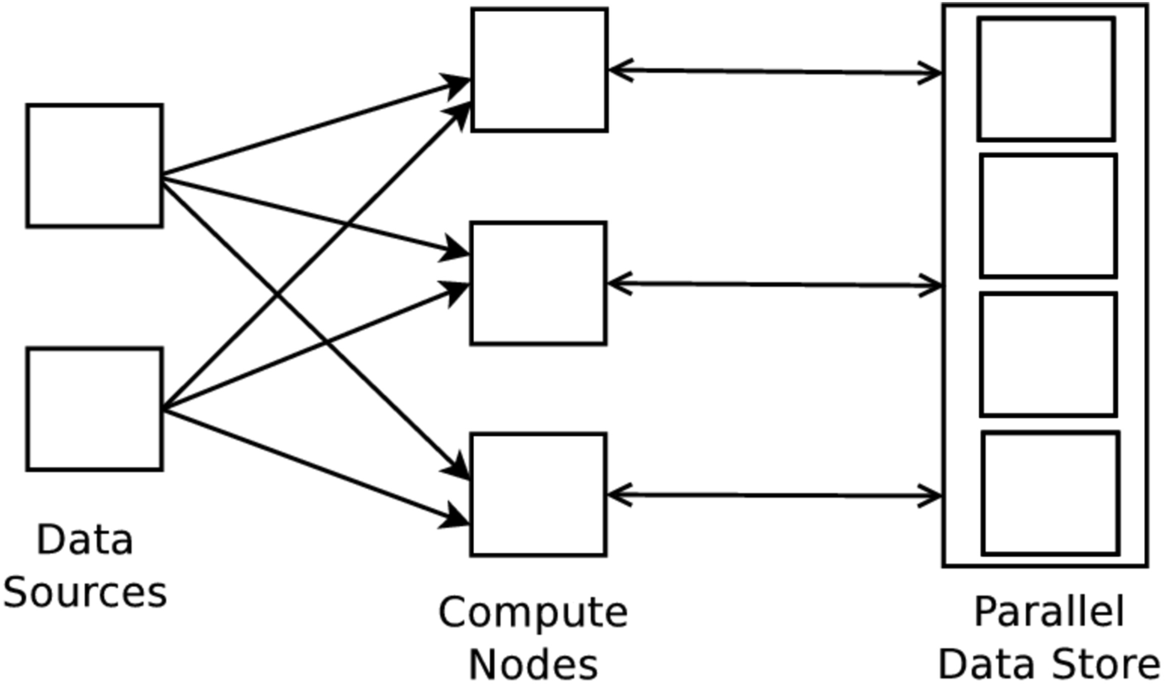
Optimizing Remote Calls in Parallel Data Management Systems

Bikash Chandra

Supervisor: Prof. S. Sudarshan

Architecture Overview

- Read data items from (streaming or stored) data sources
- Compute the function f(k, p), where
  - k- key for fetching values
  - p- list of parameters
- Fetch values from the parallel data store to compute the function

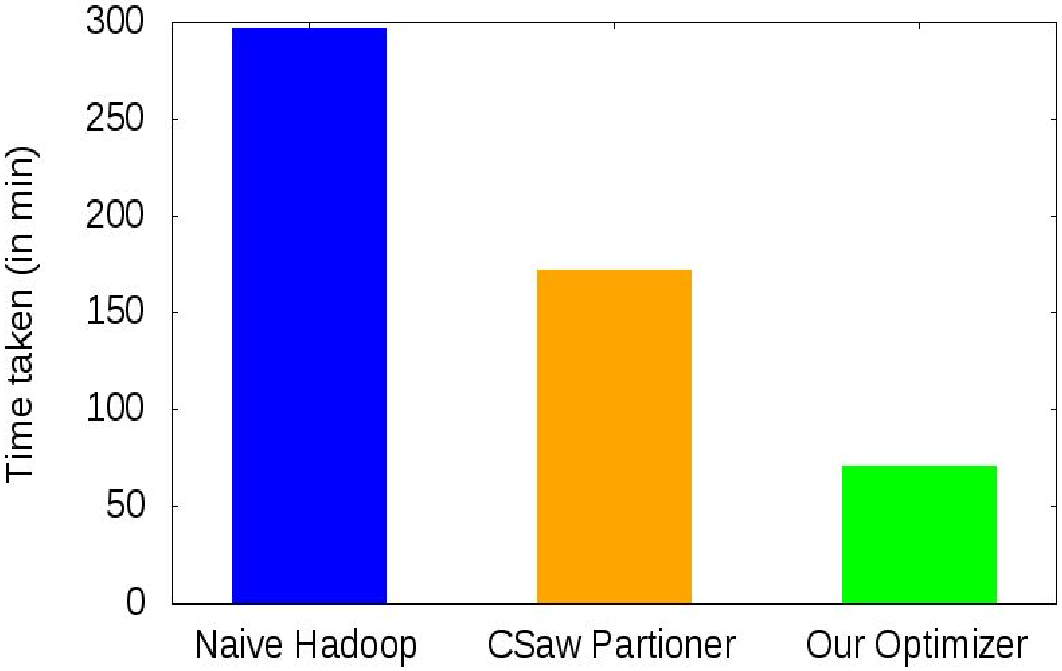


Optimization Techniques

- **Online optimization** - no statistics are available in advance
- **Optimize data access**
  - Use prefetching and batching
  - APIs to enable prefetch calls in a different thread
- **Function execution can be done at the compute nodes or the data nodes**
  - Push non frequent computations to the data nodes
  - For each batch of compute requests data node
    - Consider CPU and network costs and the load at the compute and data node
    - Determines the fraction of computations sent uncomputed to the compute node (in a load balanced way)
  - For frequent keys, fetch and cache the values at the compute nodes
- **Helps mitigate skew**

Performance

- 20 node cluster
- Entity annotation on 35,000 documents sampled from the ClueWeb09 dataset to annotate over 4.5 million entities
- Other experiments show that our techniques provide up to 5 times throughput as compared to a naive implementation


Time taken (in min)
Naive Hadoop
CSaw Partioner
Our Optimizer