1N-3

Hadoopを用いた大規模空間シミュレーションのためのフレームワークの提案

杉山 武至[†]　横山 拓也[‡]　鈴木 優[§]　石川 佳治[§‡¶]
[†]名古屋大学工学部電気電子・情報工学科
[‡]名古屋大学大学院情報科学研究科
[§]名古屋大学情報基盤センター
[¶]国立情報学研究所

1　はじめに

今日，e-サイエンスの分野では，観測やシミュレーションの結果得られる大規模データを活用する**データ集約的**（data-intensive）なアプローチが注目されている [2]．このため，**Hadoop** [7] に代表される大規模データの並列分散処理のフレームワークの活用が注目を浴びている．ただし，Hadoop の直接的な利用においては，単純な処理であってもプログラミングの労力が必要である．そのため，Pig [1] などのように高レベルのタスク記述言語を提供しようとする取組みがある．

本研究では，**データ集約的なシミュレーション**の支援に焦点を当てる．Hadoop 環境において稼働する，シミュレーションのタスクを記述するための宣言的な高レベルの言語を提案し，その実現をはかる．本フレームワークの対象としては，特に**空間データ**（spatial data）を対象としたシミュレーションを想定する．大規模なデータを対象としたシミュレーションのための宣言的言語を提案するアプローチとしては，オンラインゲームを対象とした [5] が例として挙げられる．記述の容易さとスケーラブルな処理が研究の焦点となる．

2　フレームワークについて

2.1　フレームワークの概略

本フレームワークは，Hadoop 環境上におけるミドルウェアとして稼働することを想定している．フレームワークの概念図を図 1 に示す．

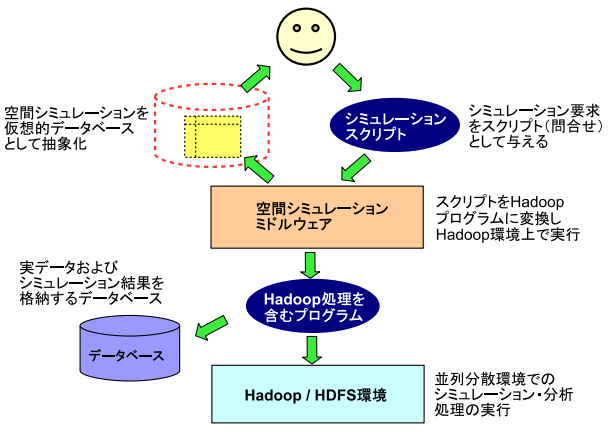

図 1　フレームワークの概念図

本フレームワークの特徴は，シミュレーション内容を一種の**データベース**として仮想化してユーザに提示することにある．実行処理したい一連のシミュレーションに対し，まず，ユーザは本フレームワークが提供する機能を用いて仮想的な**テーブル**を構築する．その定義方式については後述するが，ユーザはシミュレーション内容を一種のデータベースであるかのように捉えることができる．ユーザによる把握・操作を容易にするために仮想的なデータベースビューを用いて抽象化するアプローチは，科学データベース（例：FunctionDB [6]）や確率的データベース（例：MCDB [3]）でも用いられている．

本フレームワークでは，特に地図などのデータに代表される，空間データを対象としたシミュレーションを想定する．空間データ固有のセマンティクスを考慮し，また，空間データのシミュレーションに適した機能を盛り込むことでユーザの利便性の向上を図る．

2.2　問合せ処理

ユーザは，シミュレーション内容を表す仮想的なテーブルに対して，スクリプトにより記述される一種の**問合せ**を与えることができる．これは，シミュレーションの一部を条件指定して抽出・加工する処理である．条件指定にはシミュレーションのパラメータなども含まれる．実際には，問合せが与えられた時点で，指定された条件に基づくシミュレーションが実行され，結果が求められる．

問合せ実行の過程は次のようになる．まず，シミュレーション内容を条件指定したスクリプトをシミュレーション実行システムに与えると，システムはそれを変換し**実行プラン**を構築する．実行プランは Hadoop 処理を含むスクリプトである．シミュレーション実行システムは，実行プランを実行し，その実行状況をモニタリングしユーザに提示する．なお，データベースにはシミュレーションの際に用いるデータも蓄積可能とする．

3　空間シミュレーションの例

3.1　シミュレーションのシナリオ

ここでは，空間シミュレーションの例として，樹木が空間上にどのように分布しているかをシミュレーションをベースに分析するシナリオを考える．このような分析は，固着性・定住性の生物を統計的に分析する計算生態学などにおいてしばしば行われる．

ここでは `Trees(id, point)` というリレーションに，着目している種類の樹木の空間的な分布情報が記録されているとする．`id` は樹木の識別番号を表し，`point` はその位置を表す点データである．分析を行うユーザは，このデータセットのある一部の領域を指定して，そ

Proposal of a Large-Scale Spatial Simulation Framework Based on Hadoop
Takeshi Sugiyama[†], Takuya Yokoyama[‡], Yu Suzuki[§], Yoshiharu Ishikawa[§‡¶]
[†] School of Engineering, Nagoya University
[‡] Graduate School of Information Science, Nagoya University
[§] Information Technology Center, Nagoya University
[¶] National Institute of Informatics

Copyright ©2012 Information Processing Society of Japan.
All Rights Reserved.

の領域における樹木がランダムに分布するかを検定したいとする．統計の用語を用いると,「樹木の分布パターンはランダムに分布している」ということを帰無仮説として検定することになる．

このときによく用いられるのが ***K* 関数** [4] およびそれを拡張した ***L* 関数**である．K 関数は

$$K(r) = \frac{A\sum_i \sum_j k(u_{ij})/w_{ij}}{n^2} \quad (1)$$

と定義される．r は任意の距離，A は指定領域の面積，n は指定領域内の個体数，u_{ij} は個体 i と個体 j の距離，w_{ij} は edge effect の補正係数である．$k(u_{ij})$ は $u_{ij} \leq r$ のとき 1，$u_{ij} > r$ のとき 0 となる関数である．個体がランダムに分布しているときには，K 関数の期待値は $\mathbb{E}[K(r)] = \pi r^2$ となる．L 関数は

$$L(r) = \sqrt{K(r)/\pi} - r \quad (2)$$

と定義され，半径 r のスケールで見たとき，点がランダムに集中していれば 0，集中していれば正，分散していれば負の値をとる．

3.2 仮想テーブルの記述

まず，仮想的なテーブル RandTrees の定義を示す．この書式は MCDB [3] におけるアプローチを拡張したものとなっている．

```
CREATE TABLE RandTrees(id, point)
       WITH PARAMETER area Rect AS
  FOREACH x IN (SELECT * FROM Trees t
                WHERE Covers($area, t.point))
            WITH POINT AS RandPoint($area)
    SELECT seqid(), p.point
    FROM POINT p
```

このテーブルは，問合せ処理時に実体化されるテーブルであり，その際 Rect 型のパラメータ area を受け取る．area は分析対象の空間領域を表す．内側の SQL 文は，Trees テーブルから area 内に位置する個体 x を一つずつ抽出する．各 x に対し以下を行う．1) RandPoint 関数により area 領域内の点を一つ生成し，POINT 変数に代入する．2) 末尾の SQL 文により，RandTrees テーブルにシーケンス ID（seqid 関数で順に生成）と生成された点を挿入する．ここでは x の値は使っていないが，結果的に area 内の個体数と同じ数のランダムなサンプルが生成されることがわかる．

3.3 *L* 関数の評価のためのスクリプト

本研究では，仮想テーブル上の処理においては，Hadoop 環境における実行を想定して Pig [1] の構文をベースとしたスクリプト言語を想定する．以下がその記述例である．L_fun がメイン関数であるが，計算処理の中心は K_fun で行われる．

K_fun では，まず指定されたパラメータ area により仮想テーブル RandTrees を実体化し，それを二つのデータストリーム t1, t2 に代入する．次に SP_JOIN は一種の空間結合であり，各点について距離 r 以下にある点とのペアを作成する．これは Pig には含まれていない空間データ処理機能である．その後の 2 行では空間結合結果のレコード数 $c = \sum_i \sum_j k(u_{ij})$ をカウントする．なお，ここでの処理では簡単のため $w_{ij} = 1$ としている．

```
function K_fun(area, r) {
  t1 = t2 = LOAD(RandTrees($area));
  j = SP_JOIN t1, t2
      WHERE dist(t1.point, t2.point) <= $r;
  g = GROUP j ALL;
  c = FOREACH g GENERATE COUNT($1);
  ... // 途中略
  return K;
};
function L_fun(area, r) {
  return sqrt(K_fun(area, r)/PI) - r;
};
```

検定においては，このような L_fun を異なる乱数で何回も実行し，それをもとに統計的検定のための信頼区間（たとえば 99%という設定のもとで）を作る．実際の観測データからも同様に L 関数の値を計算し，その値が信頼区間に入っていなければ，距離 r の設定のもとで分布がランダムであるという帰無仮説が棄却されることになる．

4 まとめと今後の課題

ここでは，Hadoop 環境における高レベル言語による空間シミュレーションを行うためのフレームワークの提案を行った．さらに高レベルの構文の導入によるタスク記述の容易化を図りたい．また，このフレームワークの実装のための実現技術，特にスクリプトの Hadoop 上の実行処理への展開手法についての開発を行いたい．さらに将来的には，繰り返し処理を含む空間シミュレーションのために，本フレームワークを拡張したいと考えている．

謝辞

本研究の一部は科学研究費（22300034）および「地球環境情報統融合プログラム」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 Endowment (PVLDB)*, 2(2):1414–1425, 2009.

[2] T. Hey, et al. eds. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.

[3] R. Jampani, et al. MCDB: A Monte Carlo approach to managing uncertain data. In *Proc. ACM SIGMOD*, pp. 687–700, 2008.

[4] D. Repley. The second-order analysis of stationary point process. *Journal of Applied Probability*, 13:255–266, 1976.

[5] B. Sowell, et al. From declarative languages to declarative processing in computer games. In *Proc. CIDR*, 2009.

[6] A. Thiagarajan and S. Madden. Querying continuous functions in a database system. In *Proc. ACM SIGMOD*, pp. 791–804, 2008.

[7] T. White. Hadoop 第 2 版. オライリー・ジャパン, 2011.

Copyright ©2012 Information Processing Society of Japan.
All Rights Reserved.