Specification of different Graphalytics Competitions

Ahmed Musaafir
Vrije Universiteit Amsterdam
a.a.m.musaafir@vu.nl

Wing Lung Ngai
Vrije Universiteit Amsterdam
w.l.ngai@vu.nl

Tim Hegeman
TU Delft
t.m.hegeman@student.tudelft.com

Alexandru Uta
Vrije Universiteit Amsterdam
A.Uta@vu.nl

Alexandru Iosup
Vrije Universiteit Amsterdam
A.Iosup@vu.nl

1 INTRODUCTION

Responding to increasingly larger and more diverse graphs, and the need to analyze them, both industry and academia are developing and tuning graph analysis software platforms. Already tens of such platforms exist, among them PowerGraph [3], GraphX [4], PGX.D [6], and GraphMat [5], but their performance is often difficult to compare. Moreover, the random, skewed, and correlated access patterns of graph analysis, caused by the complex interaction between input datasets and applications processing them, expose new bottlenecks on the hardware level, as hinted at by the large differences between Top500[1] and Graph500[2] rankings. Approaches such as Graph500 do not capture the complex behavior of graph workloads when comparing graph-analysis platforms. To overcome this issue, in this paper we introduce our approach at achieving an in-depth, fair, and objective comparison of graph processing platforms.

To perform in-depth performance evaluation and analysis, we have developed the *LDBC Graphalytics* [2] benchmark suite, which has been adopted by companies and researchers in the field. Central to Graphalytics is the idea of objective comparison between graph-processing platforms by controlling the key parameters, using (i) a comprehensive suite of real-world algorithms, and synthetic and real-world datasets, (ii) an extensive set of metrics to quantify system performance, scalability (we quantify horizontal/vertical and weak/strong scalability), and robustness (we quantify failures and performance variability), and (iii) a renewal process to curate and possibly change the algorithms, datasets, and gathered metrics. It is symptomatic that other de-facto standard benchmarks in the field do not have the properties (i)–(iii).

Based on the LDBC Graphalytics suite, in this paper, we introduce the *Graphalytics Global Competition*, a platform which enables the fair and objective comparison of graph-analysis platforms. It makes it possible for graph processing platform vendors to benchmark their own systems, evaluate their performance in a fine grained manner, and compare the results to other similar platforms. To allow vendors to perform such comparisons in a realistic manner, a systematic and periodic evaluation is required. Our competition framework allows vendors to upload their results and participate in competitions to compare the platforms according to a specified scoring system per competition. At this moment, the Graphalytics Global Competition consists of two types of competitions where vendors can participate in: (1) the *Graphalytics* competition and (2) the *relative-performance* competition. The results of a few competitions have been currently published and can be accessed online through the Graphalytics website[3].

In this document, we explain both types of competitions.

2 GRAPHALYTICS GLOBAL COMPETITION

After completing a benchmark using Graphalytics, an output file containing the results is generated. This output file can be submitted on the Graphalytics Global Competition website to a competition. After reviewal by the Graphalytics team, the benchmark can be accepted to the competition. In this section, we briefly describe two types of competitions that currently exist on Graphalytics: the Tournament Competition and the Relative-Performance Competition.

2.1 Tournament Competition

The Graphalytics competition method focuses on two primary scores: *performance* and *cost-performance*.

Furthermore, this competition follows a tournament-based approach where the system performance is ranked by means of competitive tournaments [7]. Generally, a Round-Robin pair-wise tournament [1] (from hereon, *tournament*) of p participants involves a balanced set of (pair-wise) comparisons between the results of each pair of participants; if there are c criteria to compare the participants, there will be $\frac{1}{2} \times c \times p(p-1)$ pair-wise comparisons. In a pair-wise comparison, a predefined amount of points (often, 1 or 3) is given to the better (*winner*) participant, from the pair. It is also common to give zero points to the worse (*loser*) participant, from the pair, and to split the points between participants with equal performance. Similar tournaments have been used for decades in chess competitions, in professional sports leagues such as (European and American) football, etc.

The Tournament competition consists of a number of *matches*, where each match represents a type of experiment that focuses on a specific performance characteristic that is common across all systems, for example, the EVPS of the BFS algorithm on a Datagen dataset. Each match consists of a set of instances, with the *tournament score* being for each system the sum of *instance scores* accumulated by the platform across all matches in which it participates. Each *instance* is a head-to-head comparison between two systems, for example, comparing the EVPS of any algorithm-dataset

[1]https://top500.org
[2]https://graph500.org
[3]Accessible through http://beta.graphalytics.org/

ICT.OPEN18, June 2018, Houston
2018. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

for the pair (Giraph, GraphX): the winner receives 1 point, the loser 0 points, and a draw rewards each platform with 0.5 points each.

2.2 Algorithm

Algorithm 1 shows a simplified example of how the scores are obtained from a set of cells for the tournament competition. A set of cells is provided as an argument to the *runMatchesGraphalytics* function. Each cell corresponds to a single platform and dataset, for a particular metric and algorithm. An example of cells can be seen in Table 1 where a set of cells represents the values in a single column for a dataset.

Algorithm 1: Match cells (Graphalytics)

```
1 Function runMatchesGraphalytics(cells, metric)
      Data: (1) cells: a set where each cell is the (result) value
            from a given platform and dataset, corresponding to
            an algorithm. (2) metric: The current metric of the
            cells.
      Result: A score corresponding to the value of each cell
              entry.
2     for i <cells.length do
3         for j in range(i+j, cells.length) do
4             cellA = cells[i]
5             cellB = cells[j]
6             if metric == 'evps' then
7                 matchHigherValueIsBetter(cellA, cellB)
8             else
9                 matchLowerValueIsBetter(cellA, cellB)
```

Each cell is compared to one another (*lines 2-3*). Depending on whether the given metric represents a value that is better when it is higher (*line 6* for EVPS) or lower (*line 8* for PPP, TL, TP, TM), the appropriate method is called to compare the values of the two cells. Algorithm 2 shows the *matchLowerValueIsBetter* function, which assigns a winning score to the cell where the value is lower or an equal score when the two values (of the two cells) are equal.

The function *matchHigherValueIsBetter* is not shown in this document, but is similar to the *matchLowerValueIsBetter* function. The only difference is that a higher value of a cell would receive the winning score.

2.2.1 Exceptional cases.

Zero points are given to cells that do not have a value. Cells that do have a value and are being compared against a cell without value, would receive a point.

2.3 Example

An example of a published Tournament competition can be currently viewed online. A raw snippet of some of the results is shown in table 1. To further elaborate on the scoring scheme based on the example table: each platform in the column of dataset *Graph500-25* is compared to one another. Since GraphX does not beat GraphMat, but does beat PowerGraph, it gets a score of 1. GraphMat is

Algorithm 2: Match lower value is better

```
Function matchLowerValueIsBetter(cellA, cellB)
    Data: cellA and cellB: both used to compare against one
          another.
    Result: Updated score for cellA and cellB.
    valA = cellA.value
    valB = cellB.value
    if valA <valB then
        cellA.score += 1
        cellB.score += 0
    else if valA >valB then
        cellA.score += 0
        cellB.score += 1
    else
        cellA.score += 0.5
        cellB.score += 0.5
```

faster than both GraphX and PowerGraph and gets a score of 2. PowerGraph doesn't beat any platform and gets a score of 0. If we repeat this process for each dataset (column), we can compute the the total score for each platform.

Table 1: An example snippet of the Tournament competition for BFS. The scores have been prefixed by +.

System name	Total score	Graph500-25	Datagen-8_5-Fb
GraphX	3	281.98 (s) *+1*	142.79 (s) *+2*
GraphMat	2	44.80 (s) *+2*	- (s) *+0*
PowerGraph	1	405.19 (s) *+0*	221.83 (s) *+1*

Note that a platform value (and therefore its score) could not have been calculated due to several reasons, such as a time-out error. In this case, a score of 0 is assigned.

2.4 Relative Performance Competition

The drawback of the method used in the Tournament competition is that it does not take into account the scale of how much a system is better or worse than another system. To tackle this problem, the relative-performance competition can be used. A detailed description and example is provided below.

2.5 Algorithm

Initially, the minimum and maximum values of the cells are obtained (*line 2-3*). The condition on *line 4* checks whether a value in the cells is missing by determining if the minimum value is not a negative value. If that is the case (*line 6*), a similar structure (*line 8, 10*) as the Graphalytics competition is used to determine whether the type of metric for a given value is better if the value is lower or higher. In the case where a lower value is better (as in table 2), take the lowest observed value per column *min*, take the multiplicative inverse (i.e. $1/value$) and map the resulting values in range $[0, 1/min]$ to scores in range $[0, 1]$. For values where a higher one is better (e.g., for the EVPS metric), take the highest observed value *max* and map the values in range $[0, max]$ to scores in range $[0, 1]$.

Algorithm 3: Match cells (Relative-performance)

```
Function runMatchesRelativePerformance(cells, metric)
    Data: (1) cells: a set of cells where each cell is the (result)
          value from a given platform and dataset,
          corresponding to a particular metric and algorithm.
          (2) metric: the current metric of the cells.
    Result: A score (between 0 and 1) corresponding to the
            value of each cell entry.
    minVal = min(cells)
    maxVal = max(cells)
    if minVal <0 then
        scaleToZeroOneMissingVal(cells, metric, maxVal)
    else
        for cell in cells do
            if metric == 'evps' then
                cell.score = 1 / cell.value / (1 / maxVal)
            else
                cell.score = 1 / cell.value / (1 / minVal)
```

2.5.1 Exceptional cases.

Similar as with the tournament competition, zero points are given to cells that do not have a value. Cells that do have a value and are being compared against a cell without value, would receive a point.

2.6 Example

An example of this type of competition is shown in table 2.

To further elaborate on the scoring scheme: consider the values for dataset (column) *Graph500-25*. The fastest performer for this dataset is GraphMat, which has a value of 44.80. We take the multiplicative inverse of this value: $1/44.80 = 0.022$. Then we map each value *in this particular column* for the other datasets according to: $1/value/0.022$. E.g., for GraphX, this would become: $1/281.98/(1/44.8) = 0.16$. By repeating this process for every column (and taking the best from each column as well), we can calculate the total score.

Table 2: An example of the Relative-Performance competition for BFS. The scores have been prefixed by +.

System name	Total score	Graph500-25	Datagen-8_5-Fb
GraphX	1.16	281.98 (s) *+0.16*	142.79 (s) *+1.00*
GraphMat	1.00	44.80 (s) *+1.00*	- (s) *+0.00*
PowerGraph	0.75	405.19 (s) *+0.11*	221.83 (s) *+0.64*

3 CONCLUSION

The LDBC Graphalytics is an industrial-grade benchmark for graph analysis platform. The main goal of Graphalytics is to enable the fair and objective comparison of graph-analysis platforms. Users of this benchmark suite can submit their benchmark results on the Graphalytics Global Competition website to a competition. At this moment, the website contains two types of published competitions: (1) Tournament competition, which compares each platform with one another in a pairwise manner, and the (2) Relative-Performance competition, which emphasizes the relative differences between the performance of the best-performing systems. After review by the Graphalytics team, the result can be seen and compared along other graph processing platforms.

REFERENCES

[1] H. A. David. 1960. The Method of Paired Comparisons. In Fifth Conference on the Design of Experiments in Army Research Development and Testing. US Army Office of Ordnance Research, 1–17.

[2] A. Iosup et al. 2016. LDBC Graphalytics: A benchmark for large-scale graph analysis on parallel and distributed platforms. (2016).

[3] J.E. Gonzalez et al. 2012. PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs. In OSDI.

[4] J.E. Gonzalez et al. 2014. GraphX: Graph Processing in a Distributed Dataflow Framework. In OSDI.

[5] N. Sundaram et al. 2015. GraphMat: High performance graph analytics made productive. (2015).

[6] S. Hong et al. 2015. PGX.D: a fast distributed graph processing engine. In SC.

[7] Louis Leon Thurstone. 1927. Psychophysical Analysis. American Journal of Psychology 38 (1927), 368–89.