A Framework for Data Intensive Computing with Cloud Bursting

Tekin Bicer David Chiu* and Gagan Agrawal
Department of Computer Science and Engineering, Ohio State University, Columbus, OH 43210
*School of Engineering and Computer Science, Washington State University, Vancouver, WA 98686

Introduction and Motivation

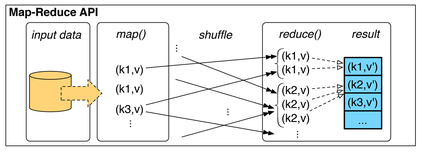
- Tremendous amounts of data to process in today's applications
- Many users have in-house computing resources
  - e.g., local clusters, storage networks
- But the cloud can be used in conjunction to help:
  - Store data remotely
  - Large-scale computation

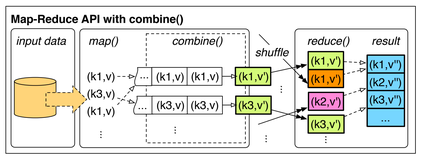
Cloud Bursting Challenges

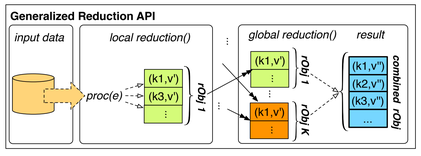
- Workload may demand more resources over time
- How best to manage a cooperation of cloud and local resources?
  - Data distribution
  - Job scheduling

Our Processing Framework vs. Map-Reduce

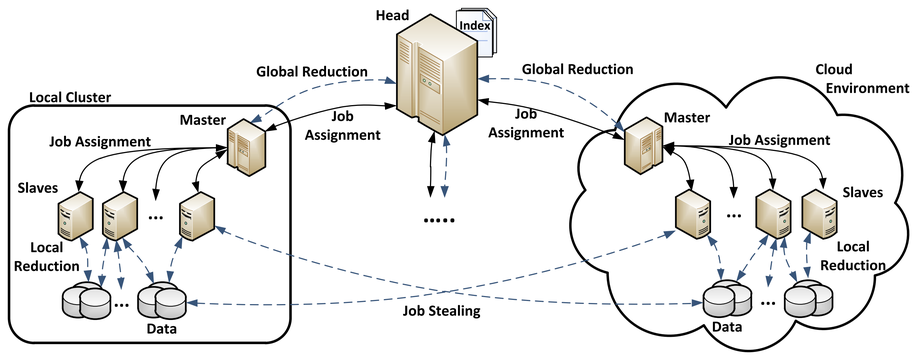
- We focus on a variant of map-reduce class applications
- **Reduction Object**: Data structure which holds the aggregated result from the reduction phases
- **Local Reduction**: The local reduction function specifies how, after processing one data element, a reduction object is updated.
- **Global Reduction**: The multiple reduction objects are combined to form the final results.







Cloud Bursting Processing System



Overall System Architecture

- Data is stored in each cluster, but can be stolen and processed by another cluster
- Data is split into fixed chunks (jobs), and pooled at the **Head Node**
- **Master Node** at each cluster request a bundle of jobs from the Head Node and assigns each job to the slaves
- **Slave Nodes** perform the local reduction on the assigned data chunk.
  - The assigned data may be from a different cluster
- After all data has been processed, the Head Node invokes the global reduction

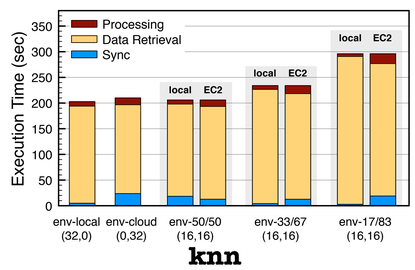
Experimental Setup

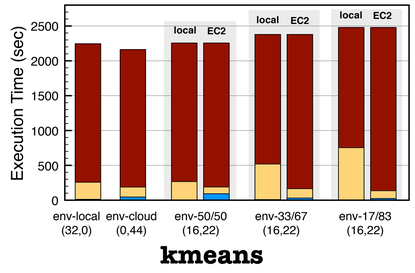
- **Compute Environment:**
  - **Ohio State cluster**
    - Compute Nodes: Intel Xeon (8 cores) and 6 GB RAM
    - Interconnect: Compute nodes are connected via Infiniband
    - Storage: Dedicated 4TB storage node (SATA-SCSI)
  - **Amazon Web Services cloud**
    - Compute Nodes: m1.large instance (2 VC's, each VC contains 2 elastic compute units = 1.7Ghz) and 7.5 GB RAM
    - Interconnect: (High AWS I/O) ??
    - Storage: S3 key-value store
- **Data Intensive Applications and Characteristics:**
  - KNN (low comp, high I/O, small reduction obj)
  - K-Means (heavy comp, low I/O, small reduction obj)
  - PageRank (low comp, high I/O, large reduction obj)
  - 120 GB data for each application

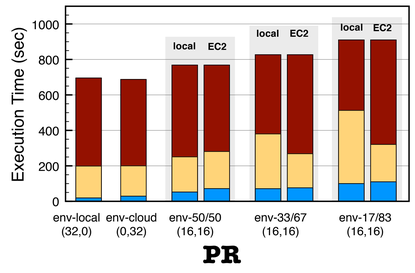
Env.	Data Dist. All app.		Cores knn & pagerank		Cores kmeans	
	Local	S3	Local	EC2	Local	EC2
local	100%	0%	32	0	32	0
cloud	100%	0%	0	32	0	44
50/50	50%	50%	16	16	16	22
33/67	33%	67%	16	16	16	22
17/83	17%	83%	16	16	16	22

Experimental Results

Feasibility







Scalability