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

Aril B. Ovesen*
Department of Computer Science,
UiT The Arctic University of Norway, Tromsø, Norway
aril.b.ovesen@uit.no

Amin M. Khan†
INESC-ID. Instituto
Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
amin.khan@tecnico.ulisboa.pt

Phuong N. Chau‡
Department of Computer Science,
UiT The Arctic University of Norway, Tromsø, Norway
phuong.n.chau@uit.no

Phuong Hoai Ha
Department of Computer Science,
UiT The Arctic University of Norway, Tromsø, Norway
phuong.hoai.ha@uit.no

Abstract

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. 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++ for PGAS and RMA implementation. 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. Graph analytics and non-iterative bulk transformation benchmarks using SaddlebagX show significant performance gains (up to 40×) compared to the Apache Spark counterparts. To evaluate the overheads incurred by SaddlebagX, we compare SaddlebagX with UPC++ using a basic sparse matrix vector multiplication (SpMV) benchmark, and observe no noticeable slowdown. These preliminary results show that SaddlebagX is a high-performance and highly-programmable data processing framework.

Keywords high-performance computing, data-intensive computing, programming frameworks

1 Introduction

Today scientific discovery typically involves both high performance computing (HPC) and big data analytics (BDA), demanding unified frameworks for HPC and BDA. Big data frameworks, such as Apache Spark, among many others, provide simpler programming models such as bulk synchronous parallel (BSP). However, applications using the BDA frameworks are often orders of magnitude slower than the counterparts using HPC frameworks such as MPI.

We present SaddlebagX, a new data-centric framework that offers the high programmability of BDA frameworks and the high performance of HPC frameworks. SaddlebagX provides typical programming features of BDA frameworks such as the bulk synchronous parallel (BSP) programming model for BDA applications. SaddlebagX provides 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 partitioned globally addressable space (PGAS) and remote memory access (RMA) implementation. The PGAS paradigm provides an efficient mechanism to develop high performance distributed applications, and UPC++ offers high-performance communication for asynchronous computation.

SaddlebagX provides the BSP programming model similar to other BDA frameworks. Programmers can use the communication primitives from SaddlebagX to batch the communication in the BSP model for efficiency. SaddlebagX supports new `push`, `pull`, and `broadcast` operations that are optimized for SaddlebagX's BSP programming model.

SaddlebagX's BSP programming model is based on a new data abstraction called *autonomous locality-aware distributed tables* (ALADT) that exploits data locality. 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, and can exchange data using `push`, `pull`, and `broadcast` operations. 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.

*Student author.
†Also with UiT The Arctic University of Norway.
‡Student author who will present the poster.

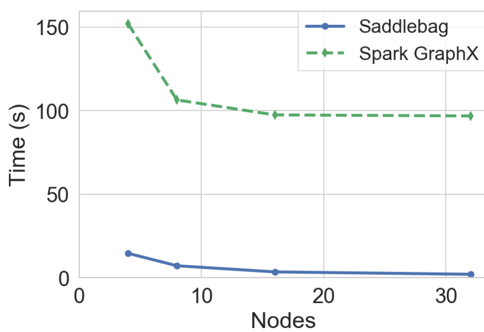

Figure 1. PageRank Benchmark.

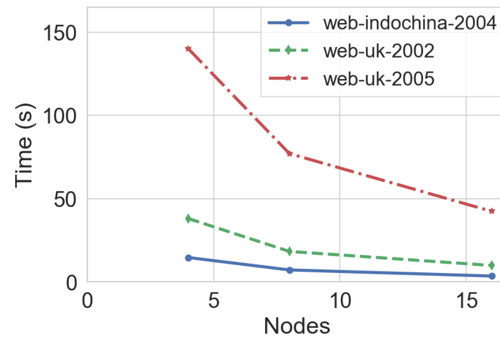

Figure 2. Scaling PageRank Benchmark with SaddlebagX.

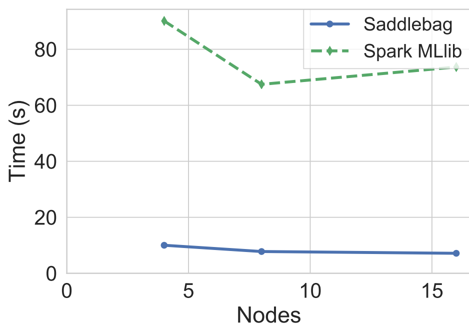

Figure 3. TF-IDF Benchmark.

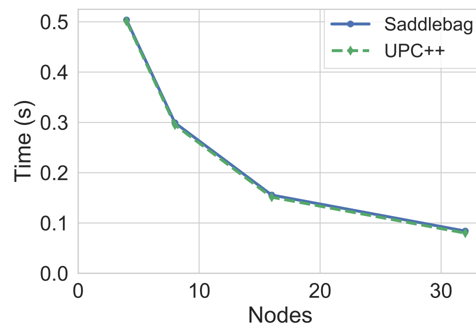

Figure 4. SpMV Benchmark.

2 Evaluation

We compare SaddlebagX, a BSP-based data-centric framework, with Apache Spark, a seminal BSP-based BDA framework, and UPC++, a HPC framework, using different benchmarks on a HPC computing cluster comprising of more than 650 compute nodes, each consisting of two Intel E5-2670 CPU @ 2.6 GHz with 16 cores, and 64 GiB RAM, connected through FDR InfiniBand. Figure 1 shows the comparison of SaddlebagX with Apache Spark GraphX for **PageRank** using the web dataset[1], where SaddlebagX vastly outperforms Apache Spark GraphX, up to 40× faster. Figure 2 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 3 shows the comparison of SaddlebagX with Apache Spark MLlib using the the term frequency–inverse document frequency (**TF-IDF**) benchmark with Wikipedia corpus [2], where SaddlebagX significantly outperforms the Apache Spark MLlib, up to 8× faster. Figure 4 shows the comparison of SaddlebagX with UPC++ using a basic sparse matrix vector multiplication (**SpMV**) benchmark, demonstrating that SaddlebagX introduces negligible overhead.

In conclusion, SaddlebagX is 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 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.

Acknowledgments

The authors would like to thank Prof. Scott Baden and Prof. Xing Cai for their helpful and thorough comments. This work is funded by the Research Council of Norway under PREAPP project (grant n° 231746) and eX3 project (grant n° 270053). The computations were performed on resources provided by UNINETT Sigma2, Norway.

References

[1] J. Bachan, S. Baden, S. Hofmeyr, M. Jacquelin, A. Kamil, D. Bonachea, P. Hargrove, H. Ahmed. UPC++: A High-Performance Communication Framework for Asynchronous Computation. In *IPDPS* (2019).

[1] http://networkrepository.com/networks.php

[2] https://dumps.wikimedia.org/