トップエスイー修了制作

Hadoop MapReduce デザインパターンのカタログ化

日本ユニシス株式会社　横石　潔和　kiyokazu.yokoishi@unisys.co.jp

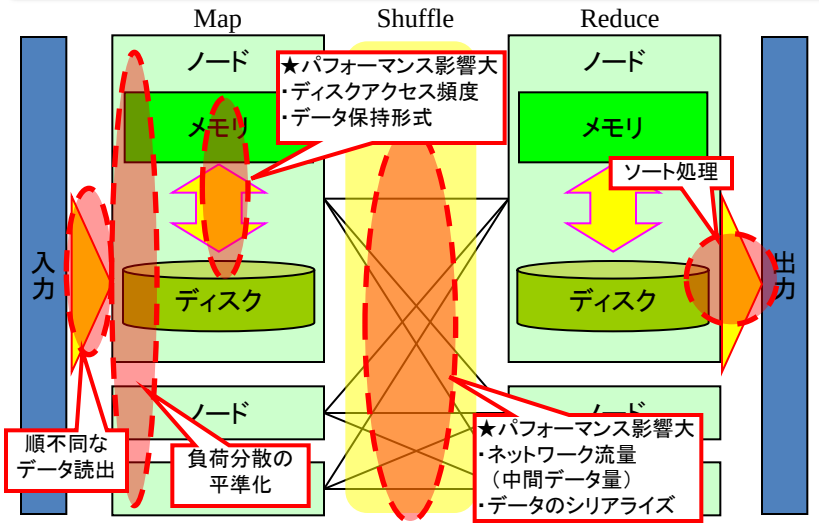
Hadoop活用における問題点

書籍「Hadoop MapReduce デザインパターン[1]」にはMapReduceプログラムの設計テクニックがパターンとして示されており、それを活用することで、品質の高いMapReduceプログラムを比較的容易に作成することができる。しかし、パターンの適用シーンや、パターンを適用することにより発生するメリット／デメリット等の情報が未整理のまま記述されているため、パターンの適用判断を容易に行うことが出来ない。

カタログ化による解決

・書籍に示されているパターンのカタログ化を行い、パターン適用場面や適用による期待効果の把握を容易にする。
・パターンカタログを元に、パターン適用判断チャートを作成し、複数あるパターンの適用判断を漏れなく実施できるようにする。

MapReduceプログラムの難しさ

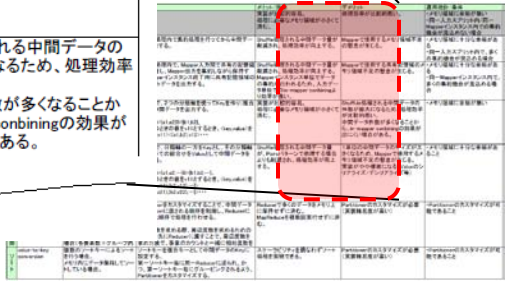


プログラムフレームワークが特異で、かつ、パフォーマンスを意識したプログラム設計が求められる
⇒**非常に難易度が高い**

パターンカタログ

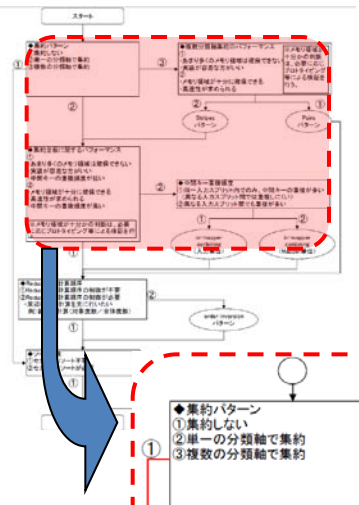
MapReduceプログラミングのパターン（テクニック）をカタログ化しておくことで、適用場面の明確化やメリット／デメリットの比較を容易にする

メリット/効果	デメリット
実装が比較的容易。 処理に必要なメモリ領域が小さくて済む。	処理効率が比較的悪い。
Shuffle処理される中間データ量が削減され、処理効率が向上する。	Mapperで使用するメモリ領域不足の懸念が生じる。
Shuffle処理される中間データ量が削減され、処理効率が向上する。 Mapperインスタンス単位でデータの集約が行われるため、入力データ単位でのin-mapper conbiningより効率が良い。	Mapperで使用する共有記憶域のメモリ領域不足の懸念が生じる。
実装が比較的容易。 処理に必要なメモリ領域が小さくて済む。	Shuffule処理される中間データの件数が膨大になるため、処理効率が比較的悪い。 中間データ件数が多くなることから、in-mapper conbiningの効果が出にくい場合がある。

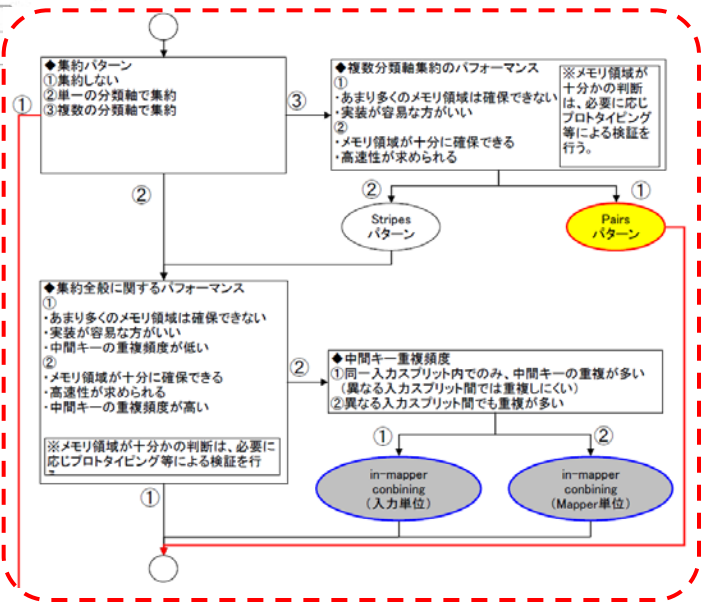


適用判断チャート

複数存在するパターンの適用判断をチャートすることで、「効率の悪い組合せ」の選択を避ける



例）
Pairsパターンを適用する場合、in-mapper convining パターンを併用しても効果が低い



評価

未経験者の作成したMapReduceアプリケーションに対し、カタログ・チャートを利用してブラッシュアップを実施した結果、実行効率を大きく改善することができた

参考文献
[1] Jimmy Lin, Cbris Dyer, 玉川 竜司訳:Hadoop MapReduce デザインパターン MapReduceによる大規模テキストデータ処理, 株式会社オライリージャパン, 2011