A Coding Based Optimization for Hadoop

Zakia Asad[1], Mohammad Asad Rehman Chaudhry[2,*], and David Malone[3]

[1]University of Toronto, [2]Soptimizer, [3]Hamilton Institute, Maynooth University

Abstract

- Big Data resulting in movement of massive volumes of data within data center
  - Testing the limits of data center networks
- Reducing volume of communication is pivotal for embracing greener data exchange
- We propose the use of coding technique as a means of dynamically-controlled reduction in volume of communication
  - Works in tandem with software-defined network control
- We implement a proof-of-concept prototype and a testbed implementation for Hadoop MapReduce
- We evaluate performance of the proposed scheme by comparing with:
  - Vanilla Hadoop implementation
  - In-network combiner
  - Combine-N-Code

Motivation

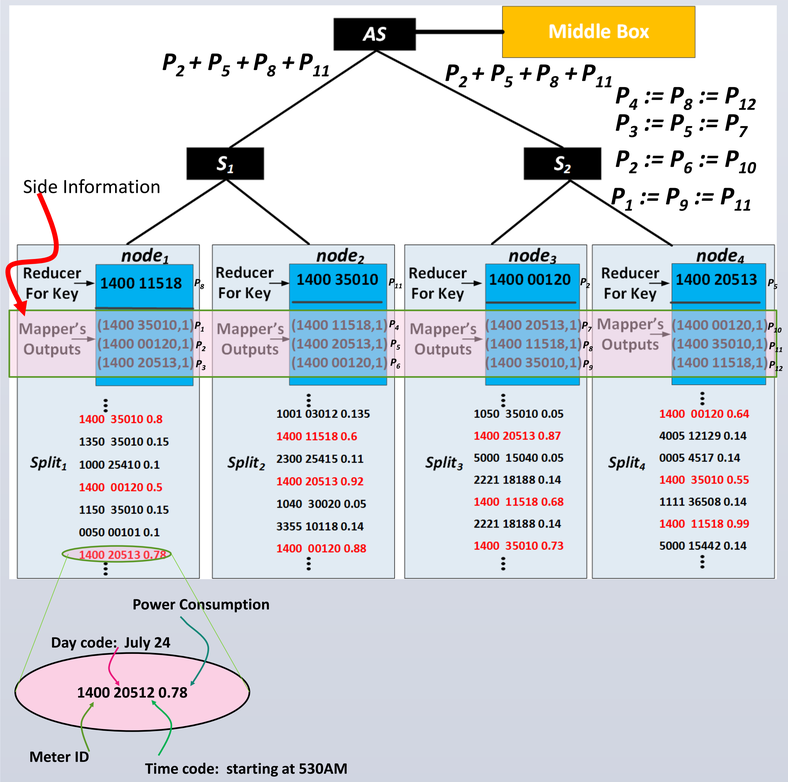
- By 2017 cloud traffic will represent 69% of data center traffic
- An unprecedented growth in data center traffic
  - 76% of the aggregate traffic not exiting the data center [2]
- Highly oversubscribed links
  - 240:1 bisectional bandwidth [3]
    - ❖ Cap the rate at which different servers can communicate with each other [4]

  Reducing volume of communication improves performance [3-4]
- Network devices alone consumed around 3 billion KWh in year 2006 [5]
- Network devices are responsible for 20% -30% of the total energy [6-7]

  Reducing volume of communication improves energy efficiency [8-10]

Hadoop Use Case: Theft detection

- Hadoop MapReduce consists of:
  - Map
  - Shuffle
  - Reduce
- Count the number of times the power consumption was higher than a threshold (0.5).
  - Smart meter with ID 1400
    - ❖ 11518: (April 25th for 30 minute interval starting at 830am)
    - ❖ 35010 (December 16th for 30 minute interval starting at 530am)
    - ❖ 00120 (January 1st for 30 minute interval starting at 930am)
    - ❖ 20513 (July 24th for 30 minute interval starting at 6am)



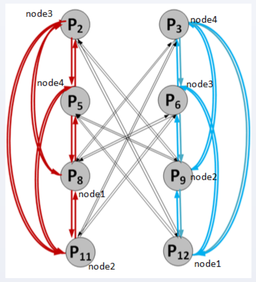
Vanilla Shuffle: 16 link-level packet transmissions at bisection

Coding based Shuffle: 12 link-level packet transmissions at bisection

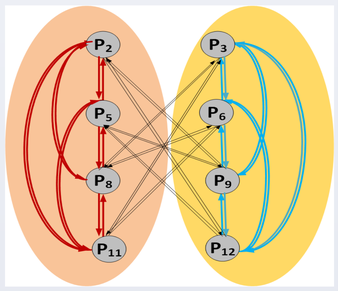
25% reduction in network bisection traffic

Dependency Graph and Cliques

- Captures mutual-Information relationship
- A vertex for each client
  - Each client wants one packet only
    - ❖ If not split it into multiple clients
    - ❖ A directed edge between two clients c_i and c_j
      - – c_i and c_j resides on different nodes(physical machines)
      - – c_j can satisfy demand of c_i



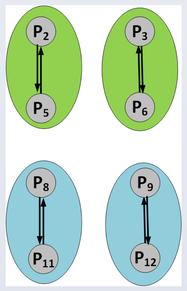
$$P_i \in W_i, \exists\ P \in H_j\ :\ P \equiv P_i$$



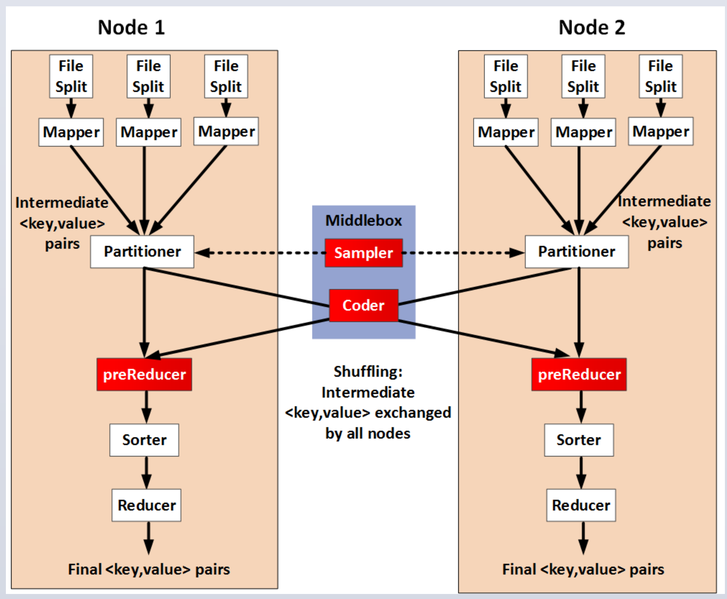
- A clique in the dependency graph represents a group of clients
  - Can be satisfied by one transmission
  - Xoring packets in their want sets

Proposed Solution

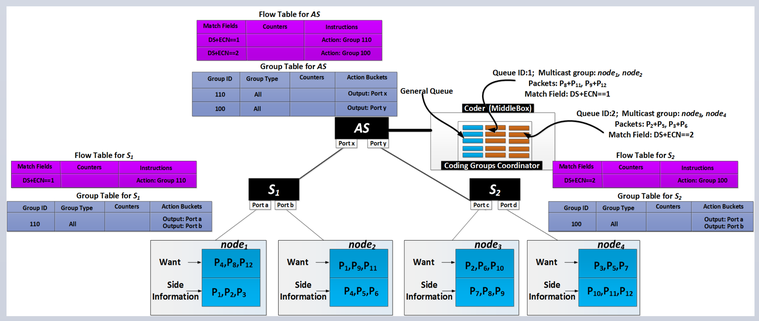
- Given the dependency graph
  - Pack cliques greedily
    - ❖ Staring from the largest clique λ
    - ❖ Clique Packing is NP-Hard
      - – λ=4
- Clique Splitting based on topology
  - Group packets belonging to same subtree only
    - ❖ Reduces the packet overhead
    - ❖ Does not affect solution's validity
    - ❖ A sub clique is also a clique
- Instantaneously decodable
  - No buffering required



Proposed Data Flow for Hadoop [11]



Seamless Integration using Openflow



Performance Evaluation

- Benchmarks used
  - Grep
  - TeraSort

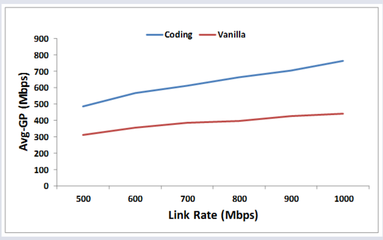
Prototype

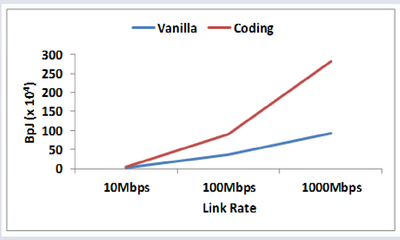
- 8 identical blade-servers each with
  - twelve x86_64 Intel Xeon cores
  - 128 GB of RAM
  - 1 TB hard disk drive
- The servers were arranged in three racks
- OpenFlow enabled IBM RackSwitch G8264 as Top-of-Rack switches
- OpenFlow enabled Pronto 3780 as Aggregation switches
- Job Gain: % increase in the number of parallel jobs that can be run compared to standard Hadoop implementation
- Utilization Ratio: ratio of link-level packet transmissions when employing coding-based shuffle to the number of link-level packet transmission incurred by the Vanilla Hadoop

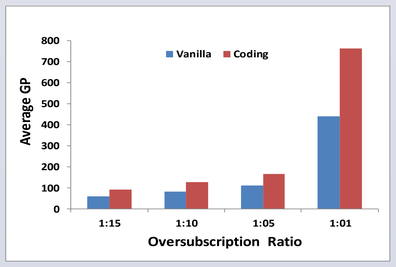
	Job Gain	Utilization Ratio
Terasort	29%	0.71
Grep	31%	0.69

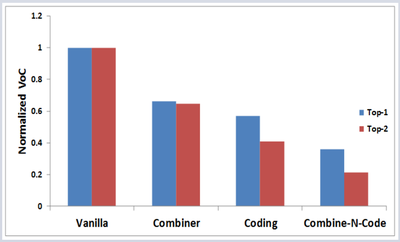
Testbed

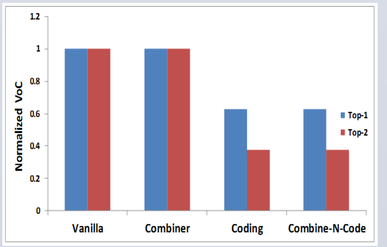
- Citrix XenServer 6.5 with openVswitch
- Eight virtual machines (VMs) each running CentOS 7
- Two commonly-used data center topologies:
  - Tree topology with middlebox at bisection(Top-1).
  - Tree topology with NIC-Teaming and middlebox is placed at first L2-switch.(Top-2)
- Parameters of interest:
  - Volume-of-Communication (VoC)
  - Goodput (GP), defined as the number of useful information bits delivered to the receiver service instance per unit of time.
  - Bits-per-Joule (BpJ), defined as the number of bits that can be transmitted per Joule of energy.











References

[1] Cisco global cloud index: Forecast and methodology, 2013
[2] Camdoop: Exploiting in-network aggregation for big data applications. USENIX NSDI'12.
[3] VI2: a scalable and flexible data center network, ACM SIGCOMM Computer Communication Review.
[4] Towards a next generation data center architecture: scalability and commoditization, ACM PRESTO, 2008.
[5] Report to congress on server and data center energy efficiency: Public law 109-431, Lawrence Berkeley National Laboratory, 2008.
[6] ElasticTree: Saving Energy in Data Center Networks, USENIX NSDI'10
[7] Energy-aware routing in data center network, ACM SIGCOMM workshop on Green networking.
[8] Energy proportional datacenter networks, ACM SIGARCH Computer Architecture News 2010.
[9] Using low-power modes for energy conservation in ethernet lans, INFOCOM 2007.
[10] Traffic merging for energy-efficient datacenter networks, SPECTS 12
[11] Codhoop: A system for optimizing big data processing, IEEE SysCon 2015

Authors would like to thank Kostas Katrinis from IBM Research for his help and support.

Part of the work was performed when the author was with IBM Research and Hamilton Institute.