Report on Mrs Ge SONG PhD thesis, entitled
« Parallel and Continuous Join Processing for Data Streams »

Johan Montagnat

Overall evaluation

In this document Mrs Ge SONG addresses the timely problem of big data streams processing in near real-time. The work is positioned in the context of very large and quickly evolving data sets that are rather commonly available today. The “Volume” and “Velocity” of such data streams makes complete data storage impractical and enforces the use of distributed computing resources and approximate algorithms to harness the amount of computations needed. Two driving cases are considered: (1) *k*-Nearest Neighbours (kNN) computation in multi-dimensional data sets, which is representative of quickly evolving data sets with “fixed query” to be computed on the data, and (2) SPARQL querying of RDF graphs, which exhibits a variable data query pattern. In both cases, the work includes (1) extensive literature surveys of the methods used to compute kNN and SPARQL queries over large data sets, although not necessarily in a streaming context, (2) theoretical complexity considerations, (3) distributed and approximate computing algorithmic contributions, and (4) performance evaluation studies. The manuscript presented shows a significant amount of work including interesting scientific contributions. The writing is sometimes terse but the overall manuscript is well structured enough and accessible. The thesis defended sometimes raises discussion points as detailed below, but there are no flaws in the work presented. Overall, the manuscript is presenting an interesting and inspiring work.

Detailed comments

This manuscript is composed by 5 chapters, including the introduction, background material, a first contribution on kNN computation, a second contribution on SPARQL queries computation, and the conclusion.

The **introduction chapter** is providing the high-level background information on the kind of problem addressed, the objectives of this work and the use cases tackled. The kNN computation and SPARQL query use cases are both presented as “join” operations on the data streams, although from quite different nature (distance-based vs equality-based), and representative of two different data analysis patterns (fixed query vs variable query). There is no real discussion on the motivation for selecting these two particular use cases, although both are “primitive” algorithms susceptible to be used in a wide variety of contexts. Otherwise this chapter provides an appropriate level of detail for introducing the work done.

Chapter 2 progressively provides background information on parallel computing challenges, Big Data processing, common data streams processing environments, and the two use cases addressed. It includes an extensive overview of existing distributed computing environments for Big Data and programming techniques to tackle quickly evolving data streams efficiently.

Chapter 3 describes the first contribution of the thesis on kNN computation. It introduces both centralized and distributed methods developed to compute kNN on static data sets, and centralized methods to compute kNN on data streams. The contribution is then introduced as the only method able to compute data streams using distributed resources.

The algorithm proposed to process data streams on a distributed system is inspired from the existing techniques used to distribute kNN computation in the case of static data sets (data partitioning strategies) and the techniques used to process data streams in a centralized context (approximations and dimension reduction techniques). Two variants are detailed, to address a single variable stream or two variable streams.

The focus of the state-of-the-art is on map-reduce-based techniques running on shared-nothing distributed computing infrastructures. Alternative techniques using parallel implementations (MPI-based of GPU-based) are mentioned but discarded as not scalable enough. There is no objective information on scale achievable in both cases though (in terms of number of computing units that can be efficiently exploited in parallel, size of the data sets processed or speed of the data streams generation for instance).

An interesting analysis of 5 existing Map-Reduce-based algorithms is proposed, introducing a generic kNN computation workflow that includes a pre-processing step (data seeds selection and dimension reduction), a partitioning step and the actual kNN computation. Existing algorithms can be seen as different variants of this generic workflow, which is also used to make a theoretical analysis of these algorithms complexity.

An extensive experimental performance analysis of the 5 state-of-the-art techniques analysed and the contributed kNN algorithm is proposed. It yields an interesting discussion on the impact of various parameters and characteristics of input data on the algorithms performance. The algorithm contributed is not compared to state-of-the-art approaches though, neither theoretically nor practically, as the only representative of a distributed algorithm applying to data streams. The evaluation metrics are different than for the other algorithms and it is difficult to assess the improvement over the state-of-the-art for this contribution.

Chapter 4 describes the second contribution on SPARQL querying over large distributed knowledge graphs. This chapter starts with an analysis of the different solutions to compute SPARQL queries in parallel, by decomposing the original query in multiple sub-queries, partitioning the RDF graph on several servers, or both. The work is then positioned in the context of a dynamically evolving RDF graphs for which oldest data tends to be outdated. The contribution proposes to process a sub-set of most recent data using a temporal sliding window mechanism. The data graph is partitioned in several storage servers on the RDF triple predicates. Queries are decomposed into triple patterns applied to each data partition, and matched triples are exchanged between servers using compact Bloom Filters to minimize communication overheads. Optimizations are considered depending on the number of join variables between triple patterns. As the Bloom filters used are based on RDF triples hashing, collisions might occur and trigger false positives. The trade-off between filter size and query accuracy therefore needs to be tuned. Objective parameterization indicators are discussed.

Apart from the sliding window mechanism, the data partitioning and query decomposition strategies described represent an interesting effort to efficiently compute SPARQL queries on a large RDF data set, approximately but with controlled accuracy. The method can only apply to a sub-set of SPARQL queries (all examples are based on Basic Graph Patterns), but there is no real discussion on the exact part of SPARQL that can be processed by the contributed algorithm. It should also be noted that subdividing a query into atomic subqueries (triple patterns here) might induce large data transfers between servers and post join evaluation. Distributed data base systems often attempt to process the larger possible chunks of a query on each sub-system to alleviate these performance issues. This aspect is not discussed in the context of this work.

To process RDF data streams, a sliding window model is applied both to the data sets and the Bloom Filters. The query computing is then incremental, periodically updating the Bloom filters to match the new generation of data with query triple patterns and recomputing the joins between different data shares. A theoretical analysis gives objective information on the Bloom filter parameters to achieve a given level of accuracy, and the query recomputation cost.

An experiments campaign using synthetically generated data is proposed. It analyses the impact of different algorithm parameters (sliding window size, number of generations processed, number of computing nodes…) on the contributed algorithm throughput. These experiments give insights on the algorithm behaviour and parameterization. However, the scalability of the method is tested to a rather limited scale (10 nodes) in spite of the efforts invested in building an efficient distributed SPARQL querying system. It would be interesting to have more insights on the ability to scale to larger sliding window / number of generations with more computing power. In addition, a high-level analysis of the experimental results is lacking. The contributed algorithm is also not compared to any state-of-the-art methods and this experimental part only serves as algorithm tuning indicators.

The **concluding chapter 5** provides a manuscript-summary conclusion and a brief section on future directions for this work. It acknowledges some of the limitations of the work done and outlines potential application domains.

Summary

In her thesis, Mrs Ge SONG introduced two contributions to the domain of distributed computing over dynamic data streams. The problem addressed is very timely. The contributions proposed are significant, backed by sound theoretical analysis and experimented in real, up-to-date distributed execution systems. This thesis presents a solid body of work on the theoretical and algorithmic aspects of the problems addressed. Equally important, the concrete implementation and confrontation to reality is not neglected. Mrs Ge SONG demonstrated strong abilities to innovate and conduct scientific investigations.

For these reasons, I recommend Mrs Ge SONG for obtaining the title of Doctor of the University Paris Saclay.

Sophia Antipolis, September 23, 2016.

[illegible]

Johan MONTAGNAT
DR CNRS
Laboratoire I3S, Université de Nice – Sophia Antipolis
930 Route des Colles
06903 Sophia Antipolis cedex
phone: +33 492 96 51 03 / email: johan.montagnat@cnrs.fr