

Instructions on Running Query Mesh

Author: Chuan Lei (chuanlei@cs.wpi.edu)

Modified: May 25, 2010 by Karen Works

Last Modified: October 27, 2010 by Chuan Lei

Contents

Get Started	Page 3
Necessary system configuration files	Page 4
How to run a query mesh query	Page 5
Step A. Collect statistic samples	Page 5
Step B. Create a training tuple set	Page 7
Step C. Create the decision tree classifier	Page 7
Step D. Execute the query	Page 8
Input File Layouts	Page 11
queryplanSampleStream.xml	Page 11
SystemConfigSampleStream.xml	Page 12
queryplanCollectTrainingSetStream.xml	Page 15
SystemConfigCollectTrainingSetStream.xml	Page 17
StreamConfig.xml	Page 20
QueryMeshStreams.xml	Page 21
optimizer_Input.xml	Page 22
SystemConfigQueryMeshPlan.xml	Page 24
AutoGeneratedQueryMeshQueryPlan.xml	Page 27

0. Get Started

Getting started with Query Mesh is pretty easy if you start playing around with the example provided in our package and read the following introductory material. To run Query Mesh, you will need the software: Download [Eclipse](#) and install it and get the latest version [Java SDK](#). You can just check out the Query Mesh [source code](#) and import it into your eclipse to get started. Figure 1 shows Query Mesh source packages imported in Eclipse.

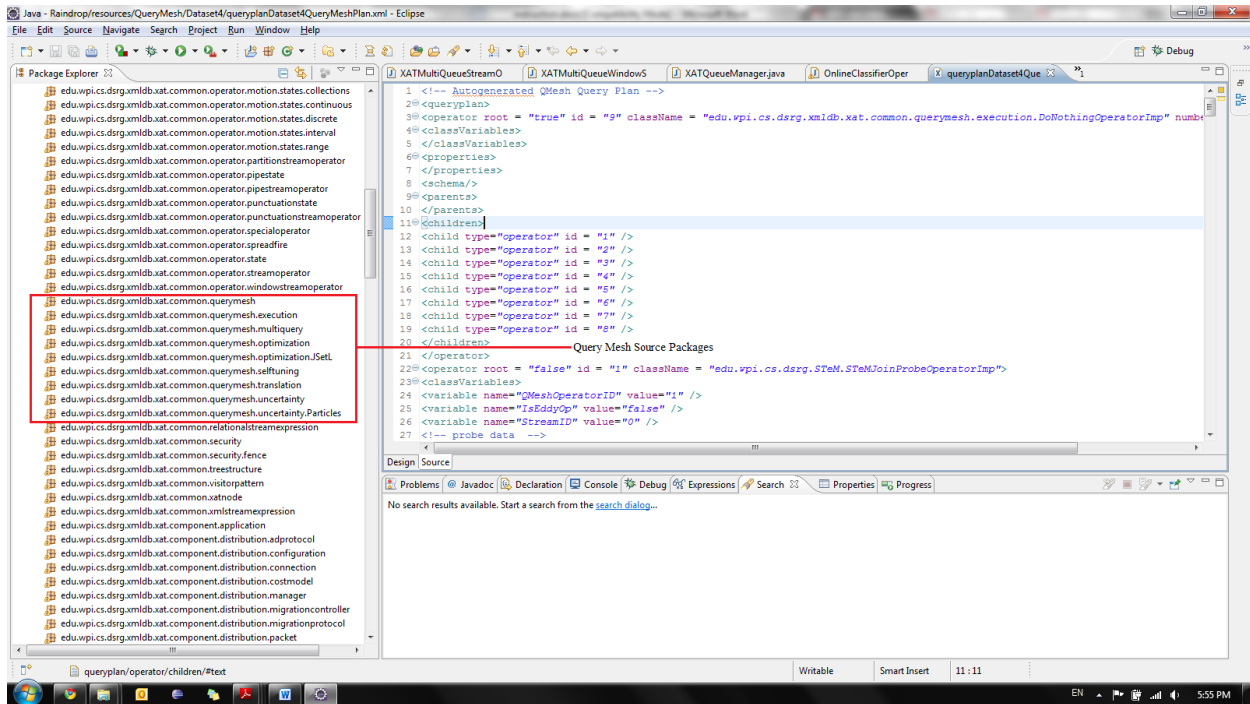


Figure 1. Query Mesh Source Packages in Eclipse

Note: we only tested Query Mesh on eclipse. You might need to make changes to Query Mesh configurations if you plan to run it on other IDEs.

Before you start playing with Query Mesh, you need to create certain folders for Query Mesh to save generated results. The following lists the default locations of the folders used by Query Mesh. DO create them before you run Query Mesh.

- C:\QueryMesh\config
- C:\QueryMesh\execution_stats
- C:\QueryMesh\experiments
- C:\QueryMesh\optimizer_output
- C:\QueryMesh\statistics_samples
- C:\QueryMesh\stats
- C:\QueryMesh\tmp
- C:\QueryMesh\training_sets

1. Necessary system configuration files

Query Mesh needs configuration files. Now people use XML for everything and so do we. This section describes configuration files that we made for Query Mesh and the underneath stream engine on which it runs.

Note: The end of document is appended with an example of each file and all files can be found in the resource folder in Query Mesh.

queryPlanSampleStream.xml: the file is used to collect sample data from streams for statistics purposes. It contains information about the sample rate, sampling heuristic, the output file which will contain the sampled data. The sample data set contains ALL attributes, and it is used for overall statistics estimations, e.g., when trying to estimate overall stream statistics. The number of queryPlanSampleStream.xml files is equal to the number of streams.

queryPlanCollectTrainingSetStream.xml: the file is used for decision tree construction. It contains a subset of tuple attributes (only the ones that will be used for building decision tree). If classifier model changes, we may substitute (modify) the training tuple set structure without effecting the overall statistics computation. The number of queryPlanCollectTrainingSetStream.xml files is equal to the number of streams.

optimizer_input.xml: the file is used for running the query. It contains information about the operators and the location of sample data sets.

queryplanQueryMeshPlan.xml: the file is auto-generated by query mesh optimizer. It contains all the information about a query plan: operators, parameters for the operators, streams and so forth. It is the schema for the query plan to be executed.

StreamsConfig.xml: the file contains the following information, the name of the Data File, schema and the format of the file (enable the parser to parse the file and generate the tuples), and inter-arrival distribution (information about what distribution to use or what attribute to use in the schema as a time stamp of the file).

QueryMeshStreams.xml: the file specifies the number of streams used in query mesh.

2. How to run a Query Mesh query

There are four steps in executing a Query Mesh query.

- Step A. Collect statistic samples for every stream in the query
- Step B. Create a training tuple set for every stream in the query
- Step C. Create the decision tree classifier
- Step D. Execute the query

In the following example, you will see Query Mesh generates and executes a query plan for a query over 5 streams. In order to run all the following applications, you will need to set up the run configurations correctly as shown in Figure 2. After each run, all running applications need to be stopped manually.

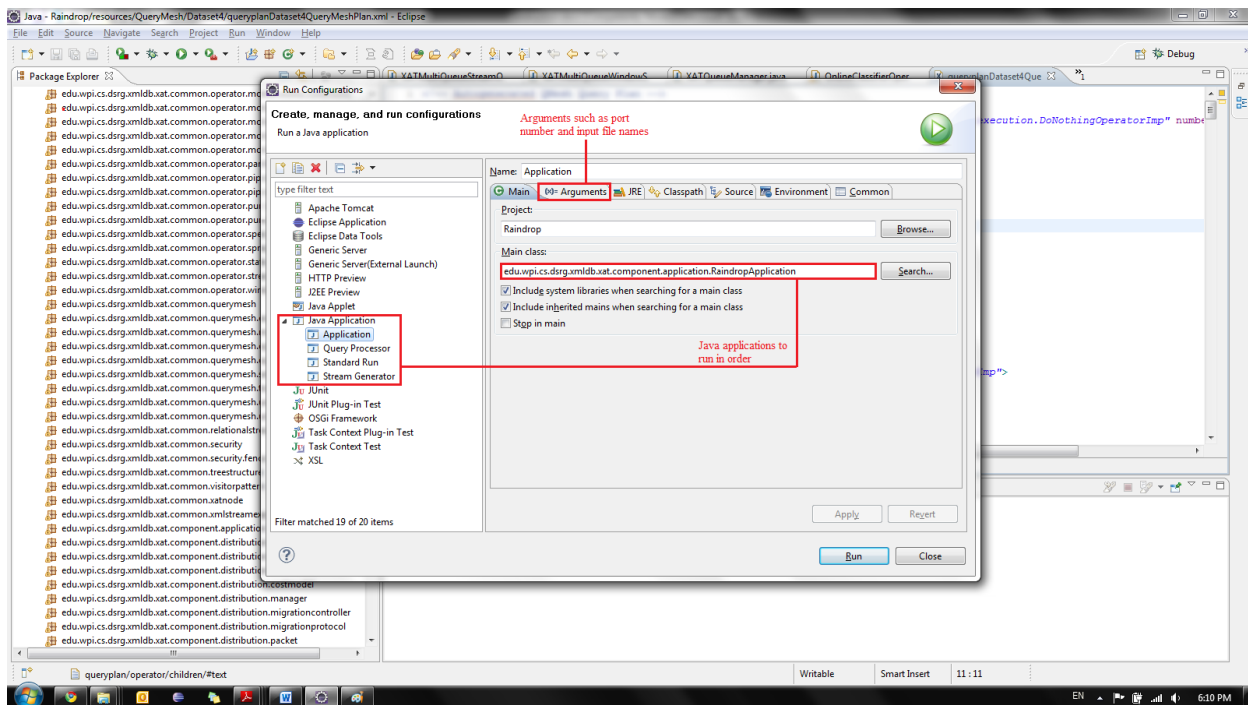


Figure 2. Run Configurations for Query Mesh

Step A. Collect statistics samples for every stream in the query

To run the sample statistics files collection, components of the system that need to be started in the respective order in Eclipse.

Query Processor

(edu.wpi.cs.dsrg.xmlldb.xat.component.executioncontroller.DistributedExperiment Setup 8001)

Note: 8001 is the port number.

Stream Generator

(edu.wpi.cs.dsrg.xmlldb.xat.component.streamgenerators.server.XATStreamGenerat or 15000 resources\QueryMesh\Example\StreamsConfig.xml)

Note: 15000 is the port number and resources\QueryMesh\Example\StreamsConfig.xml is the stream configuration.

Application

(edu.wpi.cs.dsrg.xmlldb.xat.component.application.RaindropApplication 16001)

Note: 16001 is the port number.

Standard Run

(edu.wpi.cs.dsrg.xmlldb.xat.component.executioncontroller.DistributedExperimentSetup resources\QueryMesh\Example\SystemConfigSampleStream0.xml)

Note: resources\QueryMesh\Example\SystemConfigSampleStream0.xml is the system configuration.

Repeat running the above applications five times and each time change the arguments of Standard Run in accordance with the stream number. If I am collecting statistic samples for stream 3, I need to change my argument from resources\QueryMesh\Example\SystemConfigSampleStream0.xml to resources\QueryMesh\Example\SystemConfigSampleStream3.xml

Input: The following files are required as inputs:

- query plan (e.g., queryplanSampleStream.xml)
- system configuration (e.g., SystemConfigSampleStream.xml)
- stream layout (e.g., StreamsConfig.xml)
- stream generator (e.g., QueryMeshStreams.xml).

Note: An example of each file can be found at the end of this document and example files are provided in resources folder in Query Mesh.

Expected Output: The output file for each stream will be a file named SAMPLE_STREAM#.txt, where # is the stream number. After successfully running the above applications, five sample stream files will be generated and placed under C:\QueryMesh\statistics_samples (if you keep the default directory path in your configuration file). The name of these files will be used in Step D.

The file structure of outputs is shown in Figure 3.

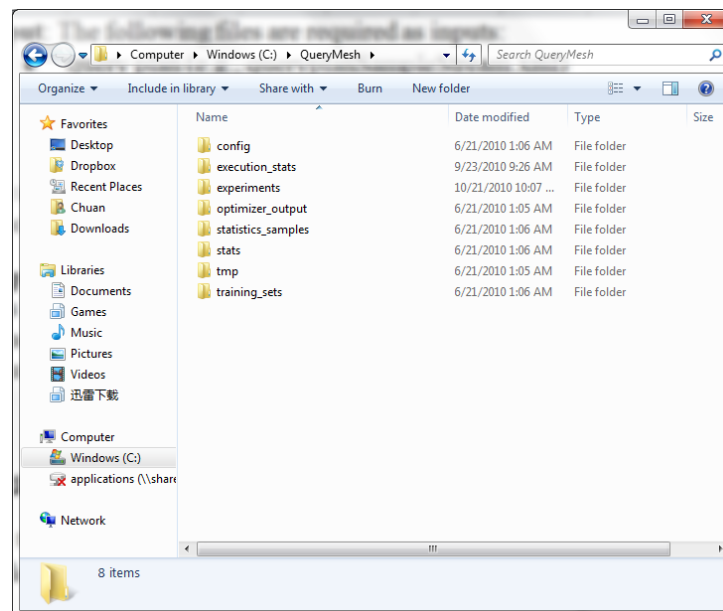


Figure 3. File Structure of Query Mesh Outputs

Step B. Collect training tuple set for every stream in the query

To run the training tuple set collection, components of the system that need to be started in the respective order in your eclipse.

Query Processor

```
(edu.wpi.cs.dsrg.xmlldb.xat.component.executioncontroller.DistributedExperiment  
Setup 8001)
```

Stream Generator

```
(edu.wpi.cs.dsrg.xmlldb.xat.component.streamgenerators.server.XATStreamGenerat  
or 15000 resources\QueryMesh\Example\StreamsConfig.xml)
```

Application

```
(edu.wpi.cs.dsrg.xmlldb.xat.component.application.RaindropApplication 16001)
```

Standard Run

```
(edu.wpi.cs.dsrg.xmlldb.xat.component.executioncontroller.DistributedExperimen  
tSetup resources\QueryMesh\Example\SystemConfigCollectTrainingSetStream0.xml)
```

Repeat running the above applications five times and each time change the arguments of Standard Run in accordance with the stream number. If I am collecting statistic samples for stream 3, I need to change my argument from

```
resources\QueryMesh\Example\SystemConfigCollectTrainingSetStream0.xml to  
resources\QueryMesh\Example\SystemConfigCollectTrainingSetStream3.xml.
```

Input: The following files are required as inputs:

- query plan (e.g., queryplanSampleStream.xml)
- system configuration (e.g., SystemConfigCollectTrainingSetStream0.xml)
- stream layout (e.g., StreamsCon_g.xml)
- stream generator (e.g., QueryMeshStreams.xml).

Note: A sample of each file can be found at the end of this document and example files are provided in resources folder in Query Mesh.

Expected Output: The output file for each stream will be a file named TRAINING_SET_STREAMS#.txt, where # is the stream number. After successfully running the above applications, five sample stream files will be generated under C:\QueryMesh\training_sets (if you keep the default directory path in your configuration file). These files will be used in Step D.

Step C. Create the decision tree classifier

The following needs to be executed to run Query Mesh optimizer.

```
(edu.wpi.cs.dsrg.xmlldb.xat.common.querymesh.optimization.QueryMeshOptimizer  
resources\QueryMesh\Example\optimizer_input.xml 0 "SA"  
C:\QueryMesh\optimizer_output\optimizer_output.xml  
C:\QueryMesh\config\qm_query_plan.xml  
C:\QueryMesh\stats\stats.txt)
```

Input: The input parameters are:

- optimizer input file (e.g., "optimizer_input.xml")
- heuristic (e.g., "SA")

- the directory for the final query mesh configuration file (e.g., C:\QueryMesh\optimizer_output\optimizer_output.xml)
- the query plan file (e.g., C:\QueryMesh\config\qm_query_plan.xml)
- the statistics output file (e.g., C:\QueryMesh\stats\stats.txt).

The following are heuristic options. "SA" is the simulated annealing algorithm and "II" is the iterative improvement algorithm.

Expected Output: The output file will be a query plan configuration file. It can be run in Query Mesh executor (i.e., "qm_query_plan.xml" file). Figure 4 shows the console output during the execution of Query Mesh optimizer.

```

Java EE - Eclipse
File Edit Navigate Search Project Run Design Window Help
<terminated> Query Mesh Optimizer [Java Application] C:\Program Files\Java\jdk1.6.0_25\bin\javaw.exe (Oct 24, 2010 6:32:52 PM)
-----END INPUT DETAILS-----
Cleaning up working directory...
Removed: C:\Rimma\Research\QueryMesh\tmp\qmInput0.txt
Removed: C:\Rimma\Research\QueryMesh\tmp\qmInput1.txt
Removed: C:\Rimma\Research\QueryMesh\tmp\qmOutput0.txt
Removed: C:\Rimma\Research\QueryMesh\tmp\qmOutput1.txt
Optimizer inputs:
  search strategy: SIMULATED_ANNEALING
  value of m: 0
  start solution: CONTENT_DRIVEN
Creating local query meshes...

-----QUERY MESH [1], Stream [1]-----
# of training tuples: 101
--- New Stream QM Search ---
-----
[87, 28]
[1, 2, 3, 4, 7, 8, 10, 11, 12, 15, 16, 23, 25, 24, 29, 31, 30, 32, 39, 36, 37, 43, 40, 46, 47, 45, 50, 49, 48, 55, 53, 52, 56, 62, 61, 60, 70, 71, 67, 76, 78, 79, 80, 83, 82, 93, 92, 89, 88,
[68, 69, 64, 65, 5, 66, 6, 77, 9, 72, 75, 13, 74, 14, 75, 85, 17, 84, 19, 86, 18, 81, 21, 20, 22, 35, 27, 94, 26, 91, 34, 35, 33, 98, 38, 96, 42, 41, 44, 51, 54, 59, 58, 57, 63]
Current partition [87, 28][1, 2, 3, 4, 7, 8, 10, 11, 12, 15, 16, 23, 25, 24, 29, 31, 30, 32, 39, 36, 37, 43, 40, 46, 47, 45, 50, 49, 48, 55, 53, 52, 56, 62, 61, 60, 70, 71, 67, 76, 78, 79, 80, 83, 82, 93, 92, 89, 88,
-----
1) Iteration
  Decision tree absolute cost: 198.0
  Decision tree normalized cost: 1.98
  ruster[87, 28]:
    Op id = 1, w/ inputs: 0, 1: Partition-based selectivity: 0.0
    Current stream: 1:
  Left Cost 2.0 Right Cost 100.0 Selectivity 0.0 JOIN: 0.0
  , Route cost: 0.0
  ruster[1, 2, 3, 4, 7, 8, 10, 11, 12, 15, 16, 23, 25, 24, 29, 31, 30, 32, 39, 36, 37, 43, 40, 46, 47, 45, 50, 49, 48, 55, 53, 52, 56, 62, 61, 60, 70, 71, 67, 76, 78, 79, 80, 83, 82, 93, 92, 89, 88,
    Op id = 1, w/ inputs: 0, 1: Partition-based selectivity: 0.0
    Current stream: 1:
  
```

Figure 4. Output When Running Query Mesh Optimizer

Step D. Execute the query

To run Query Mesh executor, the following components need to be started in the respective order:

Query Processor

(edu.wpi.cs.dsrg.xmlldb.xat.component.executioncontroller.DistributedExperiment Setup 8001)

Stream Generator

(edu.wpi.cs.dsrg.xmlldb.xat.component.streamgenerators.server.XATStreamGenerator or 15000 resources\QueryMesh\Example\StreamsConfig.xml)

Application

(edu.wpi.cs.dsrg.xmlldb.xat.component.application.RaindropApplication 16001)

Standard Run

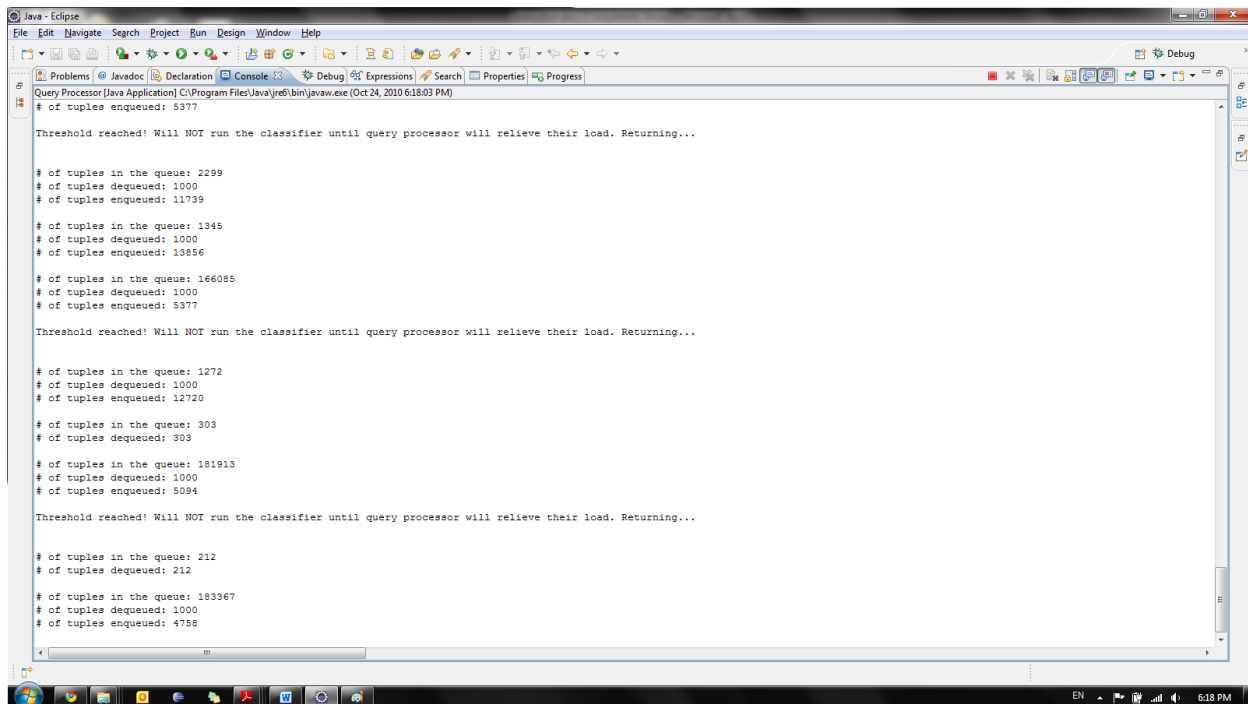
(edu.wpi.cs.dsrg.xmlldb.xat.component.executioncontroller.DistributedExperiment
Setup resources\QueryMesh\Example\SystemConfigQueryMeshPlan.xml)

Input: The following files are required as inputs:

- query plan (e.g., qm_query_plan.xml)
- system configuration (e.g., SystemConfigQueryMeshPlan.xml)
- stream layout (e.g., StreamsConfig.xml)
- stream generator (e.g., QueryMeshStreams.xml).

Note: A sample of each file can be found at the end of this document and example files are provided in resources folder in Query Mesh.

Expected Output: The results of the query plan execution will be displayed in your Eclipse console as shown in Figure 5. The experiment results will be saved under C:\QueryMesh\experiments and C:\QueryMesh\execution_stats, if you use the default setting. Figure 6 shows one of the experiment results generated by Query Mesh. For better view, you may import the file to Excel.



```
Query Processor [Java Application] C:\Program Files\Java\jre6\bin\java.exe (Oct 24, 2010 6:18:03 PM)
# of tuples enqueued: 5377

Threshold reached! Will NOT run the classifier until query processor will relieve their load. Returning...

# of tuples in the queue: 2299
# of tuples dequeued: 1000
# of tuples enqueued: 11739

# of tuples in the queue: 1345
# of tuples dequeued: 1000
# of tuples enqueued: 13856

# of tuples in the queue: 166085
# of tuples dequeued: 1000
# of tuples enqueued: 5377

Threshold reached! Will NOT run the classifier until query processor will relieve their load. Returning...

# of tuples in the queue: 1272
# of tuples dequeued: 1000
# of tuples enqueued: 12720

# of tuples in the queue: 303
# of tuples dequeued: 303

# of tuples in the queue: 181913
# of tuples dequeued: 1000
# of tuples enqueued: 5094

Threshold reached! Will NOT run the classifier until query processor will relieve their load. Returning...

# of tuples in the queue: 212
# of tuples dequeued: 212

# of tuples in the queue: 183367
# of tuples dequeued: 1000
# of tuples enqueued: 4758
```

Figure 5. Output When Running Query Mesh Executor

runtime_stats_128524823819.txt - Notepad

OPERATOR_ID	NUMBER_OF_TUPLES_OUTPUTTED_TOTAL	NUMBER_OF_TIMES_RUN	NUMBER_OF_TUPLES_IN_INPUT_QUEUES	NUMBER_OF_TUPLES_DEQUEUED_TOTAL	NUMBER_OF_TUPLES_DEQUEUED
TIME_TOOK_TO_RUN_TOTAL	NUMBER_OF_TUPLES_IN_OUTPUT_QUEUES	SELECTIVITY	USED_MEMORY	TOTAL_MEMORY	AVERAGE_TUPLE_PROCESSING_TIME
GREEDY_PRIORITY	AVERAGE_OUTPUT_RATE	THROUGHPUT	OUTPUT_RATE	NUMBER_OF_TIMES_OPERATORS_WERE_RUN	TOTAL_TUPLES_IN_QUEUES
AVERAGE_TUPLE_DELAY	AVERAGE_TUPLE_DELAY	AVERAGE_TUPLE_DELAY	AVERAGE_TUPLE_DELAY	AVERAGE_TUPLE_DELAY	AVERAGE_TUPLE_DELAY
OnlineClassifierOperatorImp 00 ACTIVE	20	1	19	19	102.0
5.368421052631579	1.2941176470588236	0.19607843137254902	0	0	1
OnlineClassifierOperatorImp 00 ACTIVE	60	2	57	38	196.0
3.43859649122807	0.740708086253368	0.20408163265306123	0	0	2
StemJoinInProbOperatorImp 56 ACTIVE	0	1	20	20	7.0
22.857142857142858	0.0	0	0	0	1.0
StemJoinInProbOperatorImp 11 ACTIVE	0	3	59.0	0	0
0.0	0	0	1285248244838	0	0
StemJoinInProbOperatorImp 44 ACTIVE	0	1	20	20	1.0
0.0	0	0	1285248244838	0	0
OnlineClassifierOperatorImp 00 ACTIVE	80	3	76	19	274.0
3.6052631578947367	0.48678492919584887	0.072992700729927	0	0	6
StemJoinInProbOperatorImp 33 ACTIVE	0	2	20	20	39.0
4.102564102564103	0.0	0	0	0	1.0
OnlineClassifierOperatorImp 00 ACTIVE	142	4	135	59	500.0
3.7037037037037037	0.34514078316813257	0.124	0	8	62.0
StemJoinInProbOperatorImp 88 ACTIVE	0	1	20	20	9.0
17.777777777777778	0.0	0	0	0	1.0
StemJoinInProbOperatorImp 66 ACTIVE	0	2	162.0	0	0
0.34146341463414637	12.351141445045894	0.0	0	0	10.0
StemJoinInProbOperatorImp 44 ACTIVE	0	2	41	21	6.0
0.14634146341463414	42.0850040096231	0.0	0	0	11.0
OnlineClassifierOperatorImp 00 ACTIVE	307	5	295	160	661.0
2.240677966101695	0.286377270660982	0.24962178517397882	0	0	12
StemJoinInProbOperatorImp 88 ACTIVE	0	2	61	41	115.0
1.885245906393444	3.510317852807625	0.0	0	0	13
StemJoinInProbOperatorImp 66 ACTIVE	0	3	268	227	415.0
1.5485074626865671	3.782046877876019	0.0	0	0	14
StemJoinInProbOperatorImp 77 ACTIVE	0	1	120	120	173.0
1.4416666666666667	5.5491329479768785	0.0	0	0	15
StemJoinInProbOperatorImp 33 ACTIVE	0	2	61	41	55.0
0.9016393442621951	3.067614819471182	0.0	0	0	16
StemJoinInProbOperatorImp 11 ACTIVE	0	2	61	41	26.0
0.4262295081967213	17.022018748637453	0.0	0	0	17
StemJoinInProbOperatorImp 55 ACTIVE	0	1	214	214	374.0
1.7476635514018692	4.577540106951871	0.0	0	0	18
StemJoinInProbOperatorImp 44 ACTIVE	0	3	1041	1000	471.0
0.4524495677234294	12.928674479778403	0.0	0	0	19
OnlineClassifierOperatorImp 00 ACTIVE	307	6	980	1255	960
0.526692270916335	0.39427080809344386	0.0	0	0	20
StemJoinInProbOperatorImp 22 ACTIVE	0	1	16735	1000	334.0
23.95209580838323	0.0	0	0	0	1.0
StemJoinInProbOperatorImp 77 ACTIVE	0	2	1953	1120	1000
4.98377676833252	0.0	0	0	0	21
StemJoinInProbOperatorImp 33 ACTIVE	0	2	282	221	65.0
0.23049645380070822	3.1842088033422543	0.0	0	0	23
StemJoinInProbOperatorImp 55 ACTIVE	0	2	3135	1214	1000
0.47364085667215816	3.994315162545415	0.0	0	0	24
StemJoinInProbOperatorImp 44 ACTIVE	0	3	3079	2041	1000
0.327290543810534	9.208922560719778	0.0	0	0	25
OnlineClassifierOperatorImp 00 ACTIVE	307	7	1780	3015	1760
0.21923714759535656	0.43689358239072845	0.0	0	0	26
doNothingOperatorImp 99 ACTIVE	16359	1	18540	488.0	16359
0.026321467088166126	270.4118854590166	33.522540883606056	16359	1090.6	27
StemJoinInProbOperatorImp 22 ACTIVE	0	2	51823	2000	1000
13.646055437100214	0.0	0	95600.0	0	0
StemJoinInProbOperatorImp 77 ACTIVE	0	3	16571	2120	1000
0.47735849056603774	4.250987252207855	0.0	16359	1022.4375	29
StemJoinInProbOperatorImp 55 ACTIVE	0	3	2123	2214	1000
0.4069557362240289	3.7046532057127926	0.0	16359	962.2941176470588	30

Figure 6. Experiment Result Generated by Query Mesh Executor

Example of queryplanSampleStream.xml

```
<queryplan>
<operator root = "true" id = "1" className =
"edu.wpi.cs.dsrg.xmlldb.xat.common.querymesh.execution.DoNothingOperatorImp">
<classVariables>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
</parents>
<children>
<child type="operator" id = "2"/>
</children>
</operator>
<operator root = "false" id = "2" className =
"edu.wpi.cs.dsrg.xmlldb.xat.common.querymesh.SampleStatsCollectorOperatorImp">
<classVariables>
<!-- Generic Properties -->
<variable name="window_size" value="100"/>
<variable name="operator_state"
value="edu.wpi.cs.dsrg.xmlldb.xat.common.querymesh.SingleStreamSamplingState"/>
<!-- Variable used to determine the sampling strategy -->
<variable name="num_sample_windows" value="10"/>
<variable name="sample_windows_size" value="100"/>
<variable name="tuples_per_window_to_sample" value="10"/>
<variable name="sampling_heuristic" value="simple_random"/>
<!-- Variable used to specify location of the sample DUMP -->
<!-- This file will be used in the construction of the decision tree -->
<variable name="sample_dump_file"
value="C:\\QueryMesh\\statistics_samples\\SAMPLE_STREAM0.txt"/>
<!-- Decision tree header contains the names of the attributes to be used -->
<!-- in the decision tree. You can just copy these attributes directly from -->
<!-- the stream specification (but you will need to add an additional parameter to
ignore/to use)-->
<!-- tuple attribute in the decision tree)-->
<decisionTreeHeader>
<attribute name="counter" type="int" use="false" is_target="false"/>
<attribute name="value" type="string" use="true" is_target="true"/>
</decisionTreeHeader>
<!-- There are 2 types of histograms: (a) number-based, (b) string-based -->
<!-- parameters for the number-based histogram: # of buckets, min and max values -->
<!-- parameters for the string-based histogram: nothing, each new string will be
assigned its own bucket -->
<histograms>
<hist id="1" attr_idx="1" type="string" num_buckets="-1" min_val="-1" max_val="-1" />
</histograms>
</classVariables>
<properties></properties>
<schema/>
<parents>
<parent id = "1"/>
</parents>
<children>
<child type="stream" name="Stream0" queueId="0"/>
</children>
</operator>
</queryplan>
```

Note: please make sure that you have an existing directory that matches the highlighted part.

Example of SystemConfigSampleStream.xml

```
<config>
<system>
<property name="StatisticsGatherer" value="on"/>
<property name="AVERAGE_WEIGHT" value=".875"/>
<property name="EXECUTION_CONTROLLER"
value="edu.wpi.cs.dsrg.xmldb.xat.component.executioncontroller.DistributedExecutionCon
troller"/>
<property name="DATA_MODEL"
value="edu.wpi.cs.dsrg.xmldb.xat.common.dag.XATMemoryQueueImp"/>
</system>
<distribution>
<property name="DISTRIBUTION_PATTERN"
value="edu.wpi.cs.dsrg.xmldb.xat.component.distribution.pattern.GroupingDistribution"/>
<property name="WORKLOAD_COST_MODEL"
value="edu.wpi.cs.dsrg.xmldb.xat.component.distribution.costmodel.NumTuplesInQueue"/>
<property name="REDISTRIBUTION_POLICY"
value="edu.wpi.cs.dsrg.xmldb.xat.component.distribution.redistribution.Balance"/>
<property name="REDISBRIBUTION_TIME" value="15000"/>
<property name="DISTRIBUTION_DELAY" value="10000"/>
<property name="STATE_SIZE_THRESHOLD" value="-1"/>
<property name="REDISTRIBUTION_PERCENT" value="110"/>
<property name="REDISTRIBUTION_SCOPE" value="global"/>
</distribution>
<experiment>
<property name="EXECUTION_DURATION" value="60000"/>
<property name="PRINT_OUT_META_INFORMATION" value="true"/>
<property name="STREAM_CONFIG_FILE_NAME"
value="resources\QueryMesh\Example\QueryMeshStreams.xml"/>
<property name="STREAM_DURATION" value="60000"/>
<treeProperties/>
</experiment>
<machines>
<machine>
<property name="NAME" value="Machine 1"/>
<property name="HOST_ADDRESS" value="localhost"/>
<property name="PORT" value="8001"/>
<property name="TUPLE_RECEIVER_PORT" value="9001"/>
<property name="CONNECTION_LISTENER_PORT" value="10001"/>
<property name="ADAPTIVE_HEURISTIC"
value="edu.wpi.cs.dsrg.xmldb.xat.component.scheduler.NeverRotateAdapter"/>
<property name="DEBUG" value="false"/>
<property name="UPDATE_OPERATOR_PROPERTY_FREQUENCY" value="1000"/>
<property name="UPDATE_TREE_PROPERTY_FREQUENCY" value="1000"/>
<property name="STATUS_CHECK_FREQUENCY" value="300000000"/>
<property name="STATS_TABLE_STATS" value="false"/>
<property name="GUI" value="off"/>
<!-- OPTIMIZATION_STRATEGY defines the migration strategy used during execution. The
possible value can be "off", "MS" stands for moving state and "PT" stands for parallel
track -->
<property name="MIGRATION_STRATEGY" value="off"/>
<property name="MIGRATION_INTERVAL" value="30000"/>
<scheduling>
<property name="WORKLOAD_RATIO" value="1"/>
<property name="WORKLOAD_THRESHOLD" value="50"/>
<preferences>
<preference statistic="TOTAL_TUPLES_IN_QUEUES" quantifier="min" weight="1"/>
<!--<preference statistic="OUTPUT_RATE" quantifier="max" weight=".5"/>-->
</preferences>
<algorithms>
```

```

<property name="RoundRobin"
value="edu.wpi.cs.dsrg.xmldb.xat.component.scheduler.RoundRobinScheduler"/>
</algorithms>
</scheduling>
</machine>
</machines>
<QueryPlans>
<QueryPlan>
<property name="QUERY_ID" value="1"/>
<property name="edu.wpi.cs.dsrg.xmldb.xat.component.queryplangenerator"
value="edu.wpi.cs.dsrg.xmldb.xat.component.queryplangenerator.DistributedFromXMLFileQueryPlanGenerator"/>
<property name="FILE_NAME"
value="resources\QueryMesh\Example\queryplanSampleStream0.xml"/>
<property name="QUERY_FILE_NAME" value="resources\QueryMesh\Example\query.txt"/>
</QueryPlan>
</QueryPlans>
<Applications>
<Application>
<property name="HOST_ADDRESS" value="127.0.0.1"/>
<property name="PORT" value="16001"/>
<property name="CONNECTS_TO" value="1"/>
</Application>
</Applications>
<outputFormat>
<property name="FORMAT" value="csv"/>
<property name="FILE_NAME" value="outputQueryMesh_SinglePlan.csv"/>
<property name="PRINT_EMPTY_ROW" value="false"/>
<property name="ALWAYS_PRINT_HEADERS" value="false"/>
<property name="OVERALL_FILENAME" value="outputQueryMesh_Overall.csv"/>
<property name="ALWAYS_PRINT_OVERALL" value="false"/>
<property name="FREQUENCY" value="5000"/>
<outputColumnNames>
<property name="TIME_TOOK_TO_RUN_TOTAL" value=""/>
<property name="USED_MEMORY" value=""/>
<property name="AVERAGE_TUPLE_DELAY" value=""/>
<property name="AVERAGE_TUPLE_PROCESSING_TIME" value=""/>
<property name="NUMBER_OF_TIMES_OPERATORS_WERE_RUN" value=""/>
<property name="TOTAL_TUPLES_IN_QUEUES" value=""/>
<property name="SELECTIVITY" value=""/>
<property name="THROUGHPUT" value=""/>
<property name="OUTPUT_RATE" value=""/>
</outputColumnNames>
</outputFormat>
<!-- Some of the Statistics to Gather. It is important that the everyTimeOperator
properties are kept intact (including order). Altering the order or makeup could
result in either.
1. a scheduler not working correctly
2. another property not being updated correctly
The value wont be used, but it keeps it consistent with the rest of the document
If a property appears in the printout (above), then it should be listed here.
The 2nd group of properties is optional metrics.
Not all properties can be specified here because some rely on outside information.
The everyTimeOperator element contains properties that are updated after every time an
operator runs. The periodicOperator element lists all properties that can be updated
at regular intervals (defined as UPDATE_PROPERTY_FREQUENCY property). -->
<statisticsToGather>
<operatorStatistics>
<everyTimeTree>
<!-- I cant think of any tree properties that would need to be updated every time so
this isnt supported -->
</everyTimeTree>
</periodicTree>

```

```

<property name="THROUGHPUT" value=""/>
<property name="OUTPUT_RATE" value=""/>
<property name="NUMBER_OF_TIMES_OPERATORS_WERE_RUN" value=""/>
<property name="TOTAL_TUPLES_IN_QUEUES" value=""/>
<property name="AVERAGE_TUPLE_DELAY" value=""/>
<property name="TOTAL_TUPLES_IN_STATES" value=""/>
</periodicTree>
<everyTimeOperator>
<!-- These properties are updated every time an operator runs -->
<property name = "NUMBER_OF_TUPLES_OUTPUTTED_TOTAL" value = ""/>
<property name="NUMBER_OF_TIMES_RUN" value=""/>
<property name="NUMBER_OF_TUPLES_IN_INPUT_QUEUES" value=""/>
<property name="NUMBER_OF_TUPLES_DEQUEUED_TOTAL" value=""/>
<property name="NUMBER_OF_TUPLES_DEQUEUED" value=""/>
<property name="TIME_TOOK_TO_RUN_TOTAL" value=""/>
<property name="NUMBER_OF_TUPLES_IN_OUTPUT_QUEUES" value=""/>
<property name="SELECTIVITY" value=""/>
</everyTimeOperator>
<periodicOperator>
<!-- These properties will be updated at regular intervals -->
<property name="AVERAGE_TUPLE_PROCESSING_TIME" value=""/>
<property name="GREEDY_PRIORITY" value=""/>
<property name="AVERAGE_OUTPUT_RATE" value=""/>
</periodicOperator>
<everyTimeSystem>
<property name="USED_MEMORY" value=""/>
<property name="FREE_MEMORY" value=""/>
<property name="TOTAL_MEMORY" value=""/>
<property name="USED_MEMORY_PERCENTAGE" value=""/>
</everyTimeSystem>
</operatorStatistics>
</statisticsToGather>
</config>

```

Note: please make sure that all highlighted parts have their matching files on your local drive.

Example of queryplanCollectTrainingSetStream.xml

```
<queryplan>
<operator root = "true" id = "1" className =
"edu.wpi.cs.dsrg.xmldb.xat.common.querymesh.execution.DoNothingOperatorImp">
<classVariables>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
</parents>
<children>
<child type="operator" id = "2"/>
</children>
</operator>
<operator root = "false" id = "2"
className = "edu.wpi.cs.dsrg.xmldb.xat.common.querymesh.HistogramBuilderOperatorImp">
<classVariables>
<!-- Generic Properties -->
<variable name="window_size" value="100"/>
<variable name="operator_state"
value="edu.wpi.cs.dsrg.xmldb.xat.common.querymesh.SingleStreamSamplingState"/>
<!-- Variable used to determine the sampling strategy -->
<variable name="num_sample_windows" value="10"/>
<variable name="sample_windows_size" value="100"/>
<variable name="tuples_per_window_to_sample" value="10"/>
<variable name="sampling_heuristic" value="simple_random"/>
<!-- Variable used to specify location of the sample DUMP -->
<!-- This file will be used in the construction of the decision tree -->
<variable name="sample_dump_file"
value="C:\\QueryMesh\\training_sets\\TRAINING_SET_STREAM0.txt"/>
<!-- Decision tree header contains the names of the attributes to be used -->
<!-- in the decision tree. You can just copy these attributes directly from -->
<!-- the stream specification (but you will need to add an additional parameter to
ignore/to use)-->
<!-- tuple attribute in the decision tree)-->
<decisionTreeHeader>
<attribute name="counter" type="int" use="false" is_target="false"/>
<attribute name="value" type="string" use="true" is_target="true"/>
</decisionTreeHeader>
<!-- There are 2 types of histograms: (a) number-based, (b) string-based -->
<!-- parameters for the number-based histogram: # of buckets, min and max values -->
<!-- parameters for the string-based histogram: nothing, each new string will be
assigned its own bucket -->
<histograms>
<hist id="1" attr_idx="1" type="string" num_buckets="-1" min_val="-1" max_val="-1" />
</histograms>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
<parent id = "1"/>
</parents>
<children>
<child type="stream" name="Stream0" queueId="0"/>
</children>
</operator>
</queryplan>
```

Note: Please make sure that you have an existing directory that matches the highlighted part.

For training tuple sampling, the sampling operator expects the schemas of the streams to be specified, as shown in gray above. In addition to the attribute names, “use” and “is_target” attributes must be specified. These parameters are specific to the decision tree classifier. “use” indicates whether to use that attribute in the decision tree algorithm and “is_target” indicates if it is a target attribute i.e., the leaf node attribute value.

Example of SystemConfigCollectTrainingSetStream.xml

```
<config>
<system>
<property name="StatisticsGatherer" value="off"/>
<property name="AVERAGE_WEIGHT" value=".875"/>
<property name="EXECUTION_CONTROLLER"
value="edu.wpi.cs.dsrg.xmldb.xat.component.executioncontroller.DistributedExecutionController"/>
<property name="DATA_MODEL"
value="edu.wpi.cs.dsrg.xmldb.xat.common.dag.XATMemoryQueueImp"/>
</system>
<distribution>
<property name="DISTRIBUTION_PATTERN"
value="edu.wpi.cs.dsrg.xmldb.xat.component.distribution.pattern.GroupingDistribution"/>
<property name="WORKLOAD_COST_MODEL"
value="edu.wpi.cs.dsrg.xmldb.xat.component.distribution.costmodel.NumTuplesInQueue"/>
<property name="REDISTRIBUTION_POLICY"
value="edu.wpi.cs.dsrg.xmldb.xat.component.distribution.redistribution.Balance"/>
<property name="REDISBRIBUTION_TIME" value="15000"/>
<property name="DISTRIBUTION_DELAY" value="10000"/>
<property name="STATE_SIZE_THRESHOLD" value="-1"/>
<property name="REDISTRIBUTION_PERCENT" value="110"/>
<property name="REDISTRIBUTION_SCOPE" value="global"/>
</distribution>
<experiment>
<property name="EXECUTION_DURATION" value="60000"/>
<property name="PRINT_OUT_META_INFORMATION" value="true"/>
<property name="STREAM_CONFIG_FILE_NAME"
value="resources\QueryMesh\Example\QueryMeshStreams.xml"/>
<property name="STREAM_DURATION" value="60000"/>
<treeProperties/>
</experiment>
<machines>
<machine>
<property name="NAME" value="Machine 1"/>
<property name="HOST_ADDRESS" value="localhost"/>
<property name="PORT" value="8001"/>
<property name="TUPLE_RECEIVER_PORT" value="9001"/>
<property name="CONNECTION_LISTENER_PORT" value="10001"/>
<property name="ADAPTIVE_HEURISTIC"
value="edu.wpi.cs.dsrg.xmldb.xat.component.scheduler.NeverRotateAdapter"/>
<property name="DEBUG" value="false"/>
<property name="UPDATE_OPERATOR_PROPERTY_FREQUENCY" value="1000"/>
<property name="UPDATE_TREE_PROPERTY_FREQUENCY" value="1000"/>
<property name="STATUS_CHECK_FREQUENCY" value="300000000"/>
<property name="STATS_TABLE_STATS" value="false"/>
<property name="GUI" value="off"/>
<!-- OPTIMIZATION_STRATEGY defines the migration strategy used during
execution. The possible value can be "off", "MS" stands for moving state and
"PT" stands for parallel track -->
<property name="MIGRATION_STRATEGY" value="off"/>
<property name="MIGRATION_INTERVAL" value="30000"/>

```

```

<scheduling>
<property name="WORKLOAD_RATIO" value="1"/>
<property name="WORKLOAD_THRESHOLD" value="50"/>
<preferences>
<preference statistic="TOTAL_TUPLES_IN_QUEUES" quantifier="min" weight="1"/>
<!--<preference statistic="OUTPUT_RATE" quantifier="max" weight=".5"/>-->
</preferences>
<algorithms>
<property name="RoundRobin"
value="edu.wpi.cs.dsrg.xmlldb.xat.component.scheduler.RoundRobinScheduler"/>
</algorithms>
</scheduling>
</machine>
</machines>
<QueryPlans>
<QueryPlan>
<property name="QUERY_ID" value="1"/>
<property name="edu.wpi.cs.dsrg.xmlldb.xat.component.queryplangenerator"
value="edu.wpi.cs.dsrg.xmlldb.xat.component.queryplangenerator.DistributedFrom
XMLFileQueryPlanGenerator"/>
<property name="FILE_NAME"
value="resources\QueryMesh\Example\queryplanCollectTrainingSetStream0.xml"/>
<property name="QUERY_FILE_NAME"
value="resources\QueryMesh\Example\query.txt"/>
</QueryPlan>
</QueryPlans>
<Applications>
<Application>
<property name="HOST_ADDRESS" value="127.0.0.1"/>
<property name="PORT" value="16001"/>
<property name="CONNECTS_TO" value="1"/>
</Application>
</Applications>
<outputFormat>
<property name="FORMAT" value="csv"/>
<property name="FILE_NAME" value="outputQueryMesh_SinglePlan.csv"/>
<property name="PRINT_EMPTY_ROW" value="false"/>
<property name="ALWAYS_PRINT_HEADERS" value="false"/>
<property name="OVERALL_FILENAME" value="outputQueryMesh_Overall.csv"/>
<property name="ALWAYS_PRINT_OVERALL" value="false"/>
<property name="FREQUENCY" value="5000"/>
<outputColumnNames>
<property name="TIME_TOOK_TO_RUN_TOTAL" value=""/>
<property name="USED_MEMORY" value=""/>
<property name="AVERAGE_TUPLE_DELAY" value=""/>
<property name="AVERAGE_TUPLE_PROCESSING_TIME" value=""/>
<property name="NUMBER_OF_TIMES_OPERATORS_WERE_RUN" value=""/>
<property name="TOTAL_TUPLES_IN_QUEUES" value=""/>
<property name="SELECTIVITY" value=""/>
<property name="THROUGHPUT" value=""/>
<property name="OUTPUT_RATE" value=""/>
</outputColumnNames>
</outputFormat>
<!-- Some of the Statistics to Gather. It is important that the
everyTimeOperator properties are kept in tact (including order).
Altering the order or makeup could result in either.
1. a scheduler not working correctly

```

2. another property not being updated correctly

The value wont be used, but it keeps it consistent with the rest of the document. If a property appears in the printout (above), then it should be listed here.

The 2nd group of properties is optional metrics.

Not all properties can be specified here because some rely on outside information.

The everyTimeOperator element contains properties that are updated after every time an operator runs. The periodicOperator element lists all properties that can be updated at regular intervals (defined as UPDATE_PROPERTY_FREQUENCY property). -->

```
<statisticsToGather>
<operatorStatistics>
<everyTimeTree>
<!-- I cant think of any tree properties that would need to be updated every
time so this isnt supported -->
</everyTimeTree>
<periodicTree>
<property name="THROUGHPUT" value=""/>
<property name="OUTPUT_RATE" value=""/>
<property name="NUMBER_OF_TIMES_OPERATORS_WERE_RUN" value=""/>
<property name="TOTAL_TUPLES_IN_QUEUES" value=""/>
<property name="AVERAGE_TUPLE_DELAY" value=""/>
<property name="TOTAL_TUPLES_IN_STATES" value=""/>
</periodicTree>
<everyTimeOperator>
<!-- These properties are updated every time an operator runs -->
<property name = "NUMBER_OF_TUPLES_OUTPUTTED_TOTAL" value = ""/>
<property name="NUMBER_OF_TIMES_RUN" value=""/>
<property name="NUMBER_OF_TUPLES_IN_INPUT_QUEUES" value=""/>
<property name="NUMBER_OF_TUPLES_DEQUEUED_TOTAL" value=""/>
<property name="NUMBER_OF_TUPLES_DEQUEUED" value=""/>
<property name="TIME_TOOK_TO_RUN_TOTAL" value=""/>
<property name="NUMBER_OF_TUPLES_IN_OUTPUT_QUEUES" value=""/>
<property name="SELECTIVITY" value=""/>
</everyTimeOperator>
<periodicOperator>
<!-- These properties will be updated at regular intervals -->
<property name="AVERAGE_TUPLE_PROCESSING_TIME" value=""/>
<property name="GREEDY_PRIORITY" value=""/>
<property name="AVERAGE_OUTPUT_RATE" value=""/>
</periodicOperator>
<everyTimeSystem>
<property name="USED_MEMORY" value=""/>
<property name="FREE_MEMORY" value=""/>
<property name="TOTAL_MEMORY" value=""/>
<property name="USED_MEMORY_PERCENTAGE" value=""/>
</everyTimeSystem>
</operatorStatistics>
</statisticsToGather>
</config>
```

Note: please make sure that all highlighted parts have their matching files on your local drive.

Example of StreamsConfig.xml

This is the Stream Generator Configuration File

This file gives the following information:

- The name of the Data File.
- Schema and the format of the file: To enable the parser to parse the file and generate the tuples.
- Inter-arrival distribution: Information about what distribution to use or what attribute to use in the schema as a time stamp of the file.

```
<streams>
<!--Note the stream name has to be unique for each stream-->
<stream name="Stream0">
<files>
<file name="resources\QueryMesh\Example\Stream0.txt"/>
</files>
<!--Gives the format of the file-->
<delimiter attribute="|" record="\n"/>
<schema>
<table name="Stream0"/>
<attribute name="counter" type="int"/>
<attribute name="value" type="string"/>
</schema>
<inter_arrival>
<distribution value="poisson" seed="0">
<interval start_time="0" mean="200"/>
</distribution>
</inter_arrival>
</stream>
<stream name="Stream1">
<files>
<file name="resources\QueryMesh\Example\Stream1.txt"/>
</files>
<!--Gives the format of the file-->
<delimiter attribute="|" record="\n"/>
<schema>
<table name="Stream1"/>
<attribute name="counter" type="int"/>
<attribute name="value" type="string"/>
</schema>
<inter_arrival>
<distribution value="poisson" seed="0">
<interval start_time="0" mean="200"/>
</distribution>
</inter_arrival>
</stream>
<total_time value="-1"/>
</streams>
```

Example of QueryMeshStreams.xml

This file gives the following information:

- For each server the ip address and the port number.
- For each stream which server it is coming from.

```
<client_config>
<servers>
<server name="HeadServer" ip_address="localhost" port="15000"/>
</servers>
<streams>
<stream name="Stream0" server="HeadServer"/>
<stream name="Stream1" server="HeadServer"/>
<stream name="Stream2" server="HeadServer"/>
<stream name="Stream3" server="HeadServer"/>
<stream name="Stream4" server="HeadServer"/>
</streams>
</client_config>
```

Example of optimizer_input.xml

```
<?xml version="1.0"?>
<QueryMeshOptimizerInput>
  <QueryPlan>
    <Operators>
      <Operator id="1" type="join">
        <OpInputs>
          <OpInput id="0" attr_idx="0"></OpInput>
          <OpInput id="1" attr_idx="0"></OpInput>
        </OpInputs>
      </Operator>
      <Operator id="2" type="join">
        <OpInputs>
          <OpInput id="1" attr_idx="0"></OpInput>
          <OpInput id="2" attr_idx="0"></OpInput>
        </OpInputs>
      </Operator>
      <Operator id="3" type="join">
        <OpInputs>
          <OpInput id="2" attr_idx="0"></OpInput>
          <OpInput id="3" attr_idx="0"></OpInput>
        </OpInputs>
      </Operator>
      <Operator id="4" type="join">
        <OpInputs>
          <OpInput id="3" attr_idx="0"></OpInput>
          <OpInput id="4" attr_idx="0"></OpInput>
        </OpInputs>
      </Operator>
    </Operators>
  </QueryPlan>
  <Inputs>
    <Input id="0" name="Stream0"
      statistics_file="C:\\QueryMesh\\statistics_samples\\SAMPLE_DS4_STREAM0.txt"
      training_tuples_file="C:\\QueryMesh\\training_sets\\TRAINING_SET_DS4_STREAM0.txt">
      <Schema>
        <attribute name="val0" type="int"/>
        <attribute name="val1" type="int"/>
        <attribute name="val2" type="int"/>
        <attribute name="val3" type="int"/>
      </Schema>
    </Input>
    <Input id="1" name="Stream1"
      statistics_file="C:\\QueryMesh\\statistics_samples\\SAMPLE_DS4_STREAM1.txt"
      training_tuples_file="C:\\QueryMesh\\training_sets\\TRAINING_SET_DS4_STREAM1.txt">
      <Schema>
        <attribute name="val0" type="int"/>
        <attribute name="val1" type="int"/>
        <attribute name="val2" type="int"/>
        <attribute name="val3" type="int"/>
      </Schema>
    </Input>
    <Input id="2" name="Stream2"
      statistics_file="C:\\QueryMesh\\statistics_samples\\SAMPLE_DS4_STREAM2.txt"
      training_tuples_file="C:\\QueryMesh\\training_sets\\TRAINING_SET_DS4_STREAM2.txt">
      <Schema>
        <attribute name="val0" type="int"/>
        <attribute name="val1" type="int"/>
        <attribute name="val2" type="int"/>
        <attribute name="val3" type="int"/>
      </Schema>
    </Input>
  </Inputs>
</QueryMeshOptimizerInput>
```

```
<Input id="3" name="Stream3"
statistics_file="C:\\QueryMesh\\statistics_samples\\SAMPLE_DS4_STREAM3.txt"
training_tuples_file="C:\\QueryMesh\\training_sets\\TRAINING_SET_DS4_STREAM3.txt">
<Schema>
<attribute name="val0" type="int"/>
<attribute name="val1" type="int"/>
<attribute name="val2" type="int"/>
<attribute name="val3" type="int"/>
</Schema>
</Input>
<Input id="4" name="Stream4"
statistics_file="C:\\QueryMesh\\statistics_samples\\SAMPLE_DS4_STREAM4.txt"
training_tuples_file="C:\\QueryMesh\\training_sets\\TRAINING_SET_DS4_STREAM4.txt">
<Schema>
<attribute name="val0" type="int"/>
<attribute name="val1" type="int"/>
<attribute name="val2" type="int"/>
<attribute name="val3" type="int"/>
</Schema>
</Input>
</Inputs>
</QueryPlan>
</QueryMeshOptimizerInput>
```

Example of SystemConfigQueryMeshPlan.xml

```
<config>
<system>
<property name="StatisticsGatherer" value="on"/>
<property name="AVERAGE_WEIGHT" value=".875"/>
<property name="EXECUTION_CONTROLLER"
value="edu.wpi.cs.dsrg.xmldb.xat.component.executioncontroller.QueryMeshExecutionContr
oller"/>
<property name="DATA_MODEL"
value="edu.wpi.cs.dsrg.xmldb.xat.common.dag.XATMemoryQueueImp"/>
</system>
<distribution>
<property name="DISTRIBUTION_PATTERN"
value="edu.wpi.cs.dsrg.xmldb.xat.component.distribution.pattern.GroupingDistribution"/>
<property name="WORKLOAD_COST_MODEL"
value="edu.wpi.cs.dsrg.xmldb.xat.component.distribution.costmodel.NumTuplesInQueue"/>
<property name="REDISTRIBUTION_POLICY"
value="edu.wpi.cs.dsrg.xmldb.xat.component.distribution.redistribution.Balance"/>
<property name="REDISBRIBUTION_TIME" value="15000"/>
<property name="DISTRIBUTION_DELAY" value="10000"/>
<property name="STATE_SIZE_THRESHOLD" value="-1"/>
<property name="REDISTRIBUTION_PERCENT" value="110"/>
<property name="REDISTRIBUTION_SCOPE" value="global"/>
</distribution>
<experiment>
<property name="EXECUTION_DURATION" value="6000000"/>
<property name="PRINT_OUT_META_INFORMATION" value="true"/>
<property name="STREAM_CONFIG_FILE_NAME"
value="resources\QueryMesh\Example\QueryMeshStreams.xml"/>
<property name="STREAM_DURATION" value="6000000"/>
<treeProperties/>
</experiment>
<machines>
<machine>
<property name="NAME" value="Machine 1"/>
<property name="HOST_ADDRESS" value="localhost"/>
<property name="PORT" value="8001"/>
<property name="TUPLE_RECEIVER_PORT" value="9001"/>
<property name="CONNECTION_LISTENER_PORT" value="10001"/>
<property name="ADAPTIVE_HEURISTIC"
value="edu.wpi.cs.dsrg.xmldb.xat.component.scheduler.NeverRotateAdapter"/>
<property name="DEBUG" value="false"/>
<property name="UPDATE_OPERATOR_PROPERTY_FREQUENCY" value="1000"/>
<property name="UPDATE_TREE_PROPERTY_FREQUENCY" value="1000"/>
<property name="STATUS_CHECK_FREQUENCY" value="300000000"/>
<property name="STATS_TABLE_STATS" value="false"/>
<property name="GUI" value="off"/>
<!-- OPTIMIZATION_STRATEGY defines the migration strategy used during execution. The
possible value can be "off", "MS" stands for moving state and "PT" stands for parallel
track -->
<property name="MIGRATION_STRATEGY" value="off"/>
<property name="MIGRATION_INTERVAL" value="30000"/>
<scheduling>
<property name="WORKLOAD_RATIO" value="1"/>
<property name="WORKLOAD_THRESHOLD" value="50"/>
<preferences>
<preference statistic="TOTAL_TUPLES_IN_QUEUES" quantifier="min" weight="1"/>
<!--<preference statistic="OUTPUT_RATE" quantifier="max" weight=".5"/>-->
</preferences>
<algorithms>
```

```

<property name="RoundRobin"
value="edu.wpi.cs.dsrg.xmldb.xat.component.scheduler.RoundRobinScheduler"/>
</algorithms>
</scheduling>
</machine>
</machines>
<QueryPlans>
<QueryPlan>
<property name="QUERY_ID" value="1"/>
<property name="edu.wpi.cs.dsrg.xmldb.xat.component.queryplangenerator"
value="edu.wpi.cs.dsrg.xmldb.xat.component.queryplangenerator.DistributedFromXMLFileQueryPlanGenerator"/>
<property name="FILE_NAME" value="resources\QueryMesh\Example\qm_query_plan.xml"/>
<property name="QUERY_FILE_NAME" value="resources\QueryMesh\Example\query.txt"/>
</QueryPlan>
</QueryPlans>
<Applications>
<Application>
<property name="HOST_ADDRESS" value="127.0.0.1"/>
<property name="PORT" value="16001"/>
<property name="CONNECTS_TO" value="1"/>
</Application>
</Applications>
<outputFormat>
<property name="FORMAT" value="csv"/>
<property name="FILE_NAME" value="outputQueryMesh_QueryMeshPlan.csv"/>
<property name="PRINT_EMPTY_ROW" value="false"/>
<property name="ALWAYS_PRINT_HEADERS" value="false"/>
<property name="OVERALL_FILENAME" value="outputQueryMesh_QueryMeshPlanOverall.csv"/>
<property name="ALWAYS_PRINT_OVERALL" value="false"/>
<property name="FREQUENCY" value="5000"/>
<outputColumnNames>
<property name="TIME_TOOK_TO_RUN_TOTAL" value=""/>
<property name="USED_MEMORY" value=""/>
<property name="AVERAGE_TUPLE_DELAY" value=""/>
<property name="AVERAGE_TUPLE_PROCESSING_TIME" value=""/>
<property name="NUMBER_OF_TIMES_OPERATORS_WERE_RUN" value=""/>
<property name="TOTAL_TUPLES_IN_QUEUES" value=""/>
<property name="SELECTIVITY" value=""/>
<property name="THROUGHPUT" value=""/>
<property name="OUTPUT_RATE" value=""/>
</outputColumnNames>
</outputFormat>
<!-- Some of the Statistics to Gather. It is important that the everyTimeOperator
properties are kept in tact (including order). Altering the order or makeup could
result in either.
1. a scheduler not working correctly
2. another property not being updated correctly
The value wont be used, but it keeps it consistent with the rest of the document
If a property appears in the printout (above), then it should be listed here.
The 2nd group of properties is optional metrics.
Not all properties can be specified here because some rely on outside information.
The everyTimeOperator element contains properties that are updated after every time an
operator runs. The periodicOperator element lists all properties that can be updated
at regular intervals (defined as UPDATE_PROPERTY_FREQUENCY property). -->
<statisticsToGather>
<operatorStatistics>
<everyTimeTree>
<!-- I cant think of any tree properties that would need to be updated every time so
this isnt supported -->
</everyTimeTree>
<periodicTree>
<property name="THROUGHPUT" value=""/>

```

```
<property name="OUTPUT_RATE" value=""/>
<property name="NUMBER_OF_TIMES_OPERATORS_WERE_RUN" value=""/>
<property name="TOTAL_TUPLES_IN_QUEUES" value=""/>
<property name="AVERAGE_TUPLE_DELAY" value=""/>
<property name="TOTAL_TUPLES_IN_STATES" value=""/>
</periodicTree>
<everyTimeOperator>
<!-- These properties are updated every time an operator runs -->
<property name = "NUMBER_OF_TUPLES_OUTPUTTED_TOTAL" value = ""/>
<property name="NUMBER_OF_TIMES_RUN" value=""/>
<property name="NUMBER_OF_TUPLES_IN_INPUT_QUEUES" value=""/>
<property name="NUMBER_OF_TUPLES_DEQUEUED_TOTAL" value=""/>
<property name="NUMBER_OF_TUPLES_DEQUEUED" value=""/>
<property name="TIME_TOOK_TO_RUN_TOTAL" value=""/>
<property name="NUMBER_OF_TUPLES_IN_OUTPUT_QUEUES" value=""/>
<property name="SELECTIVITY" value=""/>
</everyTimeOperator>
<periodicOperator>
<!-- These properties will be updated at regular intervals -->
<property name="AVERAGE_TUPLE_PROCESSING_TIME" value=""/>
<property name="GREEDY_PRIORITY" value=""/>
<property name="AVERAGE_OUTPUT_RATE" value=""/>
</periodicOperator>
<everyTimeSystem>
<property name="USED_MEMORY" value=""/>
<property name="FREE_MEMORY" value=""/>
<property name="TOTAL_MEMORY" value=""/>
<property name="USED_MEMORY_PERCENTAGE" value=""/>
</everyTimeSystem>
</operatorStatistics>
</statisticsToGather>
</config>
```

Example of auto-generated query plan a.k.a qm_query_plan.xml

```
<!-- Autogenerated QMesh Query Plan -->
<queryplan>
<operator root = "true" id = "9" className =
"edu.wpi.cs.dsrg.xmlldb.xat.common.querymesh.execution.DoNothingOperatorImp"
numberOfOutputQueue = "1">
<classVariables>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
</parents>
<children>
<child type="operator" id = "1" />
<child type="operator" id = "2" />
<child type="operator" id = "3" />
<child type="operator" id = "4" />
<child type="operator" id = "5" />
<child type="operator" id = "6" />
<child type="operator" id = "7" />
<child type="operator" id = "8" />
</children>
</operator>
<operator root = "false" id = "1" className =
"edu.wpi.cs.dsrg.STeM.STeMJoinProbeOperatorImp">
<classVariables>
<variable name="QMeshOperatorID" value="1" />
<variable name="IsEddyOp" value="false" />
<variable name="StreamID" value="0" />
<!-- probe data -->
<variable name="ProbeSTREAMId" value="0" />
<variable name="NumTupleIndex" value="0"/>
<!-- stored data -->
<variable name="STeMSTREAMId" value="1" />
<variable name="NumSTeMIndex" value="0"/>
<expressions>
</expressions>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
<parent id = "9"/>
</parents>
<children>
<child type="operator" id = "0" queueId="0"/>
</children>
</operator>
<operator root = "false" id = "2" className =
"edu.wpi.cs.dsrg.STeM.STeMJoinProbeOperatorImp">
<classVariables>
<variable name="QMeshOperatorID" value="2" />
<variable name="IsEddyOp" value="false" />
<variable name="StreamID" value="1" />
<!-- probe data -->
<variable name="ProbeSTREAMId" value="1" />
<variable name="NumTupleIndex" value="0"/>
<!-- stored data -->
<variable name="STeMSTREAMId" value="0" />
<variable name="NumSTeMIndex" value="0"/>
```

```

<expressions>
</expressions>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
<parent id = "9"/>
</parents>
<children>
<child type="operator" id = "0" queueId="1"/>
</children>
</operator>
<operator root = "false" id = "3" className =
"edu.wpi.cs.dsrg.STeM.STeMJoinProbeOperatorImp">
<classVariables>
<variable name="QMeshOperatorID" value="3" />
<variable name="IsEddyOp" value="false" />
<variable name="StreamID" value="1" />
<!-- probe data -->
<variable name="ProbeSTREAMId" value="1" />
<variable name="NumTupleIndex" value="0"/>
<!-- stored data -->
<variable name="STeMSTREAMId" value="2" />
<variable name="NumSTeMIndex" value="0"/>
<expressions>
</expressions>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
<parent id = "9"/>
</parents>
<children>
<child type="operator" id = "0" queueId="2"/>
</children>
</operator>
<operator root = "false" id = "4" className =
"edu.wpi.cs.dsrg.STeM.STeMJoinProbeOperatorImp">
<classVariables>
<variable name="QMeshOperatorID" value="4" />
<variable name="IsEddyOp" value="false" />
<variable name="StreamID" value="2" />
<!-- probe data -->
<variable name="ProbeSTREAMId" value="2" />
<variable name="NumTupleIndex" value="0"/>
<!-- stored data -->
<variable name="STeMSTREAMId" value="1" />
<variable name="NumSTeMIndex" value="0"/>
<expressions>
</expressions>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
<parent id = "9"/>
</parents>
<children>
<child type="operator" id = "0" queueId="3"/>
</children>
</operator>

```

```

<operator root = "false" id = "5" className =
"edu.wpi.cs.dsrg.STeM.STeMJoinProbeOperatorImp">
<classVariables>
<variable name="QMeshOperatorID" value="5" />
<variable name="IsEddyOp" value="false" />
<variable name="StreamID" value="2" />
<!-- probe data -->
<variable name="ProbeSTREAMId" value="2" />
<variable name="NumTupleIndex" value="0"/>
<!-- stored data -->
<variable name="STeMSTREAMId" value="3" />
<variable name="NumSTeMIndex" value="0"/>
<expressions>
</expressions>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
<parent id = "9"/>
</parents>
<children>
<child type="operator" id = "0" queueId="4"/>
</children>
</operator>
<operator root = "false" id = "6" className =
"edu.wpi.cs.dsrg.STeM.STeMJoinProbeOperatorImp">
<classVariables>
<variable name="QMeshOperatorID" value="6" />
<variable name="IsEddyOp" value="false" />
<variable name="StreamID" value="3" />
<!-- probe data -->
<variable name="ProbeSTREAMId" value="3" />
<variable name="NumTupleIndex" value="0"/>
<!-- stored data -->
<variable name="STeMSTREAMId" value="2" />
<variable name="NumSTeMIndex" value="0"/>
<expressions>
</expressions>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
<parent id = "9"/>
</parents>
<children>
<child type="operator" id = "0" queueId="5"/>
</children>
</operator>
<operator root = "false" id = "7" className =
"edu.wpi.cs.dsrg.STeM.STeMJoinProbeOperatorImp">
<classVariables>
<variable name="QMeshOperatorID" value="7" />
<variable name="IsEddyOp" value="false" />
<variable name="StreamID" value="3" />
<!-- probe data -->
<variable name="ProbeSTREAMId" value="3" />
<variable name="NumTupleIndex" value="0"/>
<!-- stored data -->
<variable name="STeMSTREAMId" value="4" />
<variable name="NumSTeMIndex" value="0"/>
<expressions>

```

```

</expressions>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
<parent id = "9"/>
</parents>
<children>
<child type="operator" id = "0" queueId="6"/>
</children>
</operator>
<operator root = "false" id = "8" className =
"edu.wpi.cs.dsrg.SteM.SteMJoinProbeOperatorImp">
<classVariables>
<variable name="QMeshOperatorID" value="8" />
<variable name="IsEddyOp" value="false" />
<variable name="StreamID" value="4" />
<!-- probe data -->
<variable name="ProbeSTREAMId" value="4" />
<variable name="NumTupleIndex" value="0"/>
<!-- stored data -->
<variable name="SteMSTREAMId" value="3" />
<variable name="NumSteMIndex" value="0"/>
<expressions>
</expressions>
</classVariables>
<properties>
</properties>
<schema/>
<parents>
<parent id = "9"/>
</parents>
<children>
<child type="operator" id = "0" queueId="7"/>
</children>
</operator>
<operator root="false" id="0"
className="edu.wpi.cs.dsrg.xmldb.xat.common.querymesh.execution.OnlineClassifierOperat
orImp" numberOfOutputQueue = "8">
<classVariables>
<variable name="Num_Streams" value="5" />
<variable name="Num_Operators" value="8" />
<variable name="Num_SendOff" value="100" />
<variable name="TupleCountThreshold" value="10000" />
<!-- variables needed for each stram -->
<variable name = "Stream0" QueueId = "0" window_type="CountBased" window_size="10000"
/>
<variable name = "Stream1" QueueId = "1" window_type="CountBased" window_size="10000"
/>
<variable name = "Stream2" QueueId = "2" window_type="CountBased" window_size="10000"
/>
<variable name = "Stream3" QueueId = "3" window_type="CountBased" window_size="10000"
/>
<variable name = "Stream4" QueueId = "4" window_type="CountBased" window_size="10000"
/>
<globalDecisionTree>
<localQM id="0" stream_id="4">
<localDecisionTree id="0" stream_id="4" is_empty="true" />
<allRoutes>
<route id="1" is_default="true" path="8|6|4|2" logical_plan="[3, 4],[2, 3],[1, 2],[0,
1]" />
</allRoutes>

```

```

</localQM>
<localQM id="1" stream_id="3">
<localDecisionTree id="1" stream_id="3" is_empty="false" >
<DTnode id="2" type="leaf" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation="" test_value="" route_id="1">
<parents>
<parent node_id="1"/>
</parents>
<children/>
</DTnode>
<DTnode id="1" type="internal" attr_idx="0" attr_name="val0" is_bucket_value="false"
bucket_range_start="" bucket_range_end="" operation="EQ" test_value="333" route_id
="">
<parents>
<parent node_id="0"/>
</parents>
<children>
<child node_id="2"/>
</children>
</DTnode>
<DTnode id="4" type="leaf" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation="" test_value="" route_id="1">
<parents>
<parent node_id="3"/>
</parents>
<children/>
</DTnode>
<DTnode id="3" type="internal" attr_idx="0" attr_name="val0" is_bucket_value="false"
bucket_range_start="" bucket_range_end="" operation="EQ" test_value="888" route_id
="">
<parents>
<parent node_id="0"/>
</parents>
<children>
<child node_id="4"/>
</children>
</DTnode>
<DTnode id="6" type="leaf" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation="" test_value="" route_id="1">
<parents>
<parent node_id="5"/>
</parents>
<children/>
</DTnode>
<DTnode id="5" type="internal" attr_idx="0" attr_name="val0" is_bucket_value="false"
bucket_range_start="" bucket_range_end="" operation="EQ" test_value="999" route_id
="">
<parents>
<parent node_id="0"/>
</parents>
<children>
<child node_id="6"/>
</children>
</DTnode>
<DTnode id="0" type="root" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation="" test_value="" route_id="">
<parents/>
<children>
<child node_id="1"/>
<child node_id="3"/>
<child node_id="5"/>
</children>
</DTnode>

```

```

</localDecisionTree>
<allRoutes>
<route id="4" is_default="true" path="6|7|4|2" logical_plan="[2, 3],[3, 4],[1, 2],[0, 1]" />
<route id="1" is_default="false" path="7|6|4|2" logical_plan="[3, 4],[2, 3],[1, 2],[0, 1]" />
</allRoutes>
</localQM>
<localQM id="2" stream_id="2">
<localDecisionTree id="2" stream_id="2" is_empty="false" >
<DTnode id="2" type="leaf" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation ="" test_value ="" route_id = "1">
<parents>
<parent node_id="1"/>
</parents>
<children/>
</DTnode>
<DTnode id="1" type="internal" attr_idx="0" attr_name="val0" is_bucket_value="false"
bucket_range_start="" bucket_range_end="" operation = "EQ" test_value = "222" route_id
= "">
<parents>
<parent node_id="0"/>
</parents>
<children>
<child node_id="2"/>
</children>
</DTnode>
<DTnode id="4" type="leaf" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation ="" test_value ="" route_id = "2">
<parents>
<parent node_id="3"/>
</parents>
<children/>
</DTnode>
<DTnode id="3" type="internal" attr_idx="0" attr_name="val0" is_bucket_value="false"
bucket_range_start="" bucket_range_end="" operation = "EQ" test_value = "999" route_id
= "">
<parents>
<parent node_id="0"/>
</parents>
<children>
<child node_id="4"/>
</children>
</DTnode>
<DTnode id="6" type="leaf" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation ="" test_value ="" route_id = "1">
<parents>
<parent node_id="5"/>
</parents>
<children/>
</DTnode>
<DTnode id="5" type="internal" attr_idx="0" attr_name="val0" is_bucket_value="false"
bucket_range_start="" bucket_range_end="" operation = "EQ" test_value = "888" route_id
= "">
<parents>
<parent node_id="0"/>
</parents>
<children>
<child node_id="6"/>
</children>
</DTnode>
<DTnode id="0" type="root" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation ="" test_value ="" route_id = "">

```

```

<parents/>
<children>
<child node_id="1"/>
<child node_id="3"/>
<child node_id="5"/>
</children>
</DTnode>
</localDecisionTree>
<allRoutes>
<route id="1" is_default="false" path="5|7|4|2" logical_plan="[2, 3],[3, 4],[1, 2],[0, 1]" />
<route id="2" is_default="true" path="4|5|7|2" logical_plan="[1, 2],[2, 3],[3, 4],[0, 1]" />
</allRoutes>
</localQM>
<localQM id="3" stream_id="1">
<localDecisionTree id="3" stream_id="1" is_empty="false" >
<DTnode id="2" type="leaf" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation="" test_value="" route_id="1">
<parents>
<parent node_id="1"/>
</parents>
<children/>
</DTnode>
<DTnode id="1" type="internal" attr_idx="0" attr_name="val0" is_bucket_value="false"
bucket_range_start="" bucket_range_end="" operation="EQ" test_value="111" route_id="">
<parents>
<parent node_id="0"/>
</parents>
<children>
<child node_id="2"/>
</children>
</DTnode>
<DTnode id="4" type="leaf" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation="" test_value="" route_id="3">
<parents>
<parent node_id="3"/>
</parents>
<children/>
</DTnode>
<DTnode id="3" type="internal" attr_idx="0" attr_name="val0" is_bucket_value="false"
bucket_range_start="" bucket_range_end="" operation="EQ" test_value="888" route_id="">
<parents>
<parent node_id="0"/>
</parents>
<children>
<child node_id="4"/>
</children>
</DTnode>
<DTnode id="6" type="leaf" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation="" test_value="" route_id="1">
<parents>
<parent node_id="5"/>
</parents>
<children/>
</DTnode>
<DTnode id="5" type="internal" attr_idx="0" attr_name="val0" is_bucket_value="false"
bucket_range_start="" bucket_range_end="" operation="EQ" test_value="999" route_id="">
<parents>
<parent node_id="0"/>

```

```

</parents>
<children>
<child node_id="6"/>
</children>
</DTnode>
<DTnode id="0" type="root" attr_idx="" attr_name="" is_bucket_value=""
bucket_range_start="" bucket_range_end="" operation="" test_value="" route_id="">
<parents/>
<children>
<child node_id="1"/>
<child node_id="3"/>
<child node_id="5"/>
</children>
</DTnode>
</localDecisionTree>
<allRoutes>
<route id="3" is_default="false" path="3|5|2|7" logical_plan="[1, 2],[2, 3],[0, 1],[3,
4]" />
<route id="1" is_default="true" path="3|5|7|2" logical_plan="[1, 2],[2, 3],[3, 4],[0,
1]" />
</allRoutes>
</localQM>
<localQM id="4" stream_id="0">
<localDecisionTree id="4" stream_id="0" is_empty="true" />
<allRoutes>
<route id="1" is_default="true" path="1|3|5|7" logical_plan="[0, 1],[1, 2],[2, 3],[3,
4]" />
</allRoutes>
</localQM>
</globalDecisionTree>
</classVariables>
<properties/>
<schema/>
<parents>
<parent id = "8" queueId = "7"/>
<parent id = "7" queueId = "6"/>
<parent id = "6" queueId = "5"/>
<parent id = "5" queueId = "4"/>
<parent id = "4" queueId = "3"/>
<parent id = "3" queueId = "2"/>
<parent id = "2" queueId = "1"/>
<parent id = "1" queueId = "0"/>
</parents>
<children>
<child type="stream" id="0" name = "Stream0" />
<child type="stream" id="1" name = "Stream1" />
<child type="stream" id="2" name = "Stream2" />
<child type="stream" id="3" name = "Stream3" />
<child type="stream" id="4" name = "Stream4" />
</children>
</operator>
</queryplan>

```