

大規模空間情報処理のための分散コンピューティング基盤に関する検討

横山 拓也[†] 石川 佳治^{††}[†] 名古屋大学 情報科学研究科 ^{††} 名古屋大学 情報基盤センター

1 はじめに

近年、クラウドコンピューティングが注目されており、Google における MapReduce や GFS [3], Yahoo! が開発した Hadoop [2, 6] などの大規模分散コンピューティング技術の研究開発が進んでいる。本稿では、このような分散コンピューティング環境における空間情報 (spatial information) の処理方式について述べる。

例として、図1のデータを考える。ソーシャルネットワークサービスにおけるデータ処理を想定する。ファイル `usrs.txt` は、ユーザ情報を保持しており、ユーザ ID、ユーザ名、ユーザの住所の座標値などからなる。一方、`msgs.txt` は、ユーザが書きこんだメッセージの情報を保持する。ただし、ここでは前処理があり、地理情報 (例: 渋谷駅) がメッセージ中に含まれる場合、対応する座標値が抽出されているとする。図1では、たとえば1, 2番目の行は、メッセージ1が3月10日に記述され、その中には2つの点 (5, 69) と (25, 2) に対応する地理情報が含まれていたことを意味する。これらのファイルは分散ファイルシステムに格納されているとする。

```
> cat usrs.txt          | > cat msgs.txt
1 John (76, 77) ...      | 1 03/10/2010 (5, 69) ...
2 Mary (66, 69) ...      | 1 03/10/2010 (25, 2) ...
3 Mike (58, 32) ...      | 2 03/11/2010 (86, 69) ...
...                      | ...
```

図 1: 空間情報を含むデータ

ここで、SNS の各ユーザに対し、最近記述されたメッセージの中から、そのユーザの住所に近い (例: 2km 以内) 地理情報に触れているものを提示することを考える。SNS のデータは大規模であるため、バッチ処理で効率的に対応関係を抽出するものとする。この処理は、データベースの用語における空間結合 (spatial join) の機能があれば容易に実行可能である。しかし、分散コンピューティング環境では、一般に等結合 (equi-join) しか支援されていない。

この問題に対処するため、これまでの研究 [7] では、分散コンピューティング環境として、Hadoop [2, 6] の使用を想定し、Hadoop 上に位置する、高レベルのデータ処理言語を含む大規模データ処理プラットフォームである Pig [4] との親和性の高い実現方式の提案を行った。その研究は、空間結合問合せが行えるよう拡張した Pig 用のプログラムを、ユーザー定義関数や外部プログラムを利用して空間結合問合せを実現する通常のプログラムに変換し、Pig 上で動かすというものだった。これは Pig の上位レイヤーでの拡張であったが、本研究では Pig の下位レイヤーである内部処理に手を加えることで、空間結合問合せが行えるよう拡張することを考える。

以下では、先の例で示したように、一方のファイル (`usrs.txt`) の各座標値に対し、ある一定の距離にあるエントリをもう一方のファイル (`msgs.txt`) から抽出し結合する空間結合について考える。しかし、空間結合の処理はこの形式に限定されるわけではない。以下で述べる手法は、矩形領域を設定して結合する場合などにも容易に一般化が可能である。

2 Pig の拡張

Hadoop 環境では MapReduce 機能をじかに用いて Java などの言語で処理を記述することが基本である。しかし、これにはユーザの負荷が高いという問題がある。そこで本研究では、Hadoop 上に位置するプラットフォームである Pig の利用を考える。Pig では Pig Latin と呼ばれる、データフロー処理を記述する高レベルのデータ処理言語が提供されている。本研究では Pig Latin を拡張して、空間結合条件を記述することを考える。

図2に空間結合の記述例を示す。最初に各ファイルを読み出し、FILTER 演算により3月10日のメッセージのみを抜き出している。そして JOIN 演算により、2つのファイルの距離による結合処理を行っている。最後の STORE 演算は結果の保存を指示している。

```
usrs = LOAD 'users.txt'
      AS (uid:int, name:chararray, uloc:point);
msgs = LOAD 'msgs.txt'
      AS (mid:int, date:chararray, mloc:point);
new_msgs = FILTER msgs BY date == '09/09/2010';
R = SPATIALJOIN usrs BY uloc, new_msgs BY mloc
    WHERE within(2.0);
STORE R INTO 'output';
```

図 2: 拡張した Pig Latin による問合せ記述

この問合せの記述では、以下の2つの拡張を想定している。

1. LOAD 演算でファイルをロードする際に型の指定において、空間上の点データであることを示すための `point` データ型が指定できる。
2. 空間結合の条件が指定できる SPATIALJOIN 演算

を定義している: 条件は WHERE within(2.0) と指定している。

以下では, このような問合せをどのように処理するかについて述べる。

3 空間充填曲線について

Pig における結合処理はいわゆる等結合に限定されている。よって, 以下では空間結合処理の一部を等結合で処理可能な形に変換し, Hadoop 上で実行するアプローチを示す。そのために空間充填曲線のアイデアを用いる。

空間充填曲線 (space-filling curve) [5] とは, 空間上を埋め尽くす曲線であり, ヒルベルト曲線や Z オーダなどが知られている。2次元空間をセルに分割したとき, 空間充填曲線の順序に従ってセルに番号を付与すると, 2次元空間のセル集合を1次元の番号で順序付けできる。空間上で近くに位置するセルには一般に近い値が付与されるという特徴がある。本稿では, Z オーダ (Z-ordering) を例として考える。

Z オーダでは, $2^n \times 2^n$ に区切られた2次元空間のセルを, アルファベットの Z のような形で順番にたどる。たどった順に 0 から $2^n \times 2^n - 1$ までの番号を付けることで, 2次元空間を1次元のデータで表すことができる。 n を Z オーダの次数と呼ぶ。図3に2次の Z オーダの例を示す。Z オーダの番号付けは簡単な規則に則っており, セルの位置と Z オーダ値の対応が容易に計算でき, 異なる次数の Z オーダ値の対応付けが容易という利点がある。

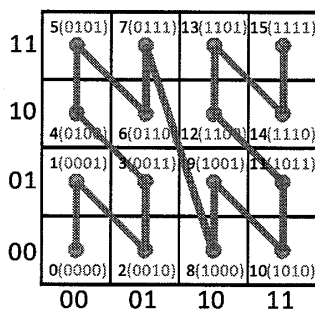


図 3: 2 次の Z オーダ

4 Pig の問合せ処理の拡張

Pig では Pig Latin のプログラムを論理プラン, 物理プラン, MapReduce プランへと順に変換することで, データフローを一連の MapReduce のジョブへと変換している [1]。図2の Pig Latin プログラムを論理プランに変換した例を図4に示す。

図4の左の論理プランは, データの結合を通常の JOIN 演算で行った場合の例である。これに対し, 本研究では, 図2の問合せを図4の右で示したような論理プランに変換する。ここでは論理プランから先の物

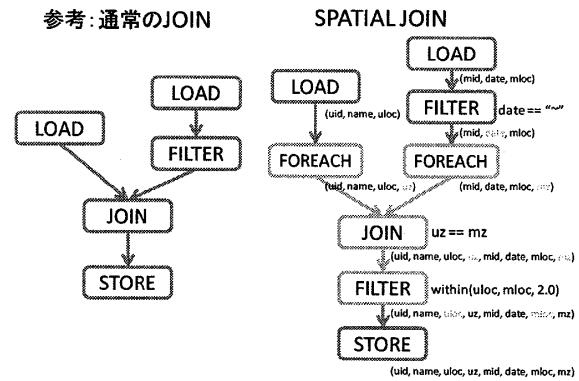


図 4: Pig Latin プログラムの論理プランの例

理プラン, MapReduce プランへの変換は, Pig の処理に任せることとする。

結合処理をする際に単純にすべてのデータを比較するような設計にすると, データ量が増えるにしたがって爆発的に計算量が増えてしまう。それを防ぐために, JOIN 演算の前に FOREACH 演算を使いそれぞれのタプルについて Z オーダ値を用いて JOIN 演算を行う。JOIN 演算の結果には空間結合の距離の条件を満たしているタプルが含まれるが, Z オーダ値による結合のため, 条件を満たさないタプルも同時に含まれてしまっている。そこで, 適切な値のみを取り出すために距離を条件に FILTER 演算を適用することで, 条件を満たすタプルのみで構成される空間結合を実現している。

5 まとめと今後の課題

本稿では, 特に Pig を用いた Hadoop 上での空間結合処理を表現し処理するためのアプローチを提案した。今後はその実装を進め, また, 最近傍問合せへの拡張なども図りたい。

謝辞

本研究の一部は科学研究費 (21013023, 22300034) の助成による。

参考文献

- [1] A.F.Gates et al. Building a High-Level Dataflow System on top of Map-Reduce: The Pig Experience. *Proc. of VLDB*, pp. 1414-1425, 2009.
- [2] Hadoop homepage. <http://hadoop.apache.org/>.
- [3] 西田圭介. Googleを支える技術: 巨大システムの内側の世界. 技術評論社, 2005.
- [4] Pig homepage. <http://hadoop.apache.org/pig/>.
- [5] Hanan Samet. Object-based and image-based object representations. *ACM Computing Surveys*, Vol. 36, No. 2, pp. 159-217, 2004.
- [6] Tom White. *Hadoop: The Definitive Guide*. O'Reilly, 2009.
- [7] 横山拓也, 石川佳治. 分散コンピューティング環境における空間結合問合せについて. 情報処理学会創立 50 周年記念 (第 72 回) 全国大会, 6ZP-6, 2010.