BSMA: A Benchmark for Analytical Queries over Social Media Data

Fan Xia Ye Li Chengcheng Yu Haixin Ma Weining Qian

Institute for Data Science and Engineering Software Engineering Institute
East China Normal University, Shanghai, China
{fanxia,yeli,chengchengyu,haixinma}@ecnu.cn, wnqian@sei.ecnu.edu.cn

ABSTRACT

The demonstration of a benchmark, named as BSMA, for Benchmarking Social Media Analytics, is introduced in this paper. BSMA is designed to benchmark data management systems supporting analytical queries over social media. It is different to existing benchmarks in that: 1) Both real-life data and a synthetic data generator are provided. The real-life dataset contains a social network of 1.6 million users, and all their tweeting and retweeting activities. The data generator can generate both social networks and synthetic timelines that follow data distributions determined by predefined parameters. 2) A set of workloads are provided. The data generator is in responsible for producing updates. A workload generator produces queries based on predefined query templates by generating query arguments online. BSMA workloads cover a large amount of queries with graph operations, temporal queries, hotspot queries, and aggregate queries. Furthermore, the argument generator is capable of sampling data items in the timeline following power-law distribution online. 3) A toolkit is provided to measure and report the performance of systems that implement the benchmark. Furthermore, a prototype system based on dataset and workloads of BSMA is also implemented. The demonstration will include two parts, i.e. the internals of data and workload generator, as well as the performance testing of reference implementations.

1. INTRODUCTION

Social media has become a kind of important source for sensing events in the real world and understanding users for many commercial applications. Successful social-media-based applications rely on efficient social media data analytics.

Social media data analysis tasks have their own characteristics. First, social media data are large scale networks and data streams, in essence. Though the data can be represented in relations with relatively simple schema, the reference structure and data distributions are different to data in many other applications. Most existing benchmarks are not capable of generating such kind of data. Secondly, analytical queries over social media may contain subqueries on network patterns and temporal attributes. Furthermore, due to the skewed distributions of social media data, the performance of those queries is sensitive to query arguments. Intuitively, for example, a query on an opinion leader produces much more intermediate and final results than that on a common user. Last but not the least, different kinds of systems, including traditional RDBMS and so called NoSQL systems [3][6][9], can be used to implement social media data analytics. Therefore, to avoid comparing apples and oranges, a benchmark is required.

BSMA is designed for benchmarking performance of analytical queries over social media[1]. An early version of BSMA was used in WISE 2012 Challenge Performance Track[2], and was reported in [5].

The framework of BSMA is shown in Fig. 1. It contains three components:

Data feeding The data feeding component is in responsible for feeding data to the tested system. It has three modules. 1) A *real-life dataset* with a social network and user activities, i.e. tweeting and retweeting, of 1.6 million users is provided. The dataset is crawled from Sina Weibo[3], the most popular microblogging service in China. 2) A *synthetic data generator*, named as BSMA-Gen[10], is provided. It can continuously generate synthetic social networks and timeline structures based on predefined parameters. 3) A *data importer* is provided to use external social media data generators, such as that provided by LinkBench[1] and S3G2[7].

Workload generator The workload generator generates workloads, feeds them to the tested system, and collects results. Workloads of updates are from the data feeding component, while workloads of queries are generated based on twenty four query templates. An important module of workload generator is *argument generator*. It samples the dynamic data online. The sampled data is used as query arguments.

Performance testing toolkit A performance testing toolkit is provided by BSMA. The toolkit contains a set of

[1]Available at: `https://github.com/c3bd/BSMA`.
[2]For details, please refer to: `http://www.wise2012.cs.ucy.ac.cy/challenge.html` and `https://wnqian.wordpress.com/research/wise2012challenge/`.
[3]`http://weibo.com/`.

Proceedings of the VLDB Endowment, Vol. 7, No. 13

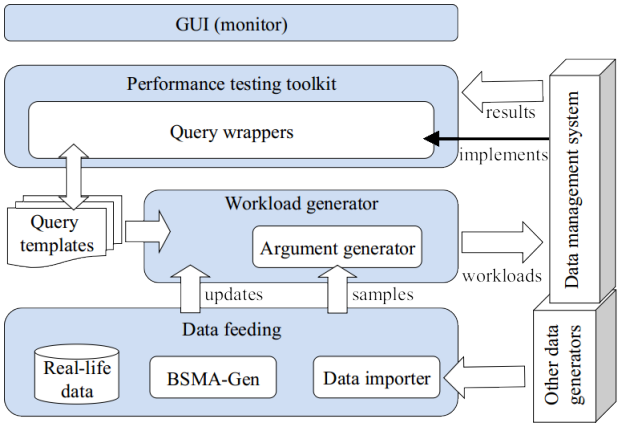

Figure 1: The framework of BSMA.

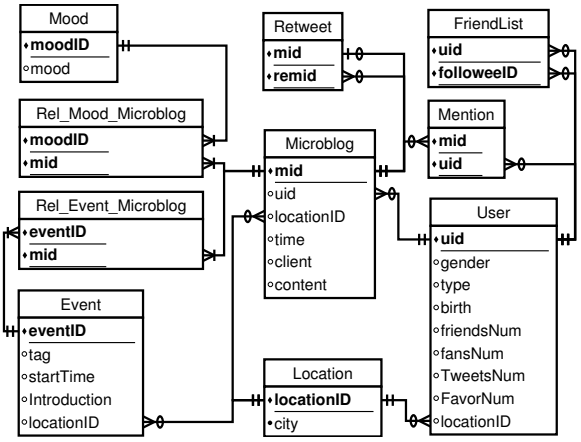

Figure 2: Schema of the Dataset

query wrappers that should be implemented by systems to be tested. Thus, BSMA can be used to benchmark systems with different interfaces. Query latency, system throughput, and scalability to the data volume can be tested by the toolkit. The toolkit also contains a monitor on dynamic data, workloads, and system performance.

We will demonstrate the internals of data feeding and workload generator. The benchmarking of reference implementations will also be demonstrated.

In the rest part of this demonstration proposal, three components of BSMA are introduced briefly in order. Finally, the demonstration outline is given in Section 5.

2. DATA FEEDING

BSMA provides both a real-life dataset and a synthetic data generator. To be simplicity, the schema of data is defined by the relational model, as it is shown in Fig. 2. It should be noted that systems to be benchmarked do not necessarily provide relational interfaces. The BSMA performance testing toolkit can be used to benchmark any systems, as long as the query wrappers are implemented.

The schema defines relations for storing both raw data, such as *Microblog*, *Retweet*, and *FriendList*, etc, and processed data, e.g. *Mood* and *Event*. The meaning of the relations are straightforward. Terms used by Twitter[4] are used to name relations and attributes.

2.1 The real-life dataset

The real-life dataset is crawled from Sina Weibo, the most popular microblogging service in China. The dataset contains a followship network of about 1.6 million users. All activities, i.e. tweeting and retweeting, of these users during August 2009 and January 2012 are provided in the dataset. There are totally 481 million tweets and retweets, and 1.2 billion followship relationships. Note that August 2009 is the month that Sina Weibo began to provide services.

The dataset is a historical partial snapshot of Sina Weibo. The raw data is preprocessed to meet the requirements of benchmarking.

- Due to legal and privacy reasons, content of tweets are removed. However, annotations on events are preserved. Furthermore, mood of tweets are detected and annotated.
- User identifiers are anonymized. All mentions are preserved. Thus, the references from tweets to users are preserved.
- The retweeting paths are recovered in a best effort approach. Note that Sina Weibo does not record complete retweeting paths. The recovered paths are useful for study on information diffusion.

The details of the data crawling can be found in [4].

2.2 Synthetic data generator

The synthetic data generator, named as BSMA-Gen, is in responsible for generating the followship network and timeline structures[10]. Here, timeline structures mean the timeline within which each data item is a quadruple $< u, t, f, c >$, where u is an user identifier, t is a timestamp, c is the content, and f is a pointer that can be *nil* or pointing to an earlier data item. All items in the timeline are sorted on timestamps. Currently, BSMA-Gen can only generate annotations for content. Thus, the generated timeline is a sketch that captures relationships of tweets and retweets.

Note that to efficiently generate realistic social media data is a non-trivial task. First, data items are correlated. For example, the probability of retweeting depends on both following relationships and the time gap between two activities. Secondly, both historical data and followship network are large. They are difficult to be stored in main memory completely. Last but not the least, social media data follows heavy-tailed distributions in many aspects. Thus, existing data generators often fail in producing realistic social media data[10].

The architecture of BSMA-Gen is shown in Fig. 3. It utilizes a nonhomogeneous Poisson process to generate activities of users. Two buffer pools, i.e. next-tweet pool and recent-tweet pool, are used to store information that will be accessed to generate new data items.

To further scale up the data generator, a parallel architecture is used. The master partitions the social network,

[4]http://twitter.com/

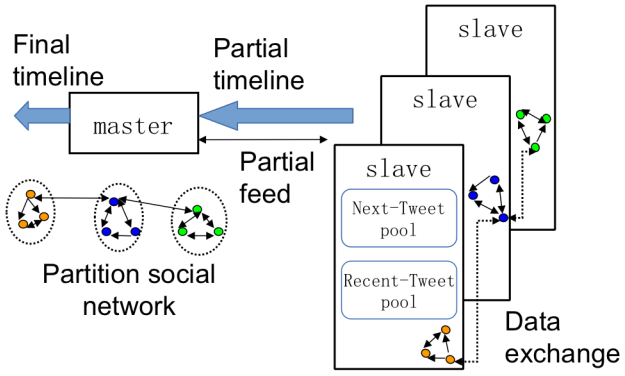

Figure 3: The architecture of BSMA-Gen.

and assigns partitions to slaves. Each slave is in responsible for generating partial feeds of users in that partition. While remote information is needed, asynchronized communications and a delayed update strategy is used, so that the throughput of data generator is not affected by remote data access. Experiments show that, given predefined parameters, BSMA-Gen can generate realistic synthetic data efficiently[10].

3. WORKLOAD GENERATOR

The workload generator transforms timeline from data generator into workloads on updates. The workload generator further generates queries based on 24 predefined query templates. Analytical queries over social media data often involve queries over social networks and temporal attributes. Furthermore, they may query hotspots, i.e. data items that have large reference numbers. There are four types of subqueries:

Graph operations Some queries may retrieve patterns from the followship network or the retweet trees. Graph operations may result in joins of very long tables.

Temporal queries They are subqueries on the timestamp attribute. Temporal queries often require that data items in intermediate and final results are ordered temporally.

Aggregate queries Aggregate queries compute one or more aggregate values based on the data items retrieved.

Hotspot queries Hotspot queries are a specific type of aggregate queries. They retrieve data items with top-k aggregate values. Since social media data often follows heavy-tailed distributions, hotspot queries often involve much more data items in intermediate and final results.

The characteristics of query templates are listed in Tab. 1. The specification of query definition is provided in the BSMA website. Again, note that though the definition is written in SQL, query wrappers are provided to enable the benchmarking of systems without standard relational interfaces.

3.1 Argument generator

All queries defined by query templates need arguments. Argument generator is an important module in workload generator. It online samples data items stored in buffer pools of data generator.

For data items following heavy-tailed distribution on some measurements, the method introduced in [8] for sampling power-law distributions is used, so that representative data items are used as query argument values. Thus, our benchmark is capable of measuring performance of systems on different kind of workloads.

4. PERFORMANCE TESTING TOOLKIT

BSMA is shipped with a performance testing toolkit, which is adapted from YCSB[2]. As long as query wrappers are implemented and argument generation policy is determined, the toolkit is capable of reporting three measurements, i.e, latency, throughput and scalability, as the benchmark result.

To test the performance of a system, a parameter *threadcount* is used to control the number of parallel query requests. Although eight parallel levels are recommended by BSMA toolkit, users could choose an acceptable value on behalf of their hardware and software configuration. The calculation of the three measurements is depicted as follows.

Latency For a test under a selected *threadcount*, the 99th latency of all requests is reported as the latency, which coincides with the concept of 9s in SLA (service-level agreement). For a benchmark job, we report the maximum latency among the latencies of all selected values of *threadcount*.

Throughput Similar to latency, the highest throughput among all *threadcount* is reported.

Scalability over data volume A set of pairs $(throughput, latency)$ is obtained from tests with different parallel levels. The linear regression method is adopted to find the line best fit to the dataset. The slope of the line is reported as the scalability.

The performance testing toolkit is highly configurable. Extended workloads and data feeds can be imported in our benchmark framework. Furthermore, a preliminary graphical user interface is provided to analyze the benchmarking results.

5. DEMONSTRATION OUTLINE

Our demonstration consists of the following three parts:

- First, the background of social media data analytics will be introduced. A service for collective behavior analysis based on our real-life social media data with a reference implementation of BSMA as backend will be demonstrated[5], as it is shown in Fig. 5. We will also show the characteristics of real-life social media data, including its schema and data distribution.
- The internals of BSMA will be introduced in detail, while the execution of data and workload generator will be demonstrated via the BSMA monitor.
  - The evolving of synthetic data generated will be shown in the monitor, as it is shown in Fig. 4. We will show how the generated data is dependent on historical data, and how BSMA-Gen manages

[5] `http://database.ecnu.edu.cn/microblogcube/`.

Table 1: Characteristics of query templates.

Type \ Query	Q1-Q7	Q8	Q9,Q10	Q11	Q12	Q13	Q14,Q15	Q16	Q17	Q18,Q19	Q20,Q21	Q22	Q23,Q24
Graph	✓			✓								✓	
Temporal		✓									✓		
Aggregates							✓		✓		✓		✓
Hotspot			✓	✓	✓	✓	✓	✓		✓			

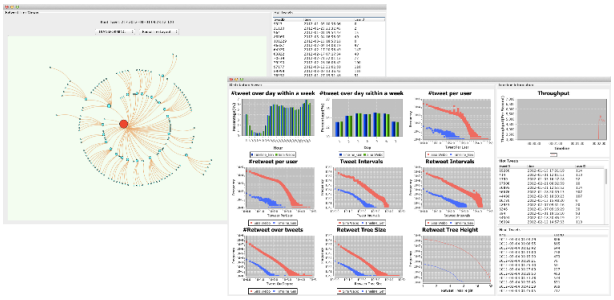

Figure 4: The monitor of BSMA.

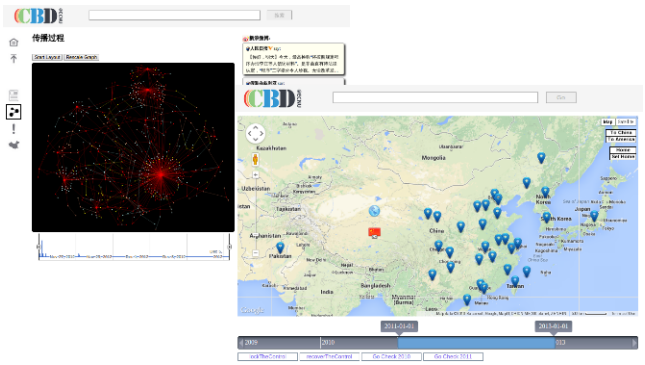

Figure 5: The GUI of a service with the reference implementation as backend.

buffer pools to support historical data access. Furthermore, the distributions of generated data will be compared with the distributions predefined by parameters.

– The workloads generated will be displayed in the monitor. The value of query arguments, especially those hotspots, will be listed. We will show how our argument generator samples the buffer pool with data following power-law distribution.

- The performance testing toolkit will be used to test a reference implementation. We will show how the toolkit reports performance measurements based on the data and workloads generated. We will also show how to configure the toolkit for various benchmarking tasks.

6. ACKNOWLEDGMENTS

This work is partially supported by National High-tech R&D Program (863 Program) under grant number 2012AA011003, National Basic Research (973 program) under grant number 2010CB731402, and National Science Foundation of China under grant number 61170086.

7. REFERENCES

[1] T. G. Armstrong, V. Ponnekanti, D. Borthakur, and M. Callaghan. Linkbench: a database benchmark based on the facebook social graph. In *SIGMOD Conference*, pages 1185–1196, 2013.

[2] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears. Benchmarking cloud serving systems with ycsb. In *SoCC*, pages 143–154, 2010.

[3] C. Engle, A. Lupher, R. Xin, M. Zaharia, M. J. Franklin, S. Shenker, and I. Stoica. Shark: fast data analysis using coarse-grained distributed memory. In *SIGMOD Conference*, pages 689–692, 2012.

[4] H. Ma, W. Qian, F. Xia, X. He, J. Xu, and A. Zhou. Towards modeling popularities of microblogs. *Frontiers of Computer Science*, 7(2), 2013.

[5] H. Ma, J. Wei, W. Qian, C. Yu, F. Xia, and A. Zhou. On benchmarking online social media analytical queries. In *GRADES*, page 10, 2013.

[6] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig latin: a not-so-foreign language for data processing. In *SIGMOD Conference*, pages 1099–1110, 2008.

[7] M.-D. Pham, P. A. Boncz, and O. Erling. S3g2: A scalable structure-correlated social graph generator. In *TPCTC*, pages 156–172, 2012.

[8] G. Pickering, J. Bull, and D. Sanderson. Sampling power-law distributions. *Tectonophysics*, 248(1):1–20, 1995.

[9] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy. Hive - a warehousing solution over a map-reduce framework. *PVLDB*, 2(2):1626–1629, 2009.

[10] C. Yu, F. Xia, W. Qian, and A. Zhou. On efficiently generating realistic social media timeline structures. Technical Report, C^{3}BD@ECNU, April 2014. Also available as `http://wnqian.drivehq.com/publications/TR-C3BD-20140301.pdf`.