

FastMR: Fast Processing for Large Distributed Data Streams

Liting Hu, Karsten Schwan, Hrishikesh Amur
Georgia Institute of Technology
{foxting, amur}@gatech.edu, schwan@cc.gatech.edu

Abstract

FastMR is a graph-style framework for stream-oriented applications to realize near real-time streaming data record processing, and more importantly, complex coordinations between those applications. We introduces two components — compressed buffer trees (CBTs) and shared reducer trees (SRTs) to assist with this task. CBTs address the problem of maintaining a significant amount of application-specific “accumulator” state in memory so that streaming data processing can combine current data with historical data. They do so by employing a novel, batch-oriented approach to updating the accumulator state. SRTs are basically P2P-based reducer trees that enable fine-grained queries (both one-shot and continual) to be efficiently rolled up concurrently. CBT’s intermediate results are aggregated to the root of SRT via network aggregation. The roots of SRTs are analogous to vertices and anycast/multicast message transmission between the vertices (roots of SRTs) are analogous to edges in the graph-style computation model.

Categories and Subject Descriptors

D.4.7 [Operating Systems]: Organization and Design
— Batch processing systems, Distributed systems

Keywords

compressed buffer tree, shared reducer tree

Copyright © 2013 by the Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SoCC’13, 1–3 Oct. 2013, Santa Clara, California, USA.
ACM 978-1-4503-2428-1. <http://dx.doi.org/10.1145/2523616.2525946>

Introduction

As shown in Figure 1, FastMR operates in two stages. In the first stage, analogous to “map” in MapReduce, log events are received by web servers and converted into key-value pairs. Each web server then sends these key-value pairs to its locally collocated CBT server for hash-based in-memory aggregation. The second stage is analogous to “reduce” in MapReduce, where all of the web servers are organized into a single topology of distributed processing agents. The system is embedded into this topology in ways that automatically maintain multiple logical reducer trees, each periodically processing and aggregating the data from many CBTs distributed across the cluster.

With an implementation extending the Flume stream processing framework, FastMR operates at large scale and offers rapid query responses, the novel CBT- and SRT-methods bypassing the storage tier. FastMR is evaluated on a cluster of 1000 agents with representative web data inputs. Experiments demonstrate that FastMR can provide significant improvements in both throughput and query latency over traditional MapReduce implementations as well as recent streaming MapReduce variants.

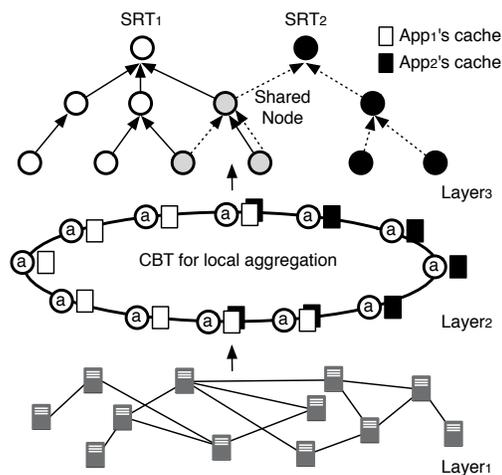


Figure 1: FastMR’s layered structure. In the example, two SRTs for two stream applications are built upon the structured overlay. The gray nodes are the shared analytic nodes. Each root reports each application’s final output.