BSMA Performance Testing Tool Manual

• Overview

BSMA, benchmark for social media analysis, will be used as the benchmark in WISE 2012 Challenge.

The goal of the BSMA project is to develop a framework and common set of workloads for evaluating the performance of different systems dealing with social media analysis.

In the case of WISE 2012 Challenge Performance Track, the base of workloads is 19 typical queries, semantics of which can be found in the specification of queries. Attendees are expected to build a data analysis system covering those 19 queries and add code to the interface layer in BSMA (BSMA/db/idb/…/DBClient.java). After loading experimental data and completing the interface layer, query results and performance (latency, throughput) can be obtained via BSMA.

• Getting Started

BSMA is a framework for benchmarking systems. Particularly, this version is developed for WISE 2012 Challenge, expected solutions of which are dealing with social media analysis. By itself, it is not that useful; only when you add code to interface with a data analysis system is it useful**.**

1. Obtain BSMA source

BSMA.zip http://d.yun.io/1JNFWn

2. Build the package

BSMA requires Java 1.5 or newer to build.

```
cd ~/BSMA
ant
ant dbcompile-idb
```

3. Try a workload in debug mode

```
java -cp .:build/bsma.jar:db/idb/lib/* edu.ecnu.imc.bsma.Client -P
workloads/workloadquery9 –s –p debug=true > debugquery9.txt
```

sample output:

1. status report in the terminal

```
Loading workload...
Starting test.
 0 sec: 0 operations; [Query9 AverageLatency(ms)=0]
 0 sec: 20 operations; 487.8 current ops/sec; [Query9 AverageLatency(ms)=0]
```

2. debugquery9.txt

```
BSMA Client 0.1
Command line: -P workloads/workloadquery9 -p operationcount=20 -threads 2 -s -p debug=true
 current returncount: 10    current timespan: h

query results and latencies:
Debuging...
Debug BSMAQuery9:userID = 971032859720065 tag = Qian Yunhui datetime = 2010-12-28 09:00:00
[OVERALL], RunTime(ms), 4.0
[OVERALL], Throughput(ops/sec), 5000.0
[Query9], Operations, 20
[Query9], AverageLatency(ms), 0.0
[Query9], MinLatency(ms), 0
[Query9], MaxLatency(ms), 0
[Query9], 50thPercentileLatency(ms), 100
[Query9], 75thPercentileLatency(ms), 100
[Query9], 95thPercentileLatency(ms), 100
[Query9], 99thPercentileLatency(ms), 100
[Query9], [0ms-100ms), 20
[Query9], [100ms-200ms), 0
[Query9], [200ms-300ms), 0
....
```

• Completing interface layer

Assume that the data analysis system has been built and experimental data has been loaded properly. The next step is to complete the interface layer.

DBClient (BSMA/db/idb/…/DBClient.java) is designed to link your system to BSMA. Check the source code in DBClient.java, you'll get many "TODO" flags. Add code to the APIs remaining to be done following the comments there. You are required to complete the initialization of state for your system and the cleanup of system state. Besides, all the 19 queries in BSMA, namely BSMAQuery1, BSMAQuery2, …, BSMAQuery19, should be implemented for their semantics. Also, the result of each query is required to be a formatted string, which will be parsed to verify correctness. Queries' semantics and result patterns can be found in specification of queries. Take BSMAQuery9 for example, result strings should conform to the following format:

uid=A, datetime=YYYY-MM-DD HH:MM:SS, tag=?, USERID, USERID, USERID…

• Running BSMA

Below is the usage of BSMA Client (edu.ecnu.imc.bsma.Client)

Options:

-threads n	execute using n threads (default: 1) - can also be specified as the "threadcount" property using –p
-target n	attempt to do n operations per second (default: unlimited) - can also be specified as the "target" property using -p
-db dbname	specify the name of the DB to use (default: edu.ecnu.imc.bsma.db.DBClient) - can also be specified as the "db" property using –p
-P propertyfile	load properties from the given file
-p name=value	specify a property to be passed to the DB and workloads; multiple properties can be specified, and override any values in the propertyfile
-s	show status during run (default: no status)

Required properties:

workload	the name of the workload class to use (e.g. edu.ecnu.imc.bsma.workloads.CoreWorkload)
operationcount	the number of queries to execute
queryNpropertion(N =1,2,3...19)	the proportion of QueryN in all queries

• WISE 2012 Challenge Performance Track

To participant in WISE 2012 Challenge Performance Track, first of all, make sure your system can deal with all the 19 queries, namely BSMAQuery1, BSMAQuery2, …., BSMAQuery19.

Assume your system works and the interface layer is all right. Now you are expected to run each query under all its settings below:

Query#	Workload File (workloads/)	Property Settings			Output File Format
		threadcount	returncount	Timespan	
1	workloadquery1	α	β	/	$q1\_\alpha\_\beta.txt$
2	Workloadquery2	α	β	/	$q2\_\alpha\_\beta.txt$
3	Workloadquery3	α	β	/	$q3\_\alpha\_\beta.txt$
4	Workloadquery4	α	/	/	$q4\_\alpha.txt$
5	Workloadquery5	α	/	/	$q5\_\alpha\_\beta\_\gamma.txt$
6	Workloadquery6	α	β	γ	$q6\_\alpha\_\beta\_\gamma.txt$
7	Workloadquery7	α	β	γ	$q7\_\alpha\_\beta\_\gamma.txt$
8	Workloadquery8	α	/	/	$q8\_\alpha.txt$

9	Workloadquery9	α	β	γ	$q9\_\alpha\_\beta\_\gamma.txt$
10	workloadquery10	α	β	γ	$q10\_\alpha\_\beta\_\gamma.txt$
11	workloadquery11	α	β	γ	$q11\_\alpha\_\beta\_\gamma.txt$
12	workloadquery12	α	β	γ	$q12\_\alpha\_\beta\_\gamma.txt$
13	workloadquery13	α	β	γ	$q13\_\alpha\_\beta\_\gamma.txt$
14	workloadquery14	α	β	γ	$q14\_\alpha\_\beta\_\gamma.txt$
15	workloadquery15	α	β	γ	$q15\_\alpha\_\beta\_\gamma.txt$
16	workloadquery16	α	β	γ	$q16\_\alpha\_\beta\_\gamma.txt$
17	workloadquery17	α	β	γ	$q17\_\alpha\_\beta\_\gamma.txt$
18	workloadquery18	α	β	γ	$q18\_\alpha\_\beta\_\gamma.txt$
19	workloadquery19	α	β	γ	$q19\_\alpha\_\beta\_\gamma.txt$

The directory of workload files is BSMA/workloads/.

α is the number of threads used in the execution. It can be 1, 10, 50, 100…. Since we can't predict performance of your system and we have no idea what kind of server(s) your system is deployed on, we'll not limit the options of α. However, you are required to start with threadcount=1 and choose another 7 values including the maximum one. Knowing that the target is to achieve low response time and high throughput (operations/second), you should carefully choose a proper maximum value of threadcount for your system and server(s) to obtain a convincing performance.

β is the maximum number of returned records, which can be used in the SELECT TOP clause. Its options are 10, 50 and 100.

γ is the time span specified in the query. Its options are h (one hour), d (one day), w (one week) and y (one year).

As is shown in the above table, β and γ are not necessary in some queries. For example, Query4 need α only.

To assign values of α, β and γ, several means are available, as can be seen in running BSMA. Note that "-p name=value" can override any values in the property file. For the purpose of unification, we here specify the three properties via "-p name=value".

–p threadcount= α *-p returncount=* β *-p timespan=* γ

When threadcount, returncount and timespan are all specified properly, the name of output file is decided. Considering the case for query 19, if threadcount=10, returncount=100 and timespan=y, the output file should be q19_10_100_y.txt.

Another important specification is about operationcount, which means the number of operations(queries) should be performed in one case. Setting it to a small value may bring negative effect to the performance report while a rather big one means a

larger output file. You are expected to assign a proper value to operationcount in the workload file.

After all the specifications are done, you can easily give the final command to run the case. It should comfort to the format below (N is 1,2,3,…,19),

java -cp .:build/bsma.jar:db/idb/lib/ edu.ecnu.imc.bsma.Client -P workloads/workloadqueryN -s –p threadcount=α -p returncount=β -p timespan=γ > $qN\_\alpha\_\beta\_\gamma.txt$*

Above 19 queries will result in 8*(1*3+3*3+12*13)=1344 cases, and thus, 1344 text files. Each file contains query results and performance report of the corresponding case.

To sum up, for one case, you should do as follows,

step 1, choose a proper value for operationcount, and specify it in the workload file.

step 2, choose proper values for threadcount, returncount(optional) and timespan(optional), and specify them via “-p name=value”.

step 3, name the output text file using values of threadcount, returncount(optional) and timespan(optional).

step 4, generate the final command to run the case.

• Acknowledgement

The BSMA Performance Testing Tool is adapted from Yahoo! Cloud Serving Benchmark (YCSB) (https://github.com/brianfrankcooper/YCSB/). We would like to thank the authors of YCSB.

Contact us via bsma.users@gmail.com