SaddlebagX : High Performance Data Processing with PGAS and UPC++

Aril B. Ovesen, Amin M. Khan, Phuong N. Chau, Phuong Hoai Ha
aril.b.ovesen@uit.no, amin.khan@tecnico.ulisboa.pt,
phuong.n.chau@uit.no, phuong.hoai.ha@uit.no
Department of Computer Science, UiT The Arctic University of Norway, Tromsø, Norway

UiT NORGES ARKTISKE UNIVERSITET

UNIVERSITY OF TROMSØ

Introduction - Background

- The ever growing scale of the data and the emerging challenges of processing it efficiently at that scale are driving the need for high-performance data processing frameworks that are efficient yet highly programmable across big data analytics (BDA) and high-performance computing (HPC) communities.
- Big data analytics (BDA) frameworks, such as Apache Spark, among many others, provide simpler programming models, such as bulk synchronous parallel (BSP), for application development. However, applications using the BDA frameworks are often orders of magnitude slower than the counterparts using HPC frameworks such as MPI.
- We introduce SaddlebagX , a new data-centric framework that offers the high programmability of BDA frameworks, such as the simple BSP programming model, while at the same time striving for high performance by leveraging the Partitioned Globally Address Space (PGAS) computing paradigm and the Remote Memory Access (RMA) capability provided in HPC frameworks.
- SaddlebagX employs UPC++ [1] for PGAS and RMA implementation. UPC++ offers high-performance communication for asynchronous computation.
- SaddlebagX complements UPC++, a HPC framework, by providing typical features of BDA frameworks such as the **bulk synchronous parallel (BSP)** programming model for BDA applications.

Methods

- SaddlebagX 's BSP programming model is based on a new data abstraction called *autonomous locality-aware distributed tables* (ALADT) that exploits data locality (see Figure 1).
- SaddlebagX supports new `push`, `pull`, and `broadcast` operations that are optimized for SaddlebagX 's BSP programming model.
- A workflow brings together various table types, and an application can consist of multiple workflows.
- All the items from different tables within a single workflow are accessible to each other, irrespective of their locality within the process or across the nodes.
- Each item consists of both data and its associated computation functions, exploiting data locality while simplifying load-balancing problems.
- SaddlebagX provides the in-order delivery of the messages, allowing messages to arrive at the recipient in the exact order they are sent;

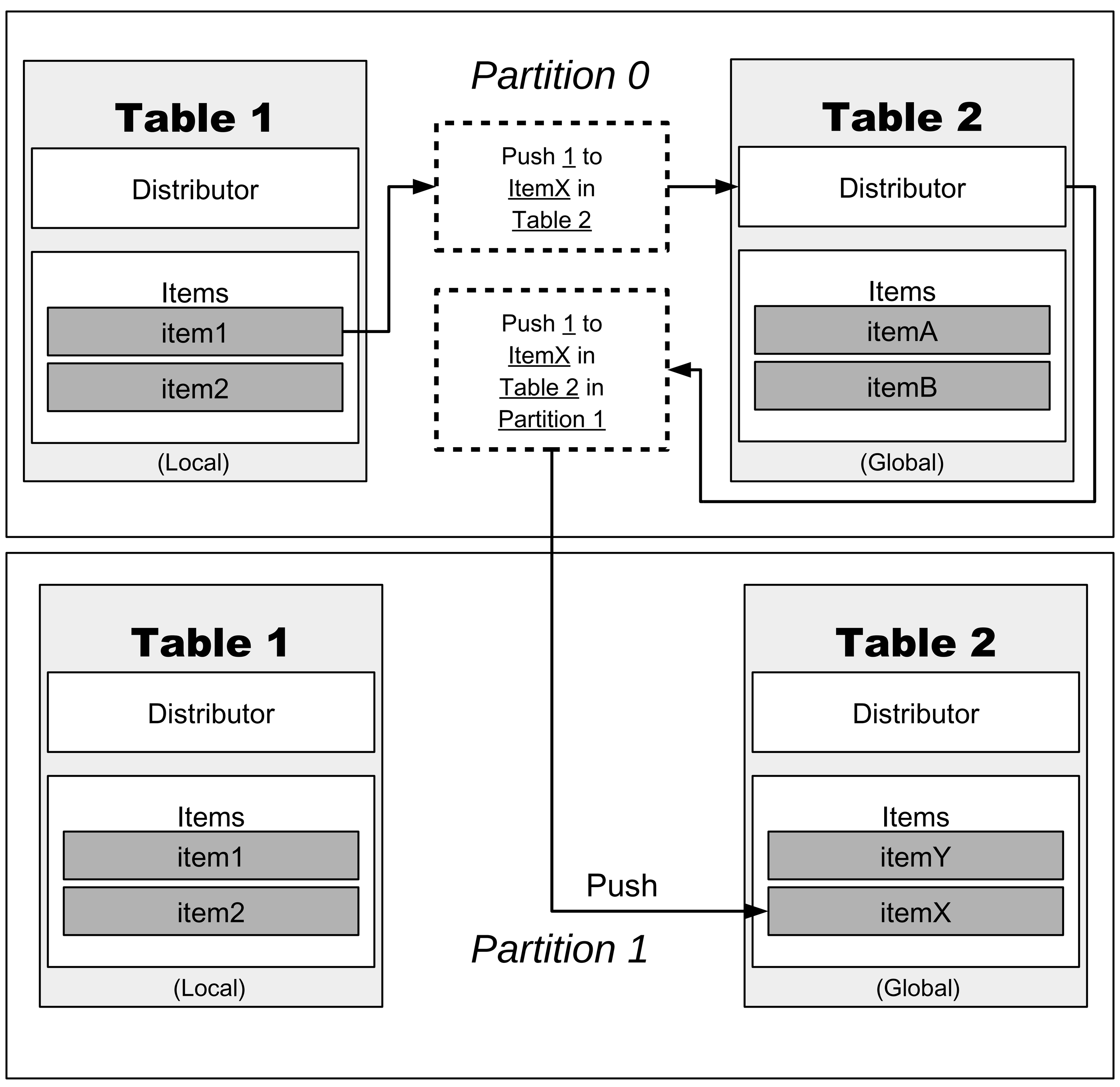

Figure 1: Example of communication between tables on different partitions.

Experimental Results

We compare SaddlebagX , a BSP-based data-centric framework, with Apache Spark, a seminal BSP-based BDA framework, and UPC++, a HPC framework.

- Figure 2 shows the comparison of SaddlebagX with Apache Spark GraphX using the **PageRank** benchmark with web datasets,where SaddlebagX vastly outperforms the Apache Spark GraphX, up to 40× faster.
- Figure 3 shows the running time in seconds for SaddlebagX , as it scales with different dataset sizes. This high performance is achieved by using the push operation with RMA for data transfer in SaddlebagX , which leverages the Remote Direct Memory Access (RDMA) capability of InfiniBand network hardware.

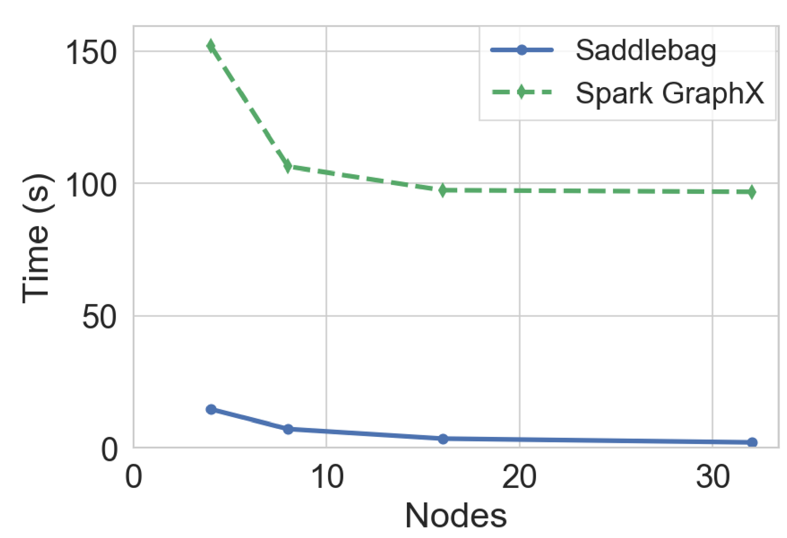

Figure 2: PageRank Benchmark.

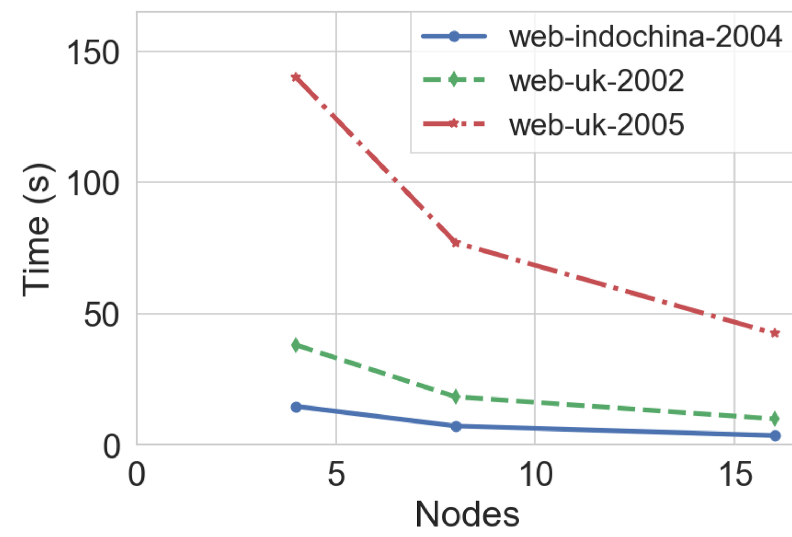

Figure 3: Scaling PageRank Benchmark with SaddlebagX .

- Figure 4 shows the comparison of SaddlebagX with Apache Spark MLlib using the the term frequency–inverse document frequency (**TF-IDF**) benchmark with Wikipedia corpus, where SaddlebagX significantly outperforms the Apache Spark implementation, up to 8× faster.
- Figure 5 shows the comparison of SaddlebagX with UPC++ using a basic sparse matrix vector multiplication (**SpMV**) benchmark, demonstrating that SaddlebagX introduces negligible overhead.

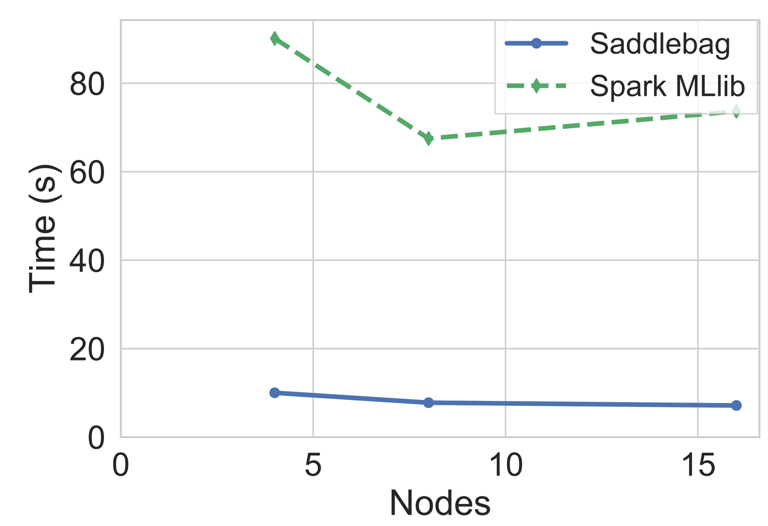

Figure 4: TF-IDF Benchmark.

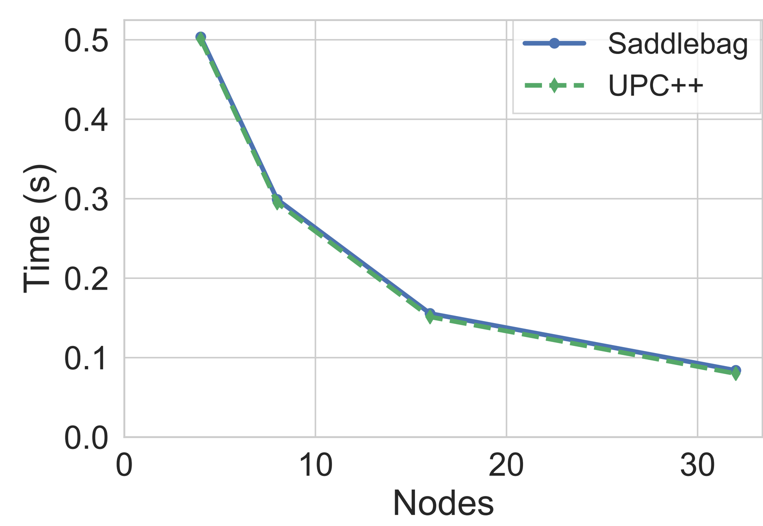

Figure 5: SpMV Benchmark.

Conclusions

- We have presented SaddlebagX , a high-performance data-centric framework that offers **programmability** via supporting the simple BSP programming model and **performance** via leveraging the PGAS computing paradigm and RMA capability. SaddlebagX is implemented as a C++ library, facilitating portability across platforms, and interoperability with other libraries (e.g., CUDA).
- SaddlebagX provides key features to ease development of data-intensive applications, like implicit distribution of data, co-locating computation with relevant data, and an easy-to-use programming model.
- Our preliminary results show that the new locality-aware data abstraction and communication features of SaddlebagX allow it to exceed in performance while SaddlebagX 's simple BSP programming model facilitates BDA application development without incurring additional performance overheads.

References

[1] Bachan, John and Baden, Scott B and Hofmeyr, Steven and Jacquelin, Mathias and Kamil, Amir and Bonachea, Dan and Hargrove, Paul H and Ahmed, Hadia.
UPC++: A High-Performance Communication Framework for Asynchronous Computation.
In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 963–973, Rio de Janeiro, Brazil, May 2019. IEEE.

This work is funded by the Research Council of Norway under PREAPP project (grant n° 231746) and eX3 project (grant n° 270053).