The DEBS 2016 Grand Challenge

Vincenzo Gulisano
Chalmers University of Technology
Hörsalsvägen 11
41296 Gothenburg, Sweden
vincenzo.gulisano@chalmers.se

Zbigniew Jerzak
SAP SE
Münzstraße 15
10178 Berlin, Germany
zbigniew.jerzak@sap.com

Spyros Voulgaris
Vrije Universiteit Amsterdam
De Boelelaan 1081A
1081HV Amsterdam, The Netherlands
spyros@cs.vu.nl

Holger Ziekow
Hochschule Furtwangen
Robert-Gerwig-Platz 1
78120 Furtwangen, Germany
zie@hs-furtwangen.de

ABSTRACT

Pending...

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed Applications*

General Terms

Algorithms, Design

Keywords

event processing, streaming, utilities, geo-spatial

1. INTRODUCTION

The ACM DEBS 2016 Grand Challenge is the sixth in a series [4, 9, 5, 6] 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 [2]. DEBS 2016 Grand Challenge takes up the general scenario from the 2014 SIGMOD Programming Contest [1], however, in contrasts 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.

DEBS'15, June 29 – July 3, 2015, OSLO, Norway.

http://dx.doi.org/10.1145/2675743.2772598.

2. DATA

The data for the 2016 Grand Challenge is organized in four separate streams, each provided as a text file. The first input stream indicates when two users enter a "friendship" relationship – see Table 1 and Listing 1. The first input stream file name is *friendships.dat*.

Table 1: The set of attributes used in the *friendships.dat* input file

Attribute	Description
ts	timestamp indicating when a friendship was established
user_id_1	id of one of the users
user_id_2	id of the other user

Listing 1: First line from the *friendships.dat* file – one attribute per line of listing

```
1 XXX
2 YYY
3 ZZZ
4 PLEASE ADD ACTUAL DATA HERE
```

The second input stream indicates when a users creates a new post – see Table 2 and Listing 2. The second input stream file name is *posts.dat*.

The third input stream indicates when a users commnets on a post – see Table 3 and Listing 3. The third input stream file name is *comments.dat*.

Each of the data files is sorted chronologically based on the timestamp (*ts*) attribute.

Table 2: The set of attributes used in the *posts.dat* input file

Attribute	Description
ts	timestamp indicating when a post was created
post_id	unique id of the post
user_id	unique id of the user who created the post
post	string containing the post's content
user	string containing the user name of the post creator

Listing 2: First line from the *posts.dat* file – one attribute per line of listing

```
1 XXX
2 YYY
3 ZZZ
4 PLEASE ADD ACTUAL DATA HERE
```

3. FREQUENT ROUTES QUERY

The goal of the frequent routes query is to find the top ten most frequent routes during the last 30 minutes. A route is represented by a starting grid cell and an ending grid cell. All routes completed within the last 30 minutes are considered for the query. The output query results must be updated whenever any of the 10 most frequent routes changes. The output format for the result stream is shown in Listing 4.

Where pickup_datetime, dropoff_datetime are the timestamps of the trip report that resulted in an update of the result stream, start_cell_id_X the starting cell of the X^{th}-most frequent route, end_cell_id_X the ending cell of the X^{th}-most frequent route. If less than 10 routes can be identified within the last 30 minutes, then NULL is to be output for all routes that lack data.

The attribute "delay" captures the time delay between reading the input event that triggered the output and the time when the output is produced. Participants must determine the delay using the current system time right after reading the input and right before writing the output. This attribute will be used in the evaluation of the submission.

The cells for this query are squares of 500 meters by 500 meters. The cell grid starts with cell 1.1, located at 41.474937, -74.913585 in Barryville. The coordinate 41.474937, -74.913585 marks the center of the first cell. Cell numbers increase towards the east and south, with the shift to east being the first and the shift to south the second component of the cell, i.e., cell 3.7 is 2 cells east and 6 cells south of cell 1.1. The overall grid expands 150km south and 150km east from cell 1.1 with the cell 300.300 being the last cell in the grid. All trips starting or ending outside this area are treated as outliers and must not be considered in the computation.

4. PROFITABLE AREAS QUERY

Table 3: The set of attributes used in the *comments.dat* input file

Attribute	Description
ts	timestamp indicating when a comment was created
comment_id	unique id of the comment
user_id	unique id of the user who created the comment
comment	string containing the comment's content
user	string containing the user name of the comment creator
comment_replied	id of the comment being commented (-1 if this is a comment to a post)
post_commented	id of the post being commented (-1 if this is a comment to a comment)

Listing 3: First line from the *comments.dat* file – one attribute per line of listing

```
1 XXX
2 YYY
3 ZZZ
4 PLEASE ADD ACTUAL DATA HERE
```

The goal of profitable areas query is to identify areas that are currently most profitable for taxi drivers. The profitability of an area is determined by dividing the area profit by the number of empty taxis in that area within the last 15 minutes. The profit that originates from an area is computed by calculating the median fare including tip for trips that started in the area and ended within the last 15 minutes. The number of empty taxis in an area is the sum of taxis that had a drop-off location in that area less than 30 minutes ago and had no following pickup yet.

The result stream of the query must list the ten most profitable areas in the format presented in Listing 5.

with attribute names containing cell_id_1 corresponding to the most profitable cell and attribute containing cell_id_10 corresponding to the 10^{th} most profitable cell. If less than 10 cells were identified within the last 30 minutes, then NULL is to be returned for all cells that lack data. Query results must be updated whenever any of the 10 most profitable areas change. The pickup_datetime and dropoff_datetime in the output are the timestamps of the trip report that triggered the change.

The attribute "delay" captures the time delay between reading the input event that triggered the output and the time when the output is produced. Participants must determine the delay using the current system time right after reading the input and right before writing the output, i.e., including the serialization and deserialization time but excluding the disk IO time.

Profitable areas query uses the same numbering scheme as for frequent routes query, however it uses a different

Listing 4: Output format for the frequent routes query

```
pickup_datetime
dropoff_datetime
start_cell_id_1
end_cell_id_1
...
start_cell_id_10
end_cell_id_10
delay
```

Listing 5: Output format for the profitable areas query

```
pickup_datetime
dropoff_datetime
profitable_cell_id_1
empty_taxies_in_cell_id_1
median_profit_in_cell_id_1
profitability_of_cell_1
...
profitable_cell_id_10
empty_taxies_in_cell_id_10
median_profit_in_cell_id_10
profitability_of_cell_10
delay
```

resolution. In this query one should assume a cell size of 250m X 250m, i.e., the area to be considered spans from cell 1.1 to cell 600.600.

5. ADDITIONAL REMARKS

6. LICENSE

7. ACKNOWLEDGEMENTS

The DEBS Grand Challenge Organizing Committee would like to explicitly thank WSO2 (http://wso2.com) for sponsoring the DEBS 2016 Grand Challenge prize and the LDBC Council (http://www.ldbcouncil.org) for their help in preparing the test data set.

8. REFERENCES

[1] Curtis E. Dyreson, Feifei Li, and M. Tamer Özsu, editors. *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*. ACM, 2014.

[2] Orri Erling, Alex Averbuch, Josep-Lluis Larriba-Pey, Hassan Chafi, Andrey Gubichev, Arnau Prat-Pérez, Minh-Duc Pham, and Peter A. Boncz. The LDBC social network benchmark: Interactive workload. In Timos K. Sellis, Susan B. Davidson, and Zachary G. Ives, editors, *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 619–630. ACM, 2015.

[3] Ihab F Ilyas, George Beskales, and Mohamed A Soliman. A survey of top-k query processing techniques in relational database systems. *ACM Computing Surveys (CSUR)*, 40(4):11, 2008.

[4] Zbigniew Jerzak, Thomas Heinze, Matthias Fehr, Daniel Gröber, Raik Hartung, and Nenad Stojanovic. The DEBS 2012 grand challenge. In François Bry, Adrian Paschke, Patrick Th. Eugster, Christof Fetzer, and Andreas Behrend, editors, *Proceedings of the Sixth ACM International Conference on Distributed Event-Based Systems, DEBS 2012, Berlin, Germany, July 16-20, 2012*, pages 393–398. ACM, 2012.

[5] Zbigniew Jerzak and Holger Ziekow. The DEBS 2014 grand challenge. In Umesh Bellur and Ravi Kothari, editors, *The 8th ACM International Conference on Distributed Event-Based Systems, DEBS '14, Mumbai, India, May 26-29, 2014*, pages 266–269. ACM, 2014.

[6] Zbigniew Jerzak and Holger Ziekow. The DEBS 2015 Grand Challenge. In Frank Eliassen and Roman Vitenberg, editors, *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems, DEBS '15, Oslo, Norway, June 29 - July 3, 2015*, pages 266–268. ACM, 2015.

[7] Nikos Mamoulis, Huiping Cao, George Kollios, Marios Hadjieleftheriou, Yufei Tao, and David W Cheung. Mining, indexing, and querying historical spatiotemporal data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 236–245. ACM, 2004.

[8] Mohamed F Mokbel, Xiaopeing Xiong, and Walid G Aref. Sina: Scalable incremental processing of continuous queries in spatio-temporal databases. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 623–634. ACM, 2004.

[9] Christopher Mutschler, Holger Ziekow, and Zbigniew Jerzak. The DEBS 2013 grand challenge. In Sharma Chakravarthy, Susan Darling Urban, Peter Pietzuch, and Elke A. Rundensteiner, editors, *The 7th ACM International Conference on Distributed Event-Based Systems, DEBS '13, Arlington, TX, USA - June 29 - July 03, 2013*, pages 289–294. ACM, 2013.

[10] Donna J Peuquet and Niu Duan. An event-based spatiotemporal data model (estdm) for temporal analysis of geographical data. *International journal of geographical information systems*, 9(1):7–24, 1995.