parallel SECONDO.

[1] It is published with SECONDO since the version 3.3.2.

2 Parallel Generation

In BerlinMOD, the data generation is processed with a single SECONDO script. In contrast, the parallel generation looks more complex as it includes the following components:

- A generation bash script: genParaBerlinMOD.sh
- A Hadoop program: GenMOD.jar
- Four SECONDO scripts prepared to generate the data in master and slave databases on different stages:
  - BerlinMOD_DataGenerator_master1.SEC
  - BerlinMOD_DataGenerator_master2.SEC
  - BerlinMOD_DataGenerator_map.SEC
  - BerlinMOD_DataGenerator_reduce.SEC
- A README file prepared to explain the usage of the generator.

In practice, the complete generation can be easily finished only with the bash script, while the other components are invoked by the script automatically. This script provides the following arguments:

- -h : It prints out a help information about the usage of this script.
- -d: It indicates the name of the created database, with a default value of berlinmod.
- -s: It defines the scale factor of the created data set, with a default value of 0.01.
- -p: It limits the observing period for simulated vehicles. By default, it is set as null so that the observing period is then decided by the given scale factor, as we introduced before.
- -l: It indicates whether the data are generated in a parallel way. If it is true, then the database is sequentially created on the local computer only. By default it is false.

For example, creating a database with a scale factor of 1.0 in the user's installed Parallel SECONDO on a cluster can be simply processed with the command:

```
$ ./genParaBerlinMOD.sh -s 1
```

At last, a database named **berlinmod** is created in the user's master database, in which the following query scripts can be processed.

3 Parallel Query

Here all the queries are expressed in SECONDO executable language, in three parts:

1. Two scripts prepared for processing the sequential OBA queries: BerlinMOD_Sequential_CreateObjects.SEC and BerlinMOD_Sequential_OBA-Queries.SEC.
2. Two scripts prepared for processing OBA queries in parallel way: BerlinMOD_Parallel_CreateObjects.SEC and BerlinMOD_Parallel_OBA-Queries.SEC.
3. One script BerlinMOD_Parallel_Verification.SEC used to compare the sequential and parallel query results. If all above queries are correctly executed, the queries in this script should return TRUE.

All these scripts only run in the database created above with the parallel generator. The CreateObjects scripts must be processed before their respective Queries scripts, in order to create auxiliary objects.