

DEBS 2016

The 10th ACM International Conference on Distributed and Event-Based Systems

Irvine, CA, June 20 - June 24, 2016

[Home \(index.html\)](#)

[Calls](#)

[Conference
Information](#)

[Program](#)

[Registration](#)

[Committees](#)

[Support DEBS 2016 \(DEBS2016_sponsoring.pdf\)](#)

Call for Grand Challenge Solutions

The ACM DEBS 2016 Grand Challenge is the sixth in a series of challenges which seek to provide a common ground and uniform evaluation criteria for a competition aimed at both research and industrial event-based systems. The goal of the 2016 DEBS Grand Challenge competition is to evaluate event-based systems for real-time analytics over high volume data streams in the context of graph models.

The underlying scenario addresses the analysis metrics for a dynamic (evolving) social-network graph. Specifically, the 2016 Grand Challenge targets following problems: (1) identification of the posts that currently trigger the most activity in the social network, and (2) identification of large communities that are currently involved in a topic. The corresponding queries require continuous analysis of a dynamic graph under the consideration of multiple streams that reflect updates to the graph.

The data for the DEBS 2016 Grand Challenge is based on the dataset provided together with the [LDBC Social Network Benchmark](#). DEBS 2016 Grand Challenge takes up the general scenario from the [2014 SIGMOD Programming Contest](#), however, in contrast to the SIGMOD contest, it explicitly focuses on processing streaming data and thus dynamic graphs. Details about the data, queries for the Grand Challenge, and information about evaluation are provided below.

Prize

Participants in the 2016 DEBS Grand Challenge will have the chance to win two prizes. The first prize is the “Grand Challenge Award” for the best performing, correct submission. The winner (or winning team) of the “Grand Challenge Award” will be additionally awarded with a monetary reward of \$1000. The monetary reward and the whole DEBS 2016 Grand Challenge is sponsored by the [WSO2](#).

The second prize is the “Grand Challenge Audience Award” - it is determined based on the audience feedback provided after the presentation of the Grand Challenge solutions during the DEBS 2016 conference. The “Grand Challenge Audience Award”, as opposed to the overall “Grand Challenge Award” does not entail any additional monetary reward as it is based purely on the audience perception of the

the overall Grand Challenge Award, does not entail any additional monetary reward as it is based purely on the audience perception of the presentation of a given solution.

Support

We would like to explicitly thank WSO2 for sponsoring the DEBS 2016 Grand Challenge prize and the LDBC Council for their help in preparing the test data set. We also like to thank the BICC lab at the Hochschule Furtwangen University for providing the resources to run the evaluation system.



License

All solutions submitted to the DEBS 2016 Grand Challenge must be made open source under the BSD license:

<https://opensource.org/licenses/BSD-3-Clause>. A solution incorporates concepts, queries, and code developed for the purpose of solving the Grand Challenge. If a solution is developed within the context of, is built on top of, or is using an existing system or solution which is licensed under a different license than BSD, then such an existing solution or system needs not have its license changed.

Input Data Streams

The input data is organized in four separate streams, each provided as a text file. Namely, we provide the following input data files:

friendships.dat: <ts, user_id_1, user_id_2>

ts	is the friendship's establishment timestamp
user_id_1	is the id of one of the users
user_id_2	is the id of the other user

posts.dat: <ts, post_id, user_id, post, user>

ts	is the post's timestamp
post_id	is the unique id of the post
user_id	is the unique id of the user
post	is a string containing the actual post content
user	is a string containing the actual user name

comments.dat: <ts, comment_id, user_id, comment, user, comment_replied, post_commented2>

ts	is the comment's timestamp
----	----------------------------

comment_id	is the unique id of the comment
user_id	is the unique id of the user
comment	is a string containing the actual comment
user	is a string containing the actual user name
comment_replied	is the id of the comment being replied to (-1 if the tuple is a reply to a post)
post_commented	is the id of the post being commented (-1 if the tuple is a reply to a comment)

likes.dat: <ts, user_id, comment_id>

ts	is the like's timestamp
user_id	is the id of the user liking the comment
comment_id	is the id of the comment

Please note that as the contents of files represent data streams, each file is sorted based on its respective timestamp field.

A small sample set of input data streams can be downloaded from this URL: <https://www.dropbox.com/s/vo83ohrgcgfqg27/data.tar.bz2>. These files are only meant for development and debugging. A bigger data set can be downloaded from here: <https://www.dropbox.com/s/dcy21uc745pv98i/data.zip?dl=0>. Our tests will be based on other files (about the same size as the provided larger data set), but strictly following the same format.

Query 1

The goal of query 1 is to compute the top-3 scoring active posts, producing an updated result every time they change.

The total score of an active post P is computed as the sum of its own score plus the score of all its related comments. Active posts having the same total score should be ranked based on their timestamps (in descending order), and if their timestamps are also the same, they should be ranked based on the timestamps of their last received related comments (in descending order). A comment C is related to a post P if it is a direct reply to P or if the chain of C's preceding messages links back to P.

Each new post has an initial own score of 10 which decreases by 1 each time another 24 hours elapse since the post's creation. Each new comment's score is also initially set to 10 and decreases by 1 in the same way (every 24 hours since the comment's creation). Both post and comment scores are non-negative numbers, that is, they cannot drop below zero. A post is considered no longer active (that is, no longer part of the present and future analysis) as soon as its total score reaches zero, even if it receives additional comments in the future.

Input Streams: *posts, comments*

Output specification:

```
<ts,top1_post_id,top1_post_user,top1_post_score,top1_post_commenters,
top2_post_id,top2_post_user,top2_post_score,top2_post_commenters,
top3_post_id,top3_post_user,top3_post_score,top3_post_commenters>
```

ts: the timestamp of the tuple event that triggers a change in the top-3 scoring active posts appearing in the rest of the tuple

topX_post_id: the unique id of the top-X post

topX_post_user: the user author of top-X post

topX_post_commenters: the number of commenters (excluding the post author) for the top-X post

Results should be sorted by their timestamp field. The character "-" (a minus sign without the quotations) should be used for each of the fields (post id, post user, post commenters) of any of the top-3 positions that has not been defined. Needless to say, the logical time of the query advances based on the timestamps of the input tuples, not the system clock.

Sample output tuples for the query

2010-09-19 12:33:01.923+0000,25769805561,Karl Fischer,115,10,25769805933,Chong Liu,83,4,-,-,-,-

2010-10-09 21:55:24.943+0000,34359739095,Karl Fischer,58,7,34359740594,Paul Becker,40,2,34359740220,Chong Zhang,10,0

2010-12-27 22:11:54.953+0000,42949673675,Anson Chen,127,12,42949673684,Yahya Abdallahi,69,8,42949674571,Alim Guliyev,10,0

Query 2

This query addresses the change of interests with large communities. It represents a version of query type 2 from the 2014 SIGMOD Programming contest. Unlike in the SIGMOD problem, the version for the DEBS Grand Challenge focuses on the dynamic change of query results over time, i.e., calls for a continuous evaluation of the results.

Goal of the query:

Given an integer k and a duration d (in seconds), find the k comments with the largest range, where the range of a comment is defined as the size of the largest connected component in the graph defined by persons who (a) have liked that comment (see *likes*, *comments*), (b) where the comment was created not more than d seconds ago, and (c) know each other (see *friendships*).

Parameters: k, d

Input Streams: *likes*, *friendships*, *comments*

Output:

The output includes a single timestamp ts and exactly k strings per line. The timestamp and the strings should be separated by commas. The k strings represent comments, ordered by range from largest to smallest, with ties broken by lexicographical ordering (ascending). The k strings and the corresponding timestamp must be printed only when some input triggers a change of the output, as defined above. If less than k strings can be determined, the character "-" (a minus sign without the quotations) should be printed in place of each missing string.

The field ts corresponds to the timestamp of the input data item that triggered an output update. For instance, a new friendship relation may change the size of a community with a shared interest and hence may change the k strings. The timestamp of the event denoting the added friendship relation is then the timestamp ts for that line's output. Also, the output must be updated when the results change due to the progress of time, e.g., when a comment is older than d . Specifically, if the update is triggered by an event leaving a time window at t_2 (i.e., t_2 = timestamp of the event + window size), the timestamp for the update is t_2 . As in Query 1, it is needless to say that timestamps refer to the logical time of the input data streams, rather than on the system clock.

In summary, the output is specified as follows:

ts: the timestamp of the tuple event that triggers a change in the output.

comments_1,...,comment_k: top k comments ordered by range, starting with the largest range (comment_1).

Sample output tuples for the query with k=3 could look as follows:

```
2010-10-28T05:01:31.022+0000,I love strawberries,-,-
2010-10-28T05:01:31.024+0000,I love strawberries,what a day!,-
2010-10-28T05:01:31.027+0000,I love strawberries,what a day!,well done
2010-10-28T05:01:31.032+0000,what a day!,I love strawberries,well done
```

Submission Information

For general submission guidelines (incl. paper formatting) please refer to <http://www.debs2016.org/submission-guidelines.html>.

Below we provide the specific submissions guidelines for the Grand Challenge.

Step 1 - submission of non-binding intent for participation. The goal of this submission is to initiate the contact between the DEBS Challenge organizers and solution authors. The submission of the intent should be in form of an abstract and done via the EasyChair (<https://easychair.org/conferences/?conf=debs2016>) choosing the Grand Challenge Track. There is no deadline for this submission. **Please note that not registering your non-binding intent might result in missing of the website update notifications.**

Step 2 - 15th of January - Evaluation platform will be made available to registered participants. Registered participants will receive detailed instructions about how to provide the virtual machine with their solutions and receive the evaluation results.

Step 3 - 3rd of April - deadline for submissions of the DEBS Challenge solutions. The evaluation platform will be closed at this time. Final results will be sent to each participant (each participant's results refer to their last submitted solution).

Step 4 - 10th of April - notification of solution acceptance. All accepted solutions will be ranked based on their performance, the best solutions will be invited to submit either a long or short paper (based on their ranking) discussing their approach. In addition, the best solutions will get a presentation slot in the DEBS Grand Challenge Track at the conference or the opportunity to present a poster in the poster session (dependent on their ranking).

Step 5 - 1st of May - provision of the Camera Ready for the accepted solutions.

Step 6 - June 20 - June 24 - Announcement of the DEBS 2016 Grand Challenge winner during the conference. DEBS Grand Challenge winner will be the solution achieving the shortest answer time for both queries. In addition, an “Audience Award” will be granted to the solution which appears most interesting to the participants on the DEBS conference. It will be determined through a vote on site and factor in the presentations during the DEBS Grand Challenge track or poster session.

Important Dates

15 th December	Challenge announcement
15 th January	Evaluation platform made available to registered participants
15 17 th April	Evaluation period ends
22 nd April	Notification of acceptance
7 th May	DEBS Camera Ready deadline

Evaluation

The evaluation of the Grand Challenge will be done using VirtualBox virtual machines. The virtual machine should be setup with 4 cores and 8 GB of RAM. An automated evaluation platform will be used for this year's challenge. The evaluation platform will be made available on the 15th of January. Registered participants will be able to run their code at the evaluation platform and receive the performance results multiple times (detailed instructions about the submission process will be shared with participants starting from the 15th of January once they register to the challenge via easychair). The evaluation platform will be closed on the 3rd of April (challenge deadline).

For both queries it is assumed that the result is calculated in a streaming fashion - i.e.: (1) solutions must not make use of any pre-calculated information, such as indices and (2) result streams must be updated continuously.

The ranking of the submissions will be done using the total duration of the execution and the average delay per stream. Specifically, we will rank the solutions (a) by the total execution time as well as (b) by the average latency (combining the average results from both streams in a sum). The average latency should be computed as the average of each output tuple latency, the latter measured as the difference between the system clock taken when logging an output tuple and the system clock taken when the processing of the input tuple triggering the result starts. The final ranking will be determined by a score that is the sum of the rank for execution time and the rank for delay. For solutions with the same overall score we will favor the solution with the lower execution time.

FAQ

In order to access the Frequently Asked Questions please [follow the link to the Google Drive FAQ document](#).

Test cases

In order to access the test cases please follow the link for query [Q1](#) and [Q2](#).

Grand Challenge Co-Chairs

Vincenzo Gulisano, Chalmers University of Technology, Sweden

Zbigniew Jerzak, SAP SE, Germany

Spyros Voulgaris, Vrije Universiteit Amsterdam, The Netherlands



[_ \(https://web.archive.org/web/20170301140202/http://www.sigmod.org/\).](https://web.archive.org/web/20170301140202/http://www.sigmod.org/)



[_ \(https://web.archive.org/web/20170301140202/http://www.sigsoft.org/\).](https://web.archive.org/web/20170301140202/http://www.sigsoft.org/)

SIGMOD



PODS

San Francisco 2016



[\(scholarship.html\)](#)

DEBS will be held back-to-back with SIGMOD'2016 @ San Francisco, CA (June 26 - July 1)
(<https://web.archive.org/web/20170301140202/http://sigmod2016.org/>)



(<https://web.archive.org/web/20170301140202/https://www.ics.uci.edu/>)

Important Dates*

Research Track, Abstract Submission	February 22nd February 29 th , 2016

Research Track, Full Paper Submission	February 27th March 7 th , 2016
Industry Track, Full Paper Submission	February 27th March 7 th , 2016

* For all the dates, see [here \(important-dates.html\)](#).

DEBS on Social Media



Tweets by [@ACM_DEBS](https://web.archive.org/web/20170301140202/https://twitter.com/ACM_DEBS) (https://web.archive.org/web/20170301140202/https://twitter.com/ACM_DEBS)

DEBS 2016 Sponsors

Grand Challenge Sponsor:



(<https://web.archive.org/web/20170301140202/http://www.wso2.com/>)

© Copyright 2016 The 10th ACM International Conference on Distributed Event Based Systems (DEBS 2016)