Revisit Data Partitioning in Data-intensive workflows
Radita Liem, Shadi Ibrahim

▶ To cite this version:
Radita Liem, Shadi Ibrahim. Revisit Data Partitioning in Data-intensive workflows. PDSW 2022 - 7th International Parallel Data Systems Workshop, Nov 2022, Dallas, United States. hal-03913369

HAL Id: hal-03913369
https://inria.hal.science/hal-03913369
Submitted on 26 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License
Revisit Data Partitioning in Data-intensive workflows

Radita Liem∗§, Shadi Ibrahim†
∗ RWTH Aachen University, Germany
† Inria, Univ. Rennes, CNRS, IRISA, France

As data volumes increase (e.g., in 2020 alone, 64.2 ZettaByte of data was generated and replicated [1]), data-intensive workflows are now executed across multiple platforms from the Edge to Cloud and HPC, which is known as computing continuum. The intention behind this execution model is to exploit data locality and reduce data movement by executing data close to their sources while taking into consideration the computation capacities of different platforms.

Data-intensive workflows usually consist of multiple computation stages where data flows among them. For example, MapReduce jobs consist of two consecutive stages (Map and Reduce) and the output of the map stage is transferred and used as an input for the Reduce stage [2], [3]. The completion time of a stage strongly depends on the finishing time of the last task in this stage. Accordingly, much efforts have focused on reducing the variation in task execution times in the same stage by launching speculative copies of slow tasks [2] or evenly partitioning data across tasks to mitigate data skew [4], [5], [6], [7].

With the proliferation of data-intensive workflows such as IoT applications, machine learning applications, and deep learning applications; data skew remains a bottleneck in data analytic frameworks. Many research efforts have been dedicated to mitigating data skew by evenly distributing data across tasks while considering data locality [4], [6] and relying on high-speed networks [7], or by partitioning the data based on the computation capacities of the nodes [8]. However, most of these efforts focus on only two-stage applications and don’t consider network and I/O heterogeneity within or across platforms, making them impractical when running multi-stage data-intensive workflows on heterogeneous and shared environments.

As shown in Figure 1, PageRank application exhibits a severe skew in partitioned data (i.e., shuffled data) in stage 3 and stage 4 of the application (bottom chart). This in turn results in a noticeable variation in task execution times. The variation is represented by the gap between the average and maximum task execution times (top chart). We can also observe that the variation is more obvious when the degree of parallelism is set to 24, 32, and 40. Here, it is important to note that, the data assigned to each task (i.e., task input) includes the partitioned data and the data retrieved from the storage memory (cache).

ACKNOWLEDGMENTS

This work is supported by the ANR KerStream project (ANR-16-CE25-0014-01). Experiments presented in this paper were carried out using the Grid’5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER, and several Universities as well as other organizations (see http://www.grid5000.fr/).

§Work done while at Inria, Univ. Rennes, CNRS, IRISA, France.
REFERENCES


