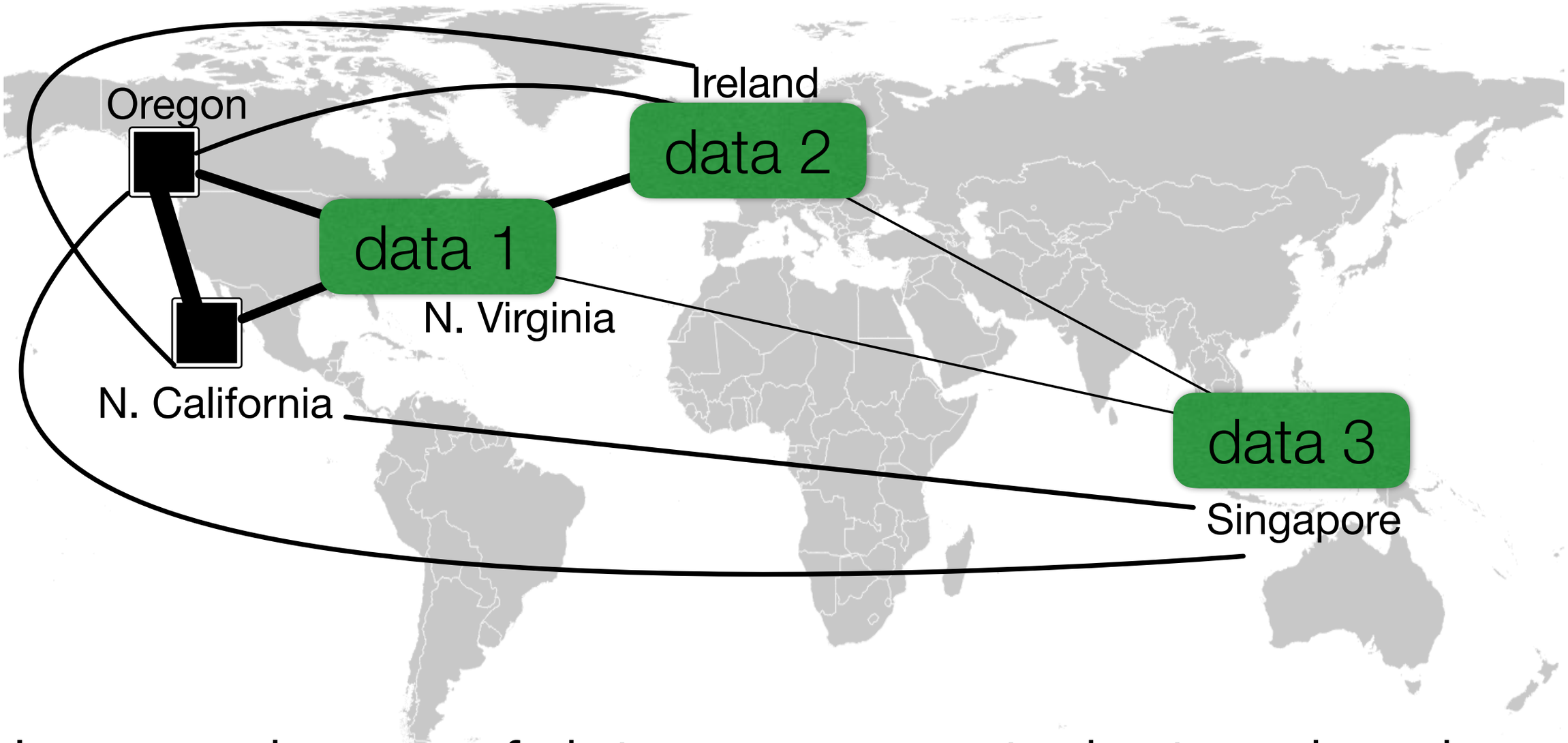
Optimizing *Shuffle* in Wide-Area Data Analytics

Shuhao Liu*, Hao Wang, Baochun Li
Department of Electrical & Computer Engineering
University of Toronto

What is:

- Wide-Area Data Analytics?
- Shuffle?

Wide-Area Data Analytics



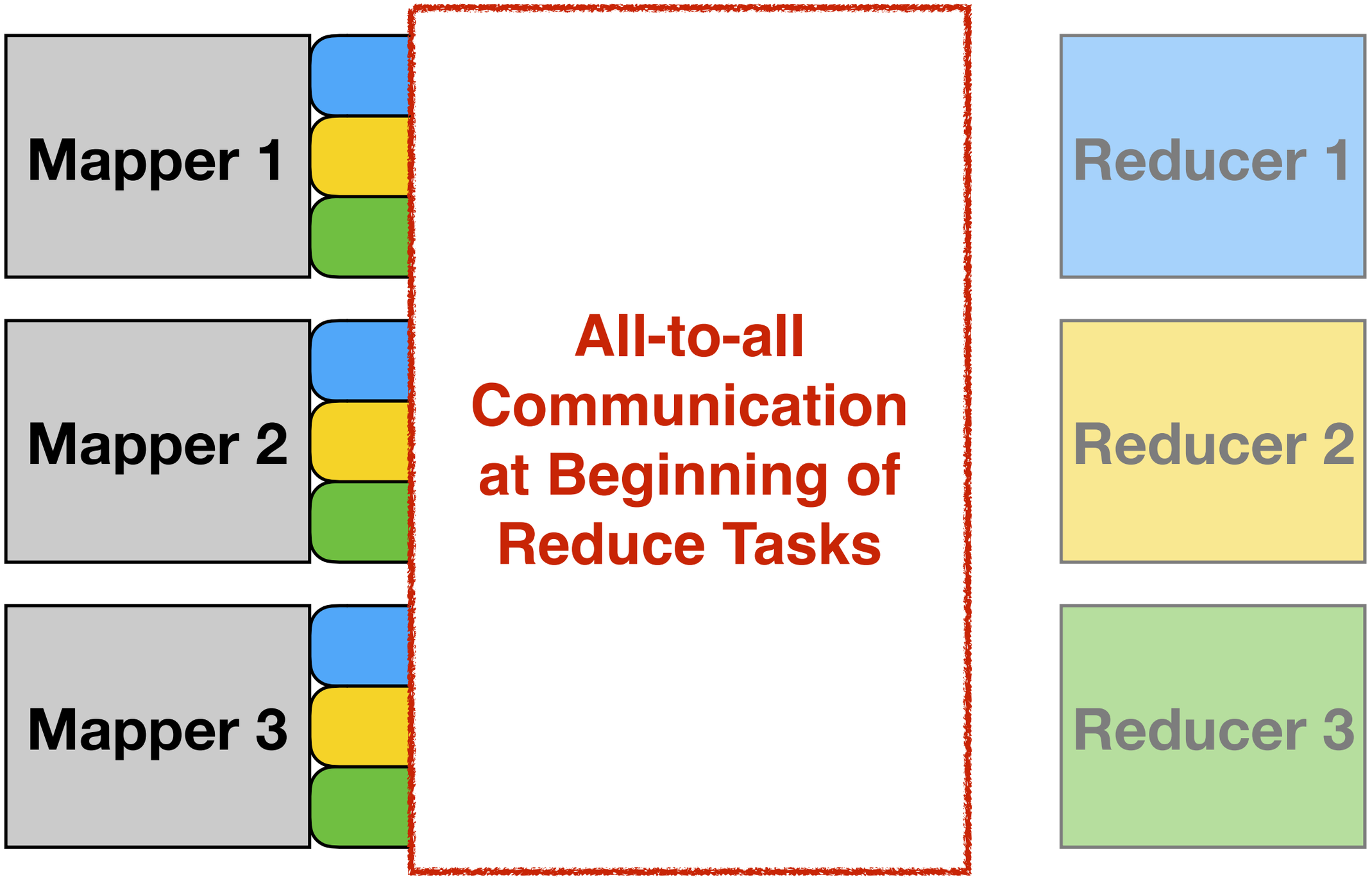
Large volumes of data are generated, stored and processed across geographically distributed DCs.

Existing work focuses on

Task Placement

Rethink the root cause of inter-DC traffic: **Shuffle**

Fetch-based Shuffle

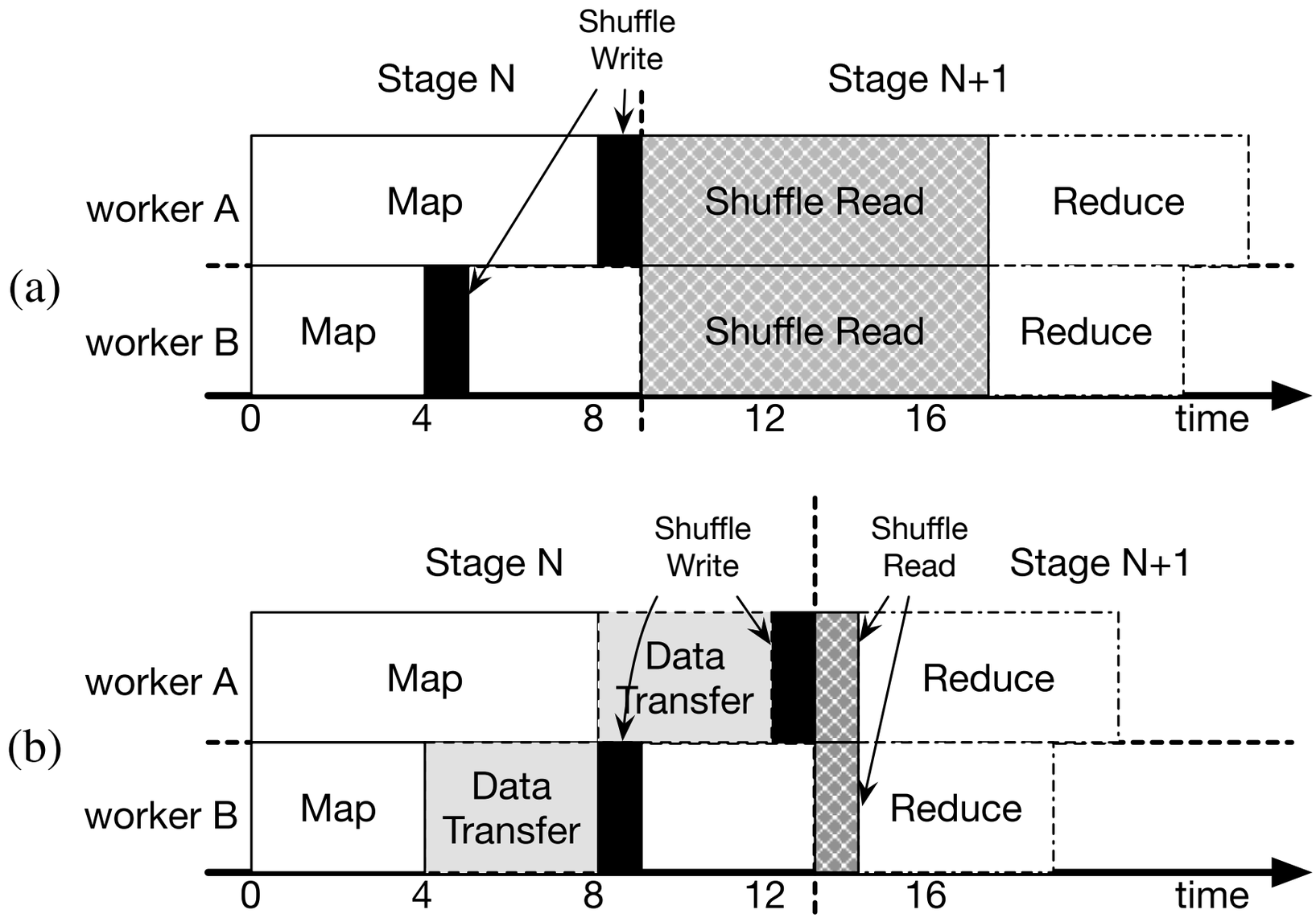


Problems with Fetch

- Under-utilize the inter-datacenter bandwidth
  - Start late: beginning of reduce
  - Start concurrently: share bandwidth
- Need for refetch
  - Possible reduce task failure

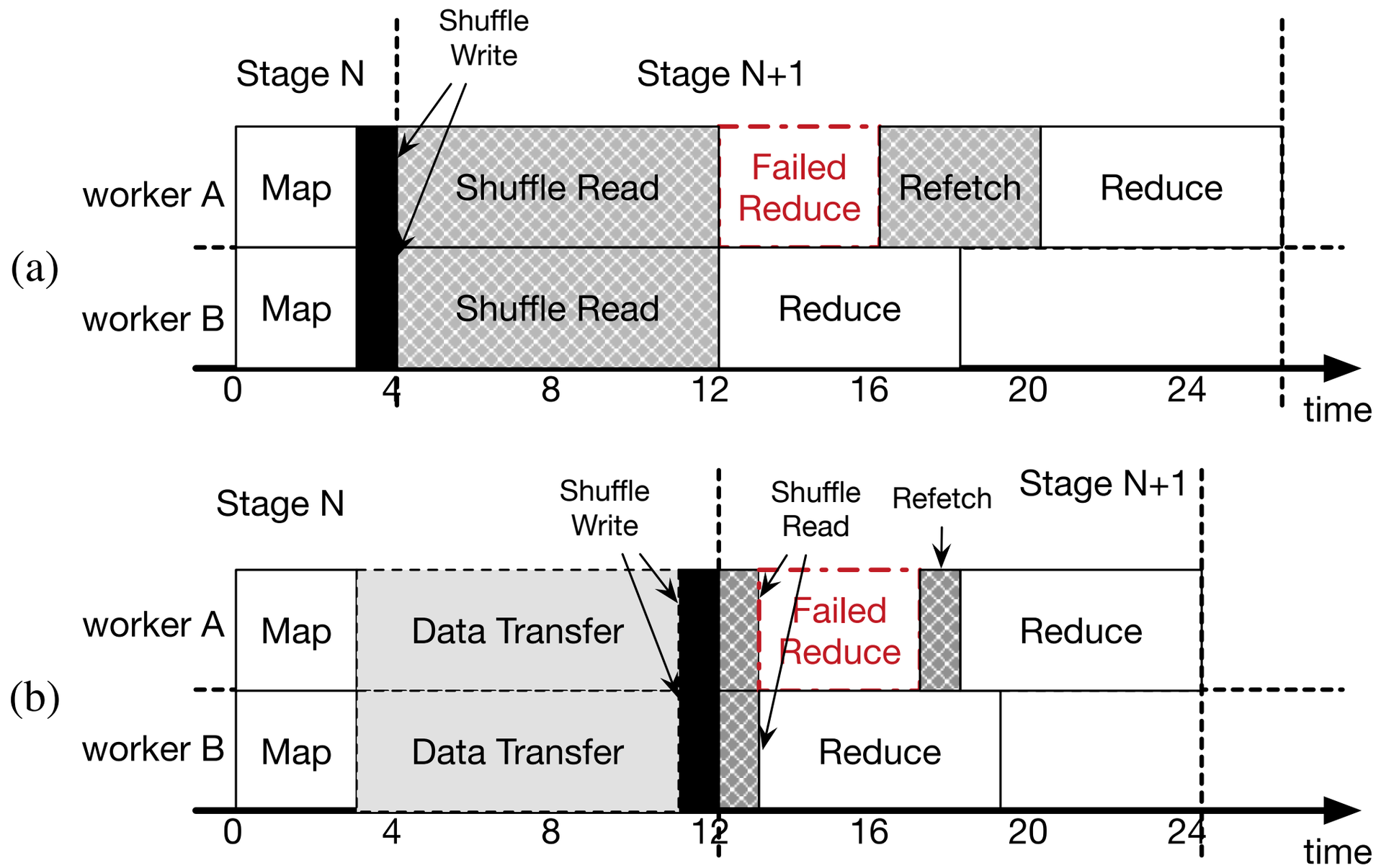
Push-based Shuffle

- Bandwidth Utilization



Push-based Shuffle

- Failure Recovery

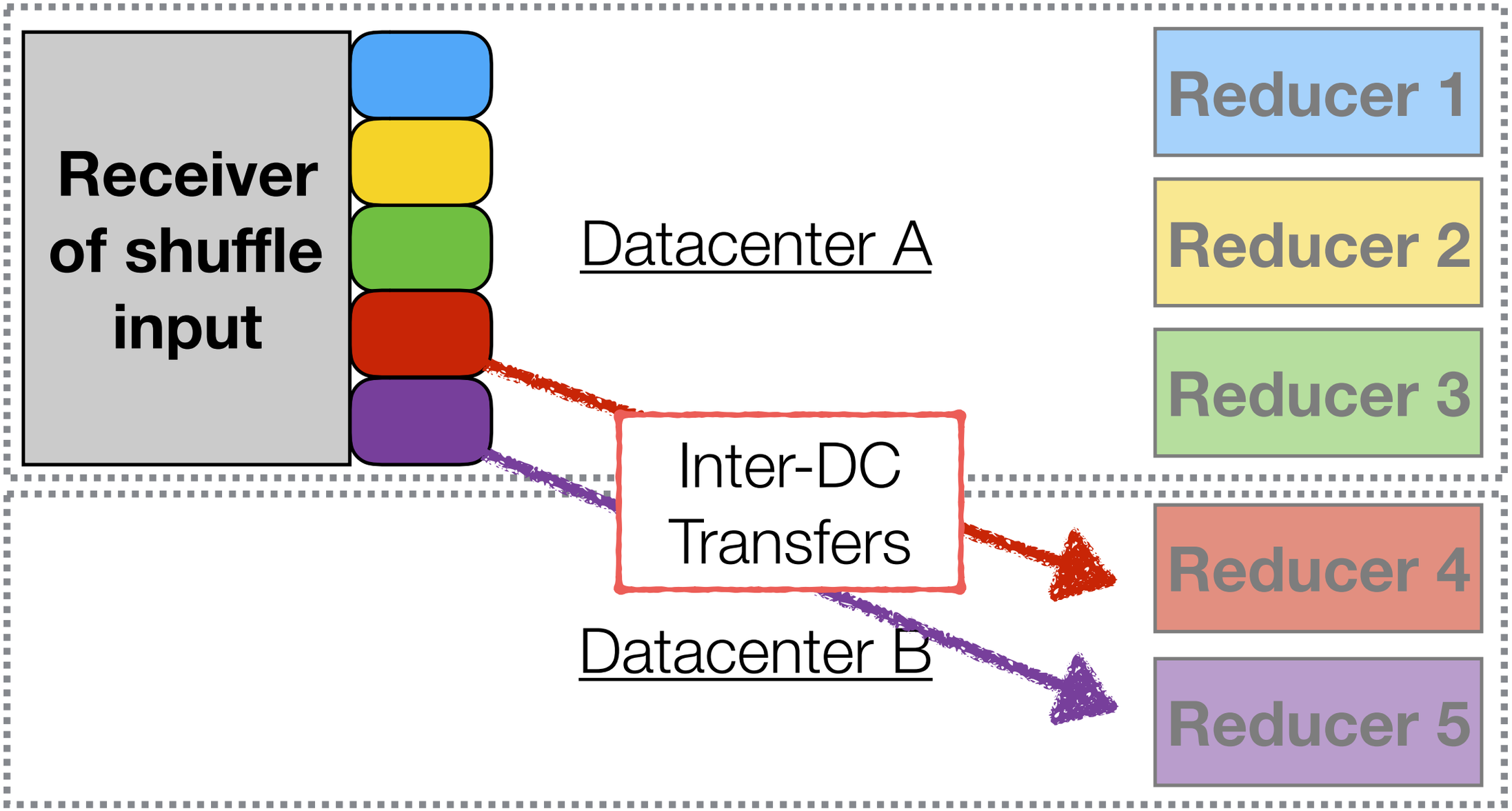


Where to Push?

- Optional: existing task placement algorithms
  - Know reducer placement before hand
  - Require prior knowledge
    - e.g., predictable jobs, inter-DC available bandwidth
- Our solution: **Push/Aggregate**

Aggregating Shuffle Input

- Send shuffle input to a subset of datacenters with a large portion of shuffle input



For any partition of shuffle input, the expected inter-datacenter traffic in next shuffle is proportional to
the number of non-colocated reducers.

Aggregating Shuffle Input

- Send shuffle input to **a subset of datacenters with a large portion of shuffle input**

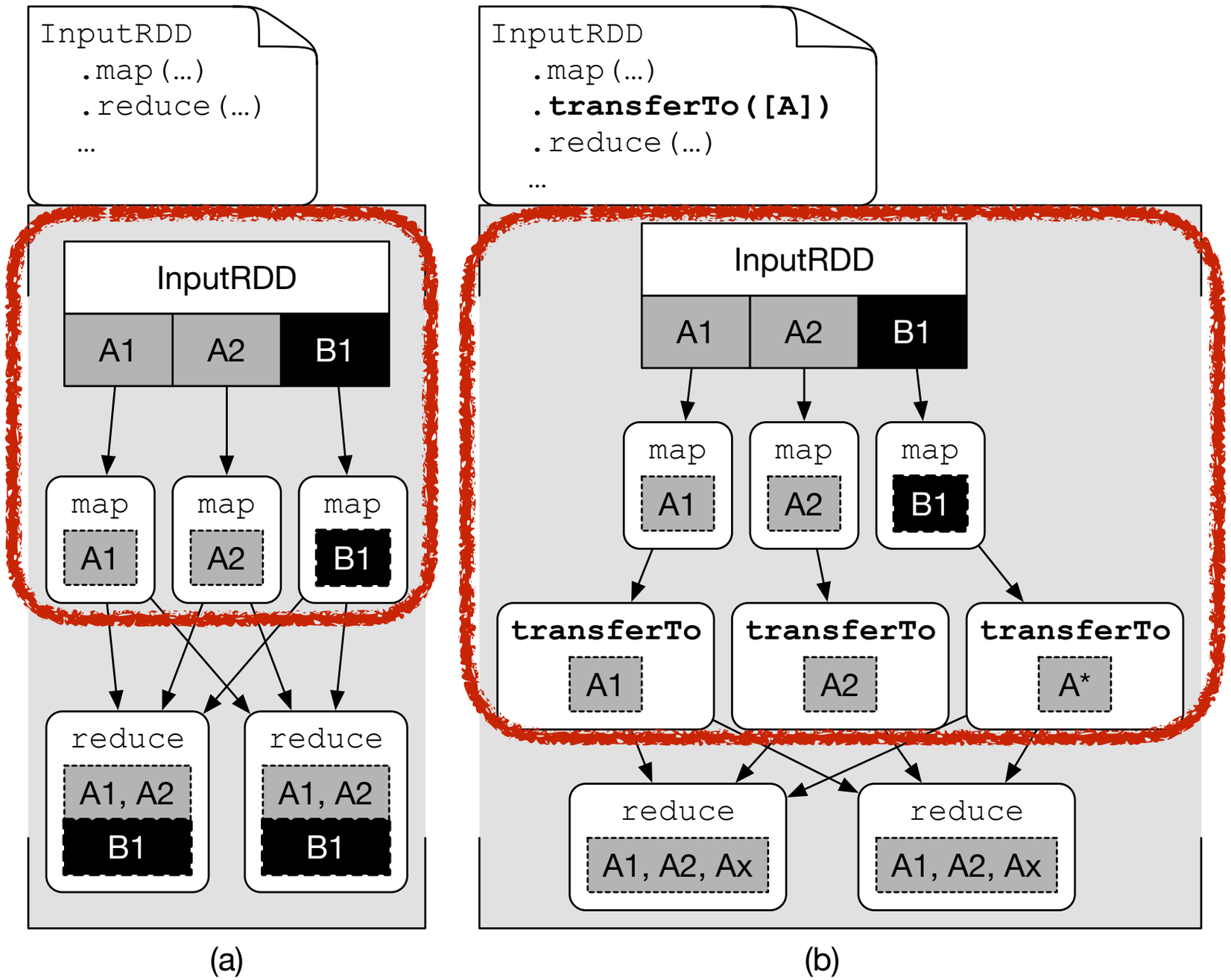
- Reducer is likely to be placed close to shuffle input

- More aggregated data -> less inter-datacenter traffic with reasonable task placement

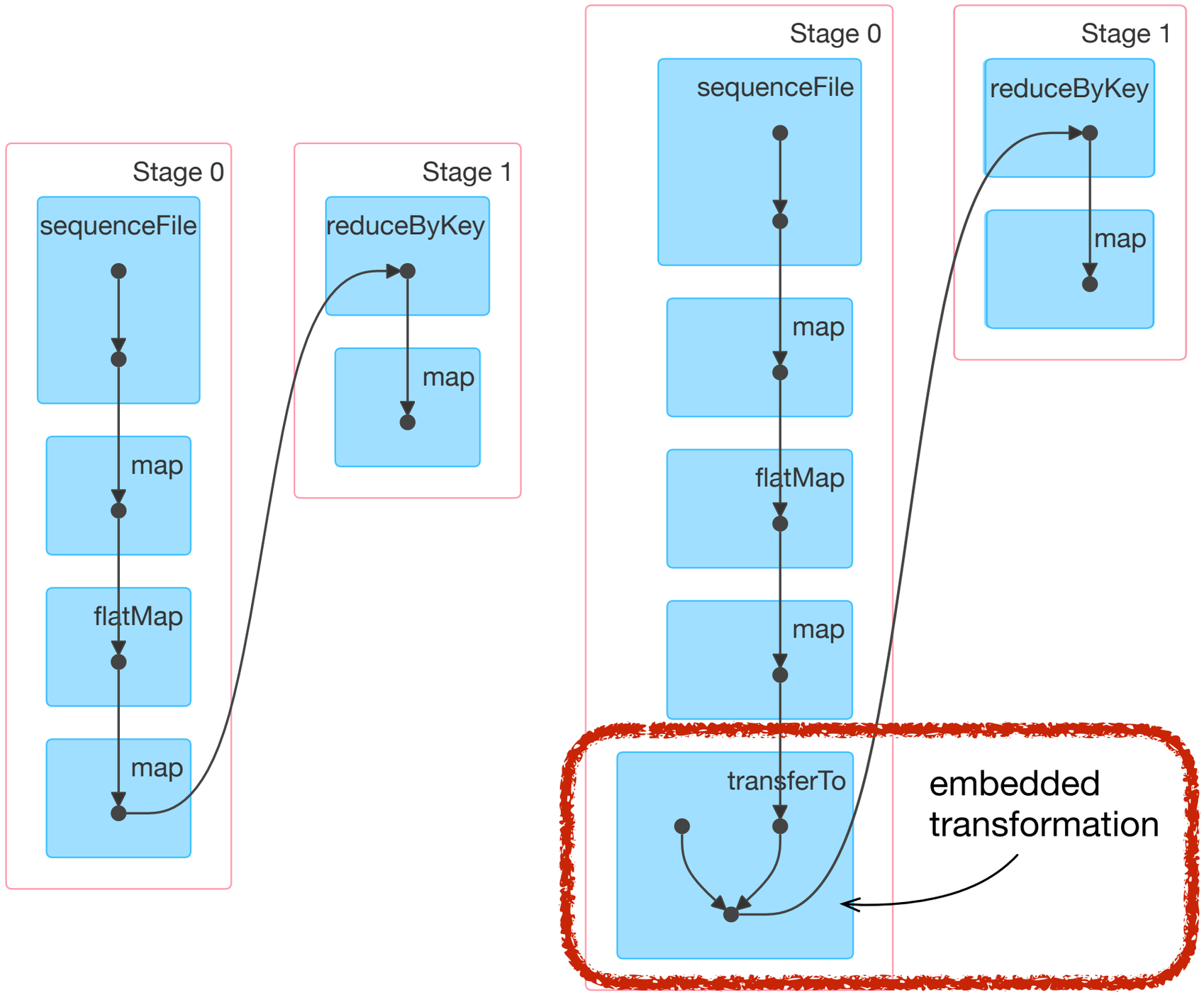
Implementation in Spark

- Requirements:
  - Push before writing to disk
  - Destined to the aggregator datacenters
- `transferTo()` as an RDD transformation
  - Allow implicit or explicit usage

Implementation in Spark

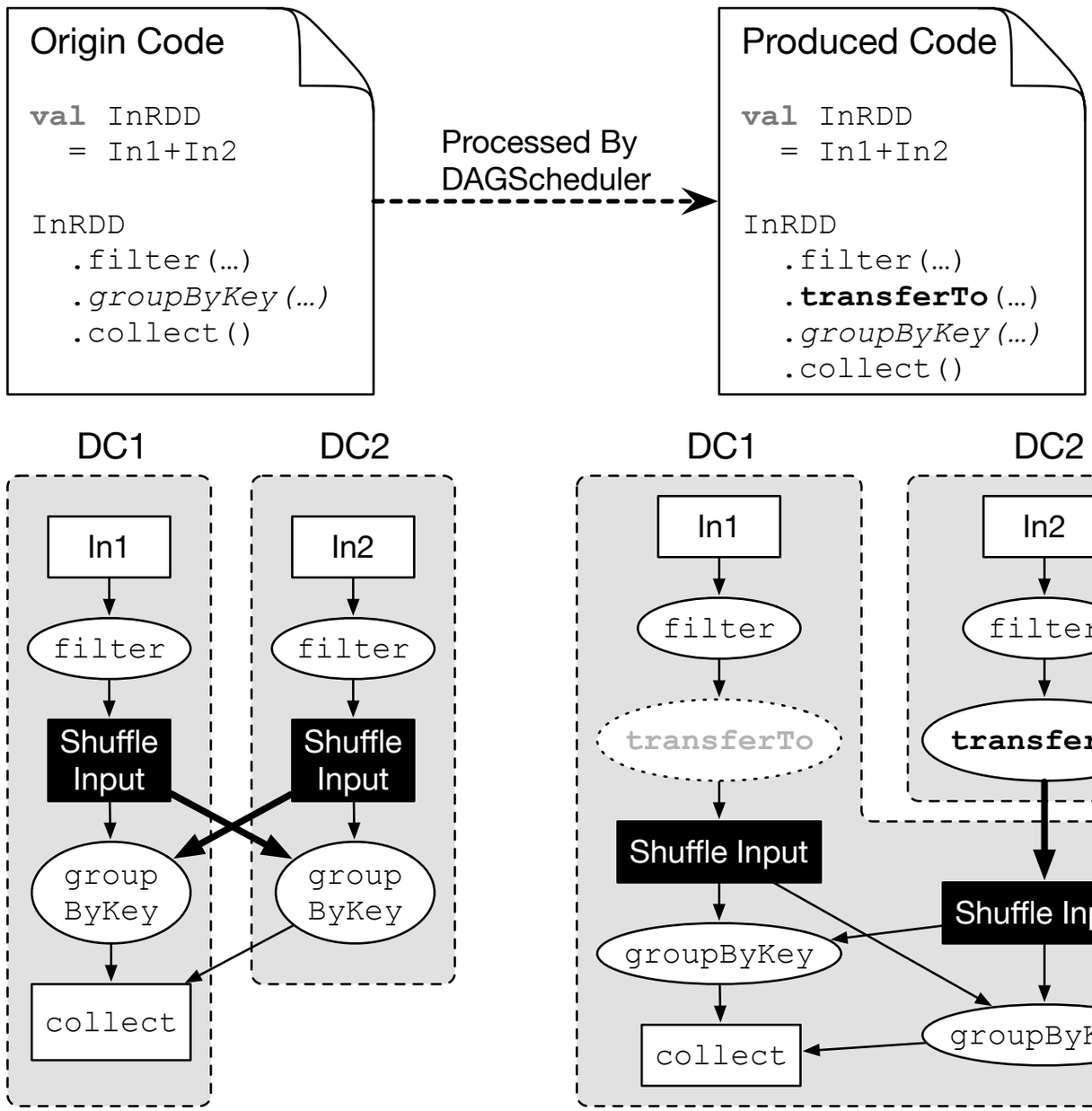


Implementation in Spark



Implementation in Spark

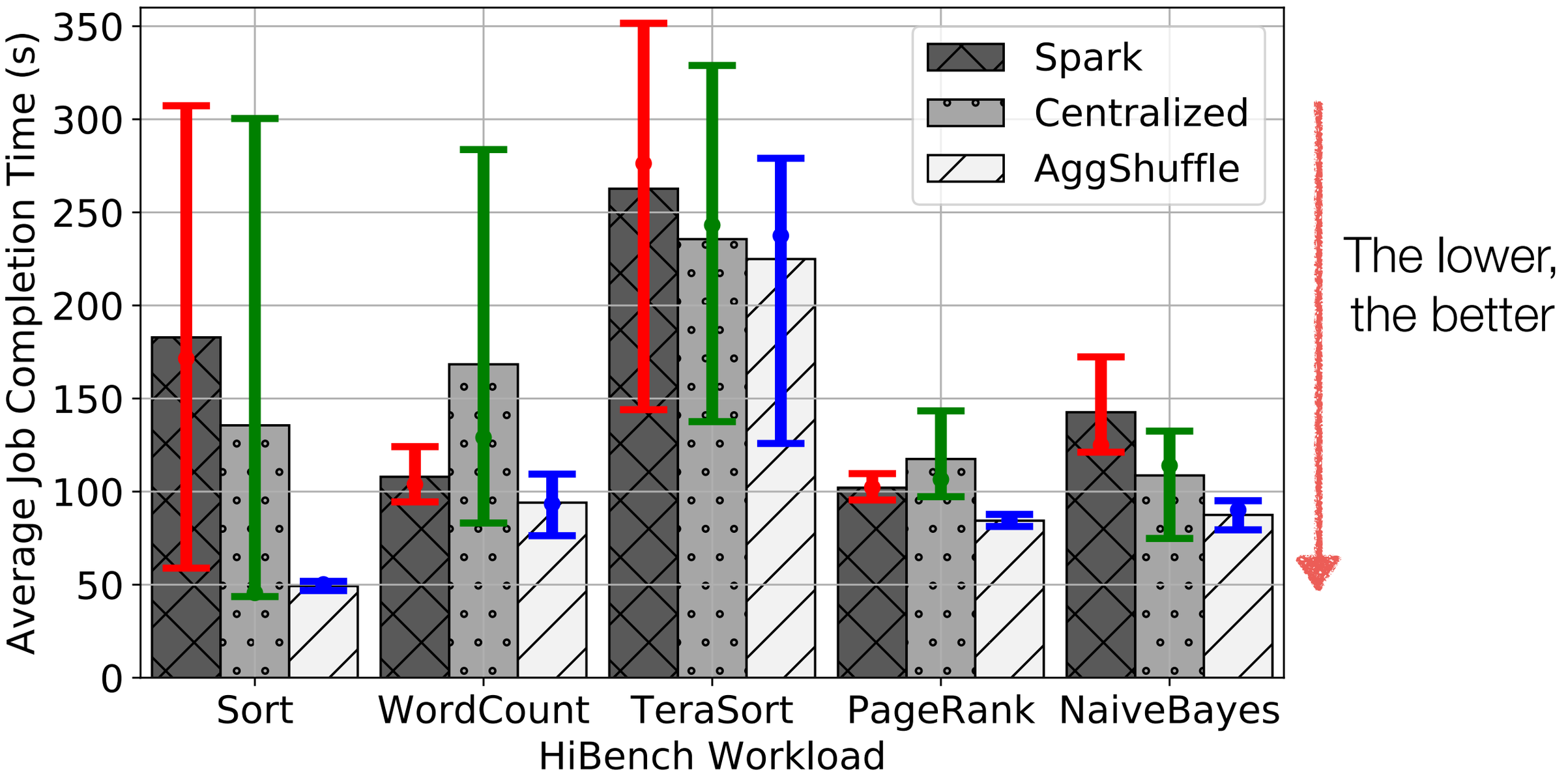
- `transferTo()` implicit insertion



Evaluation

- Amazon EC2, m3.large instances
- 26 nodes in 6 different locations

Performance



Take-Away Messages

- Push-based shuffle mechanism is beneficial in wide-area data analytics
- Aggregating shuffle input to a subset of datacenters is likely to help when you have no priori knowledge
- Implementation in Apache Spark as a data transformation
- Performance: reduced shuffle time and its variance

Thanks! Q&A