ストリーミング型クラスタリングアルゴリズムの性能評価

白幡晃一[†]　鈴村豊太郎[†,††]
佐藤仁[†]　松岡聡[†,†††]

1. はじめに

近年，センサーデバイス技術やネットワーク技術の発達，IT システムの高度化に伴って，リアルタイムに大量のセンサーデータを取得できる時代が到来している．科学技術計算や産業界においてこれらの高速なデータ処理の需要が高まっており，産業界や科学技術分野において様々な応用領域での適応が検討されているが，データ分析の時間が長大化していることが問題視されている．

大規模データ処理の高速化を行うために，ストリームコンピューティングという計算パラダイムがある．従来，大規模データ処理を高速化するにはバッチ処理によって一旦データを格納して処理する方式を取ってきたが，ストリームコンピューティングではデータを受信したと同時に逐次処理していく方式を取る．これによって，出来る限りデータの受信と同時に前処理もしくは本処理を施すことによって，ディスクに格納されるデータを少なくするか，もしくは必要に応じて圧縮処理をリアルタイムに加えることによって，ストレージへの圧縮を防ぐことができる．

しかし，ストリームコンピューティングが必ずしも優位性を持つわけではなく，一般にクラスタリング処理等の反復処理を必要とする計算には不向きである．それでも，近年アルゴリズムの改良が進んでおり，k-means クラスタリング処理においてストリーミング型の近似アルゴリズムが生み出されている．これは，本来のバッチ型処理である k-means に対し，精度を犠牲にする代わりにワンパスで処理を行うというものである．

我々は，k-means 処理においてストリーミング型処理とバッチ型処理の性能を比較することにより，ストリーミング型処理において平均 1.37 倍の高速化を，また結果精度を示すコストも厳密解に比べ平均 1.13 倍と許容できるものとなることを示した．

† 東京工業大学
†† IBM 東京基礎研究所
††† 科学技術振興機構

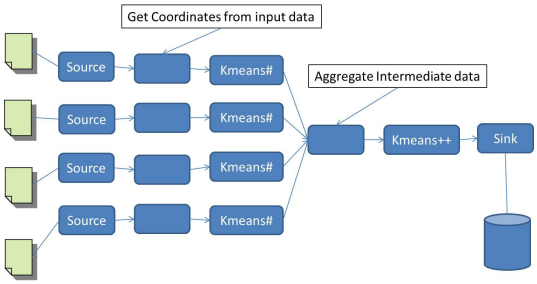

図 1　ストリーミング k-means 近似処理の概要

2. ストリーミング型およびバッチ型クラスタリング処理

2.1 ストリーミング型 k-means 近似アルゴリズム

ストリーミング向けの k-means の近似アルゴリズムの一つに，Nir Ailon らによるストリーミング分割統治クラスタリングアルゴリズム[1] がある．これは，一旦入力データを分割し，各分割データに対して k-means の近似アルゴリズムを適用し，その結果を集約したものに対して再び k-means の近似アルゴリズムを適用する，というものである．一度目の k-means 近似には k-means# と呼ばれる $3k \cdot \log k$ 個の中心を出力するアルゴリズムを，二度目には kmeans++ と呼ばれる k 個の中心を出力するアルゴリズムをそれぞれ使用することにより，ワンパスのストリーミングアルゴリズムが実現される．

2.2 ストリーミング型クラスタリング処理

ストリーミング型の k-means クラスタリング処理は**図 1** のように実行される．まず，入力データから必要な座標データを抽出する．続いて，抽出された座標データに対し各コアが k-means# を実行する．各コアの出力結果を Aggregate 処理で中間結果を集約する．最後に，集約された中間結果に対して kmeans++ を実行し，最終結果を得る．ストリーミング処理はリアルタイム性に特化しているため，オンメモリで実行されることが特徴である．

2.3 バッチ型クラスタリング処理

バッチ型の場合は通常の k-means が実行される．k-means アルゴリズムは，始めランダムな箇所に置かれた中心に対し，各点が距離計算を行い自分に一番近い中心を求めた後，各点によって求められた中心ごとにそれらの平均を計算し新たな中心を求める，という操作を結果を収束するまで反復して行う．バッチ処理の場合はストリーミング処理とは異なり，ディスクへの読み書きが発生する．

3. 評　　価

Twitter データを使用し，各 Tweet の位置情報を用いて地理的なクラスタリングを行う．ストリーミング処理とバッチ処理の性能を比較するため，ストリーミング処理には IBM System S[2] を，バッチ処理には Hadoop を使用し，実行時間と結果精度の比較を行った．なお，Hadoop では C++ラッパーライブラリである Hadoop Pipes を使用した．

入力データには 1 コア当たり Twitter のログデータ 1 日分を集めたファイル約 2GB を使用し，コア数とファイル数を一定の比率で増加させながらスケーラビリティを調べた．ログデータから各 Tweet の経度と緯度の情報をアメリカ国内に絞って抽出した．1 日分のログデータから抽出される位置情報の総数はおよそ 10000〜13000 件であった．CPU は AMD Phenom(tm) 9850 Quad-Core(1.25GHz) を，メモリを 8.17GB 搭載したシングルノードで行った．Hadoop で使用した Jave は IBM Java x86_64-60 である．

実行時間の計測は，System S では計測用のストリームを別のノードで実行させ，また Hadoop では time コマンドを使用して行った．なお，Hadoop では入力データから座標データを抽出する部分と k-means を実行する部分を二つのジョブとして連結し，両方のジョブが終了するまでの時間を計測した．また，計算精度の計測は Hadoop，System S での実行結果のコストを比較し，Hadoop に対する System S のコスト比率を求めることによって行った．ここで，コストとは各点から最も近い中心までの距離の全点の総和の値であり，コスト比率が 1 に近いほど精度が高いと言える．

中心の数を 2 から 128 まで，およびコア数を 1 から 4 まで変化させたときの実行時間を**図 2** に示す．図 2 では中心の数が 2 と 128 の場合しか掲載していないが，他の数の場合でも同様の結果が見られたので割愛する．結果より，ストリーミング処理の場合にバッチ処理と比べ，平均 1.37 倍の性能を示した．また，コア数の増加によって，1 コア当たりの入力サイズがほぼ同一であるにも関わらず実行時間が増加している現象が見られる．この原因の解明は今後の課題の一つである．また，コストについて，1 コアの場合のそれぞれのコスト，および Hadoop に対する System S のコスト比率を**表 1** に示す．ストリーミング処理の場合のコストはバッチ処理に比べ，平均 1.13 倍となった．これらの結果より，ストリーミング型クラスタリングの有効性をうかがうことができる．

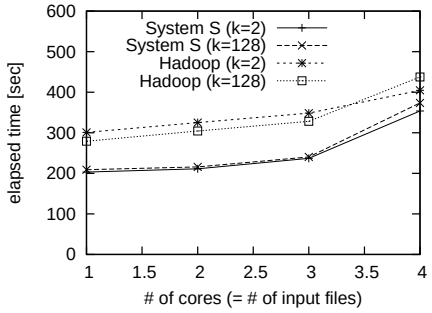

図 2　実行時間

表 1　結果精度

k	System S	Hadoop	コスト比率
2	88578	76900	1.15
4	47535	47277	1.01
8	37347	28904	1.29
16	26094	22828	1.14
32	16915	16491	1.03
64	11547	10451	1.10
128	8156	5801	1.41

4. まとめと今後の課題

典型的なクラスタリングアルゴリズムの一つである k-means に対し，ストリーミング型処理とバッチ型処理との性能比較を行った結果，ストリーミング型処理において結果精度を大きく損なうことなく高速に実行されることを確認した．

今後の課題としては，ジョブ実行時間の詳細なブレークダウンや，複数ノードでのスケーラビリティを調べる実験を行うこと等が挙げられる．

参　考　文　献

1) Ailon, N., Jaiswal, R., Monteleoni, C.: Streaming k-means approximation, NIPS 2009
2) Gedik, Bugra and Andrade, Henrique and Wu, Kun-Lung and Yu, Philip S. and Doo, Myungcheol: SPADE: the system s declarative stream processing engine. SIGMOD '08