On Benchmarking Online Social Media Analytical Queries

Haixin Ma
Institute of Massive Computing
Software Engineering Institute
East China Normal University
51111500010@ecnu.cn

Weining Qian
Institute of Massive Computing
Software Engineering Institute
East China Normal University
wnqian@sei.ecnu.edu.cn

Fan Xia
Institute of Massive Computing
Software Engineering Institute
East China Normal University
52101500012@ecnu.cn

Jinxian Wei
Institute of Massive Computing
Software Engineering Institute
East China Normal University
51101500014@ecnu.cn

Chengcheng Yu
Institute of Massive Computing
Software Engineering Institute
East China Normal University
52111500011@ecnu.cn

Aoying Zhou
Institute of Massive Computing
Software Engineering Institute
East China Normal University
ayzhou@sei.ecnu.edu.cn

ABSTRACT

Social media analytics has many applications in collective behavior sensing and monitoring, online advertisement, opinion mining, and etc. Though a number of technologies and systems are proposed for analyzing social media data, the overall performance and the advantages of those technologies and systems are not compared under similar settings. In this paper, a benchmark named as BSMA, for Benchmarking Social Media Analytics, is proposed. It distinguishes with other similar effort in that: 1) A real-life dataset with activties of more than 1.6 million users in 2 years and followship relationships of 1.2 billion users is used. The distributions of data in the dataset is different from those of data generators. 2) 19 queries fitting into three categories, i.e. social network quries, hotspot queries, and timeline queries, are used. The three categories each poses challenge to different part of testing systems. 3) Measurements of throughput, latency, and scalability are used for testing performance. A toolkit for reporting measurement values that are based on YCSB is developed. A previous version of BSMA is used in WISE 2012 Challenge. Four teams implemented all or part of the 19 queries. Their results are analyzed in this paper. The progress and future work of BSMA is also discussed.

1. INTRODUCTION

Social media services are widely used for recording and sharing of what users are seeing, hearing and thinking. Analysis of the huge volume of social media data has many applications such as collective behavior sensing and monitoring, online advertisement, opinion mining, and etc. Social media data distinguishes itself from other kind of data in that, first, it consists of both structured and unstructured data. For example, the user profile is usually structured or semi-structured. However, the content of pieces of information is usually unstructured. Furthermore, the followship (or subscription) relationships and repost relationships form huge graphs. Though these graphs can be modeled as adjacency lists, traditional data management technologies are not capable of handling them due to the huge number of rows and costly self-join operations that are often needed in query processing.

Secondly, social media data is dynamic. A social media service may continuously append pieces of information from users to the backend database in high speed. Meanwhile, analytical queries over the data may specify conditions on the time dimension. The temporal attribute gives hints on caching. However, it also poses difficulty on indexing.

Last but not the least, the distribution of social media data is highly biased. For example, opinion leaders may attract much more followers than common users, while an emerging event may result in a burst of pieces of information. Therefore, an efficient query engine should be able to handle not only ordinary users and time period, but also those hotspots.

Many systems are used for management of social media data. Hadoop, the open-source clone of Google File System[5] and MapReduce programming paradigm[5], is often used for storage of social media data. Then, MapReduce programs, scipts written in Pig Latin[7], or SQL-like queries written for Hive[11] can be used for analyzing the data. There are also proposals for using in-memory data management systems, such as Spark[15] or HANA[4], for the same purpose. Systems designed specifically for social media data management, such as the Little Engine[8] and Feed Frenzy[9] also exist.

Thus, a natural question is: *what is the advantage of each system in social media data analytics?* We propose the BSMA, for Benchmarking Social Media Analytics, in this paper. The contributions of the paper are as follows:

- BSMA uses a dataset crawled from Sina Weibo[1], which is the most popular microblogging service in China. The data set consists of a followship network and a series of user activities. The distribution of the data is different to those generated by existing social media/network data generator, such as SIB[12]. Thus, we believe that the benchmark develped based on the

[1] `http://weibo.com`

Proceedings of the First International Workshop on Graph Data Management Experience and Systems (GRADES 2013), June 23, 2013, New York, NY, USA.

real-life dataset is meaningful for testing the performance of social media data analytics.

- 19 types of queries for performance benchmarking are introduced. The queries can be classified into three categories, i.e. social network queries, timeline queries, and hotspot queries. They are designed for testing the performance of systems over different types of analytical requests. Thus, BSMA is different to graph-serving benchmarks, such as LinkBench[3].
- The performance measurements of throughput, latency, and scalability are used in BSMA. A toolkit[2] developed based on Yahoo Cloud Service Benchmark (YCSB)[1] is used in BSMA for reporting the throughput and latency values. The measurements of scalability can be determined based on reported values of other two measurements.
- A previous version of BSMA was used in WISE 2012 Challenge[3]. Four groups attended the challenge. The details on the challenge are introduced, while part of reported results are analyzed in this paper.

The rest part of this paper is organized as follows. In Section 2, the dataset used in BSMA is introduced. The schema is provided, while the statistics and distribution of the dataset is analyzed. The queries in three categories are introduced in Section 3. The challenges on processing these queries are analyzed. The performance measurements are also defined. The Section 4 is devoted to analysis of results from WISE 2012 Challenge. Finally, Section 5 is for concluding remarks and discussion on future work.

2. SOCIAL MEDIA DATASET

BSMA uses a dataset crawled via API from Sina Weibo, the most popular microblogging service in China. To ease the discussion, we adopt terms used by Twitter[4] in the rest of this paper. Though some operations with identical name in Twitter and Sina Weibo provide slightly different functions [6], the difference does not affect the discussion in this paper.

The dataset contains two parts: user activities and followship network. The basic information is as follows:

User activities: It contains about 481 million tweets (including retweets) of 1.6 million users from August 2009 to January 2012.

Followship network: It contains about 1.2 billion followship relationships.

2.1 Data Collecting and Preprocessing

A distributed crawler was developed to collect data from Sina Weibo. The crawling procedure of our system is showed in Fig. 1. In the first place, 32 users are selected as seeds and a breadth-first strategy is applied to crawl the information along the direction of followees of the selected users. The first three levels of breadth-first search result in information of 1.6 million users, who are called as core users in the rest of this paper. Then, the top 5000 followers of the core users

[2]`https://github.com/xiafan68/BSMA`
[3]`http://www.wise2012.cs.ucy.ac.cy/challenge.html`
[4]`http://twitter.com`

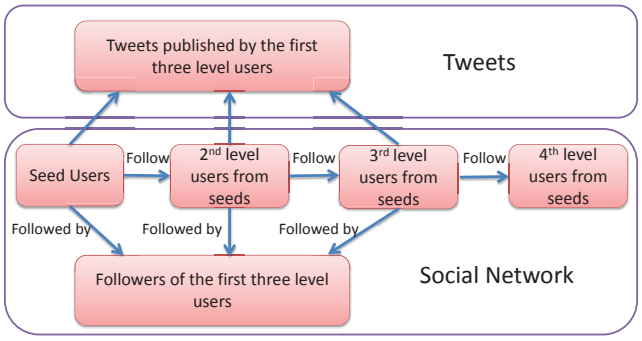


Figure 1: The crawling process of the data set used in this paper.

are crawled. Thus, about 1.2 billion followship relationships are collected.

The tweets of core users from August 2009 to January 2012 are also collected, which form the basis of the first part of the dataset.

It should be noted that the dataset is neither synchronized nor complete, which means the items in the dataset are crawled at different time, while some users' tweets and their followship relationships are missing. This issue is caused by the limitation of Sina Weibo API. However, we believe that most social media analytics tasks from users outside Sina should face this issue.

The raw data crawled from Sina Weibo are preprocessed for legal and privacy considerations. The dataset is preprocessed as follows:

- User identifiers and message identifiers are anonymized.
- Content of tweets are removed[5].
- Some tweets are annotated with events. For each event, the terms that are used to identify the event and a link to Wikipedia[6] page containing descriptions to the event are given[7].
- The retweet paths are re-constructed in a best-effort manner[8].

2.2 Schema of the Dataset

The dataset is provided in plain text files. The schema of the dataset is defined to ease the formalization of queries.

The first part of the dataset contains four tables, which are listed in Table 1, 2, 3, and 4. The `microblog` table records the message identifier, the author's user identifier, and the publish time of the tweet. The event about the tweet is recorded in the `event` table, while the users that are mentioned are recorded in the `mention` table. The retweeting information is recorded in the `retweet` table, which actually records the information of tweet propagation trees.

[5]Most tweets are in Chinese.
[6]`http://wikipedia.org`
[7]`http://115.com/file/beem15q0`
[8]Sina Weibo API does not provide retweet paths. However, a path can be re-constructed if the author of a retweet has not intentionally remove the retweeting information.

Table 1: The `microblog` Table

Attribute	Data Type	Description
MID	text(64)	Message identifier
UID	text(200)	Author's user identifier
TIME	date.time	Time when the tweet is posted

Table 2: The `event` Table

Attribute	Data Type	Description
MID	text(64)	Message identifier
TAG	text(200)	The tag of event

Table 4: The `retweet` Table

Attribute	Data Type	Description
MID	text(64)	Message identifier of the retweet
REMID	text(64)	MID of the tweet that is retweeted

Table 5: The `friendlist` Table

Attribute	Data Type	Description
UID	text(200)	User identifier
FRIENDID	text(200)	A user that is followed by UID

The second part of the dataset contains just one table. The `friendlist` table is essentially the adjacency list of the followship network. The table definition is provided in Table 5.

2.3 Data Distributions

The real-life dataset, instead of a data generator, is used in BSMA, because that it is noticed that the synthetic data often have different distributions. The Social Network Intelligence Benchmark (SIB) [12], for example, uses a generator to generate synthetic RDF data. However, it is shown in Figure 2 that the distribution of number of followees, number of retweets (or comments), user activities, and temporal properties are all different to our real-life dataset. It is shown that the real-life dataset is more biased and dynamic. The mechanics designed by the social media service also affects the distribution. For example, the steep gradient in Figure 2 (b) is actually caused by the limitation on number of followees for common users.

Since data distributions may greatly affect the strategies of cost estimation, indexing and query processing, especially when the hotspots and bursts exist, we believe that using the real-life dataset in the benchmark is meaningful for testing the performance of social media analytics.

3. WORKLOAD AND MEASUREMENTS

3.1 Overview of the Queries

The workload of our benchmark consists of nineteen queries derived from real-life social media analytical requirements. Generally, they can be classified into three categories:

Social network queries: $Q1, Q2, Q3, Q4$ and $Q5$ belong to this class. All these queries are supposed to retrieve a subset of the entire social network to find out all the users satisfying the specified constraints. In detail, $Q4$ and $Q5$ are based on intersection between the followers or followees of two users, while $Q1, Q2$ and $Q3$ are to find the top-k users that share as more as possible common followships with a given user. Clearly, the execution of all the five queries needs to pass parameter $userID$ and an additional parameter $returncount$ is to transferred to $Q1, Q2$ and $Q3$.

Timeline queries The only *timeline* query is $Q8$. A *timeline* is a sequence of items (e.g., messages) created by a certain set of users, that are ordered chronologically. Particularly, $Q8$ is to merge top-k latest items from followees or followees of them for a given user. Two related parameters are $userID$ and $returncount$.

Hotspot queries All other queries except those in above two categories are supposed to retrieve *hotspots*. *Hotspots* are users or messages or events (depending on the query) that have the largest aggregation values of some features during a specific period. Some queries, e.g. $Q7, Q10$ and $Q14$, have no filtering criteria while others need filtering by one or more properties. All queries in this category are associated with three parameters: start $datetime$,$timespan$ and $returncount$. Some also need $userID$ or event tag.

A query may contain several arguments, which are listed in Table 6. Values of $returncount$ and $timespan$ are given in workload files of BSMA. The options of $returncount$ are 10, 50 and 100. Values of $timespan$ are h, for one hour, d, for one day, and w, for one week, and y, for one year. Other arguments' values are randomly selected from each candidate set in runtime.

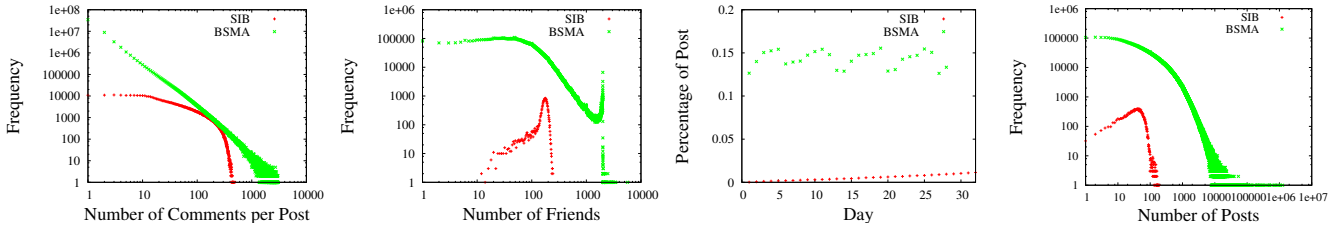
Queries are given in forms of SQL over the schema. However, the BSMA performance testing tool accepts implementations based on systems other than RDBMS, as long as the wrappers of the implementation fit the interfaces.

Table 3: The `mention` Table

Attribute	Data Type	Description
MID	text(64)	Message identifier
UID	text(200)	A user identifier that is mentioned in the message

Table 6: Arguments of queries in BSMA

Arguments	Description
userID	User identifier
tag	The tag of an event
datetime	Start timestamp
timespan	Time interval
returncount	The maximum number of returned records



(a) Distribution of comments and retweets (b) Distribution of friends (c) Distribution of #posts over time (d) Distribution of #posts of users

Figure 2: Data distribution of BSMA and Social Network Intelligence Benchmark.

3.2 Query Cases

Queries of different categories need to access different social media data and the operators involved in each query may also vary. It is non-trivial for processing these queries. Several queries are analyzed in this subsection to illustrate the difficulties.

Social media data typically contains various kinds of closely related informations, e.g. social network, generated tweets and the retweet graph. When normalized in relation model, the data would be represented with a number of large tables. It is common that analysis tasks need to integrate multiple pieces of data, which results in joins with huge tables. As a simple but appealing application, a user may want to discover those popular tweets viewed by him and his followees. $Q12$, for example, ranks the tweets appearing in somebody's followees' timelines according to the number of retweet, as it is shown in Figure 3. However, such a query need to self-join the friendlist table to retrieve the followees of his followees. Then the retrieved UIDs need to be joined with the microblog table to select the tweets published by them. At last those tweets are further joined with the retweet table to produce the input to aggregate function so that the number of times each tweet is retweeted can be computed. Hence, all the three tables involved in those joins are extremely large. Besides, two arguments, i.e, *datetime* and *timespan*, specify the segment of timeline the tweets during which need to be analyzed. The timeline dimension makes the partition of social media data more complicated apart from the essential graph structure under the data. Other queries such as $Q6$, $Q9$, and $Q13$ are similar to $Q12$. Consequently, substantial optimization are needed.

Most types of social media data adhere to the power-law distribution. Such phenomena causes queries of the same type instantiated with different argument executed with different performance. For example, $Q2$, shown in Figure 4, is designed to find the set of people who share the same followee with the specified user, which is useful for recommending potential friends. Once Q2 is provided with a user with many followee, a large set of followers will be selected and then join with the friendlist table again, which will return a even larger set of tuples. The situation becomes worse when the user follows some authorities, i.e. nodes with enormous followers. Hence, the size of input to the sort and aggregate operation varies greatly. Developers need to confirm that the system won't crash or stuck in such kind of queries such that other small queries are also blocked.

3.3 Performance Measurements and Testing

```
SELECT x.remid
FROM microblog,
 (SELECT retweet.mid AS mid,retweet.remid AS remid
  FROM microblog,retweet
  WHERE microblog.mid = retweet.remid) AS x
WHERE microblog.mid = x.mid AND
 microblog.uid IN
  (SELECT friendID
   FROM friendList
   WHERE uid = "A" OR
   uid IN
    (SELECT friendID
     FROM friendList
     WHERE uid = "A")) AND
     microblog.time BETWEEN
      TO_DAYS('YYYY-MM-DDHH:MM:SS') AND
     DATE_ADD('YYYY-MM-DD HH:MM:SS',INTERVAL 1HOUR)
GROUP BY x.remid
ORDER BY COUNT(*)DESC
LIMIT 10;
```

Figure 3: $Q12$

To test the performance of a system under different workloads, BSMA uses the parameter of *threadcount* to control the number of parallel requests. A user of BSMA may set the appropriate parameter value by himself to fit the hardware and software configuration for testing.

BSMA is developed based on YCSB[1]. Users need to implement all or part of queries and call their implementations inside wrappers of queries in BSMA. Three measurements are used for testing.

Throughput The highest throughput over eight different settings of *threadcount*. Higher value gets higher score.

Latency Average latency under second highest throughput over eight different settings of *threadcount*. Lower value gets higher score.

Scalability The slope of the line that had the best fit to the (throughput, latency) data points by least squares method. Lower slope gets higher score.

The above three measurements imply practical significance. The throughput measures the limit of number of concurrencies a system can reach, which is critical to social media naturally along with potential burst data transmission. Since low latency guarantee is key to user experience,

```
SELECT f1.uid
FROM friendList AS f1,
 (SELECT friendID
  FROM friendList
  WHERE uid = "A") AS f2
WHERE f1.uid <> "A" AND
 f1.friendID = f2.friendID AND
 f1.uid<> f2.friendID
GROUP BY f1.uid
ORDER BY COUNT(f1.friendID)DESC
LIMIT 10;
```

Figure 4: $Q2$

BSMA uses latency measurements, under which, response time under second highest throughput instead of the highest one is considered for the fact that systems are chugging along at a utilization rate of about 80% at normal state in real life. The scalability measurement is given to check whether the benchmarked systems can work well with dynamically increasing throughput.

4. WISE 2012 CHALLENGE PERFORMANCE TRACK RESULT ANALYSIS

A previous version of BSMA is used in WISE 2012 Challenge Performance Track[14]. Four teams attends the challenge[10, 2, 16, 17]. Each team implements part of the queries correctly.

To make a deep comparison and analysis of the set of queries, we filtered out all the incorrect performance reports and dealt with the remaining ones as follows: firstly, for each combination of *returncount* and *timespan* to one query, we calculated its value under the three measurements query by query and team by team. Then, for each team, we averaged its values under all combinations query by query and measurement by measurement. Finally, for each query, we made an average among all teams measurement by measurement.

Figure 5 shows the averaged highest throughputs of all the sixteen queries. Note that queries that $Q6, Q7$ and $Q18$ are missing since no team implemented them correctly. Figure 6 indicates the averaged latencies under second highest throughputs of those queries.

We now focus on $Q1, Q2, Q3, Q4, Q5, Q14$ and $Q19$ since these queries were implemented properly by most teams. Throughput of $Q1, Q2$ and $Q3$ is low while latency of them is high. Those of $Q4$ and $Q5$ are just the opposite. All the five queries are social network queries. $Q1, Q2$ and $Q3$ are supposed to find the top-k users that have common relations with a given user, while $Q4$ and $Q5$ are to find out the intersection of users related to two specified users. Consequently, the former queries need a scan and filter upon much more users than that of $Q4$ and $Q5$.

Both $Q14$ and $Q19$ are hotspots queries. However, $Q19$ results in a much lower throughput and higher latency in comparison with $Q14$ for the reason that hotspots retrieved through $Q19$ should match an extra filtering criteria.

$Q8$, the only timeline query, was processed correctly by only one team, who could not achieve a satisfactory performance of $Q8$ at first in spite of their in-memory system and finally made it after a series of optimizations[2].

The scalability measurements are reported in Figure 7.

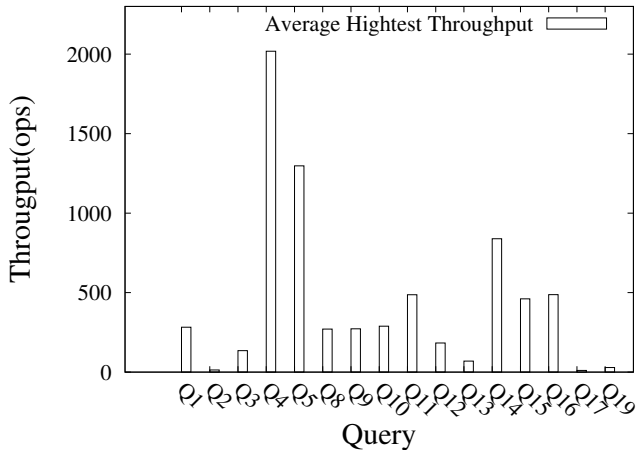


Figure 5: Average highest throughput of 16 queries

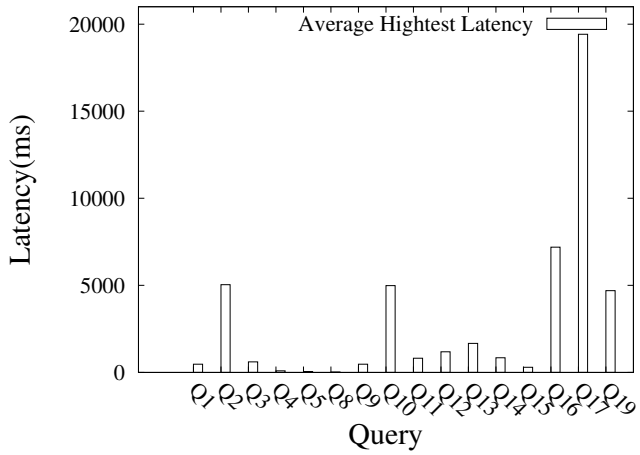


Figure 6: Average latency under second highest throughput of 16 queries

Scalability values with negative values are not shown in the figure. It is shown that all teams failed in achieve high scalability for $Q2$, which is supposed to scan a considerable big set of users.

The preliminary analysis shows that, 1) social network queries are challenging since scan of the data and self-join of a large table may be involved. 2) Hotspots queries associated with more filtering criteria tend to be more difficult. And, 3) Timeline query deserves dedicated optimization.

5. CONCLUSIONS AND DISCUSSIONS

The BSMA for benchmarking social media data analytics is introduced in this paper. BSMA uses a real-life dataset from Sina Weibo. 19 types of queries in three categories are defined, while measurements on throughput, latency, and scalability can be reported by a toolkit developed based on YCSB for performance testing. A previous version of BSMA was used in WISE 2013 Challenge. The results submitted by four teams are reported and analyzed in this paper.

BSMA is in its early stage. Our future work on the benchmark includes:

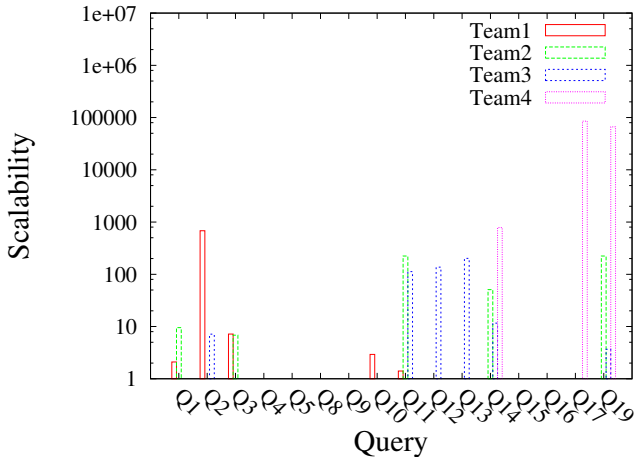


Figure 7: Scalability of 16 queries

Data generator: We are working on a distributed data generator for generating synthetic data that are consistent with the distribution of real-life social media data.

Queries related to content of tweets: Some analytical queries may have query conditions related to content of tweets. We are working on retrieval style queries using vectors and n-grams.

Other queries: We are working on other typical social media analytical queries that are to be put into the query set.

Performance testing of more systems: We are working on benchmarking more systems by using BSMA.

6. ACKNOWLEDGMENTS

This work is partially supported by National Science Foundation of China under grant No. 60925008, 61170086, and 61021004, National Basic Research (973 Program) under grant No. 2010CB731402, and National High-tech R&D Program (863 Program) under grant No. 2012AA011003.

7. REFERENCES

[1] Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R. Benchmarking cloud serving systems with ycsb. In *SoCC* (2010), J. M. Hellerstein, S. Chaudhuri, and M. Rosenblum, Eds., ACM, pp. 143–154.

[2] de Oliveira Sandes, E. F., Weigang, L., and de Melo, A. C. M. A. Logical model of relationship for online social networks and performance optimizing of queries - wise 2012 challenge - t1: Performance track scalability winner. In Wang et al. [13], pp. 726–736.

[3] Facebook Engineering. Linkbench: A database benchmark for the social graph. "https://www.facebook.com/notes/facebook-engineering/linkbench-a-database-benchmark-for-the-social-graph/10151391496443920", April 2013.

[4] Färber, F., May, N., Lehner, W., Grosse, P., Müller, I., Rauhe, H., and Dees, J. The sap hana database – an architecture overview. *IEEE Data Eng. Bull. 35*, 1 (2012), 28–33.

[5] Ghemawat, S., Gobioff, H., and Leung, S.-T. The google file system. In *SOSP* (2003), M. L. Scott and L. L. Peterson, Eds., ACM, pp. 29–43.

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

[7] Olston, C., Reed, B., Srivastava, U., Kumar, R., and Tomkins, A. Pig latin: a not-so-foreign language for data processing. In *SIGMOD Conference* (2008), J. T.-L. Wang, Ed., ACM, pp. 1099–1110.

[8] Pujol, J. M., Erramilli, V., Siganos, G., Yang, X., Laoutaris, N., Chhabra, P., and Rodriguez, P. The little engine(s) that could: Scaling online social networks. *IEEE/ACM Trans. Netw. 20*, 4 (2012), 1162–1175.

[9] Silberstein, A., Terrace, J., Cooper, B. F., and Ramakrishnan, R. Feeding frenzy: selectively materializing users' event feeds. In *SIGMOD Conference* (2010), A. K. Elmagarmid and D. Agrawal, Eds., ACM, pp. 831–842.

[10] Tang, Z., Lin, H., Li, K., Han, W., and Chen, W. Acolyte: An in-memory social network query system. In Wang et al. [13], pp. 755–763.

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

[12] W3C. Social network intelligence benchmark. `http://www.w3.org/wiki/Social_Network_Intelligence_BenchMark`. [Online, accessed 1-November-2012].

[13] Wang, X. S., Cruz, I. F., Delis, A., and Huang, G., Eds. *Web Information Systems Engineering - WISE 2012 - 13th International Conference, Paphos, Cyprus, November 28-30, 2012. Proceedings* (2012), vol. 7651 of *Lecture Notes in Computer Science*, Springer.

[14] WISE. Wise 2012 challenge. "http://www.wise2012.cs.ucy.ac.cy/challenge.html", November 2012.

[15] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M., Shenker, S., and Stoica, I. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI 2012)* (2012), USENIX Association, pp. 2–2.

[16] Zheng, L., Zhou, X., Lin, Z., and Jin, P. Accelerating queries over microblog dataset via grouping and indexing techniques. In Wang et al. [13], pp. 764–770.

[17] Zhu, F., Liu, J., and Xu, L. A fast and high throughput sql query system for big data. In Wang et al. [13], pp. 783–788.

APPENDIX

The documents of BSMA includes:

- Data format (A1.txt);
- Queires (A2.pdf);
- BSMA performance testing tool manual (A3.pdf).

The version of BSMA, including the dataset, used in WISE 2012 Challenge Performance Track is available at: `http://www.wuala.com/imc_ecnu/wise_challenge/`. A followup web page of WISE 2012 Challenge is available at: `https://wnqian.wordpress.com/research/wise2012challenge/`. The BSMA performance testing tool is maintained at: `https://github.com/xiafan68/BSMA`.