VNE-Sim Tool User Guide

Nashila Jahan
Soroush Haeri

Simon Fraser University
04/09/2018

VNE-Sim is a discrete event simulator written in C++ 11. It is publicly available under the terms of the MIT License. This tool may be used to simulate virtualization algorithms and compare their performance using various topologies of data center networks. CMake build system is used to compile all required packages before the test cases are run for various scenarios. Furthermore, being written in C++, *VNE-Sim* offers good memory management and enables batch simulations using any scripting language. A step by step procedure is provided to enable effective use of the tool.

1. Gathering Pre-Requisites

Listed steps and instructions are based on the Linux OS implementation.

1. Download the package from website: https://bitbucket.org/shaeri/vne-sim

and unzip it in home directory. *VNE-Sim* code may be also downloaded using command line:

> git clone https://bitbucket.org/shaeri/vne-sim

The root directory of *vne-sim* should be */home/username/vne-sim.*

2. Install the following pre-requisites available via Linux distribution:
   - CMake
   - Boost libraries
   - GSL: GNU scientific library
   - GLPK: GNU linear programming kit.

For Ubuntu, use specific version or name of the library:

> sudo apt-get install <name of the library>

Other required libraries to be installed as part of *VNE-Sim* package are:

- Fast Network Simulation Setup (FNSS): used to create data center topologies
- Boston University Representative Topology Generator (BRITE): for generating network topologies together with FNSS
- Hiberlite: for saving simulation results
- Adevs: for modeling virtual network embedding processes
- SQLite3: used for handling simulation results exported automatically as SQLite databases.

CMake relies on scripts to search for the listed libraries and applies the required modifications after downloading these libraries. A C++11 compiler is required to compile the code.

2. Compiling Source Code

To compile the source code, change directory to */home/username/vne-sim* created in the previous step and type in the terminal window:

```
> mkdir build && cd build
> cmake .. -DWITH_FNSS_SUPPORT=off
> make
```

This will create a *build* directory under *vne-sim* directory and install all required libraries after successful compilation. At the beginning, all required modules are checked:

```
-- The CXX compiler identification is GNU 4.9.4
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- ADEVS DOES NOT EXIST
-- HIBERLITE DOES NOT EXIST...
-- Found Git: /usr/bin/git (found version "1.9.1")
-- Boost version: 1.56.0
```

After initiating the build for the first time, CMake attempts to download and patch some of the dependencies (FNSS, Hiberlite, ADEVS). Samples of errors if this step fails or gets interrupted are:

```
make[2]: *** No rule to make target `../external-libs/hiberlite/
libhiberlite.a', needed by `lib/libcore.so'.  Stop.
```

In this case, following directories need to be deleted in the *vne-sim* root directory:

- *extranal-libs/adevs*
- *external-libs/hiberlite*
- *extenal-libs/fnss*

Delete also all CMake generated files under the *vne-sim/build* directory. Then, attempt to build the code again from scratch. If successfully compiled, the display should show:

```
[ 98%] Building CXX object src/network-file-generator/test/CMakeFiles/nfg-
test.dir/network-file-generator-test.cc.o
```

```
Linking CXX executable ../../../bin/nfg-test
[ 98%] Built target nfg-test
Scanning dependencies of target vineyard-test
[100%] Building CXX object src/Vineyard/test/CMakeFiles/vineyard-test.dir/vy-test.cc.o
Linking CXX executable ../../../bin/vineyard-test
[100%] Built target vineyard-test
```

3. Modifying Configuration File

All configuration and runtime parameters required for *VNE-Sim* execution are stored in an XML file *configuration.xml*, which resides in the *root* folder of *VNE-Sim*. It is important to set correct run-time path pointing to *vne-sim* root directory in the *configuration.xml* file. The GLPK files *<glpk></glpk>* are part of the code. Hence, change only the beginning of the paths to point to the *vne-sim* directory.

Next, download the latest network files *network_files.zip*, save them outside the *vne-sim* folder in the *home* directory, and unzip them:

> *http://www2.ensc.sfu.ca/~ljilja/cnl/projects/VNE-Sim/vne-sim-web/index.html#network*

The network files are already available and hence they do not need to be generated again. Ignore all the configurations within the tag:

> *<NetworkFileGenerator></NetworkFileGenerator>*

Set the correct path within the tags pointing to the network files you have unzipped:

> *<SubstrateNetwork></SubstrateNetwork>*

and

> *<VirtualNetRequest></VirtualNetRequest>*

NOTE: The directory listed in *configurations.xml* file refers to a set of network files for the arrival rate of 100 Erlangs. This directory name should be changed to arrival rate of 12 Erlangs because network files *network_files.zip* downloaded from the website only have a subset of files for traffic of 12 Erlangs.
Change:

> *<dir>reqs-100-1000-nodesMin-3-nodesMax-10-grid-25</dir>*

to

<dir>reqs-12-1000-nodesMin-3-nodesMax-10-grid-25</dir>

These two paths in *configurations.xml* file need to be changed to point to correct network files:

```
<SubstrateNetwork>
        <path>/home/username/network_files/SN</path>
        <filename>vy_substrate_net_n_50_outergrid_25_inner_grid_25.txt</filename>
</SubstrateNetwork>
<VirtualNetRequest>
        <path>/home/username/network_files/VNRs</path>
        <dir>reqs-12-1000-nodesMin-3-nodesMax-10-grid-25</dir>
        <reqfileExtension>.txt</reqfileExtension>
</VirtualNetRequest>
```

The configuration within the *<vineyard></vineyard>* tag should be correctly set and point to the existing files. The directory and filenames should match with actual directory and filenames in the SN and VNRs directories within the network files *network_files.zip* used in the test simulation. The modified configurations file is listed in the Appendix A1.

4. Executing the Test Case

The source code of the test cases is placed in various packages under the test folder. All package/test/testname.cc (e.g., *vineyard/test/vineyard-test.cc*) files are compiled into executables. For the first test cases, only use this "experiment-test" located at:

src/experiments/test/expriments-test.cc

It is compiled into an executable under:

build/bin/experiments-test

These test cases have been reported in References [1]-[5] listed at the end of this Guide:

The experiments and tests are written using the Boost library unit test framework. The documentation describing the framework is available at:

http://www.boost.org/doc/libs/1_53_0/libs/test/doc/html/utf/user-guide/runtime-config/run-by-name.html

Open the file *src/experiments/test/expriments-test.cc*. After the *#include* and *using* clauses, review the section:

```
BOOST_AUTO_TEST_SUITE (AlgorithmExperiments)

BOOST_AUTO_TEST_CASE(ARRIVAL_RATE_TESTS)
```

“AlgorithmExperiments” is the name of test suite and “ARRIVAL_RATE_TESTS” is the name of test case that we wish to run. Some minor modifications are required in *experiments-test.cc* file because it was originally written for a complete set of network files. However, we shall only use here a subset of these network files. Now, modify original *experiments-test.cc* file, within the scope of:

```
BOOST_AUTO_TEST_CASE(ARRIVAL_RATE_TESTS)

{

}
```

Change:

```
std::string vnr_dirs[] =

{"reqs-12-1000-nodesMin-3-nodesMax-10-grid-25",  "reqs-14-1000-nodesMin-3-nodesMax-10-grid-25","reqs-16-1000-nodesMin-3-nodesMax-10-grid-25",
"reqs-20-1000-nodesMin-3-nodesMax-10-grid-25","reqs-25-1000-nodesMin-3-nodesMax-10-grid-25","reqs-33-1000-nodesMin-3-nodesMax-10-grid-25"
"reqs-50-1000-nodesMin-3-nodesMax-10-grid-25", "reqs-100-1000-nodesMin-3-nodesMax-10-grid-25”};
```

to:

```
std::string vnr_dirs[] = {"reqs-12-1000-nodesMin-3-nodesMax-10-grid-25”};
```

and all the for loops:

```
for (int j = 0; j < 8; j++)
        { … }
```

to:

```
for (int j = 0; j < 1; j++)
        { … }
```

Alternatively, copy the code for a minimal test before the scope of ARRIVAL_RATE_TESTS:

```
BOOST_AUTO_TEST_CASE(ARRIVAL_RATE_TESTS)
{
}
```

Script for the minimal test is available in Appendix A2.

After we have modified *experiments-test.cc* and *configurations.xml* files, the entire code needs to be recompiled using *make* command as instructed in Section 2.

Finally, to run the first test, type in the terminal:

```
experiment-tests --run_test=AlgorithmExperiments/ARRIVAL_RATE_TESTS
```

or

```
experiment-tests --run_test=AlgorithmExperiments/MINIMAL_VINEYARD_GRC_TEST
```

5. Extracting Simulation Results

After successful completion of the test run, unprocessed data generated by discrete events that occur during simulations are saved as SQLite databases in *vne-sim* directory. These data may be processed using SQL queries or other statistical analysis tools. The shell script in Appendix A3 may be used to extract key performance data from database files generated through the test.

Type in command to run the script:

```
sh export_overall_avg_data.sh
```

Note that we are using only one arrival rate of 12 Erlangs for test simulation. This can be easily extended to a range of arrival rates by using *for* loop:

```
for

arrivalRate in 12 14 16 20 25 33 50 100

do

sqlite3vne-sim/mcvne_bfs_mcf_reqs-$arrivalRate-1000-nodesMin-3-nodesMax-10-grid-25.db < query.sql
```

Summary table *SummaryStat.xls* lists average performance data for various algorithms:

SummaryStat.xlsx

6. Network File Parameters

This Section describes various parameters used for generating network files for Substrate Network (SN) and Virtual Network Requests (VNRs).

Both SN and VNRs are generated using BRITE [6] library with RT Waxman algorithm. Substrate graph is composed of 50 nodes where each node is randomly connected to a maximum of 5 other nodes. The substrate graph generated for the simulation scenarios consists of 221 edges. Each node of the substrate network is randomly placed on a 25 by 25 grid as:

```
<nodePlacement>1</nodePlacement>

<numNeighbors>5</numNeighbors>

<innerGridSize>25</innerGridSize>
```

```
<outerGridSize>25</outerGridSize>

<RTWaxman>

    <growthType>2</growthType>

    <alpha>0.5</alpha>

    <beta>0.2</beta>

</RTWaxman>
```

For each VNR, number of nodes are uniformly distributed between 3 and 10 and each virtual host is connected to maximum of 3 neighbor hosts. The CPU requirements of the virtual nodes are uniformly distributed between 2 and 20 units while the bandwidth requirements of the virtual links are uniformly distributed between 0 and 50 units. Duration of each simulation scenario is 50,000 time units. The listed code shows the network file parameters for simulating VNRs:

```
<TotalTime>50000</TotalTime>

<VNTopologyType>Waxman</VNTopologyType>

<VNRLinkSplittingRate>0.1</VNRLinkSplittingRate>

<VNRNumNodesDist>0</VNRNumNodesDist>

<VNRNumNodesDistParam1>3</VNRNumNodesDistParam1>

<VNRNumNodesDistParam2>10</VNRNumNodesDistParam2>

<VNRNumNodesDistParam3>-1</VNRNumNodesDistParam3>

<VNRDurationDist>1</VNRDurationDist>

<VNRDurationDistParam1>1000</VNRDurationDistParam1>

<VNRDurationDistParam2>-1</VNRDurationDistParam2>

<VNRDurationDistParam3>-1</VNRDurationDistParam3>

<VNRArrivalDist>2</VNRArrivalDist>
```

```
<VNRArrivalDistParam1>12.5</VNRArrivalDistParam1>
<VNRArrivalDistParam2>-1</VNRArrivalDistParam2>
<VNRArrivalDistParam3>-1</VNRArrivalDistParam3>
<VNRMaxDistanceDist>1</VNRMaxDistanceDist>
<VNRMaxDistanceDistParam1>15</VNRMaxDistanceDistParam1>
<VNRMaxDistanceDistParam2>25</VNRMaxDistanceDistParam2>
<VNRMaxDistanceDistParam3>-1</VNRMaxDistanceDistParam3>
<VNCPUDist>0</VNCPUDist>
<VNCPUDistParam1>2</VNCPUDistParam1>
<VNCPUDistParam2>20</VNCPUDistParam2>
<VNCPUDistParam3>-1</VNCPUDistParam3>
<VLBWDist>0</VLBWDist>
<VLBWDistParam1>1</VLBWDistParam1>
<VLBWDistParam2>10</VLBWDistParam2>
<VLBWDistParam3>-1</VLBWDistParam3>
<VLDelayDist>0</VLDelayDist>
<VLDelayDistParam1>50</VLDelayDistParam1>
<VLDelayDistParam2>100</VLDelayDistParam2>
<VLDelayDistParam3>-1</VLDelayDistParam3
```

Various network topologies such as Two-Tier, F^{2}Tree, and Diamond may be used to create substrate networks using Fast Network Simulation Setup (FNSS) [12]. Then, these SN graphs may be used together with the VNR graphs to compare performances over various topologies as described in [2].

The *VNE-Sim* tool provides a framework of simulating new and existing virtual network embedding algorithms over a pre-defined substrate network topology. It is intended for future researchers to take advantage of existing codes while plugging in new algorithms and topologies to enhance source code base.

References

[1] S. Haeri and Lj. Trajkovic, "Virtual network embedding via Monte-Carlo tree search," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 1–12, Feb. 2017.

[2] H. Ben Yedder, Q. Ding, U. Zakia, Z. Li, S. Haeri, and Lj. Trajkovic, "Comparison of virtualization algorithms and topologies for data center networks," *The 26th International Conference on Computer Communications and Networks (ICCCN 2017)*, *2nd Workshop on Network Security Analytics and Automation (NSAA)*, Vancouver, Canada, Aug. 2017.

[3] S. Haeri and Lj. Trajkovic, "VNE-Sim: a virtual network embedding simulator," *SIMUTOOLS*, Prague, Czech Republic, Aug. 2016, pp. 112–117.

[4] S. Haeri, Q. Ding, Z. Li, and Lj. Trajkovic, "Global resource capacity algorithm with path splitting for virtual network embedding," *IEEE Int. Symp. Circuits and Systems*, Montreal, Canada, May 2016, pp. 666–669.

[5] S. Haeri and Lj. Trajkovic, "Virtual network embeddings in data center networks," *IEEE Int. Symp. Circuits and Systems*, Montreal, Canada, May 2016, pp. 874–877.

[6] (2018, Aug.) Boston University Representative Internet Topology Generator. [Online]. Available: http://www.cs.bu.edu/brite/.

[7] (2018, Aug.) SQLite: Small. Fast. Reliable. Choose any three. [Online]. Available: https://www.sqlite.org/.

[8] (2018, Aug.) CMake Build System. [Online]. Available: https://cmake.org/.

[9] (2018, Aug.) Boost C++ Libraries. [Online]. Available: http://www.boost.org/.

[10] (2018, Aug.) GSL-GNU Scientific Library. [Online]. Available: https://www.gnu.org/software/gsl/.

[11] (2018, Aug.) GLPK-GNU Linear Programming Kit. [Online]. Available: http://www.gnu.org/software/glpk/.

[12] (2018, Aug.) Fast Network Simulation Setup. [Online]. Available: http://fnss.github.io/.

[13] (2018, Aug.) Hiberlite Library. [Online]. Available: https://github.com/paulftw/hiberlite/.

A. Appendix

A1. Modified configurations.xml

```
    <utilities>
      <!--if the path does not exist, no log file will be set.-->
      <logFile></logFile>
      <!-- log level can be set using the environment variable LOG_LEVEL. If
LOG_LEVEL is not set this variable will be used.-->
      <logLevel>error</logLevel>
    </utilities>

    <core>
      <!-- This conditino can be used by all the algorithms to ignore the location
constraint -->
      <ignoreLocationConstrain>false</ignoreLocationConstrain>
      <!--The result database file-->
      <dbPath>/home/njahan/tmp/vne-sim/</dbPath>
      <!-- 0 uses the libraries standard seed -->
      <rngSeed>0</rngSeed>

<rngUseSameSeedForParallelRuns>false</rngUseSameSeedForParallelRuns>
      <!--Verious random number generator types that gsl_rng library implements
may be found at: -->
      <!--https://www.gnu.org/software/gsl/manual/html_node/Random-number-
generator-algorithms.html#Random-number-generator-algorithms -->
      <rngType>gsl_rng_mt19937</rngType>
    </core>

    <vineyard>
      <SubstrateNetwork>
        <path>/home/njahan/tmp/network_files/network_files/SN/</path>

<filename>vy_substrate_net_n_50_outergrid_25_inner_grid_25.txt</filename>
```

```
            </SubstrateNetwork>
            <VirtualNetRequest>
               <path>/home/njahan/tmp/network_files/network_files/VNRs/</path>
               <dir>reqs-12-1000-nodesMin-3-nodesMax-10-grid-25</dir>
               <reqfileExtension>.txt</reqfileExtension>
            </VirtualNetRequest>
            <Constants>
               <revenueMultiplier>1.0</revenueMultiplier>
               <costMultiplier>1.0</costMultiplier>
               <epsilon>1E-6</epsilon>
            </Constants>
            <Configs>
               <setAlpha>false</setAlpha> <!-- if set to false Alpha is automatically set
to 1-->
               <setBeta>false</setBeta>   <!-- if set to false Beta is automatically set to
1-->
               <!-- "deterministic" or "randomized" (use exact words without quotes).
                If it is not specified deterministic approach is selected by default. -->
               <nodeMappingType>deterministic</nodeMappingType>
            </Configs>
            <glpk>
               <LPmodelFile>/home/njahan/tmp/vne-sim/src/Vineyard/files/lp/CNLM-
LP.mod</LPmodelFile>
               <LPdataFile>/home/njahan/tmp/vne-sim/src/Vineyard/files/lp/CNLM-
LP.dat</LPdataFile>
               <MCFmodelFile>/home/njahan/tmp/vne-
sim/src/Vineyard/files/lp/MCF.mod</MCFmodelFile>
               <MCFdataFile>/home/njahan/tmp/vne-
sim/src/Vineyard/files/lp/MCF.dat</MCFdataFile>
               <!-- 0 disabled, 1 enabled -->
               <terminalEnabled>0</terminalEnabled>
            </glpk>
         </vineyard>
```

```
        <MCTS>
            <Simulator>
                <Knowledge>
                    <!-- 0:PURE 1:LEGAL 2:SMART -->
                    <TreeLevel>1</TreeLevel>
                    <!-- 0:PURE 1:LEGAL 2:SMART -->
                    <RolloutLevel>1</RolloutLevel>
                    <SmartTreeCount>10</SmartTreeCount>
                    <SmartTreeValue>1.0</SmartTreeValue>
                </Knowledge>
                <!-- Discount in (0,1] -->
                <discount>1</discount>
                <rewardRange>1000</rewardRange>
            </Simulator>

            <MCTSParameters>
                <MaxDepth>10</MaxDepth>
                <NumSimulations>40</NumSimulations>
                <!-- How many nodes to add at each expansion step -->
                <ExpandCount>1</ExpandCount>
                <!-- if this is set to false Exploration constant will be used otherwise
Exploration constant will be set to 0 if UseRave flag is set or Exploration constant will
be set to rewardRange.-->
                <AutoExploration>false</AutoExploration>
                <!-- One option is to set Exploration Constant c = R_{hi}-R_{lo} -->
                <ExplorationConstant>0.5</ExplorationConstant>
                <!-- Rave Parameters: 0 -> rave is not set -->
                <UseRave>false</UseRave>
                <RaveDiscount>1</RaveDiscount>
                <RaveConstant>0.01</RaveConstant>
                <!-- When set, the baseline rollout algorithm is run. -->
                <DisableTree>0</DisableTree>
                <UseSinglePlayerMCTS>false</UseSinglePlayerMCTS>
                <SPMCTSConstant>10000</SPMCTSConstant>
            <!--0: Action Root Parallelization, 1: Full Tree parallelization -->
            <!-- Setting this parameter does not enable parallelism. You need to
compile the code with option -DENABLE_MPI=on
                to enable parallelism.-->
            <ParallelizationType>1</ParallelizationType>
          </MCTSParameters>
        </MCTS>
        <MCVNE>
          <VNEMCTSSimulator>
            <!-- alpha is the substrate liks weight and beta is the substrate node
weight -->
            <!-- if set to false the weights are 1 -->
            <setAlpha>false</setAlpha>
            <setBeta>false</setBeta>
          </VNEMCTSSimulator>
          <NodeEmbeddingAlgo>
            <!-- MCF of BFS-SP -->
            <LinkEmbedder>BFS-SP</LinkEmbedder>
          </NodeEmbeddingAlgo>
        </MCVNE>

        <GRC>
          <!--unit price charged for computing resources-->
          <alpha>1</alpha>
          <!--unit price charged for bandwidth resources-->
          <beta>1</beta>
          <!--It is a positive small threshold defining the convergence of node
mapping algorithm-->
          <!--Refer to:
           L. Gong, Y. Wen, Z. Zhu, and T. Lee, “Toward profit-seeking virtual
network embedding algorithm via global resource capacity.,” INFOCOM, pp. 1–9,
2014.-->
          <sigma>0.00001</sigma>
```

```
    <dampingFactor>0.85</dampingFactor>
  </GRC>
```

A2. Script for MINIMAL_VINEYARD_GRC_TEST

```
BOOST_AUTO_TEST_CASE(MINIMAL_VINEYARD_GRC_TEST)
{
	std::string vnr_dirs[] =
	{"reqs-12-1000-nodesMin-3-nodesMax-10-grid-25”};

	//GRC BFS
	bool ret = ConfigManager::Instance()-
>setConfig("vineyard.VirtualNetRequest.dir", vnr_dirs[0]);
	assert(ret);
	vne::experiments::GRCNodeBFSLinkExp<> exp =
vne::experiments::GRCNodeBFSLinkExp<> ();

	std::string dbPath;
	std::stringstream dbName;
	dbPath = ConfigManager::Instance()->getConfig<std::string>("core.dbPath");
	dbName << dbPath <<"grc_bfs_" <<  vnr_dirs[0] << ".db";
	std::string str = dbName.str();
	std::shared_ptr<hiberlite::Database> db = DBManager::Instance()-
>createDB(str);

	exp.run();
	db->registerBeanClass<vne::experiments::GRCNodeBFSLinkExp<>>();
	db->dropModel();
	db->createModel();
	db->copyBean(exp);

	//destroy all singleton classes to start fresh for next simulations
	ConfigManager::Destroy();
	IdGenerator::Destroy();
	RNG::Destroy();

	bool ret = ConfigManager::Instance()-
>setConfig("vineyard.VirtualNetRequest.dir", vnr_dirs[0]);
	assert(ret);
	vne::experiments::VineNodeMCFLinkExp<> exp =
vne::experiments::VineNodeMCFLinkExp<> ();

	std::string dbPath;
	std::stringstream dbName;
	dbPath = ConfigManager::Instance()->getConfig<std::string>("core.dbPath");
	dbName << dbPath <<"vineyard_deterministic_" <<  vnr_dirs[0] << ".db";
	std::string str = dbName.str();
	std::shared_ptr<hiberlite::Database> db = DBManager::Instance()-
>createDB(str);

	exp.run();
	db->registerBeanClass<vne::experiments::VineNodeMCFLinkExp<>>();
	db->dropModel();
	db->createModel();
	db->copyBean(exp);

	//destroy all singleton classes to start fresh for next simulations
	ConfigManager::Destroy();
	IdGenerator::Destroy();
	RNG::Destroy();
}
```

A3. Shell Script for Results

```
#!/bin/bash
#run nsf for 1 wave length
#prepare sql query file
#echo  ".mode column" > query.sql

#naming convention:
#mcvne_SimulatorLinkEmbeder_linkEmbeder_minVNRNodeNUM_maxVNRNodeN
UM_grid_numSims.csv

echo "removing files..."
pwd
rm mcvne_bfs_mcf_3_10_25_40.csv mcvne_mcf_mcf_3_10_25_40.csv
mcvne_bfs_bfs_3_10_25_40.csv grc_mcf_3_10_25.csv  grc_bfs_3_10_25.csv
dvine_mcf_3_10_25.csv rvine_mcf_3_10_25.csv
rm single_file_results.csv
touch mcvne_bfs_mcf_3_10_25_40.csv mcvne_mcf_mcf_3_10_25_40.csv
mcvne_bfs_bfs_3_10_25_40.csv grc_mcf_3_10_25.csv  grc_bfs_3_10_25.csv
dvine_mcf_3_10_25.csv rvine_mcf_3_10_25.csv
touch single_file_results.csv

# NOTE: mcvne bfs mcf calculations ADDED to script, original script did not contain
this
echo "processing mcvne bfs mcf files..."
echo
"Arrival_Dist,Mean_Arrival_Rate,Duration_Dist,Mean_Duration,N_VNRs,N_Embedd
ed_VNRs,Acceptance_Ratio,Avg_Rev,Avg_Cost,Avg_Node_Util,Avg_Link_Util,Avg_
Proc_Time" >> mcvne_bfs_mcf_3_10_25_40.csv
#MCVNE BFS MCF
echo  ".output single_file_results.csv" >> query.sql
#echo  ".mode column" >> query.sql
echo  "SELECT t1.Experiment_params_VNRArrivalDist,
            t1.Experiment_params_VNRArrivalDistParam1,
                t1.Experiment_params_VNRDurationDist,
              t1.Experiment_params_VNRDurationDistParam1,
                SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END),
              SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END),
              SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END)*1.0/SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END)*1.0,
              AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_revenue END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_revenue END),
              AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_cost END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_cost END),
              AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_node_stress END),
              AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_link_stress END),
                AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_actual_processing_time END)
          FROM
                MCVNENodeMCFLink AS t1
          LEFT JOIN
                MCVNENodeMCFLink_statistics_items as t2
          ON
                t1.hiberlite_id = t2.hiberlite_parent_id;" >> query.sql
echo      ".quit" >> query.sql

#
                sed -i.back 's/|/,/g' single_file_results.csv
                rm single_file_results.csv.back
                #echo "$nSims \t" >>
                cat mcvne_bfs_mcf_3_10_25_40.csv single_file_results.csv  >
dummy.csv
              rm mcvne_bfs_mcf_3_10_25_40.csv
              mv dummy.csv mcvne_bfs_mcf_3_10_25_40.csv
done
rm query.sql  single_file_results.csv
echo "finished processing mcvne bfs mcf files."

echo "processing mcvne mcf files..."
echo
"Arrival_Dist,Mean_Arrival_Rate,Duration_Dist,Mean_Duration,N_VNRs,N_Embedd
ed_VNRs,Acceptance_Ratio,Avg_Rev,Avg_Cost,Avg_Node_Util,Avg_Link_Util,Avg_
Proc_Time" >> mcvne_mcf_mcf_3_10_25_40.csv
#MCVNE MCF
echo  ".output single_file_results.csv" >> query.sql
#echo  ".mode column" >> query.sql
echo  "SELECT t1.Experiment_params_VNRArrivalDist,
               t1.Experiment_params_VNRArrivalDistParam1,
               t1.Experiment_params_VNRDurationDist,
             t1.Experiment_params_VNRDurationDistParam1,
               SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END),
             SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END),
             SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END)*1.0/SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END)*1.0,
             AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_revenue END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_revenue END),
             AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_cost END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_cost END),
		AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 
t2.item_avg_node_stress END),
		AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 
t2.item_avg_link_stress END),
		 AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 
t2.item_actual_processing_time END)
	FROM
		MCVNENodeMCFLink AS t1
	LEFT JOIN
		MCVNENodeMCFLink_statistics_items as t2
	ON
		t1.hiberlite_id = t2.hiberlite_parent_id;" >> query.sql
echo	".quit" >> query.sql

# NOTE: the following line CHANGED from "for  arrivalRate in 12 14 16 20 25 33 50 
100"
for  arrivalRate in 12
do
		# NOTE: the following line CHANGED from "sqlite3 
../../mcvne_mcf_mcf_reqs-$arrivalRate-1000-nodesMin-3-nodesMax-10-grid-25.db < 
query.sql"
		sqlite3 vne-sim/mcvne_mcf_mcf_reqs-$arrivalRate-1000-nodesMin-3-
nodesMax-10-grid-25.db < query.sql
		sed -i.back 's/|/,/g' single_file_results.csv
		rm single_file_results.csv.back
		#echo "$nSims \t" >>
		cat mcvne_mcf_mcf_3_10_25_40.csv single_file_results.csv  > 
dummy.csv
	rm mcvne_mcf_mcf_3_10_25_40.csv
	mv dummy.csv mcvne_mcf_mcf_3_10_25_40.csv
done
rm query.sql  single_file_results.csv
echo "finished processing mcvne mcf files."
echo "processing mcvne bfs files..."
echo
"Arrival_Dist,Mean_Arrival_Rate,Duration_Dist,Mean_Duration,N_VNRs,N_Embedd
ed_VNRs,Acceptance_Ratio,Avg_Rev,Avg_Cost,Avg_Node_Util,Avg_Link_Util,Avg_
Proc_Time" >> mcvne_bfs_bfs_3_10_25_40.csv
#MCVNE BFS
echo  ".output single_file_results.csv" >> query.sql
#echo  ".mode column" >> query.sql
echo  "SELECT t1.Experiment_params_VNRArrivalDist,
            t1.Experiment_params_VNRArrivalDistParam1,
            t1.Experiment_params_VNRDurationDist,
          t1.Experiment_params_VNRDurationDistParam1,
            SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END),
          SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END),
          SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END)*1.0/SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END)*1.0,
          AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_revenue END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_revenue END),
          AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_cost END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_cost END),
          AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_node_stress END),
          AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_link_stress END),
            AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_actual_processing_time END)
        FROM
                MCVNENodeBFSLink AS t1
        LEFT JOIN
```

```
                    MCVNENodeBFSLink_statistics_items as t2
          ON
                    t1.hiberlite_id = t2.hiberlite_parent_id;" >> query.sql
echo      ".quit" >> query.sql

# NOTE: the following line CHANGED from "for arrivalRate in 12 14 16 20 25 33 50
100"
for  arrivalRate in 12
do
                  # NOTE: the following line CHANGED from "sqlite3
../../mcvne_bfs_bfs_reqs-$arrivalRate-1000-nodesMin-3-nodesMax-10-grid-25.db <
query.sql"
                  sqlite3 vne-sim/mcvne_bfs_bfs_reqs-$arrivalRate-1000-nodesMin-3-
nodesMax-10-grid-25.db < query.sql
                    sed -i.back 's/|/,/g' single_file_results.csv
                    rm single_file_results.csv.back
                    #echo "$nSims \t" >>
                  cat mcvne_bfs_bfs_3_10_25_40.csv single_file_results.csv  >
dummy.csv
               rm mcvne_bfs_bfs_3_10_25_40.csv
               mv dummy.csv mcvne_bfs_bfs_3_10_25_40.csv
done
rm  single_file_results.csv
echo "finished processing mcvne bfs files."

echo "processing mcvne sp bfs files..."
echo
"Arrival_Dist,Mean_Arrival_Rate,Duration_Dist,Mean_Duration,N_VNRs,N_Embedd
ed_VNRs,Acceptance_Ratio,Avg_Rev,Avg_Cost,Avg_Node_Util,Avg_Link_Util,Avg_
Proc_Time" >> mcvne_sp_bfs_bfs_3_10_25_40.csv
# NOTE: the following line CHANGED from "for arrivalRate in 12 14 16 20 25 33 50
100"
for  arrivalRate in 12
do
```

```
                # NOTE: the following line CHANGED from "sqlite3
../../mcvne_sp_bfs_bfs_reqs-$arrivalRate-1000-nodesMin-3-nodesMax-10-grid-25.db
< query.sql"
                sqlite3 vne-sim/mcvne_sp_bfs_bfs_reqs-$arrivalRate-1000-nodesMin-3-
nodesMax-10-grid-25.db < query.sql
                  sed -i.back 's/|/,/g' single_file_results.csv
                  rm single_file_results.csv.back
                  #echo "$nSims \t" >>
                cat mcvne_sp_bfs_bfs_3_10_25_40.csv single_file_results.csv  >
dummy.csv
              rm mcvne_sp_bfs_bfs_3_10_25_40.csv
              mv dummy.csv mcvne_sp_bfs_bfs_3_10_25_40.csv
done
rm query.sql  single_file_results.csv
echo "finished processing mcvne sp bfs files."

echo "processing dvine mcf files..."
echo
"Arrival_Dist,Mean_Arrival_Rate,Duration_Dist,Mean_Duration,N_VNRs,N_Embedd
ed_VNRs,Acceptance_Ratio,Avg_Rev,Avg_Cost,Avg_Node_Util,Avg_Link_Util,Avg_
Proc_Time" >> dvine_mcf_3_10_25.csv

#R-VINE D-VINE
echo  ".output single_file_results.csv" >> query.sql
#echo  ".mode column" >> query.sql
echo  "SELECT t1.Experiment_params_VNRArrivalDist,
              t1.Experiment_params_VNRArrivalDistParam1,
              t1.Experiment_params_VNRDurationDist,
            t1.Experiment_params_VNRDurationDistParam1,
              SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END),
            SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END),
            SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END)*1.0/SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END)*1.0,
            AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_revenue END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_revenue END),
            AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_cost END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_cost END),
            AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_node_stress END),
            AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_link_stress END),
              AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_actual_processing_time END)
         FROM
                  VineNodeMCFLink AS t1
         LEFT JOIN
                  VineNodeMCFLink_statistics_items as t2
         ON
                  t1.hiberlite_id = t2.hiberlite_parent_id;" >> query.sql

echo     ".quit" >> query.sql

# NOTE: the following line CHANGED from "for arrivalRate in 12 14 16 20 25 33 50
100"
for  arrivalRate in 12
do
                # NOTE: the following line CHANGED from "sqlite3
../../vineyard_randomized_reqs-$arrivalRate-1000-nodesMin-3-nodesMax-10-grid-
25.db < query.sql"
                sqlite3 vne-sim/vineyard_randomized_reqs-$arrivalRate-1000-nodesMin-
3-nodesMax-10-grid-25.db < query.sql
                  sed -i.back 's/|/,/g' single_file_results.csv
```

```
                    rm single_file_results.csv.back
                    #echo "$nSims \t" >>
                  cat rvine_mcf_3_10_25.csv single_file_results.csv  > dummy.csv
               rm rvine_mcf_3_10_25.csv
               mv dummy.csv rvine_mcf_3_10_25.csv

                    # NOTE: the following line CHANGED from "sqlite3
../../vineyard_deterministic_reqs-$arrivalRate-1000-nodesMin-3-nodesMax-10-grid-
25.db < query.sql"
                    sqlite3 vne-sim/vineyard_deterministic_reqs-$arrivalRate-1000-
nodesMin-3-nodesMax-10-grid-25.db < query.sql
                    sed -i.back 's/|/,/g' single_file_results.csv
                    rm single_file_results.csv.back
                    #echo "$nSims \t" >>
                  cat dvine_mcf_3_10_25.csv single_file_results.csv  > dummy.csv
               rm dvine_mcf_3_10_25.csv
               mv dummy.csv dvine_mcf_3_10_25.csv
done
rm query.sql  single_file_results.csv
echo "finished processing dvine mcf files."

echo "processing grc mcf files..."
echo
"Arrival_Dist,Mean_Arrival_Rate,Duration_Dist,Mean_Duration,N_VNRs,N_Embedd
ed_VNRs,Acceptance_Ratio,Avg_Rev,Avg_Cost,Avg_Node_Util,Avg_Link_Util,Avg_
Proc_Time" >> grc_mcf_3_10_25.csv

# GRC-MCF
echo  ".output single_file_results.csv" >> query.sql
#echo  ".mode column" >> query.sql
echo  "SELECT t1.Experiment_params_VNRArrivalDist,
               t1.Experiment_params_VNRArrivalDistParam1,
               t1.Experiment_params_VNRDurationDist,
             t1.Experiment_params_VNRDurationDistParam1,
        SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END),
       SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END),
       SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END)*1.0/SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END)*1.0,
       AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_revenue END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_revenue END),
       AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_cost END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_cost END),
       AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_node_stress END),
       AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_link_stress END),
        AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_actual_processing_time END)
     FROM
          GRCNodeMCFLink AS t1
     LEFT JOIN
          GRCNodeMCFLink_statistics_items as t2
     ON
          t1.hiberlite_id = t2.hiberlite_parent_id;" >> query.sql
echo    ".quit" >> query.sql

# NOTE: the following line CHANGED from "for arrivalRate in 12 14 16 20 25 33 50
100"
for  arrivalRate in 12
do
         # NOTE: the following line CHANGED from "sqlite3 ../../grc_mcf_reqs-
$arrivalRate-1000-nodesMin-3-nodesMax-10-grid-25.db < query.sql"
```

```
            sqlite3 vne-sim/grc_mcf_reqs-$arrivalRate-1000-nodesMin-3-nodesMax-
10-grid-25.db < query.sql
              sed -i.back 's/|/,/g' single_file_results.csv
              rm single_file_results.csv.back
              #echo "$nSims \t" >>
            cat grc_mcf_3_10_25.csv single_file_results.csv  > dummy.csv
          rm grc_mcf_3_10_25.csv
          mv dummy.csv grc_mcf_3_10_25.csv
done
rm query.sql  single_file_results.csv
echo "finished processing grc mcf files."

echo "processing grc bfs files..."
echo
"Arrival_Dist,Mean_Arrival_Rate,Duration_Dist,Mean_Duration,N_VNRs,N_Embedd
ed_VNRs,Acceptance_Ratio,Avg_Rev,Avg_Cost,Avg_Node_Util,Avg_Link_Util,Avg_
Proc_Time" >> grc_bfs_3_10_25.csv

# GRC-BFS1
echo  ".output single_file_results.csv" >> query.sql
#echo  ".mode column" >> query.sql
echo  "SELECT t1.Experiment_params_VNRArrivalDist,
            t1.Experiment_params_VNRArrivalDistParam1,
            t1.Experiment_params_VNRDurationDist,
          t1.Experiment_params_VNRDurationDistParam1,
            SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END),
          SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END),
          SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN 1
ELSE 0 END)*1.0/SUM (CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' OR
t2.item_event_type = 'EMBD_FAIL' THEN 1 ELSE 0 END)*1.0,
            AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_revenue END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_revenue END),
            AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_node_cost END)+AVG(CASE WHEN t2.item_event_type =
'EMBD_SUCCESS' THEN t2.item_link_cost END),
            AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_node_stress END),
            AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_avg_link_stress END),
              AVG(CASE WHEN t2.item_event_type = 'EMBD_SUCCESS' THEN
t2.item_actual_processing_time END)
        FROM
                GRCNodeBFSLink AS t1
        LEFT JOIN
                GRCNodeBFSLink_statistics_items as t2
        ON
                t1.hiberlite_id = t2.hiberlite_parent_id;" >> query.sql
echo     ".quit" >> query.sql

# NOTE: the following line CHANGED from "for arrivalRate in 12 14 16 20 25 33 50
100"
for  arrivalRate in 12
do
              # NOTE: the following line CHANGED from "sqlite3 ../../grc_bfs_reqs-
$arrivalRate-1000-nodesMin-3-nodesMax-10-grid-25.db < query.sql"
              sqlite3 vne-sim/grc_bfs_reqs-$arrivalRate-1000-nodesMin-3-nodesMax-
10-grid-25.db < query.sql
                sed -i.back 's/|/,/g' single_file_results.csv
                rm single_file_results.csv.back
                #echo "$nSims \t" >>
              cat grc_bfs_3_10_25.csv single_file_results.csv  > dummy.csv
            rm grc_bfs_3_10_25.csv
            mv dummy.csv grc_bfs_3_10_25.csv
```

```
done
rm query.sql  single_file_results.csv
echo "finished processing grc bfs files."
```