分散KVSに基づくMapReduce 処理系SSS

中田秀基、小川宏高、工藤知宏
独立行政法人産業技術総合研究所

http://sss.apgrid.org

謝辞
本研究の一部は，独立行政法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務「グリーンネットワーク・システム技術研究開発プロジェクト（グリーンITプロジェクト）」の成果を活用している．

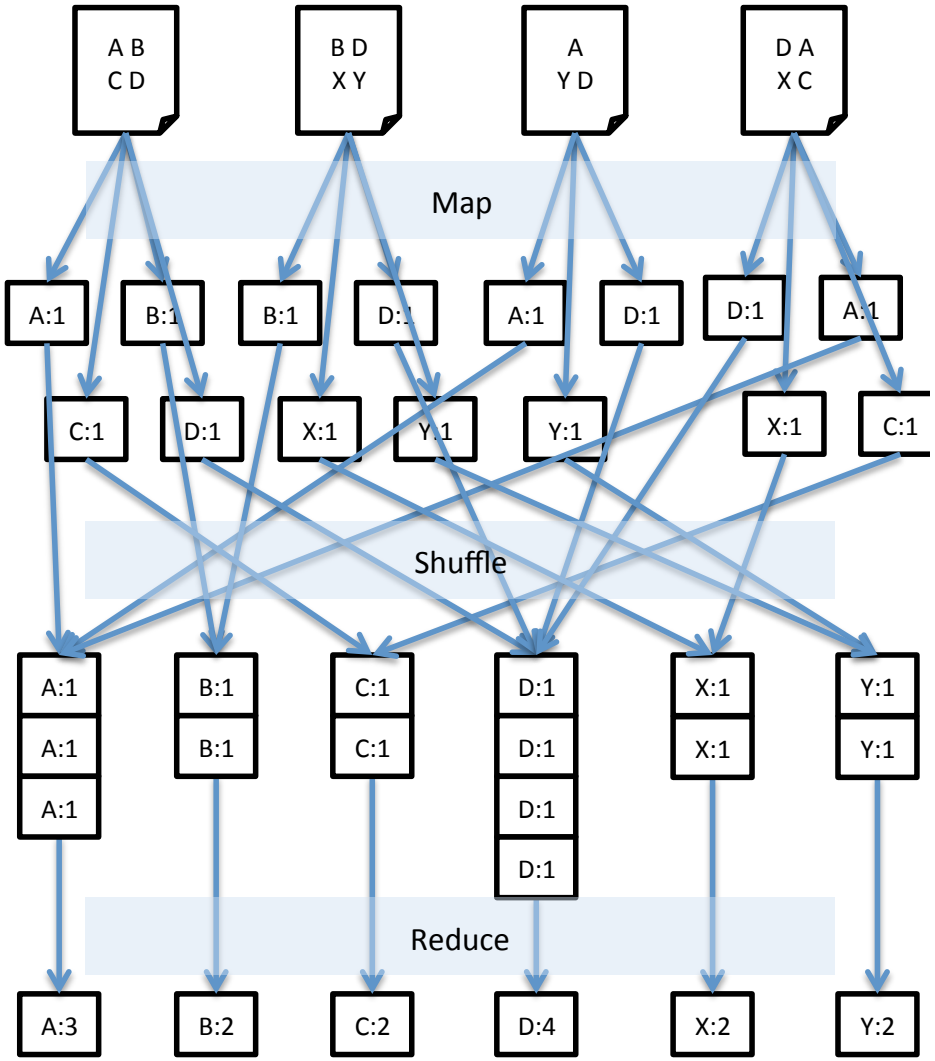
背景と目的

背景

- MapReduce の普及
  - Apache Hadoopの普及による
- SSS [Ogawa, MapReduce11]
  - 複数のMap,Reduceからなるワークフロー処理の効率的な実行

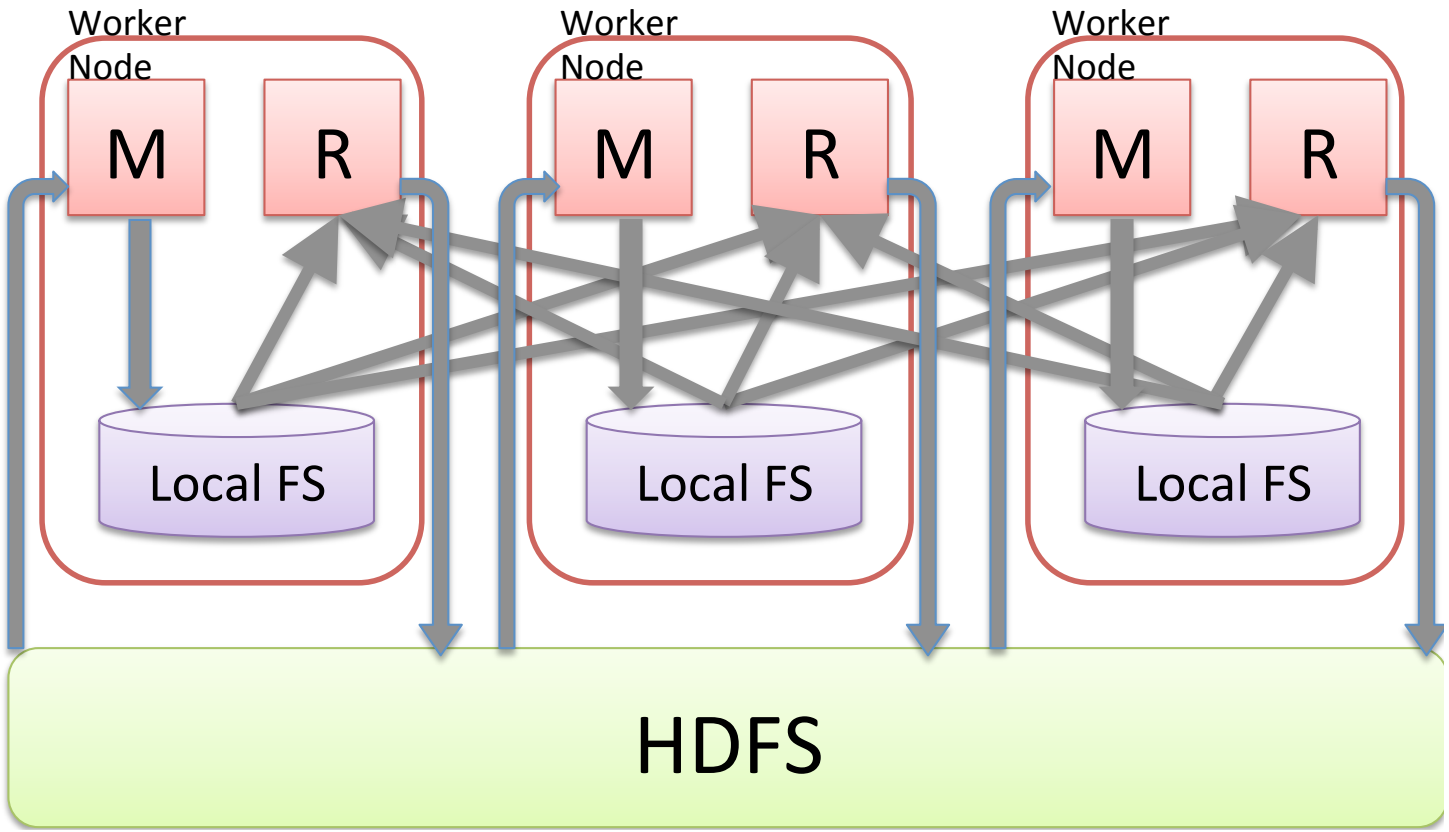
目的

- 設計と実装の詳細
- K-meansによる評価



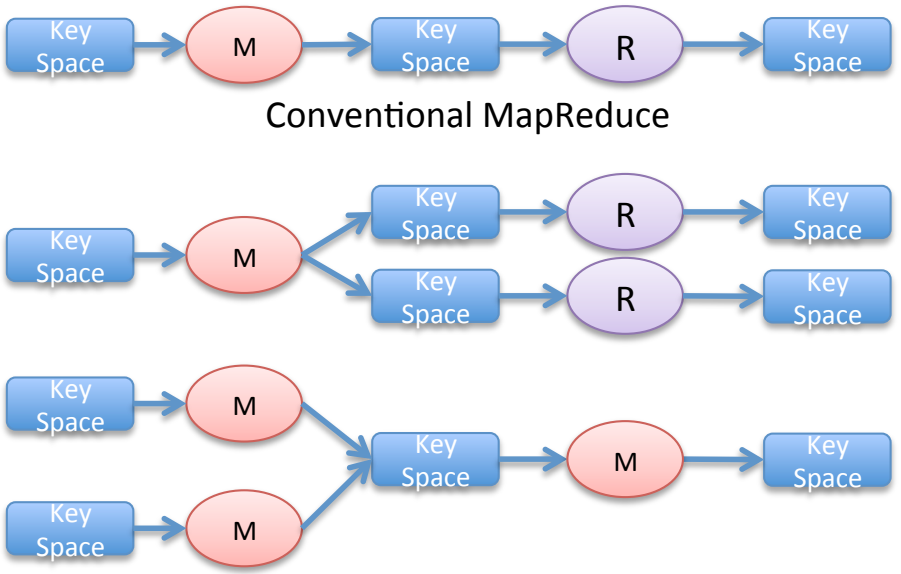
Hadoopの問題点

- Job起動のオーバヘッドが大きい
  - 10秒程度：繰り返し処理には不適
- MapとReduceのデータフローが非対称
  - MapとReduceが必ず1対1に対応



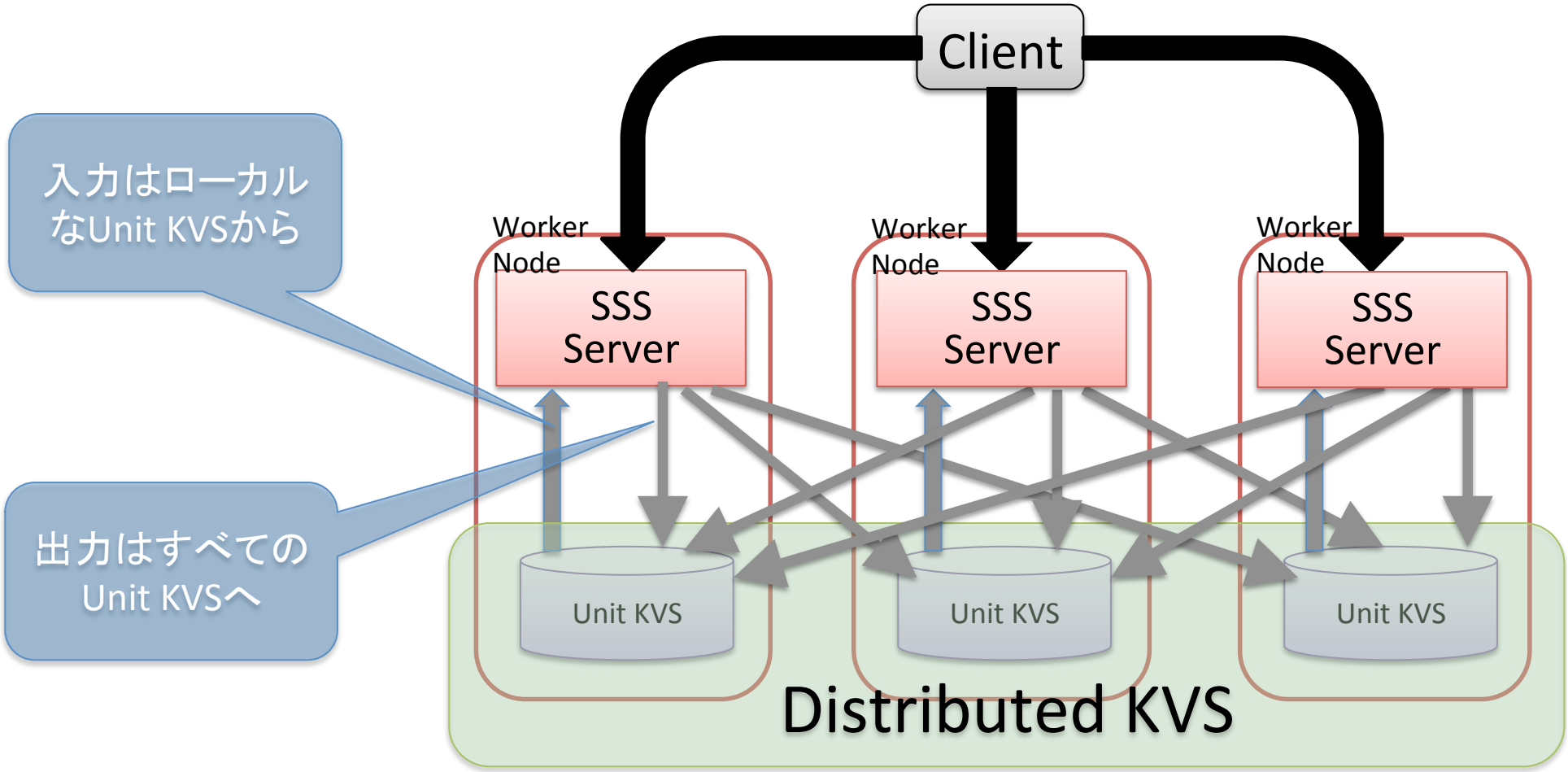
SSSの設計

- 分散ファイルシステムではなく、分散KVSを利用
  - 中間データもすべてそこに書く
  - すべてのノードにデータを薄く広くばらまく
- Owner Computes Rule
  - 自分のノード上のデータのみ処理
  - すべてのタスクをすべてのノードで処理
- Key Spaces
  - HDFSのディレクトリに対応
  - 入出力の対象
  - 節点として利用することで
    ワークフローを実現



SSSの実装

- すべてのノードにSSS サーバとUnit KVS
- SSSサーバがMapper/Reducerを実行
- Unit KVSが集まって一つの分散KVSを構成



分散KVSの実装

要請

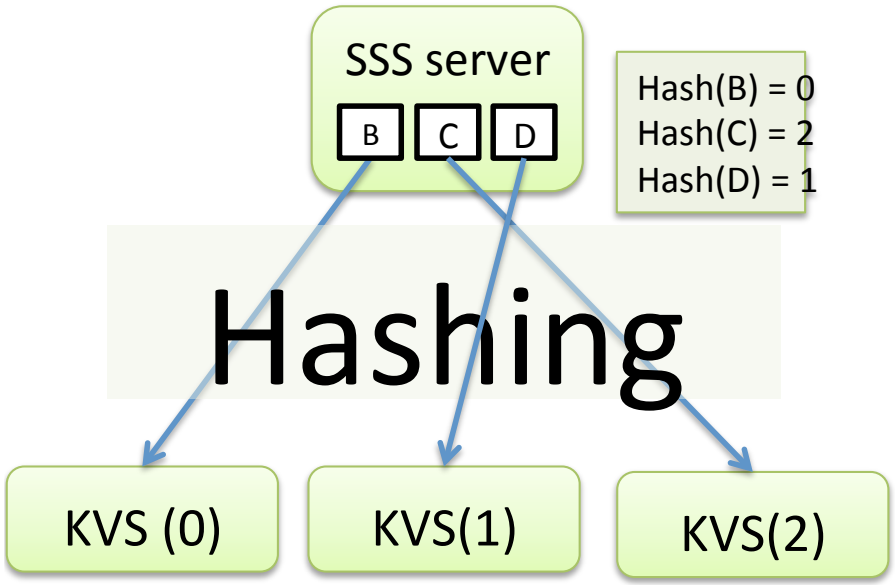
・ノードへの分散

- Key Space
- Multi-Map（一つのキーに対して複数の値を保持）
  - MapReduceでは必須

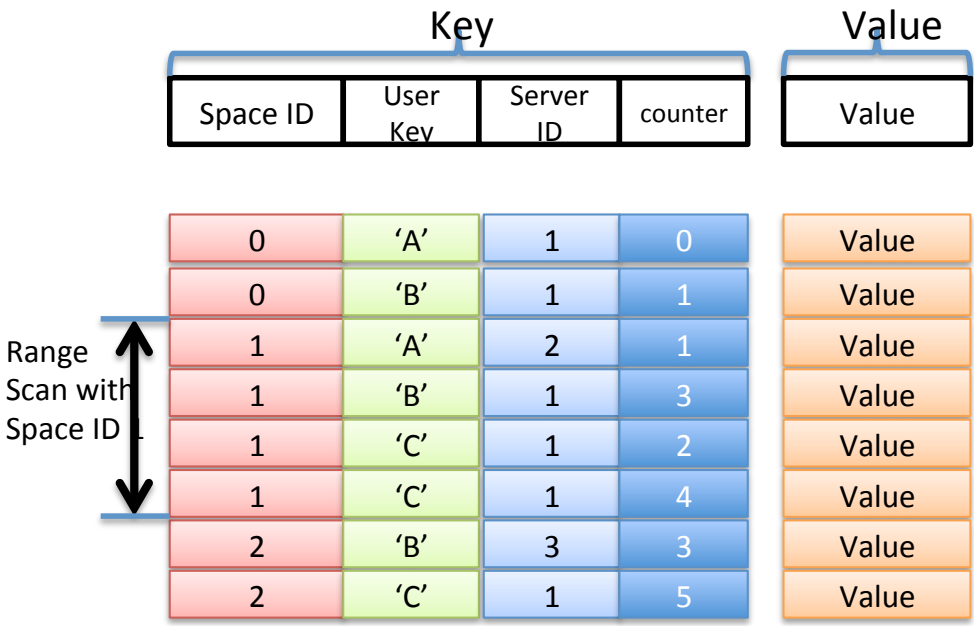
実装

- Tokyo Cabinet/Tyrantを利用
- ハッシュによる分散化
- キーエンコードによる Key Space とMulti-Mapの実現

ハッシュによる分散KVSの実装

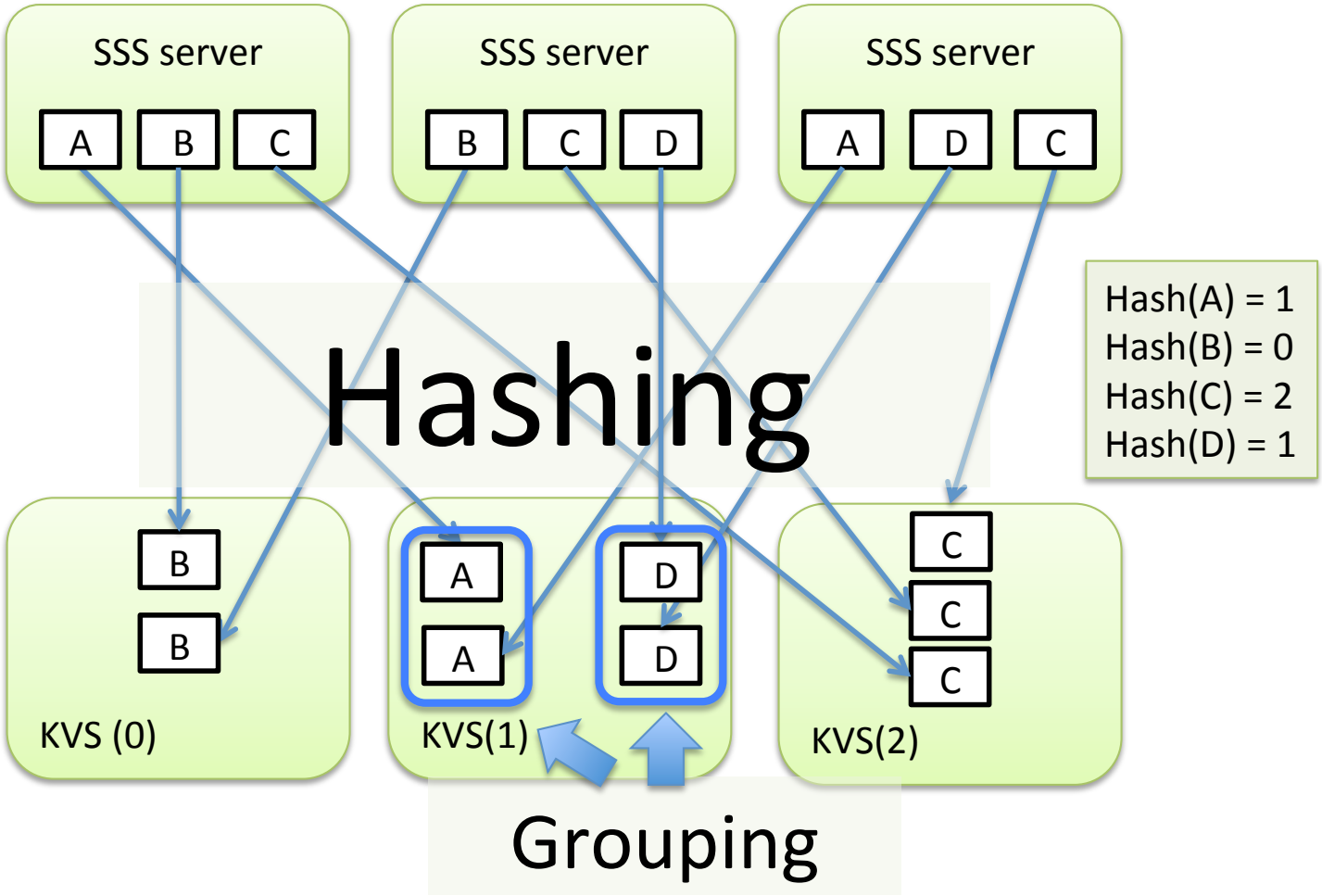


キーのエンコード



Shuffle の実装

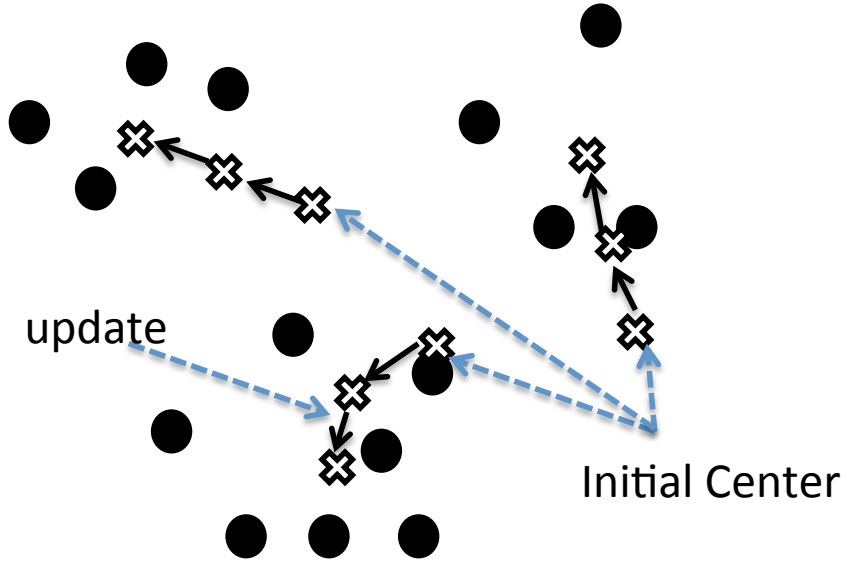
- 2段階でshuffleを実現
  - Hashによるunit KVSの決定
  - Unit KVS内部でのグルーピング

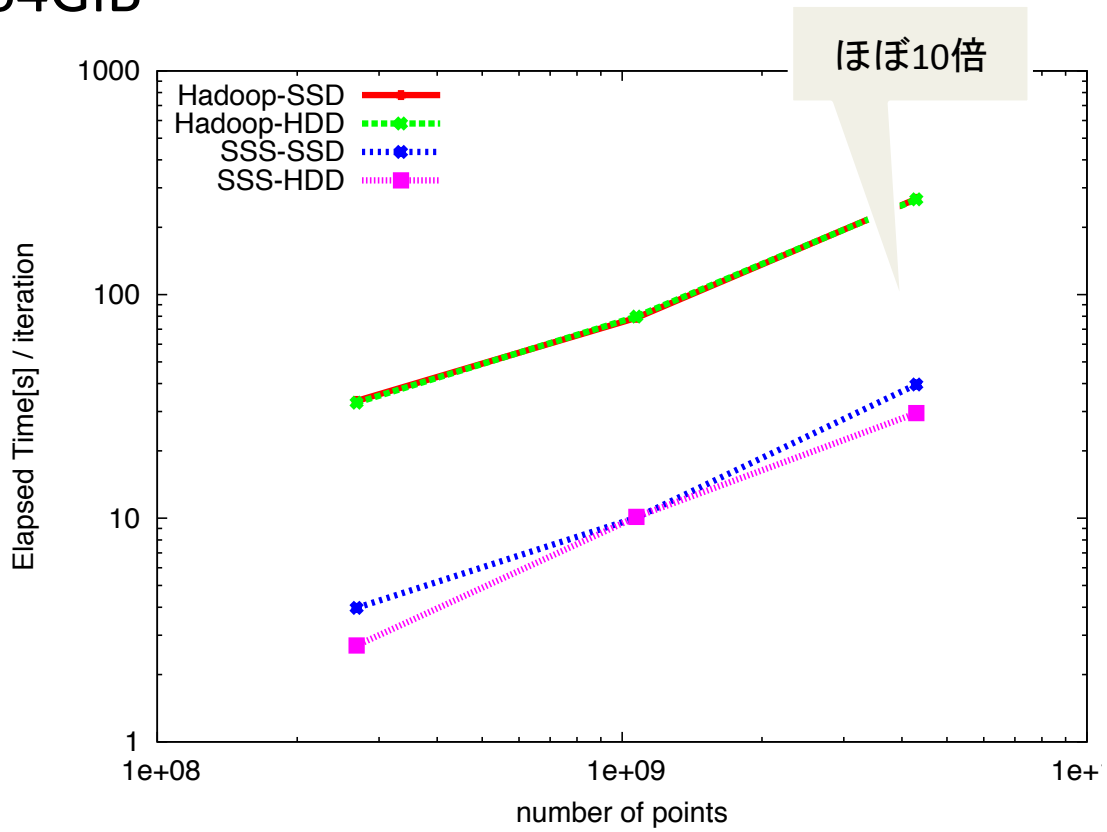


K-meansクラスタリングによる評価

- 大容量データを繰り返しスキャン
- 重心を繰り返し更新
- 収束するまで実行
- 256Mi点、1Gi点、4Gi点を処理
- データ総量は 1GiB, 16GiB, 64GiB

- 評価環境
  - Number of nodes: 16 + 1 (master)
  - CPUs/ node: Intel Xeon W5590 3.33GHz x 2
  - Memory/node: 48GB
  - Storage: Fusion-io ioDrive Duo 320GB
  - NIC: Mellanox ConnectX-II 10G





まとめ

- 分散KVSを基盤としたMapReduce処理系SSSを提案
  - Owner Computes
  - 軽量なスケジューリング
- K-meansによる評価を示した
  - 高速性を確認

今後の課題

- より広範な実アプリケーションでの評価
  - PrefixSpan法[CPSY 中田]
- リアルタイム解析へのシステム拡張