УДК 004.65
DOI: 10.25559/SITITO.14.201801.126-137

BENCHMARKING BIG SPATIAL DATA PROCESSING FRAMEWORKS

Anastasia A. Garaeva, Airat D. Kabirov, Olga V. Tikhonova

Kazan National Research Technical University named after A.N. Tupolev -KAI, Kazan, Russia

Abstract

Today, the processing of large amounts of spatial data in distributed systems plays a crucial role in many areas of our life. Large data are often unstructured, and special algorithms are required for its processing. One of the methods for analyzing large data is a spatial analysis. The source of large data in this case is often the geographical information system.

In this article, a benchmark is considered to evaluate the frameworks that work with such data. Also, the evaluation results of three frameworks according to developed benchmark are presented: GeoSpark, STARK, SpecialSpark. In the course of this paper, we considered a benchmark of two types: macrobenchmark and microbenchmark.

In the paper, testing of topological predicates on various topological data is also considered. The comparison was made using the DE-9IM model. This model is used to determine the types of topological relationships, such as intersection, equality, etc. The main problem of comparing the data frameworks was that not all of them support the operations of the selected model, which influenced the formation of scenarios for the microbenchmark and macrobenchmark, since it was impossible to compare all the DE-9IM items.

Keywords

Big data; microbenchmark; macrobenchmark; spatial data; topological relationships.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СПОСОБОВ ОБРАБОТКИ БОЛЬШИХ ПРОСТРАНСТВЕННЫХ ДАННЫХ

А.А. Гараева, А.Д. Кабиров, О.В. Тихонова

Казанский национальный исследовательский технический университет им. А.Н. Туполева - КАИ, г. Казань, Россия

About the authors:

Anastasia A. Garaeva, post-graduate student, Department Applied Mathematics and Computer science, Kazan National Research Technical University named after A.N. Tupolev –KAI (10 K. Marx St., Kazan 420111, Tatarstan, Russia); ORCID: http://orcid.org/0000-0001-8205-0324, nastya.9315@gmail.com

Airat D. Kabirov, post-graduate student, Department Security Systems, Kazan National Research Technical University named after A.N. Tupolev –KAI (10 K. Marx St., Kazan 420111, Tatarstan, Russia); ORCID: http://orcid.org/0000-0002-7262-263X, airiboy@mail.ru

Olga V. Tikhonova, post-graduate student, Department Applied Mathematics and Computer science, Kazan National Research Technical University named after A.N. Tupolev –KAI (10 K. Marx St., Kazan, 420111, Tatarstan, Russia); ORCID: http://orcid.org/0000-0002-5638-8011, olgishkaaa@gmail.com

Аннотация

Обработка больших объемов пространственно-распределенных данных играет важную роль во многих областях современной жизни. Так называемые «большие данные» - Big Data - часто неструктурированы, и для их обработки необходимо применять специальные алгоритмы. Одним из методов анализа больших данных является пространственный анализ. Источником больших данных в этом случае часто являются географические информационные системы - ГИС.

В статье приведены результаты оценки эффективности трех структур, которые работают с такими данными: GeoSpark, STARK, SpecialSpark на основе стандартного эталонного теста. Использованы макро- и микротесты. Также приводятся итоги сравнения топологических предикатов для различных топологических данных. Сравнение проводилось с использованием стандартной модели DE-9IM, используемой для определения типов топологических отношений, таких как пересечение, равенство и т. д. Основная проблема сравнения структур данных заключалась в том, что не все они поддерживают операции выбранной модели. Это повлияло на разработку сценариев сравнительного анализа, поскольку в модели DE-9IM не все элементы возможно сравнивать между собой.

Ключевые слова

Большие данные; микротесты; макротесты; пространственные данные; топологические отношения.

1. Introduction

Big data is a huge amount of heterogeneous and rapidly flowing digital information that cannot be processed by traditional tools. Big data analytics can help to see hidden patterns, invisible to the limited human perception. This gives unprecedented opportunities to optimize all areas of life: public administration, medicine, telecommunications, finance, transport, production education and so on [1-4]. Big data is often unstructured, and its processing requires special algorithms. One of the methods of big data analysis is spatial analysis. It is set partly borrowed from statistics techniques for the analysis of spatial data – topology of locality, geographical coordinates and objects geometry. The source of big data in this case often is geographic information system.

At the moment, a few frameworks allow you to work with spatial data. If you need to analyze the performance of such frameworks, you will use a benchmark. Non-profit corporation TPC developed for many systems data-centric benchmark standards that would easily complete the task, but for spatial data standards do not exist. Therefore, the main aim of our work is to develop a benchmark of frameworks, that work with spatial data. Our benchmark will consist of two parts: microbencmark and macrobenchmark. For this work were selected three spatial data framework: GeoSpark, SpatialSpark and STARK. According to the results of our work, we will be able to carry out the comparative characteristic of these frameworks and identify their advantages and disadvantages.

2. Basic knowledge

The practice of benchmarking is widespread. It is used in different fields: finance systems, computer graphic, audio systems, database processing. It allows selecting the most appropriate implementation among several choices. In our work, we consider two types of benchmarks: microbenchmark and microbenchmark. The microbenchmark is the testing of primitive topological relationship like spatial join or spatial analysis operations with different predicates [5-11]. The macrobenchmark is a sequence of topological relationship queries, which simulate the workload of certain system.

2.1. Microbenchmark

Nowadays there are several models of microbenchmark: 4-Intersection model, 9-intersection model and The Dimensionally Extended 9 Intersection Model (DE-9IM). Egenhofer and Herring developed the 4-intersection and 9-intersection models. The 4-intersection deals with two objects. Each of them is divided into interior and boundary. Therefore, this model analyzes connections between these objects. The 9-intersection adds to the 4-intersection the intersections with the two objects' complements [12].

However, models, which were considered before, had some problems with embedding to DBMS query language. Thus, Clementini and Di Felice extended the 9-Intersection model to the Dimensionally Extended 9 Intersection Model. DE-9IM is a mathematical approach that defines the pairwise spatial relationship between geometries of different types and dimensions. This model expresses spatial relationships among all types of geometry as pairwise intersections of their interior, boundary, and exterior with consideration for the dimension of the resulting intersections [13]. Dimensionally Extended Nine-Intersection Model (DE-9IM) proposes the relationships: Equals, Disjoint, Intersects, Touches, Crosses, Within, Contains and Overlaps.

During our work we used DE-9IM model for micro benchmarking, it has been adopted by the Open Geospatial Consortium.

2.2. Macrobenchmark

The macrobenchmark simulates work of real application system that is why we need to make a significant amount of stress on the underlying system. Thus, the work of benchmark indicates the system's performance.

In the article [14] six macrobenchmark scenarios are considered: Geocoding, Reverse Geocoding, Map Search and Browsing, Flood Risk Analysis, Land Information Management, Toxic Spill. This kind of models are examples of real life situations, which address typical user necessity.

In our work, we implement two models for macrobenchmark: Map Search and Browsing, Landscape analysis.

2.3. Frameworks

We chose three frameworks, which work with spatial data: GeoSpark [15-17], SpatialSpark [18] and STARK [18]. All of them have open source code, which is available on GitHub. These frameworks are based on Apache Spark. It is a software framework with open source code for implementing the distributed processing of unstructured and semi-structured data, which is included in the Hadoop ecosystem of projects. The project provides APIs for the languages Java, Scala, Python, R. Originally written in Scala, subsequently added a substantial part of the Java code to enable writing programs directly in Java. For our benchmark, we use Scala programming language. Spark use the ***resilient distributed dataset*** (RDD) concept. This is a fault-tolerant collection of elements that can be operated on in parallel.

All of mentioned frameworks can be used in an interactive Spark shell (Scala supported) by running spark-shell command with pre-compiled jar. Another way of using it is to create self-contained Spark application (Scala and Java supported), set dependencies (for example Maven dependencies in Eclipse or using sbt) and create jar file, which could be then used via spark-submit command.

2.4. GeoSpark

GeoSpark is a cluster computing system, which process large-scale spatial data. GeoSpark works with Spatial Resilient Distributed Dataset that efficiently load, process, and analyze large-scale spatial data across machines. The GeoSpark provides APIs for users to make work with it easier. Geospark has Java API, that does not integrate well (using special RDDs, which could be only of the certain type) into the Spark API [19-23].

Furthermore, GeoSpark SRDD allows to process with large-scale spatial datasets using spatial queries (spatial join, spatial aggregation, and spatial co-location). First, geometrical objects transfer to the Spatial RDD layer. After that user can apply spatial query processing operations. Next, Spatial Query Processing Layer decides what happened with object-relational tuples: how it will stored, how will be accessed and indexed. This process occurs in memory cluster. Finally, result of spatial query returns to the user.

2.5. SpatialSpark

Spatial Spark is a framework, which processes spatial queries directly on Spark [24-28]. This is a high-performance in-memory Big Data system developed using Scala and Java. SpatialSpark works as Spark library for spatial extension to process large scale spatial join operations [16].

2.6. STARK

STARK is a framework, which is based on Apache Spark. It supports spatial data types and operations. It also supports Scala language. STARK is a very convenient tool, because it works with standard RDDs of Spark. It is easy to use STARK in self-contained applications as well as directly in

Spark shell, because all functions are described well and examples of using different operations are given.

3. System design

3.1. Microbenchmark

Main concept of our benchmark is to test different topological predicates on different types of data. As we have said before we use Dimensionally Extended Nine-Intersection Model (DE-9IM). This model is used for defining our topological relation queries. In the Table 1 we describe the possible pairwise topological relationships among polygon, line and point according to the DE-9IM.

In Table 1:

Y means predicates, included in microbenchmark.

NA means not applicable

Equals means that predicates contains and within are executed consequently. We applied within, contains and touches to different type of topological relations to compare how type of data influences on performance of the system.

Intersection of geometries means non-disjoint geometries. For equally dimensional geometries intersect can be expressed by overlaps. For non-equally dimensional geometries intersect can be expressed by crosses. That is why for each possible combination (point – point and so on) we chose one of these predicates.

Table 1. Topological relations in dimensionally extended 9-intersection model

	Polygon And Polygon	Line And Line	Line And Polygon	Point And Polygon	Point And Line	Point And Point
Equals	Y		NA	NA	NA	Y
Disjoint		Y			Y	
Intersect				Y		Y
Touches	Y	Y	Y	Y	Y	NA
Crosses	NA		Y	NA	NA	NA
Overlaps	Y		NA	NA	NA	NA
Within	Y			Y	Y	NA
Contains		Y	Y	Y	Y	NA

For each pair of geometric objects times of computing is different. Table 2 summarizes all the queries that are included in the micro benchmark.

3.2. Macrobenchmark

Main concept of our macrobenchmark is to evaluate system performance under real world workload. In our macrobenchmark we include two use cases:

Use case 1 «Landscape analyses» [16-18]: I am a person in certain geoposition and I would like to know about some military installations within certain distance from me.

After that, I would like to know in which US states these filtered military installations are:

We need to filter military installations within certain distance from our geoposition.

We need to join result of previous step with dataset of US states in order to obtain names of the states, that contain filtered installations.

Use case 2 « Map search and browsing»: This use case contains following steps:

- We are in certain geoposition.
- We need to define in which polygon (country or state in real world) we are.
- After that, we need to join this found polygon with polygons, containing information about borders of land and water objects and with points, containing coordinates of different landmarks on land and on water. Thus, we get full information about geographic region, where we are.

Table 2. Microbenchmark queries

Operation	Description	Query
Equals	Polygon equals Polygon	Find the polygons that are spatially equal to other polygons
Equals	Point equals Point	Find the points that are spatially equal to other points

Disjoint	Line disjoint Line	Find the lines that are spatially disjoint from other lines
Disjoint	Point disjoint Line	Find the points that are spatially disjoint from other lines
Intersect	Point intersect Polygon	Find the points that intersect polygons
Intersect	Point intersect Point	Find the points that intersect points
Touches	Polygon touches Polygon	Find the polygons that touch polygons
Touches	Line touches Line	Find the lines that touch lines
Touches	Line touches Polygon	Find the lines that touch polygons
Touches	Point touches Polygon	Find the points that touch polygons
Touches	Point touches Line	Find the points that touch lines
Crosses	Line crosses Polygon	Find the lines that cross polygons
Overlaps	Polygon overlaps Polygon	Find the polygons that overlap other polygons
Within	Polygon within Polygon	Find the polygons that are within other polygons
Within	Point within Polygon	Find the points that are inside the polygons
Within	Point within Line	Find the points that are inside the lines
Contains	Line contains Line	Find the lines that contain other lines
Contains	Line contains Polygon	Find the lines that contain other polygons
Contains	Point contains Polygon	Find the points that contain other polygons
Contains	Point contains Line	Find the points that contain other lines

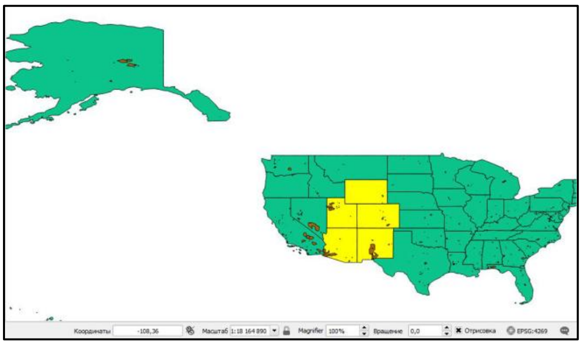

Figure 1. Visualization of use case#1

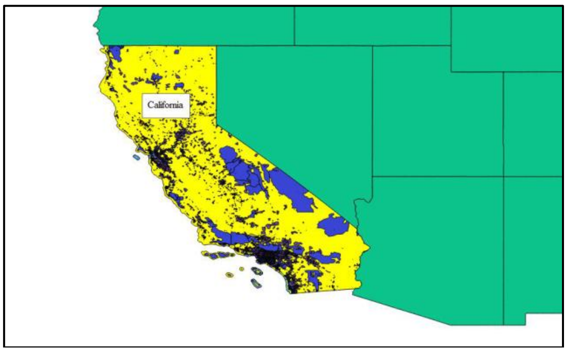

Figure 2. Visualization of use case#2

Table 3. Feature comparison

	GeoSpark	SpatialSpark	STARK
Spatial Partitioning	Yes	Yes	Yes
Indexing	Yes	Yes	Yes
Filter			
Contains	Yes	Yes	Yes
ContainedBy	No	Yes	Yes
Intersects	No	Yes	Yes
WithinDistance	No	Yes	Yes
Join			
Contains	Yes	Yes	Yes
ContainedBy	No	Yes	Yes
Intersects	No	Yes	Yes
WithinDistance	No	Yes	Yes

For our microbenchmark we consider following relation:

Equals means consequently applied within (containedBy) and contains operations. SpatialSpark, GeoSpark and STARK supports both these operations, thus it is possible to realize this query.

Disjoint means that objects do not intersect. SpatialSpark and STARK support intersect predicate, thus it is possible to realize this query in these frameworks.

It is not possible to implement following operation called "touches" in evaluated frameworks:

Crosses and overlaps are specific case of intersect relation.Overlap compares two geometries of the same dimension and returns t (TRUE) if their intersection set results in a geometry different from both but of the same dimension. Cross returns t (TRUE) if the intersection results in a geometry whose dimension is one less than the maximum dimension of the two source geometries and the intersection set is interior to both source geometries.

Evaluated frameworks only support simple intersect operation, thus we cannot ensure execution of such condition. So, none of evaluated frameworks support neither cross nor overlaps operation.

Our macrobenchmark is supported in SpatialSpark and STARK framework. Neither use case 1 nor use case 2 is supported by GeoSpark due to the following reasons:

In first use case we use filter operation with withinDistance predicate (not supported in GeoSpark); in second use case we use join operation with intersect predicate (also not supported by GeoSpark).

Evaluation results

In our work we use spark-submit with following parameters:

```
--master yarn --num-executors 32 --executor-cores 2 --executor-memory 7G
```

All our queries run on the cluster of our department, which has 16 machines (the table 4 shows information about 1 machine in a cluster).

4. Performance evaluation

4.1. Microbenchmark performance evaluation

We represent some of results of our queries on three big spatial data frameworks.

We show execution time of filtering operation with predicate contains for all mentioned frameworks. In this query we give certain query point and dataset of polygons. As the result, we expect to have a number of polygons, that contain this point. We do this operation with different number of points.

We show execution time of filtering operation with predicate containedBy for all mentioned frameworks. In this query we give certain query polygon and dataset of points. As the result, we expect to have number of points, that are contained by this polygon. We do this operation with different size of polygons.

We show execution time of join operation with predicate contains for all mentioned frameworks. In this query we give dataset of points and dataset of polygons.

As the result, we expect to have number of pairs of joined points and polygons. We do this operation with different number of points and polygons.

Our datasets of points and polygons for following queries were obtained from data generator provided by our department.

It generates points and polygons with uniform distribution. We created datasets of points of size 10000 (494KB), 50000 (2511KB), 100000 (5033KB), 250000 (13MB). For polygons datasets of size 10000 (19MB), 50000 (96MB), 100000 (192MB), 250000 (480MB).

Table 4. Cluster parameters

Parameter name	Value
CPU	Intel Core i5-3470S @ 2.90 Ghz (4 logical cores)
Memory	16 GB DDR3 1600 Mhz
Storage	HDD 1Tb

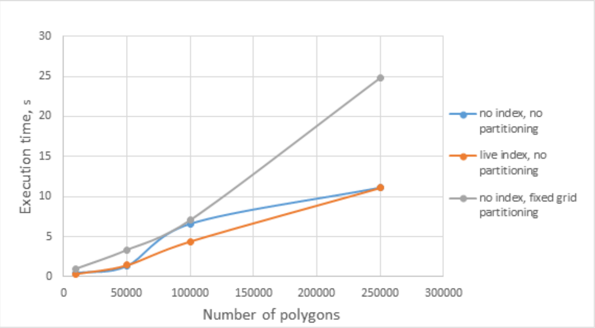

Figure 3. STARK Filter Contains

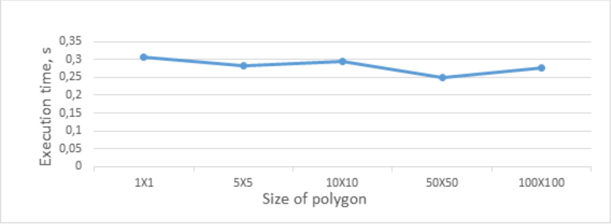

Figure 4. STARK Filter ContainedBy

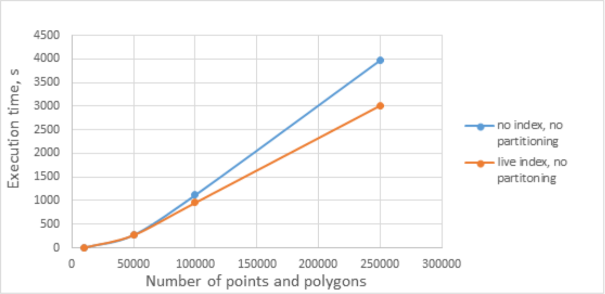

Figure 5. STARK Join Contains

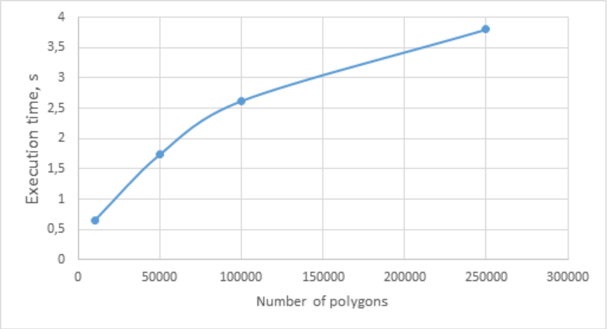

Figure 6. SpatialSpark Filter Contains

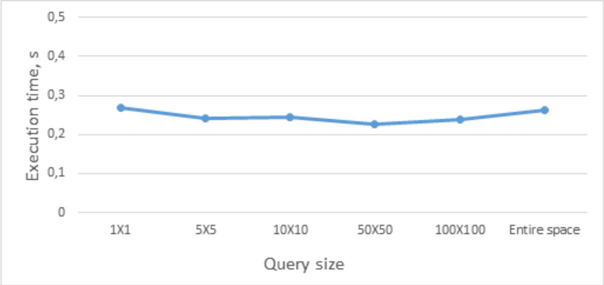

Figure 7. SpatialSpark Filter ContainedBy

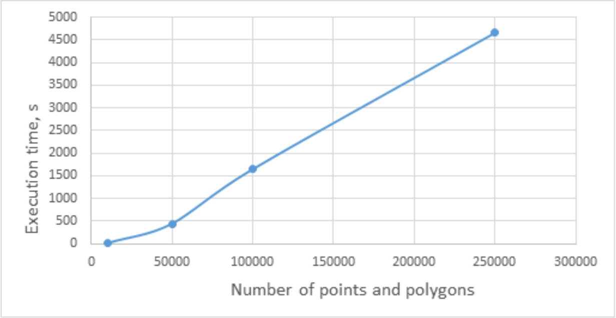

Figure 8. SpatialSpark Join Contains

STARK

As we can see, the fastest results for filtering on Contains predicate are with live index and no partitioning (results are at the figure 3).

In Filter ContainedBy the size of polygon doesn't affect too much the performance of the system (results are at the figure 4).

The operation Join Contains work faster if we have not got index and partitioning (results are at the figure 5).

Spatial Spark

As we can see at the figure 6, the number of polygons obviously affect the performance of framework for filtering on Contains predicate.

In Filter ContainedBy the size of polygon doesn't affect too much the performance of the system (figure 7).

In this query, the number of points and polygons affect performance of framework.

GeoSpark

In case of filter Contains, egualgrid, rtree and hilbert grid partitioning almost have the same results, the difference is in milliseconds. If we don't apply partitioning the execution time is better (figure 9).

We can see that in case of Voronoi partitioning if we choose different type of indexes the execution time doesn't change significantly (figure 10). In case of increasing the polygon's size the execution time does not change sufficiently. In case we have rtree index, but different types of partitioning, the time of execution depends on type of partitioning. In case we have Voronoi type of partitioning the execution time depends on type of indexes. The best execution time we have in case of lack of index.

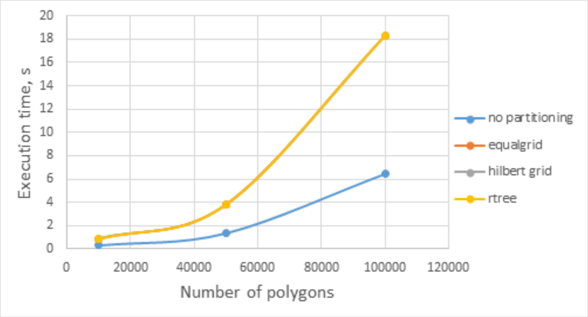

Figure 9. GeoSpark Filter Contains. No index, different types of partitioning

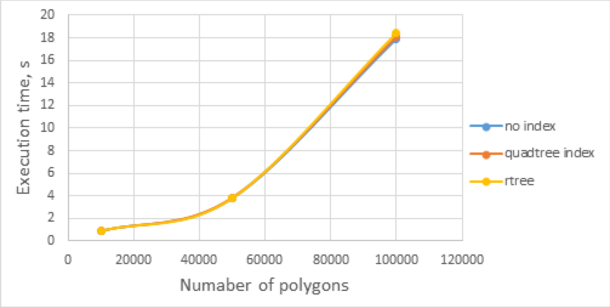

Figure 10. GeoSpark Filter Contains. Voronoi partitioning, different types of indexes

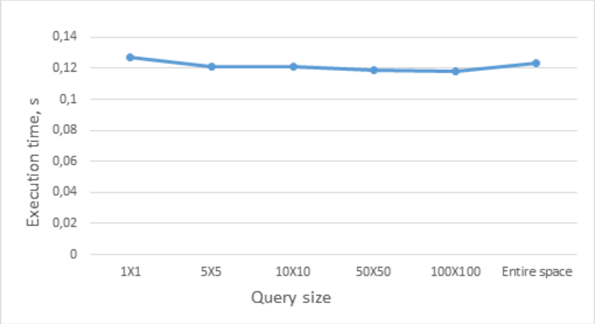

Figure 11. GeoSpark Filter ContainedBy

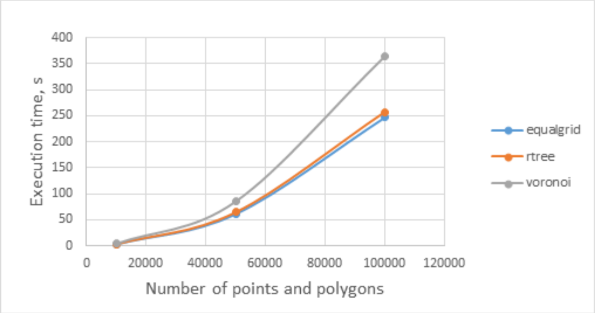

Figure 12. GeoSpark Filter Contains. Rtree index, different types of partitioning

Cross-framework comparison

Further we compare performance of three frameworks on a same picture. We took best results for each framework with datasets of 100000 points and 100000 polygons. As we can see GeoSpark at the figure 14 with quadtree index and rtree partitioning showed best performance.

After that, we compared the performance of three frameworks on a same graph. We took best results for each framework with datasets of 100000 points and 100000 polygons. As we can see at the figure 15, SpatialSpark with no indexing and no partitioning showed best performance.

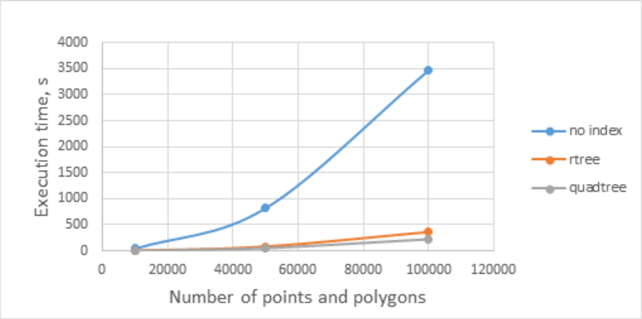

Figure 13. GeoSpark Filter Contains. Voronoi partitioning, different types of indexes

4.2. Macrobenchmark performance evaluation

We represent results of our macrobencmarks on SpatialSpark and STARK.

Fistly, evaluation was performed on datasets of 100000 points (5MB) and 100000 polygons (192MB) obtained from data generator to check

frameworks behavior on relatively big data (uniformly distributed).

As we can see at the figure 16, the best one is STARK without index and partitioning. For second use case, the best one is STARK with live indexing (results are represented at the figure 17).

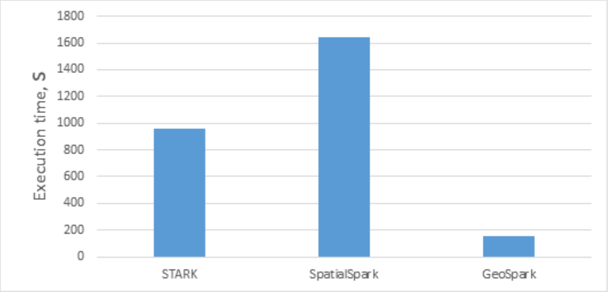

Figure 14. Comparison of different frameworks with operation Join Contains

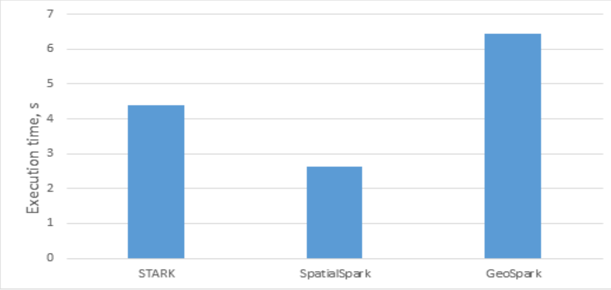

Figure 15. Comparison of different frameworks with operation Filter Contains

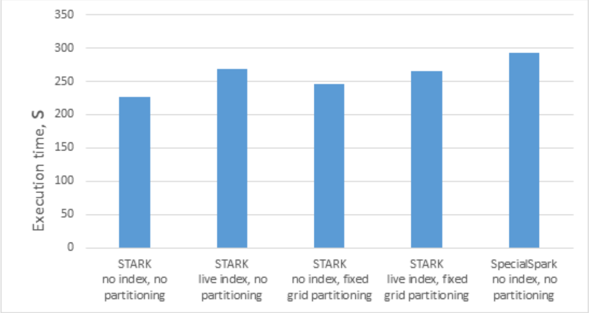

Figure 16.Use case #1

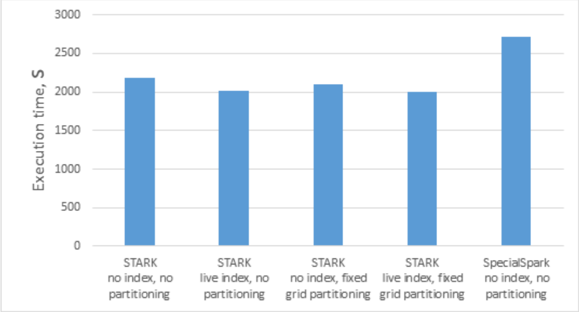

Figure 17. Use case #2

5. Conclusion

As a result of our work, we introduced micro- and macrobenchmarks for big data spatial frameworks. Applied these benchmarks to three certain frameworks SpatialSpark, GeoSpark and STARK. In our opinion, STARK is the most convenient to process big spatial data. SpatialSpark lacks of documentation, there is no option of live indexing and it is hard to apply it with your own code. GeoSpark has many opportunities for partitioning and indexing, but it supports limited number of spatial relation operations (its filtering and join operators have already predefined predicate contains).

We also found that SpatialSpark works better with Filter Contains. GeoSpark works better with Join Contains.

REFERENCES

[1] Zakharova I., Kuzenkov O., Soldatenko I., Yazenin A., Novikova S., Medvedeva S., Chukhnov A. Using SEFI framework for modernization of requirements system for mathematical education in Russia. *Proceedings of the 44th SEFI Annual Conference 2016 - Engineering Education on Top of the World: Industry University Cooperation (SEFI 2016)*. 12-15 September 2016, Tampere, Finland. 15 p. Available at: http://sefibenvwh.cluster023.hosting.ovh.net/wp-content/uploads/2017/09/zakharova-using-sefi-framework-for-modernization-of-requirements-system-for-mathematical-education-155.pdf (accessed 10.02.18)

[2] Soldatenko I., Kuzenkov O., Zakharova I., Balandin D., Biryukov R., Kuzenkova G., Yazenin A., Novikova S. Modernization of math-related courses in engineering education in Russia based on best practices in European and Russian universities. *Proceedings of the 44th SEFI Annual Conference 2016 - Engineering Education on Top of the World: Industry University Cooperation (SEFI 2016)*. 12-15 September 2016, Tampere, Finland. 16 pp. Available at: http://sefibenvwh.cluster023.hosting.ovh.net/wp-content/uploads/2017/09/soldatenko-modernization-of-math-related-courses-in-engineering-education-in-russia-based-133.pdf (accessed 10.02.18)

[3] Zakharova I., Kuzenkov O. Experience in implementing the requirements of the educational and professional standards in the field of ICT in the Russian education. *Modern information technologies and IT-education*. 2016; 12(3)-1:17-31. Available at: https://elibrary.ru/item.asp?id=27411971 (accessed 10.02.18) (In Russian)

[4] Bedny A., Erushkina L., Kuzenkov O. Modernising educational programmes in ICT based on the Tuning methodology. *Tuning Journal for Higher Education*. 2014; 1(2):387-404. Available at: http://www.tuningjournal.org/article/view/32/20 (accessed 10.02.18)

[5] Yoo J.S., Boulware D., Kimmey D. A Parallel Spatial Co-location Mining Algorithm Based on MapReduce. *Proceedings of 2014 IEEE International Congress on Big Data*. 27 June - 2 July 2014, Anchorage, AK, USA, 2014. p. 25-31. DOI: https://doi.org/10.1109/BigData.Congress.2014.14

[6] Refaye E.M., Hegazy O. Parallel Co-location Pattern Mining Discovery: Constraint Neighborhood Approach. *International Journal of Applied Engineering Research*. 2016; 11(1):586-591. Available at: https://www.ripublication.com/ijaer16/ijaerv11n1_86.pdf (accessed 10.02.18)

[7] Shekhar S., Zhang P., Huang Y., Vatsavai R.R. Trends in Spatial Data Mining. In: Kargupta H., Joshi A., Sivakumar K., Yesha Y. (eds.), Data Mining: Next Generation Challenges and Future Directions. Cambridge, USA: AAAI/MIT Press, 2003.

[8] Shekhar S., Huang Y. Co-location Rules Mining: A Summary of Results. In: Jensen C.S., Schneider M., Seeger B., Tsotras V.J. (eds.), Advances in Spatial and Temporal Databases – 2001. 7th International Symposium (SSTD 2001), LNCS 2121, July 12–15, 2001, Redondo Beach, CA, USA, 2001. p. 236-256.

[9] Huang Y., Zhang P. On The Relationships Between Clustering and Spatial Co-location Pattern Mining. *Proceedings of 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*. 13-15 Nov. 2006, Arlington, VA, USA, 2006. DOI: https://doi.org/10.1109/ICTAI.2006.91

[10] Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann Publishers, 2001. 31 p.

[11] You S., Zhang J., Gruenwald L. Large-scale spatial join query processing in Cloud. *Proceedings of 31st IEEE International Conference on Data Engineering Workshops (ICDEW 2015)*. 13-17 April 2015, Seoul, South Korea, 2015. DOI: https://doi.org/10.1109/ICDEW.2015.7129541

[12] Egenhofer M., Sharma J., Mark D. A critical comparison of the 4-intersection and 9-intersection models for spatial relations: Formal analysis. *Proceedings of 1993 the AutoCarto Conference, Minneapolis*, MN, USA, 30 October – 1 November 1993. p. 1 - 12. Available at: https://pdfs.semanticscholar.org/4c7c/eeaf64f969f5bb05f07f81aa51259c246d18.pdf (accessed 10.02.18)

[13] McKenney M., Schneider M. Topological Relationships between Map Geometries. In: Haritsa J.R., Kotagiri R., Pudi V. (eds) Database Systems for Advanced Applications. DASFAA 2008. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2008. Vol. 4947. p. 110-125. DOI: https://doi.org/10.1007/978-3-540-78568-2_11

[14] Ray S., Simion B., Brown A.D. Jackpine: A benchmark to evaluate spatial database performance. *Proceedings of 2011 IEEE 27th International Conference on Data Engineering (ICDE 2011)*. 11-16 April 2011, Hannover, Germany, 2011. p. 1139 – 1150. DOI: https://doi.org/10.1109/ICDE.2011.5767929

[15] Yu J., Wu J., Sarwat M. GeoSpark: A Cluster Computing Framework for Processing Large-Scale Spatial Data. GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems (Vol. 03-06 November 2015). Association for Computing Machinery, 2015. p. 70. DOI: https://doi.org/10.1145/2820783.2820860

[16] Eldawy A., Mokbel M.F. SpatialHadoop: A MapReduce Framework for Spatial Data. *Proceedings of 2015 IEEE 31st International Conference on Data Engineering (ICDE 2015)*. 13-17 April 2015, Seoul, South Korea, 2015. DOI: https://doi.org/10.1109/ICDE.2015.7113382

[17] Yu J., Wu J., Sarwat M. A demonstration of GeoSpark: A cluster computing framework for processing big spatial data. *Proceedings of 2016 IEEE 32nd International Conference on Data Engineering (ICDE 2016)*. Institute of Electrical and Electronics Engineers Inc. 2016. p. 1410-1413. DOI: https://doi.org/10.1109/ICDE.2016.7498357

[18] You S., Gorloo K. (Eds.) Big Spatial Data Processing using Spark. Available at: http://simin.me/projects/spatialspark/ (accessed 10.02.18)

[19] Hagedorn S., Goetze P., Sattler K-U. Big Spatial Data Processing Frameworks: Feature and Performance Evaluation. Advances in Database Technology - EDBT 2017, 20th International Conference on Extending Database Technology, March 21-24, Venice, Italy, 2017. p. 490-493. Available at: https://openproceedings.org/2017/conf/edbt/paper-344.pdf (accessed 10.02.18)

[20] Rigaux P., Scholl M., Voisard A. Spatial Databases – With Application to GIS. Morgan Kaufmann Publishers. 2002. 410 p.

[21] Eldawy A., Mokbel M.F. Pigeon: A spatial MapReduce language. *Proceedings of 2014 IEEE 30th International Conference on Data Engineering (ICDE 2014)*. IEEE Computer Society, 2014. p. 1242-1245. DOI: https://doi.org/10.1109/ICDE.2014.6816751

[22] Whitman R.T., Park M.B., Ambrose S.M., Hoel E.G. Spatial indexing and analytics on Hadoop. *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'14)*. ACM, New York, NY, USA, 2014. p. 73-82. DOI: http://dx.doi.org/10.1145/2666310.2666387

[23] Hagedorn S., Sattler K.-U. Piglet: Interactive and Platform Transparent Analytics for RDF & Dynamic Data. *Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2016. 4 p. DOI: https://doi.org/10.1145/2872518.2890530

[24] Kresse W., Danko D.M. (Eds.) Springer handbook of geographic information (1. Ed. ed.). Berlin: Springer, 2010. p. 82–83. ISBN 9783540726807

[25] Shekhar S., Chawla S. Spatial Databases: A Tour. Prentice Hall, 2003. 262 p.

[26] ESRI Press. ESRI Press titles include Modeling Our World: The ESRI Guide to Geodatabase Design, and Designing Geodatabases: Case Studies in GIS Data Modeling, Ben Franklin Award winner, PMA, The Independent Book Publishers Association, 2005.

[27] Rigaux P., Scholl M., Voisard A. Spatial Databases – With Application to GIS. Morgan Kaufmann Publishers. 2001. 410 p.

[28] Amirian P., Basiri A., Winstanley A. Evaluation of Data Management Systems for Geospatial Big Data. In: Murgante B. et al. (eds) Computational Science and Its Applications – ICCSA 2014. ICCSA 2014. Lecture Notes in Computer Science, Springer, Cham, 2014. Vol. 8583. DOI: https://doi.org/10.1007/978-3-319-09156-3_47

Submitted 20.12.2017; Revised 10.02.2018; Published 30.03.2018.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Using SEFI framework for modernization of requirements system for mathematical education in Russia / I. Zakharova [et al.] // Proceedings of the 44th SEFI Annual Conference 2016 - Engineering Education on Top of the World: Industry University Cooperation (SEFI 2016). 12-15 September 2016, Tampere, Finland. 15 pp. URL:

http://sefibenvwh.cluster023.hosting.ovh.net/wp-content/uploads/2017/09/zakharova-using-sefi-framework-for-modernization-of-requirements-system-for-mathematical-education-155.pdf (дата обращения: 10.02.18).

[2] Modernization of math-related courses in engineering education in Russia based on best practices in European and Russian universities / I. Soldatenko [et al.] // Proceedings of the 44th SEFI Annual Conference 2016 - Engineering Education on Top of the World: Industry University Cooperation (SEFI 2016). 12-15 September 2016, Tampere, Finland. 16 p. URL: http://sefibenvwh.cluster023.hosting.ovh.net/wp-content/uploads/2017/09/soldatenko-modernization-of-math-related-courses-in-engineering-education-in-russia-based-133.pdf (дата обращения: 10.02.18).

[3] *Захарова И.В., Кузенков О.А.* Опыт реализаций требований образовательных и профессиональных стандартов в области ИКТ в российском образовании // Современные информационные технологии и ИТ-образование. 2016. Т. 12, № 3-1. С. 17-31. URL: https://elibrary.ru/item.asp?id=27411971 (дата обращения: 10.02.18).

[4] *Bedny A., Erushkina L., Kuzenkov O.* Modernising educational programmes in ICT based on the Tuning methodology // Tuning Journal for Higher Education. 2014. Vol. 1, no. 2. Pp. 387-404. URL: http://www.tuningjournal.org/article/view/32/20 (дата обращения: 10.02.18).

[5] A Parallel Spatial Co-location Mining Algorithm Based on MapReduce / J.S. Yoo, D. Boulware, D. Kimmey // Proceedings of 2014 IEEE International Congress on Big Data. 27 June - 2 July 2014, Anchorage, AK, USA, 2014. Pp. 25-31. DOI: https://doi.org/10.1109/BigData.Congress.2014.14

[6] *Refaye E.M., Hegazy O.* Parallel Co-location Pattern Mining Discovery: Constraint Neighborhood Approach // International Journal of Applied Engineering Research. 2016. Vol. 11, no. 1. Pp. 586-591. URL: https://www.ripublication.com/ijaer16/ijaerv11n1_86.pdf (дата обращения: 10.02.18).

[7] *Shekhar S., Zhang P., Huang Y., Vatsavai R.R.* Trends in Spatial Data Mining. In: Kargupta H., Joshi A., Sivakumar K., Yesha Y. (eds.), Data Mining: Next Generation Challenges and Future Directions. Cambridge, USA: AAAI/MIT Press, 2003.

[8] *Shekhar S., Huang Y.* Co-location Rules Mining: A Summary of Results. In: Jensen C.S., Schneider M., Seeger B., Tsotras V.J. (eds.), Advances in Spatial and Temporal Databases – 2001. 7th International Symposium (SSTD 2001), LNCS 2121, July 12–15, 2001, Redondo Beach, CA, USA, 2001. Pp. 236-256.

[9] On The Relationships Between Clustering and Spatial Co-location Pattern Mining / Y. Huang, P. Zhang // Proceedings of 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06). 13-15 Nov. 2006, Arlington, VA, USA, 2006. DOI: https://doi.org/10.1109/ICTAI.2006.91

[10] *Han J., Kamber M., Pei J.* Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann Publishers, 2001. 31 p.

[11] Large-scale spatial join query processing in Cloud / S. You, J. Zhang, L. Gruenwald // Proceedings of 31st IEEE International Conference on Data Engineering Workshops (ICDEW 2015). 13-17 April 2015, Seoul, South Korea, 2015. DOI: https://doi.org/10.1109/ICDEW.2015.7129541

[12] A critical comparison of the 4-intersection and 9-intersection models for spatial relations: Formal analysis / M. Egenhofer, J. Sharma, D. Mark // Proceedings of 1993 the AutoCarto Conference, Minneapolis, MN, USA, 30 October – 1 November 1993. Pp. 1 - 12. URL: https://pdfs.semanticscholar.org/4c7c/eeaf64f969f5bb05f07f81aa51259c246d18.pdf (дата обращения: 10.02.18).

[13] *McKenney M., Schneider M.* Topological Relationships between Map Geometries. In: Haritsa J.R., Kotagiri R., Pudi V. (eds) Database Systems for Advanced Applications. DASFAA 2008. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2008. Vol. 4947. Pp. 110-125. DOI: https://doi.org/10.1007/978-3-540-78568-2_11

[14] Jackpine: A benchmark to evaluate spatial database performance / S. Ray, B. Simion, A.D. Brown // Proceedings of 2011 IEEE 27th International Conference on Data Engineering (ICDE 2011). 11-16 April 2011, Hannover, Germany, 2011. Pp. 1139 – 1150. DOI: https://doi.org/10.1109/ICDE.2011.5767929

[15] GeoSpark: A Cluster Computing Framework for Processing Large-Scale Spatial Data / J. Yu, J. Wu, M. Sarwat // GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems (Vol. 03-06 November 2015). Association for Computing Machinery, 2015. Pp. 70. DOI: https://doi.org/10.1145/2820783.2820860

[16] SpatialHadoop: A MapReduce Framework for Spatial Data / A. Eldawy, M.F. Mokbel // Proceedings of 2015 IEEE 31st International Conference on Data Engineering (ICDE 2015). 13-17 April 2015, Seoul, South Korea, 2015. DOI: https://doi.org/10.1109/ICDE.2015.7113382

[17] A demonstration of GeoSpark: A cluster computing framework for processing big spatial data / J. Yu, J. Wu, M. Sarwat // Proceedings of 2016 IEEE 32nd International Conference on Data Engineering (ICDE 2016). Institute of Electrical and Electronics Engineers Inc. 2016. Pp. 1410-1413. DOI: https://doi.org/10.1109/ICDE.2016.7498357

[18] *You S., Gorloo K.* (Eds.) Big Spatial Data Processing using Spark. URL: http://simin.me/projects/spatialspark/ (дата обращения: 10.02.18).

[19] *Hagedorn S., Goetze P., Sattler K-U.* Big Spatial Data Processing Frameworks: Feature and Performance Evaluation. Advances in Database Technology - EDBT 2017, 20th International Conference on Extending Database Technology, March 21-24, Venice, Italy, 2017. Pp. 490-493. URL: https://openproceedings.org/2017/conf/edbt/paper-344.pdf (дата обращения: 10.02.18).

[20] *Rigaux P., Scholl M., Voisard A.* Spatial Databases – With Application to GIS. Morgan Kaufmann Publishers. 2002. 410 pp.

[21] Pigeon: A spatial MapReduce language / A. Eldawy, M.F. Mokbel // Proceedings of 2014 IEEE 30th International Conference on Data Engineering (ICDE 2014). IEEE Computer Society, 2014. Pp. 1242-1245. DOI: https://doi.org/10.1109/ICDE.2014.6816751

[22] Spatial indexing and analytics on Hadoop / R.T. Whitman, M.B. Park, S.M. Ambrose, E.G. Hoel // Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'14). ACM, New York, NY, USA, 2014. Pp. 73-82. DOI: http://dx.doi.org/10.1145/2666310.2666387

[23] Piglet: Interactive and Platform Transparent Analytics for RDF & Dynamic Data / S. Hagedorn, K.-U. Sattler // Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2016. 4 p. DOI: https://doi.org/10.1145/2872518.2890530

[24] *Kresse W., Danko D.M.* (Eds.) Springer handbook of geographic information (1. Ed. ed.). Berlin: Springer, 2010. Pp. 82–83, ISBN 9783540726807

[25] *Shekhar S., Chawla S.* Spatial Databases: A Tour. Prentice Hall, 2003. 262 p.

[26] ESRI Press. ESRI Press titles include Modeling Our World: The ESRI Guide to Geodatabase Design, and Designing Geodatabases: Case Studies in GIS Data Modeling, Ben Franklin Award winner, PMA, The Independent Book Publishers Association, 2005.

[27] *Rigaux P., Scholl M., Voisard A.* Spatial Databases – With Application to GIS. Morgan Kaufmann Publishers. 2001. 410 p.

[28] *Amirian P., Basiri A., Winstanley A.* Evaluation of Data Management Systems for Geospatial Big Data. In: Murgante B. et al. (eds) Computational Science and Its Applications – ICCSA 2014. ICCSA 2014. Lecture Notes in Computer Science, Springer, Cham, 2014. Vol. 8583. DOI: https://doi.org/10.1007/978-3-319-09156-3_47

Поступила 20.12.2017; принята к публикации 10.02.2018; опубликована онлайн 30.03.2018.

Об авторах:

Гараева Анастасия Алексеевна, аспирант кафедры прикладная математика и информатика, Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ (420111, Россия, Татарстан, г. Казань, ул. К. Маркса, д. 10); ORCID: http://orcid.org/0000-0001-8205-0324, nastya.9315@gmail.com

Кабиров Айрат Дмитриевич, аспирант кафедры системы информационной безопасности, Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ (420111, Россия, Татарстан, г. Казань, ул. К. Маркса, д. 10); ORCID: http://orcid.org/0000-0002-7262-263X, airiboy@mail.ru

Тихонова Ольга Викторовна, аспирант кафедры прикладная математика и информатика, Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ (420111, Россия, Татарстан, г. Казань, ул. К. Маркса, д. 10); ORCID: http://orcid.org/0000-0002-5638-8011