

Original citation:

Triantafillou, Peter (2018) Towards intelligent distributed data systems for scalable efficient and accurate analytics. In: 38 IEEE International Conference on Distributed Computing Systems, ICDCS, Vienna, Austria, 2-5 Jul 2018. Published in: 38 IEEE International Conference on Distributed Computing Systems, ICDCS (In Press)

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/101785>

Copyright and reuse:

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher's statement:

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting /republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

A note on versions:

The version presented here may differ from the published version or, version of record, if you wish to cite this item you are advised to consult the publisher's version. Please see the 'permanent WRAP url' above for details on accessing the published version and note that access may require a subscription.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk

Towards Intelligent Distributed Data Systems for Scalable Efficient and Accurate Analytics

Peter Triantafillou
Department of Computer Science,
University of Warwick
UK
Email: p.triantafillou@warwick.ac.uk

Abstract—Large analytics tasks are currently executed over Big Data Analytics Stacks (BDASs) which comprise a number of distributed systems as layers for back-end storage management, resource management, distributed/parallel execution frameworks, etc. In the current state of the art, the processing of analytical queries is too expensive, accessing large numbers of data server nodes where data is stored, crunching and transferring large volumes of data and thus consuming too many system resources, taking too much time, and failing scalability desiderata.

With this vision paper we wish to push the research envelope, offering a drastically different view of analytics processing in the big data era. The radical new idea is to process analytics tasks employing learned models of data and queries, instead of accessing any base data – we call this data-less big data analytics processing. We put forward the basic principles for designing the next generation intelligent data system infrastructures realizing this new analytics-processing paradigm and present a number of specific research challenges that will take us closer to realizing the vision, which are based on the harmonic symbiosis of statistical and machine learning models with traditional system techniques. We offer a plausible research program that can address said challenges and offers preliminary ideas towards their solution. En route, we describe initial successes we have had recently with achieving scalability, efficiency, and accuracy for specific analytical tasks, substantiating the potential of the new paradigm.

I. INTRODUCTION

We have all heard the heavy declarations surrounding (big) data science, such as "data is the new oil", "data analytics is the new combustion engine", etc. Behind such hype, lie two fundamental truths: First, humans and organizations are inundated with massive data volumes, of high complexity, which seriously impede their ability to manage it and make sense of it in a time-critical fashion. Second, it has been proved that intelligent analyses of data lead to profound novel insights, which bear great benefits to organizations and individuals, for practically all facets of our lives, including health, education, public governance, scientific knowledge generation, finances, business decision-making, etc.

Current trends indicate that big, complex data science will become even bigger and more complex. As one example, earth science sensors around the globe record 100s

of thousands unique correlations [1]. As another example, we expect that by 2025 2B human genomes will have been sequenced. All trends with respect to analytics at a large variety of applications, such as health informatics, astrophysics, particle physics, social media, finance, etc. reveal the same expectations. While complex data of massive volumes become omnipresent, their analyses presents insurmountable challenges to the system infrastructures that will be called upon to process them. Referring back to human genome analysis, identifying genomic variants requires 4 orders of magnitude speedups (with 100,000 CPUs running in parallel) [2]!

The system infrastructures for big data analytics entail a number of distributed systems, organized in so-called Big Data Analytics Stacks (BDASs). A distributed storage back-end is used to store and manage data sets; this is in the form of a distributed file system, distributed SQL or NoSQL modern databases, or often a combination of these systems. Additionally, a distributed resource management system and a Big Data Engine (which implements distributed processing paradigms, such as MapReduce) are utilized for the processing of analytics tasks. Finally, a number of systems/applications are typically supported as higher layers, offering specialized functionality, such as Machine Learning libraries, graph analytics, etc.

Our recent research has revealed that the state of the art methods for fundamental analytics queries over such infrastructures leaves a lot to be desired in terms of efficiency, scalability and accuracy. Armed with this knowledge, with this paper we put forward our vision for research in large-scale systems for big data analytics, which based on our recent endeavours reveals great promises.

This vision, coined *SEA* (for Scalable, Efficient, Accurate) presents a novel set of principles, goals and objectives, and a novel research program, which (individually and especially) collectively aim to push the research envelope into distributed data systems for scalable, efficient, accurate big data science and analytics, offering orders of magnitude improvements and much-needed, novel functionality.

II. THE CONTEXT AND RELATED WORK

Scalable, efficient, and accurate (SEA) analytics is becoming increasingly important in the big data era. All research communities involved in big data science have

recognized and embraced the inherent challenges en route to achieving scalability, efficiency, and accuracy. New distributed systems, facilitating scalability with massive storage management and parallel/distributed processing have been developed. The seminal Hadoop-based efforts [3]–[5] solved key scalability problems. Realising limitations for key applications (e.g., iterative and/or interactive) newer systems such as Spark [6] and Stratosphere/Flink [7], as well as extensions to Hadoop [8], [9] were proposed. In parallel, modern scalable data systems (e.g., [10]–[12]) were contributed.

Furthermore, the need for ensuring high efficiency, in addition to scalability, was becoming evident and new systems for fast analytics [13], [14] emerged to satisfy such needs. Despite the huge success of the above efforts, it is increasingly becoming apparent that something more is required if massive volumes of data are to be analysed efficiently and scalably.

A distinct and promising research thread, within the data systems community, concerns approximate query processing (AQP). In AQP the accuracy of analyses results is sacrificed for increased efficiency, relying on data sampling (e.g., [15]) and/or data synopses (e.g., [16]). New systems (be it general-purpose such as BlinkDB [17] or for specific applications, such as for scientific analysis SciBORQ [18]) emerged. More recently, systems such as DBL/VertexDB [19], which are built on top of AQP engines, leverage them and contribute ML models so the system can learn from past behavior and gradually improve performance.

With respect to systems offering exact (and not approximate answers), new methods enriching the state of the art data systems with efficient and scalable statistical analysis, such as the Data Canopy [20] have recently been proposed. Data Canopy contributes a novel semantic cache with high success.

Interestingly, at the same time, leading researchers in ML were recognizing the scalability/efficiency constraints of known methods [21]–[23]. Michael Jordan, for instance, suggested time-data trade-offs using (efficient but) less robust algorithms, where larger datasets compensate for poorer accuracy [21], [23].

Hence, one can observe different data science research communities converging in their realizations regarding the need for new research and approaches towards scalable, efficient, and accurate analytics – the holy grail of modern big data science.

Alas, current solutions are very much lacking and, as mentioned, are predicted to fall way short of requirements [1], [2], for many data-intensive applications as, for example, earth science, particle physics, astronomy, genomics, social media analytics, etc. Even the most recent research from the data systems community, although promising, also leaves much to be desired if scalability, efficiency and accuracy are to be ensured. For example, the storage required by Data Canopy [20] (as well as by traditional data system methods based on caches and materialized views) can grow prohibitively large. Also, such efforts typically only benefit previously seen queries. Similarly, the state-of-the-art AQP Engines, like BlinkDB [17] suffer

from several disadvantages: First, sample sizes can become prohibitively large. Second, accuracy can be quite low for many tasks. Finally, engines like BlinkDB arguably place its key functionality at the wrong place within the big data analytics stack: Samples are created and maintained over a distributed file system (e.g., HDFS) and are accessed through big data infrastructures (e.g., Hive on top of Hadoop or Shark/Spark). This amounts to time-consuming and resource hungry tasks and in the end it may attain scalability, but typically at the expense of efficiency (e.g., as large a la MapReduce tasks) execute over all HDFS nodes. Finally, state of the art learning approaches, like the DBL approach [19], albeit interesting and promising, are based upon an AQP Engine, such as [17]. Thus, they inherit the aforementioned limitations in storage space and an initial (typically large) error (which they try to improve). Additionally, DBL requires large storage space to manage previous queries and answers, (e.g., maintain 1000s of answer items per executed query).

A. *Where Have we Gone Wrong?*

Figure 1 exemplifies the current state of affairs in analytical query processing over large distributed big data infrastructures.

A population of analysts submits (a large number of) analytical queries to the big data system. The data sets to be analyzed are typically massive in size and are stored, managed, and accessed using distributed big data analytics stacks (BDASs) encompassing distributed storage engines (e.g., HDFS, HBase/BigTable, Cassandra, MongoDB or a combination), distributed resource manager systems (such as Yarn, Mesos, Hadoop, etc.), and a Big Data Engine (such as Spark, Hadoop, etc.).

Processing analytical queries incurs large delays and overheads for many reasons, including the following.

- First, each analytical query passes through many layers of the BDAS, with each layer adding extra overheads at all nodes engaged in task processing.
- Second, processing is typically continued (e.g., using a MapReduce style of distributed/parallel processing) across a (potentially) large number of data nodes (e.g., running HDFS and/or HBase) over which the data is distributed.
- Third, the processing of complex tasks involves several such passes, with lots of data being transferred from node to node, etc.
- Fourth, for many tasks, there are several alternative processing methods one could employ and the system fails to itself choose or guide programmers to choose the best possible method.
- Finally, as applications for emerging large-scale geo-distributed analytics proliferate, at such global-scales current solutions’ requirements either exceed available resources or simply cost too much.

The end result of the above misses, is that task processing becomes time-consuming (inefficient), resource-hungry (costly), and unscalable (the system cannot scale as query arrival rates increase). As mentioned, the state of the art offers some improvements, e.g., using caching, AQP, and

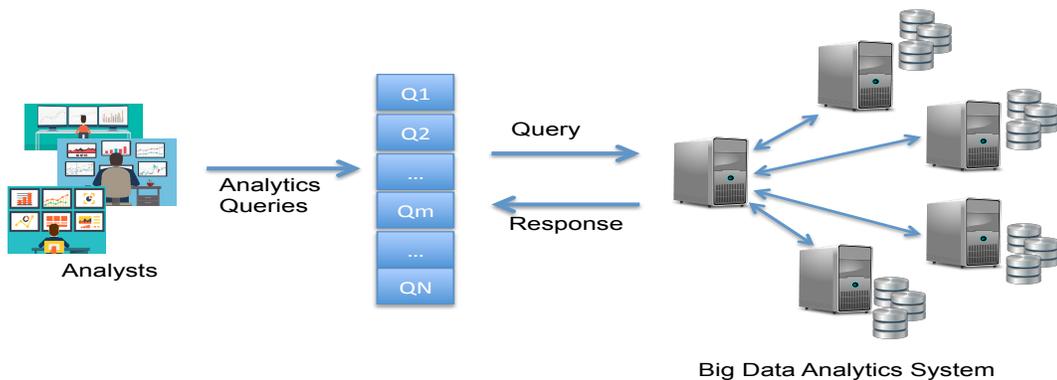


Fig. 1: Traditional Big Data Analytics Processing.

learning; but they do not go far enough – we will exemplify and detail several such shortcomings incurred when processing important types of analytical tasks throughout the remainder of the paper.

A new much bolder approach is required.

III. VISION OVERVIEW

We now first describe in more detail the types of analytical queries we focus on with this research. Subsequently, we provide the big picture of our vision.

A. The Analytics

Consider Penny, an analyst visualizing the (typically multi-dimensional) data space, which she is trying to explore and analyze. With the help of a GUI, Penny can capture a subspace of possible interest (e.g., drawing circles (hyper-spheres) or (hyper)rectangles and then issuing an analytical query over this subspace to determine if it is of interest). For example, she can use aggregation operators such as `count`, `mean`, `media`, `quartiles` etc., to derive descriptive statistics within this space (these are also referred to as roll-up operators in data warehouses and Online Analytical Processing). Alternatively, she can directly issue SQL(-like) queries, (e.g., in Hive or Pig environments implemented on top of a BDAS) involving selection-projection-join queries, with aggregations. In general, the above queries can be of different types and will consist of (a) selection operators, which identify a data subspace of interest and (b) an analytical operator over the data items within this data subspace.

A variety of selection operators important for data analytics should be considered; for example: (i) range queries, which supply a range of values for each dimension of interest, defining (hyper)-rectangles in multi-dimensional data spaces; (ii) radius queries, which supply a central point in a multi-dimensional data space and a radius, defining (hyper-)spheres, and (iii) Nearest-Neighbour queries, which select a given number of data items that are closest to a given data point, given a distance definition.

Analytics over defined subspaces should cover both descriptive statistics (e.g., aggregations) and dependence (multivariate) statistics (e.g., regressions, correlations).

Finally these analytical queries will involve in general different types of base data (e.g., SQL tables, .csv files and spreadsheets, graph data, etc.) and may in addition depend on other expensive data manipulation tasks, such as joins across tables or files.

In addition, we argue for the need of new functionality, not supported by present-day data systems and SEA methods to implement it. For example, Penny should be empowered to perform analyses based on multivariate (dependence) statistics between attributes – e.g., correlation and regression analyses, informed of model coefficients for predictive analytics and visualizations (e.g., regression coefficients) etc. These in turn may be used as building blocks for higher-level interrogations, such as “return the data subspaces where the correlation coefficient between attributes is greater than a threshold value”.

Furthermore, we argue for a new class of functionality, based on the notion of *explanations* to be defined and delivered [24]. Consider Penny receiving the answer that the population (`count`) within a data subspace is 273. Such single-scalar answers, returned by present-day data systems leave much to be desired. What is she to make of it? How would this value change if the selection operators defining the data subspace to be explored were more/less selective? Penny would have to issue a large number of queries, redefining in turn the size of the queried data subspace to gain deeper understanding. We need systems that offer rich, compact, and accurate explanations, which will accompany answers and will empower Penny to better and faster understand data analyzed data subspaces. And, approaches whereby said explanations can be derived themselves scalably and efficiently.

A key point to note is that this new functionality also benefits the system itself: Higher-level interrogations and explanations will allow analysts to understand data spaces faster, and enable their exploration without burdening the system with a large number of queries (that would otherwise be required). Therefore, as systems will be facing fewer queries, the scalability, efficiency, and accuracy desiderata are achieved indirectly!

B. The Key Paradigm

Our vision revolves around a new paradigm we coined *Data-less Big Data Analytics*. Imagine a data system, which can **process analytical queries without having to access any base data, while providing accurate answers!** This would achieve the ultimate in scalability (as query processing times become de facto insensitive to data sizes), and the ultimate in efficiency, as time consuming, resource-hungry interactions within distributed and complex big data analytics stacks are completely avoided!

Figure 2 exemplifies the central paradigm. The key idea is to develop an intelligent agent and insert it between user queries and the system. This agent develops a number of statistical machine learning (SML) models, which can learn from users the characteristics of the queried space, learn from the system the characteristics of the data-answer space per query and using them can predict with high accuracy the answer to future previously-unseen new queries.

Queries are submitted to the system as before. An initial subset of these queries are sent to the system as before in Figure 1 (with the exception that the agent 'intercepts' them and their answers). These queries are in essence treated as 'training' queries. Once the models are trained, all future queries need not access any base data and all answers are provided by the agent outside the BDAS.

But is this paradigm realistic? Can these ambitious goals be achieved? And with methods and models which themselves are efficient and scalable? And without sacrificing accuracy? And for which types of analytics tasks? And what are we to do for cases where data-less processing is not possible (e.g., for certain tasks). In the rest of this paper we start a discussion as to how to address these issues. Our vision is based on recent results, which are very encouraging. At their core, they rest upon ML advances employed within distributed data systems which alongside distributed systems techniques (e.g., indexes, statistical structures, caches, query routing, load balancing, data placement, etc.) deliver the above desiderata.

IV. SEA: PRINCIPLES, GOALS, AND OBJECTIVES

SEA is fuelled and driven by four fundamental principles, the validity of which has been verified and proven through our recent research, as will be explained. These principles are associated with goals and specific objectives, which will be delivered by a research program, which we outline later.

The first principle, P1, realizes that a human-in-the-loop approach can add the inherent intelligence of the human analysts to the models, structures, and algorithms that strive to introduce (artificial) intelligence towards scalable, efficient, and accurate analytics.

P1: Empower the human to empower the data analytics system to empower the human.

SEA will formulate a novel approach to realize this principle, viewed from a system's perspective. This rests

on the following two goals: The first goal (G1) concerns the understanding and leveraging of analyst actions vis-a-vis the system's behaviour in response to analysts' queries. There are two central objectives here. On the one hand, (O1) SEA will derive novel algorithms, structures, and models, which focus on and learn from what analysts are doing. What are their queries? Which data subspaces are they concentrating on, etc.? In essence, this will quantize the query space. At the same time, on the other hand, (O2) SEA will also develop novel models, algorithms, and structures that learn from what the system does in response to analyst queries. What are the answers to analyst queries? How were they computed? Using which data subspaces? Furthermore, SEA's third objective (O3) is to unify the results of O1 and O2, **associating specific query space quanta with methods, models, and answers used to predict results for future queries, depending on their position in the query space.**

The second equally important goal (G2) pertains to the formulation and development of novel notions of and models for explanations, which capture queried data subspaces and answers to analytical queries. The rationale here is to **not simply throw data back at analysts** in response to their queries! Instead, explain the characteristics of the queried data spaces vis-a-vis their queries. For instance, explain how query answers depend on key query parameters. Such explanations are crucial, for instance, during exploratory analytics: They will help analysts efficiently and scalably discover interesting data subspaces, exploring them while understanding them, further empowering them without the need to explicitly issue an inordinate number of specific queries to gain such understanding.

As a result, the human analyst is empowered by the system in her data analyses in a way that empowers the system to be scalable and efficient!

SEA's second principle is:

P2: Data-less big data analytics.

This principle encapsulates the key new paradigm outlined earlier. The efficiency and scalability gains of the new paradigm would obviously be dramatic, as mentioned earlier. The viability of the principle rests on leveraging known and widely accepted workload characteristics, namely that queries define overlapping data subspaces [17]–[20], [25]. Using objectives O1, O2, and O3 above, SEA essentially injects an "intelligence" in the system that achieves its SEA goals.

SEA's third goal (G3) therefore is to develop a suite of models, methods, and structures, which prove the applicability and showcase the benefits of the data-less analytics principle across various analytics tasks (query types), data formats (text, graph, tabular), and system types (from cluster-based systems, to multi-datacentre, to geo-distributed systems).

Please note that P2 puts forward a game-changing approach. It is fundamentally different than the state of the art, such as caching approaches like Data Canopy [20] and AQP approaches (e.g., stratified data-sampling based)

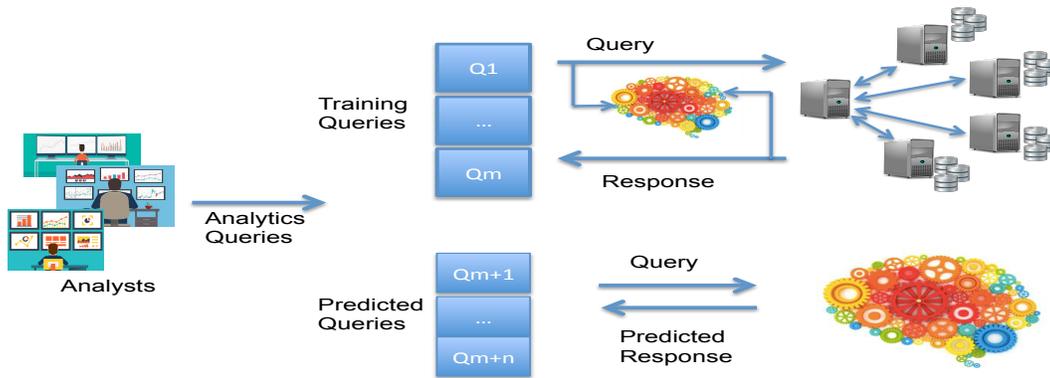


Fig. 2: The Vision for Scalable Efficient Accurate Big Data Analytics Processing.

like BlinkDB [17] and learning approaches built on top of AQP engines, like DBL [19]. As aforementioned, these make a good step forward, but are associated with several drawbacks and are not nearly as ambitious as SEA.

In addition to its obvious efficiency and scalability advantages, the salient feature of P2 is that it proposes to learn gradually, focusing only on those data subspaces, which are of interest to analysts, instead of trying to capture the whole data domain (which is time-consuming and error-prone).

Also note that data-less analytics is paramount for respecting security/privacy concerns. Typically organizations holding sensitive data, may be wary of revealing the data itself, but will nonetheless typically reveal analytical query results (over sensitive data), such as those summarizing data. Note that the basis of SEA’s ML models rests on observing queries and such analytical query answers and not the data itself.

Our group was the first to formulate and advocate this principle and obtain the first research results which showcase its validity and potential (for classes of analytics based on `count` queries [26]–[28], `average` queries and `regression` queries [29]).

SEA’s third principle is:

P3: Big-data-less big data analytics.

Depending on the analytics task at hand, it may not always be possible to apply data-less analytics processing, e.g., in cases where approximate answers are not satisfactory. For these tasks, SEA’s third principle applies with SEA’s fourth goal (G4) which aims to develop algorithms, structures, and models, which will process said analytics tasks via **surgically accessing the smallest data subset that is required to compute the answers**.

Alas, the state of the art is pervaded with approaches where key analytics tasks are processed “scalably” albeit having to access large portions (if not all) of the massive underlying base data and/or all or many data server nodes – recipes for performance disaster in the big data era! To be concrete, consider the following fundamental operators for various analytics tasks.

- Often, data analysis requires the joining of data sets spread across different tables/files, etc., and then ranking the items in the join result. State of the art solutions for this (so-called rank-join) operator were based on algorithms, which performed several MapReduce tasks (with Hadoop or Spark) over the base data sets. With our work in [30], we showed that by developing and leveraging appropriate statistical index structures, the (typically very small set of) necessary data items can be identified and only those are surgically accessed. This achieved up to 6 orders of magnitude performance improvements (in execution time, network bandwidth, and money costs)!
- k-Nearest-Neighbours (kNN) is another fundamental analytics operator. The state of the art for processing kNN queries also required processing an inordinate amount of the underlying base data either using Hadoop [31] or Spark [32]. Our work [33] introduced performance improvements of three orders of magnitude utilising novel indexes and appropriate distribution processing paradigms.
- Reaching further, subgraph matching is a fundamental operator for graph analytics. In [34], [35] novel subgraph-query semantic caches, minimized back-end stored data accesses, ensuring performance improvements up to 40X.
- The same holds for key tasks that are preparatory for analytical query answering, such as ensuring data quality. For example, our work on scalable missing value imputation [36] showed big gains in performance and scalability compared to typical BDAS/MapReduce-style processing.

Therefore, SEA’s fourth objective (O4) is to develop a suite of indexes, caches, and statistical structures, which will introduce dramatic improvements in efficiency, scalability, and money costs during analytics processing, for a wide variety of analytics tasks across multiple data formats and system types.

SEA’s fourth and final principle is:

P4: Understand the alternatives and select optimal processing methods.

Our recent research and exposure on the systems, algorithms, methods, and ML models employed for various big data analytics tasks has revealed that most tasks can be processed using a number of alternative approaches. Given the complex state-of-the-art big data analytics stacks (e.g., be it based on Hadoop or Spark) it is important to fully understand the alternatives and related trade-offs. A central question then emerges: For a specific metric (scalability, efficiency, accuracy, availability, money-costs, etc.) how should analytics tasks be processed? Using which alternative algorithms, or methods, or models?

To concretize the above, with our research on join queries over big data tables (based on [30]), we have found that sometimes applying a MapReduce based algorithm is beneficial, while other times a coordinator-cohort distributed processing model is more beneficial, depending on data distribution degrees and join selectivities. Similarly, for graph-pattern queries we have found that different algorithms [37] and different index types [38] are preferable for different graph patterns and graph Databases. The same holds for other key analytics operators, such as kNN queries, depending on the value of k and the probability distribution of the data sets (as [33] showcased).

In all above cases, the performance difference of employing different alternatives can be dramatic! P4 is associated with two goals: SEA’s fifth goal (G5) is to understand the alternatives for processing fundamental operators and their impact on key performance metrics. Objective O5 pertains to the identification of key alternatives and their comprehensive experimental evaluation. SEA’s sixth goal (G6) is to leverage what is learnt from O5. G6 is associated with objective O6, which concerns training, learning, and building optimising modules, which on-the-fly adopt the best execution method.

V. A RESEARCH PROGRAM TOWARDS SEA

Now we identify the key research challenges and their grouping into research themes, bringing to the surface specific open research problems that must be addressed, as well as some preliminary ideas to approach them based on the aforementioned paradigm and its principles.

A. The Challenges

Methodologically, SEA research should encompass the previously outlined goals and objectives, which are permeated by the above four principles. Namely, to:

- 1) Derive novel algorithms, structures, and models, which focus on and learn from what analysts are doing, tracking their interests in data spaces, as they shift with time.
- 2) Develop novel models and algorithms that learn from what the system does in response to queries.
- 3) Unify the results of 1 and 2, associating specific query space quanta with methods, models, and answers. Then develop, train, and leverage associative-learning models to predict results for future unseen queries.
- 4) Formulate and develop novel notions of and models for query-answer explanations.

- 5) Develop a suite of models, methods, and structures, which prove the applicability and showcase the benefits of the data-less analytics principle across various analytics tasks (query types), data formats (text, graph, tabular), and system types.
- 6) Develop a suite of indexes, caches, and statistical structures, towards big-data-less analytics.
- 7) Identify and evaluate key alternative algorithms, methods, and models for key analytics tasks.
- 8) Train models which learn from past task executions and build optimising modules, which, on-the-fly, adopt the best execution method for the task at hand.

The above challenges constitute open problems to be solved en route to meeting the SEA desiderata. They are grouped into the following research themes (RTs).

B. Research Theme 1: Data-less Big Data Analytics

Understand and leverage analyst queries vis-a-vis the system’s behaviour in response to analysts’ queries, in order to predict answers to future, unseen queries without access to base data. This theme entails five core challenges:

1) *Query-space Quantization*: Derive novel algorithms and models, to efficiently and scalably learn the structure of the query space, identifying analysts’ current interests.

2) *Answer-space Modelling*: Derive novel algorithms and models, to learn and understand how to describe the results provided by the system to analytical queries (i.e., model the answer data space).

3) *Predictive Models*: Combine 1) and 2) to predict results of new queries, using their position within the quantized query space and the association of query quanta with answer-space models. The models will be developed and trained to concurrently optimize query space quantization and system-answer error. Develop error estimation techniques, in order to accompany predicted answers with (accurate) error estimations so that the system (or analyst) can choose to proceed with the predicted answer or to obtain an exact answer by accessing the base data.

4) *Model Maintenance*: Develop approaches, which can ensure accuracy in the presence of updates in (i) query patterns, as analysts’ interests drift, and (ii) base data, as data is inserted, deleted, and updated. (i) will rest on appropriate definitions of distance between a query and the query quanta. (ii) will rest on this and the estimated error associated with the predicted answer.

5) *Multi-system Analytics*: Big data analytics is currently performed over big data analytics stacks. Within the same level, many alternative systems may be collaboratively employed. For instance, different types of NoSQL systems may coexist for different data types, e.g., for structured access to graphs, tables, and documents, alongside a distributed file system. Emerging applications involving analytics operators across data stored in such polystores typically wish to access data stored at different systems. Invariably this requires moving data from one system to the other, which is a time-consuming and resource wasting process. Despite the promises of recent research

on polystores (e.g., [39], [40]) these problems continue to be a major impediment in multi-system analytics.

The data-less data processing paradigm offers new insights and the potential to completely ameliorate these costs. The central idea is to develop and deploy agents within each constituent system in a polystore. The agent in essence encapsulates the ML models and functionality for data-less processing. Therefore, instead of migrating large volumes of data between constituent systems, either: (i) only approximate results of performing operators on the local data are sent, or (ii) the models themselves are migrated which are incorporated to produce the final approximate answers.

C. Research Theme 2: Big-Data-less Big Data Analytics

RT2 focuses on answering queries using surgical base data accesses to only the (small) subsets of the voluminous data sets (which are required or simply suffice) to produce the answer. This will be achieved by developing access structures like indexes as well as specialized (semantic) caches. RT2 includes three main threads.

1) *Data Manipulation Operations (Defining Subspaces of Interest)*: The key operations to be studied have been identified from our prior research, where the state of the art solutions are badly lacking and include fundamental tasks. In general, they should include fundamental operations such as join operations, (especially in distributed settings, focusing on minimising the data movements from server node to server node for its computation), kNN query processing (and its variants, such as Reverse kNN, kNN joins, all-pair and approximate kNN, etc.), spatial analytics operations (such as Spatial Joins, spatial (multi-dimensional) range queries, etc.), radius queries, etc.

2) *Ad Hoc ML Tasks*: This task focuses on traditional ML operations (such as classification, clustering, regression, etc.). Specifically, analysts are to define (using selection operators, such as those discussed above) subspaces of interest and ask for the data items within these subspaces to be clustered, classified, or to perform regressions, etc. Although a large body of research has tackled such tasks in isolation, performing these tasks efficiently and scalably on arbitrarily defined, ad hoc subspaces is an open problem. This thread will develop semantic caches and indexes to dramatically expedite such operations. Furthermore, it will target expediting other fundamental operations, namely kNN regression and kNN classification, exploiting insights gained.

3) *Raw Data Analytics*: Currently data analytics is performed on cleaned data, fitted to given data models. This requires a resource-hungry and time-consuming data wrangling process and ETL (Extract-Transform-Load) procedures. As data sizes increase, the data-to-insight times can become too high. This thread will centre its attention on developing adaptive indexing and caching techniques that operate on raw data and facilitate efficient and scalable raw-data analyses.

RT2 will consider both exact and (appropriately-defined) approximate solutions and will exploit known

properties of both real-world data sets (e.g., their distributions), known access patterns (workload characteristics) and will leverage statistical properties and ML results (such as data transformations and embeddings) to accomplish its goals.

D. Research Theme 3: Optimization – Alternatives, Evaluation, and Optimal Selection

The goal of RT3 is to derive an "on-the-fly optimized processing strategy" for the analytical query at hand. Assume we are given a performance metric (e.g., money-costs, task-processing time, bandwidth, etc.) to optimize. RT3 research aims to produce the optimal execution strategy for the given metric. There are several degrees of freedom here, which define the following main research items:

1) *Access Structure/Method Selection*: A variety of indexes (i.e., those developed in RT2) may exist or may worth to be built on the fly during query processing. Selecting the most appropriate access structure is the first degree of freedom, and a key challenge.

2) *Distributed Processing Paradigm*: Based on our experience, we differentiate between a MapReduce style of distributed task processing (on top of systems like Hadoop, Spark, Shark, Flink, etc.) versus a coordinator-cohort paradigm, whereby a coordinating node accesses directly the storage engine. Different circumstances dictate use of different paradigms. Assume there exist indexes for processing analytics queries. Then, having a coordinating node accessing the (typically distributed) index and then use it to surgically access small subsets of base data, directly from the back-end storage, may be preferable to having an all-out MapReduce processing of data nodes.

3) *Inference Model*: SEA will contain a number of inference models used for predictive analytics. Furthermore, higher-level functionality, to be provided by SEA, may directly depend on using a number of different models. Even if said models derive from the same family (e.g., regression-based), different models have been found to be best for different data subspaces: e.g., when considering using different regression base models or boosting-based ensemble models [41], [42]. Therefore, identifying the most appropriate inference model to employ is another key challenge.

RT3 involves in-depth experimentation in order to identify costs (for the aforementioned metrics) associated with the alternatives mentioned above. SEA will conduct such experiments, derive key data and perform feature selection from this data in order to develop ML models, which can be trained from this data, and accurately predict the best execution method.

E. Research Theme 4: New Functionality – Higher-level Queries and Explanations

RT4 consists of two investigation threads defining appropriate (i) higher-level queries and (ii) query-answer explanations.

1) *Higher-level Queries*: Building upon RT2, this will facilitate more complex and in-depth analyses, proposing higher-level queries, which encompass the more "basic" queries of WP2.

- The first challenge is to identify higher-level queries that can be themselves processed efficiently, scalably and accurately.
- The second challenge pertains to the development of the processing mechanisms (indexes, semantic caches, etc.) to do so, ensuring higher performance compared to executing each constituent basic query in isolation.
- The third challenge is to define appropriate hierarchical or graph structured spaces, showing how queries at lower levels can be combined to offer higher-level functionality.
- The final challenge is how to visualize and export to analysts this complex query capability set.

2) *Query-Answer Explanations*: A large number of analytical queries supported by present day data systems return a single scalar value, which as mentioned, leaves much to be desired. This thread will develop novel rich, yet concise, explanations for query-answer pairs that will facilitate deeper understanding of the queried data subspaces. These explanations are expected to be models / functions that show how the answer to the query depends on the query's parameters. As an example, an explanation can be a (piecewise) linear regression model showing how count of (or the correlation coefficient between attributes within) a data subspace depends on the size of the subspace. This will facilitate result visualizations and provide the answer of many related exploratory queries the analyst may wish to issue, without issuing them; the analyst will be able to simply plug in values for parameters to the explanation models.

The next challenges concern the ability to compute explanations in a SEA fashion and deriving appropriate explanations for the more complex queries above.

F. Research Theme 5: Global-Scale Geo-Distributed SEA

RT5 is concerned with how to best deploy the developed SEA models in a wide-scale (geo-distributed) setting and the additional research problems that this setting introduces.

Big cloud computing providers are increasingly investing in a federation of geo-distributed data centres, located around the world to provide high local-client efficiency of data access as well as robust, reliable and fault-tolerant data access to all clients around the world. For example, the Google global cloud platform employs tens of such data centres [43] and the Microsoft counterpart involves more than 100 such data centres [44]. As a result the massive data sets, expected to be involved in big data analytics scenarios, can be geo-distributed across such global cloud platforms. Providing analytics, thus, over such data sets is fraught with efficiency and scalability limitations - as has been well recognized, for example in the Iridium project [45].

Research themes RT1 – RT3 can be of real value in these environments. The target is to reduce WAN-based

inter-datacentre communication, which can introduce high response times and unpredictability during geo-distributed analytics. Figure 3 exemplifies how SEA is envisaged to operate within such a geo-distributed setting. The essential idea is to place key functionality at the edge nodes of the global network, akin to the earlier discussion about multi-system analytics. The agents encapsulating the developed intelligence (from RT1 and RT2) can be deployed at edge nodes to localize query processing as much as possible.

In this way, the system (i.e., an agent at some edge node) accesses base data (stored at remote data centres) only when expected errors of local models at the edge node is high. Conversely, analysts can be informed of expected errors in the answers received by local agents and can issue an exact query to base data stored remotely, only if estimated error of local predictions is high.

RT5 brings to the surface several research tasks, discussed below.

1) *Network System Architecture*: We envisage the network to contain **core nodes** and **edge nodes**. The core nodes store the actual data. Additionally, as we shall see below, they can also participate in building models from the data. Therefore, they possess the option of either executing an exact-answer query (e.g., scanning the base data) or providing approximate answers by answering queries based on the built models. Conversely, edge nodes typically maintain only models of the base data and can provide only approximate answers to queries.

Given this node-functionality separations, the central question is how to organize the query- and data-flow between nodes in the network. The issues here concern how to (i) organize the (intelligent agents at the) edges so that they can possibly collaborate in knowledge sharing and query answering; and (ii) how to organize edges-to-core nodes communication. A combination of peer-to-peer overlay network approaches and processing as well as a client-server model communication is possible, depending on the number and churn of edge nodes.

2) *Distributed Model Building*: In an ideal setting, analytical queries from edge nodes target different data subspaces. In this case, each edge node would build its own separate model. The initial queries would be treated as training queries and the edge would gather enough (query, answer) pairs to locally train a model. When the model is trained, the edge can then filter all queries from reaching the core, by using the local model to answer queries.

In general, however, this will not be the case. The queried subspaces from queries originating at different edge nodes will be overlapping. This affords the opportunity of distributed model building. As before, the initial training queries will reach the core nodes, but this time from different edge nodes. Said core nodes can then collaborate to train a model faster, by considering training queries from several different edge nodes. Subsequently, the core nodes can then communicate the model to the edge nodes from where relevant queries originated. From this point on, edges can enter their query-answering phase.

An important relevant issue here is how to **share the**

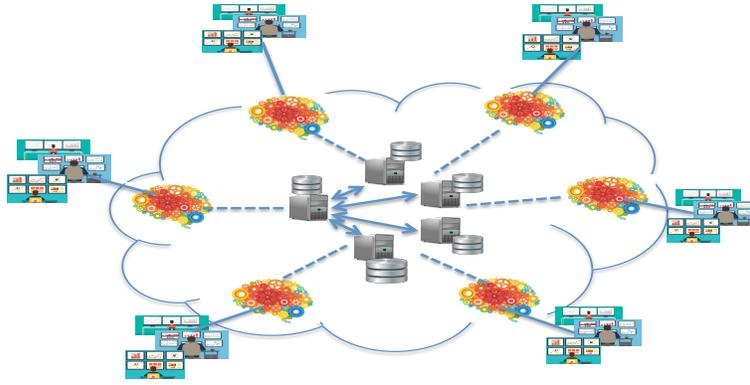


Fig. 3: The Vision for Wide-Scale Distributed SEA Big Data Analytics Processing.

model state, defining which models have been built and for which data subspaces.

3) *Distributed Model Maintenance*: The key issues here pertain to answering the following questions: (i) which models to build? (ii) where (i.e., at which edge nodes) should these models be located? (iii) how to efficiently maintain the models consistent, vis-a-vis changes in query patterns and data state updates (data insertions, deletions, and updates).

We expect that, for many applications, queries in typical workloads are overlapping and target certain subspaces which are of significantly smaller size than the total sizes of the stored data sets. Therefore, a key desideratum is that, given the massive size and number of stored data sets, *only* models for the (much smaller) data subspaces of interest are built. Hence, we expect core nodes to identify the data subspaces of (current) interest. How is this best accomplished? Certainly edge nodes can help as they are 'query-facing', but as mentioned core nodes will likely be involved as queried subspaces from different edges overlap.

The methods for query quantization, mentioned in RT1 can come handy here. But we must ensure that modelled subspaces are as small as possible in order to increase model accuracy. However, we need also reduce model training time. So this is a challenging task.

Given the need for model consistency and maintenance, said models should be carefully distributed at edge nodes so to, on the one hand, increase the filtering power of edge nodes, while on the other, reduce the costs involved in maintaining model consistency and accuracy.

Shifts in the user interests can be detected locally by edge nodes and globally by core nodes. This detection should lead to purging 'older' models, referring to data subspaces which are no longer of interest and/or to the redefinition of subspaces of interest and to the update of models to account for the extension or shrinking of said subspaces.

4) *Analytical Query Routing*: Given an analytical query at some edge node, query routing refers to deciding where should the query be answered. Should it be answered at the local edge node? Should it be sent to another edge node? If so, which one and how? Should it reach other nodes?

Depending on how the model state is being shared by the nodes of the system, several options for answering the query are possible.

5) *Model Error Maintenance*: Given the distribution of models across the edge and core networks a key requirement is to be able to accurately predict the model error. This may be extremely challenging in itself, depending on which ML models are being used. But even assuming that it is possible to predict the error associated with a deployed model, the different nodes where the model is actively used should have an accurate expectation with regard to its accuracy (in the presence of data and query changes and model updates).

VI. CONCLUSIONS

Distributed systems play a key role in big data analytics, as data analytics stacks involve different distributed systems at their various layers (for distributed/parallel dataflow, distributed resource management, distributed file systems, distributed NoSQL systems, etc.). Furthermore, given the push for global-scale geo-distributed analytics, collections of such analytics stacks are working together, answering analytical queries. In these settings, unfortunately the current state of the art often fails to address concerns with respect to efficiency, scalability, and accuracy. These three properties represent the holly grail for modern big data analytics. To overcome current limitations a new paradigm is needed.

With this vision paper we attempt to provide this missing paradigm: It revolves around the novel notion of data-less data analytics. It proposes that current approaches fail to accomplish the SEA desiderata exactly because they access too many data-storing nodes and too much data during analytics processing. In contrast, the new paradigm attempts to answer analytical queries using models of the underlying data; said models are much lighter and can be highly accurate. This style of processing avoids time-consuming and resource hungry processing over big data analytics stacks.

We have outlined the principles on which related research that will materialize this vision should be based. We have outlined successes so far from our recent work, which

testify to the high potential of the vision. We have also put forth a research program that attempts to organize the main research threads that should be undertaken, offering initial suggestions that appear promising, and identifying challenges that need be addressed.

REFERENCES

- [1] M. L. Williams, K. Fischer, J. Freymueller, B. Tikoff, and A. T. et al, “Unlocking the secrets of the north american continent: An earthscope science plan for 2010-2020,” 2010.
- [2] Z. D. Stephens, S. Y. Lee, F. Faghri, and R. H. C. et al, “Big data: astronomical or genetical?” *PLoS Biol*, 2015.
- [3] S. Ghemawat, H. Gobioff, and S.-T. Leung, “The google file system,” in *Proceeding of SOSP*, 2003.
- [4] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” in *Proceeding of OSDI*, 2004.
- [5] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in *Proceeding of IEEE 26th Symposium on Mass Storage Systems and Technologies*, 2010.
- [6] M. Zaharia, M. Chowdhury, T. Das, and A. D. et al, “Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing,” in *Proceeding of NSDI*, 2012.
- [7] A. Alexandrov, R. Bergmann, and S. E. et al, “The stratosphere platform for big data analytics,” *The VLDB Journal*, 2014.
- [8] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmelegy, and R. Sears, “Mapreduce online,” in *Proceeding of NSDI*, 2010.
- [9] V. Valilapalli, A. Murthy, C. Douglas, and S. A. et al, “Apache hadoop yarn: Yet another resource negotiator,” in *Proceeding of ACM SOCC*, 2013.
- [10] F. Chang, J. Dean, and S. G. et al, “Bigtable: A distributed storage system for structured data,” in *Proceeding of OSDI*, 2006.
- [11] L. George, *HBase: The definitive Guide*. O’Reilly, 2011.
- [12] A. Lakshman and P. Malik, “Cassandra: A decentralized structured storage system,” in *SIGOPS Operating Systems Review*, 2010.
- [13] S. Melnik, A. Gubarev, J. J. Long, G. Romer, S. Shivakumar, M. Tolton, and T. Vassilakis, “Dremel: interactive analysis of web-scale datasets,” *Communication ACM*, 2011.
- [14] C. Engle, A. Lupher, R. Xin, and M. Z. et al, “Shark: Fast data analysis using coarse-grained distributed memory,” in *Proceeding of ACM SIGMOD*, 2012.
- [15] S. Chaudhuri, G. Das, and V. Narasayya, “Optimized stratified sampling for approximate query processing,” *ACM Trans. on Database Systems, (TODS)*, 2007.
- [16] G. Cormode and S. Muthukrishnan, “An improved data stream summary: The count-min sketch and its applications,” *Journal of Algorithms*, 2005.
- [17] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica, “Blinkdb: Queries with bounded errors and bounded response times on very large data,” in *Proceeding of ACM Eurosys 2013*.
- [18] L. Sidirourgos, M. L. Kersten, and P. A. Boncz, “Sciborq: Scientific data management with bounds on runtime and quality,” in *Proceeding of CIDR*, 2011.
- [19] Y. Park, A. S. Tajik, M. Cafarella, and B. Mozafari, “Database learning: Toward a database that becomes smarter every time,” in *Proceeding of ACM SIGMOD*, 2017.
- [20] A. Wasay, X. Wei, N. Dayan, and S. Idreos, “Data canopy: Accelerating exploratory statistical analysis,” in *Proceeding of ACM SIGMOD*, 2017.
- [21] M. I. Jordan, “On statistics, computation and scalability,” *Bernoulli*, 2013.
- [22] L. Bottou and O. Bousquet, “The tradeoffs of large scale learning,” in *Proceeding of NIPS*, 2007.
- [23] J. Acharya, I. Diakonikolas, J. Li, and L. Schmid, “Fast algorithms for segmented regression,” in *Proceeding of ICML ’16*.
- [24] F. Savva, C. Anagnostopoulos, and P. Triantafillou, “Explaining analytical queries,” in *In progress*, 2018.
- [25] S. Idreos, O. Papaemmanouil, and S. Chaudhuri, “Overview of data exploration techniques,” in *Proceeding of ACM SIGMOD*, 2015.
- [26] C. Anagnostopoulos and P. Triantafillou, “Learning set cardinality in distance nearest neighbours,” in *Proceeding of IEEE International Conference on Data Mining, (ICDM15)*, 2015.
- [27] —, “Learning to accurately count with query-driven predictive analytics,” in *Proceeding of IEEE International Conference on Big Data*, 2015.
- [28] —, “Efficient scalable accurate regression queries in in-dbms analytics,” in *Proceeding of IEEE International Conference on Data Engineering, (ICDE17)*, 2017.
- [29] —, “Query-driven learning for predictive analytics of data subspace cardinality,” *ACM Trans. on Knowledge Discovery from Data, (ACM TKDD)*, 2017.
- [30] N. Ntarmos, Y. Patlakas, and P. Triantafillou, “Rank join queries in nosql databases,” in *Proceedings of VLDB Endowment (PVLDB)*, 2014.
- [31] A. Eldawy and M. F. Mokbel, “Spatialhadoop: A mapreduce framework for spatial data,” in *Proceeding of IEEE Int. Conf. on Data Engineering (ICDE)*, 2015.
- [32] D. Xie, F. Li, B. Yao, and G. L. et al, “Simba: Efficient in-memory spatial analytics,” in *Proceeding of ACM SIGMOD*, 2016.
- [33] A. Cahsai, N. Ntarmos, C. Anagnostopoulos, and P. Triantafillou, “Scaling k-nearest neighbours queries (the right way),” in *Proceeding of 37th IEEE Int. Conf. Distributed Computing Systems (ICDCS)*, 2017.
- [34] J. Wang, N. Ntarmos, and P. Triantafillou, “Indexing query graphs to speed up graph query processing,” in *Proceeding of 19th International Conference on Extending Database Technology, (EDBT)*, 2016.
- [35] —, “Graphcache: A caching system for graph queries,” in *Proceeding of 20th International Conference on Extending Database Technology, (EDBT)*, 2017.
- [36] C. Anagnostopoulos and P. Triantafillou, “Scaling out big data missing value imputations,” in *Proceeding of ACM SIGKDD Conference, (KDD14)*, 2014.
- [37] F. Katsarou, N. Ntarmos, and P. Triantafillou, “Subgraph querying with parallel use of query rewritings and alternative algorithms,” in *Proceedings of Conference on Extending Database Technology, (EDBT)*, 2017.
- [38] —, “Hybrid algorithms for subgraph pattern queries in graph databases,” in *Proceedings of IEEE International Conference on Big Data, (BigData17)*, 2017.
- [39] J. LeFevre, J. Sankaranarayanan, H. Hacigumus, J. Tatemura, N. Polyzotis, and M. J. Carey, “Miso: Souping up big data query processing with a multistore system,” in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD ’14. New York, NY, USA: ACM, 2014, pp. 1591–1602. [Online]. Available: <http://doi.acm.org/10.1145/2588555.2588568>
- [40] V. Gadepally and et al, “The bigdawg polystore system and architecture,” in *arXiv:1609.07548v1*, 2016.
- [41] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001.
- [42] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.
- [43] “Google data centre locations,” <https://www.google.com/about/data-centers/inside/locations/index.html>.
- [44] “Microsoft data centres,” <https://www.microsoft.com/en-us/cloud-platform/global-datacenters>.

- [45] Q. Pu, G. Ananthanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl, and I. Stoica, “Low latency geo-distributed data analytics,” in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15. New York, NY, USA: ACM, 2015, pp. 421–434. [Online]. Available: <http://doi.acm.org/10.1145/2785956.2787505>
- [46] L. Bottou, F. Curtis, and J. Nocedal, “Optimization for large-scale machine learning,” in *arXiv:1606.04838[stat.ML]*, 2017.
- [47] F. Katsarou, N. Ntarmos, and P. Triantafillou, “Performance and scalability of indexed subgraph query processing methods,” in *Proceedings of VLDB Endowment (PVLDB)*, 2015.
- [48] Q. Ma and P. Triantafillou, “Query-driven regression model selection,” in *In progress*, 2018.