

多次元範囲検索を可能とする キーバリューストア「MD-HBase」

西村 祥治

要 旨

Webサイトのログデータ、センサ情報などの大量データが収集、解析、利用されるようになってきました。これらの大量データの格納先として、従来よく用いられてきたリレーショナルデータベースに代わり、データサイズに対してスケールアウトさせやすいキーバリューストア（KVS）が注目されています。しかし、KVSは単純な検索機能しか提供していません。本稿では、KVSの1つであるHBaseを拡張し、そのスケーラビリティを損ねることなく、多次元データに対する効率的な範囲検索を実現した「MD-HBase」を紹介します。

キーワード

●キーバリューストア ●多次元範囲検索 ●HBase ●スケールアウト

1. はじめに

近年、大量のデータが収集可能となるだけでなく、収集したデータの活用が考えられるようになりました。例えば、位置情報サービスでは、スマートフォンからユーザーの現在位置を収集し、それらの情報を元に、そのユーザー周辺にある店舗の推薦や、クーポン券配信による店舗のプロモーションなどきめ細かなサービスをすることが考えられます。このとき、位置情報を蓄積するデータベースは、数千万を超える端末からの情報収集を可能とするだけでなく、例えば、ある店舗の周辺にいるユーザー一覧を取り出すといった検索機能が不可欠です。つまり、大量データを扱うデータベースは、データサイズに対するスケーラビリティとともに、実用的な検索処理を両立する必要があります。

従来、データベースといえば、汎用的な検索機能を提供するリレーショナルデータベースが広く用いられてきました。従来型のデータベースでスケーラビリティを確保したい場合、システムを構成するハードウェア部品をより強力なもので置き換えるスケールアップ戦略でその要求に応じてきました。この戦略は、構成を大きく変えることなくシステムを強化することができますが、ハードウェアが持つ性能限界以上に強化することはできず、コストもかさんでしまいます。そこで、サーバを追加することでシステムを増強するスケールアウト戦略が注目を集めています。スケールアウト戦略では、データサイズの増加に伴い、システムの構成を変えていきます。このため、汎用的で効率的な検索性能を達成するために、内

部構造が複雑になったデータベースにはスケールアウト戦略を適用することは一般的に困難です。そこで、構造が単純で容易にスケールアウトさせやすいキーバリューストア（Key Value Store：KVS）が注目されています。一方で、KVSはその構造の単純さゆえ、単純な検索操作しか提供していません。そのため、スケーラビリティを確保しつつ、それなりに複雑な検索を効率的に実行できることが、大量データを扱う時代におけるデータベースのチャレンジと言えます。本稿では、効率的な多次元範囲検索を実現したKVSである「MD-HBase」を紹介します。

2. キー順序型キーバリューストア（KVS）

KVSとは、キーとバリュー（値）の組で表現されるレコードを効率的に管理するデータベースです。KVSには、キーを昇順に並べた順序になるようにレコードを格納するキー順序型と呼ばれるKVSがあります（図1）。キー順序型KVSは辞書に例えられます。見出し語（キー）が昇順に並び、それぞれの見出し語に対してその内容（バリュー）が関連付けられています。

キー順序型KVSは、レコード数が多くなる場合、適当なキーの範囲でテーブルを分割することで分散化し、スケールアウトを実現できます。これは、ちょうど辞書を見出し語の範囲で区切って、冊子を分けるのに対応します。分散化により複数のテーブルができるので、どのテーブルがどの位置にくるのかを管理するためにインデックスが必要になります。

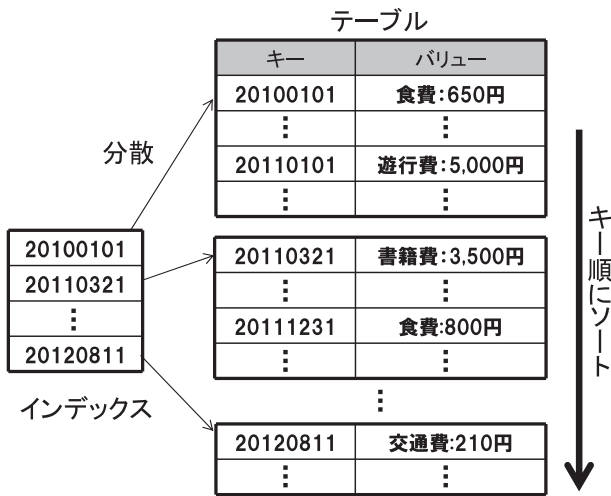


図1 キー順序型KVS

キー順序型KVSは、2つの検索機能を提供します。1つ目は、一致検索と呼ばれるもので、あるキーに関連付けられたバリューを取り出す操作です。これは、辞書において、見出し語が一致する項目を探すことと同じです。もう1つは範囲検索と呼ばれるもので、あるキーからキーまでの範囲内に含まれるすべてのレコードを取り出す操作です。これは、例えば図1において、見出し語が「20110101」から「20111231」までのレコードを抜き出すことに相当します。キー順序型KVSは、キーの昇順にレコードを並べているため、この操作を効率的に実行することができます。また、この範囲検索はキーというレコードの1つの属性の範囲に関する検索なので、一次元の範囲検索とみなすことができます。

3. 多次元範囲検索

多次元範囲検索を、位置情報サービスを例に説明します。ある店舗周辺にいるユーザーに対してクーポンを配布したいとします。配布先のユーザーの一覧は、その店舗を中心とした緯度、経度の範囲にいるユーザーを検索することで得られます。つまり、緯度、経度という2つの属性に対する範囲検索で求めることができます。このような2つ以上の属性に対する範囲を指定する検索のことを、多次元範囲検索といいます。

従来方法によるKVS上での多次元範囲検索の実現

それでは、この多次元範囲検索を、先ほどのキー順序型KVS上で実現することを試みます。キー順序型KVSは、キーが昇順に並んでいるため、キーとして選んだ属性に対する範囲検索を効率的に実行できます。しかし、この例では、キーとして選びたいものが複数（緯度、経度）あります。このため、これらの属性を何らかの方法で1つの属性としてまとめる必要があります。これは、多次元空間上の点（緯度、経度）から、一次元空間上の点（キー）へと変換する問題と言い換えることができます。それを実現する方法の1つとして、空間充填曲線による変換手法が知られています。

空間充填曲線とは、多次元空間上のすべての点をくまなく、一筆書きでなぞる（一次元の）曲線です。Zカーブという空間充填曲線を2次元空間上に適用した例を図2に挙げます。矢印で書かれた線がZカーブで、Z字を描くように2次元空間をくまなく、重なりなく、なぞりつくしています。それぞれのマス目の数字が、そのマスがZカーブによりなぞられた順番になります。つまり、多次元空間上のすべての点は、空間充填曲線によりなぞられる順番で順序付けられることとなります。

多次元値をKVSに格納する際、それぞれの属性値の組が空間充填曲線上で何番目にあたるかを計算し、それをキーとします。例えば、Xの値が1、Yの値が2である場合、キーは6となります。

それでは、上記の方法でキーを生成し、キー順序型KVSに格納されたレコードに対して、多次元範囲検索を実行します。

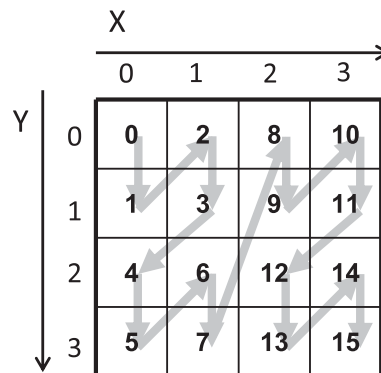


図2 空間充填曲線

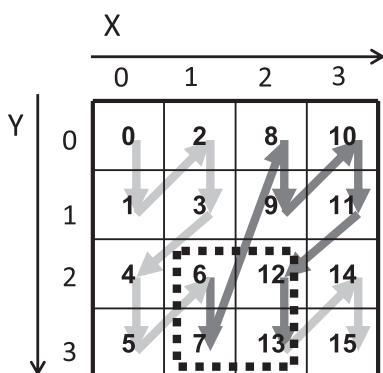


図3 X=[1, 2], Y=[2, 3]の範囲検索 (2次元上のZカーブ)

ここでは、図3の点線で囲まれた領域、すなわち、Xの範囲が[1, 2]、Yの範囲が[2, 3]に対する範囲検索を例として説明します。まず、検索領域内において、キーの最小値と最大値を求めます。この例では、キーの最小値が6、最大値は13になります。次に、キーの最小値6から最大値13までの範囲検索をKVS上で実行します。このとき、図3において濃い矢印の範囲が走査されるため、検索範囲外にある8から11をキーとするレコードも結果として含まれてしまいます。これを避けるため、各次元の属性値が範囲内に入っているかをチェックし、条件に合ったレコードだけを抽出し、それを検索結果とします。

以上の方法で、キー順序型KVS上で多次元範囲検索を容易に実現できますが、キーが8から11の区間のように検索範囲外にあるレコードも走査しています。これは、検索性能を劣化させる原因となります。

4. MD-HBase

MD-HBase¹⁾ は、キー順序型KVSの1つであるHBase²⁾ を改良し、多次元範囲検索法を効率化したKVSです。MD-HBaseは、多次元空間上のデータを効率的にアクセスする手法である多次元インデックスのアイデアを取り込むことで、先ほどのような無駄なレコードの走査を削減し、多次元範囲検索を効率化します。

4.1 多次元インデックスと空間充填曲線

多次元インデックスとは、データを組織的に構造化するこ

とで、データへのアクセスを効率化する手法です。MD-HBaseは、多次元インデックスの1つである空間分割ベースのKd-Treeの空間分割法をキー順序型KVS上で実現することで、多次元範囲検索を効率化します。

空間分割ベースのKd-Treeによる空間分割方法を、図4を用いて説明します。Kd-Treeは空間内にあるデータ点の数が一定数以上になると空間を半分に分けます。ここでは、空間分割の閾値をデータ点2つとします。図4 (a) では、二次元空間にデータ点が3つ挿入され、空間分割の閾値を超えています。そこで、この二次元空間をちょうどX次元の真ん中で半分に分けます。その結果、図4 (b) のようになります。更に、データ点を追加し、図4 (c) のようになります。右半分空間にあるデータ点の数は空間分割の閾値を超えているので、今度は、Y次元の真ん中で右半分空間を上下に分割します。その結果、図4 (d) のように分割し、それぞれの空間に含まれるデータ点の数が2以下になるように維持します。つまり、データ点が密集する領域ではより細かく分割され、それぞれの空間に含まれるデータ点の数を閾値以下に抑えます。

このKd-Treeによる空間分割の結果と、先ほどの空間充填曲線を重ねます。すると、図5で示すようにそれぞれの分割された空間内で、空間充填曲線が途切れることなく、連続性を維持したまま、なぞられていることがわかります。つまり、空間分割ベースのKd-Treeは、空間充填曲線によって順序付けられた順番を崩すことなく、データ点を分割できるという性質があることを意味します。

MD-HBaseは、この性質を利用して、テーブルを分割し、テーブルへのインデックスを作成します。MD-HBaseは、多次元空間上の点を空間充填曲線であるZカーブで一次元上にマップした点をキーとして、レコードをキー順序型KVS上に保存します。そして、テーブルに含まれるレコード数がテーブル分割の閾値を超えると、前述の空間分割法に従ってテーブルを分割します。そして、図5に示すように、Zカーブのなぞる

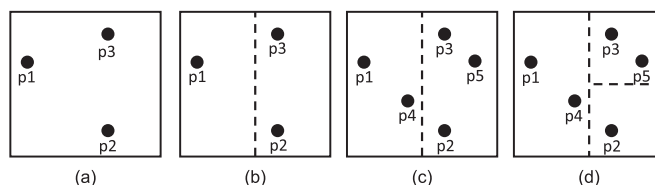


図4 Kd-Treeによる空間分割

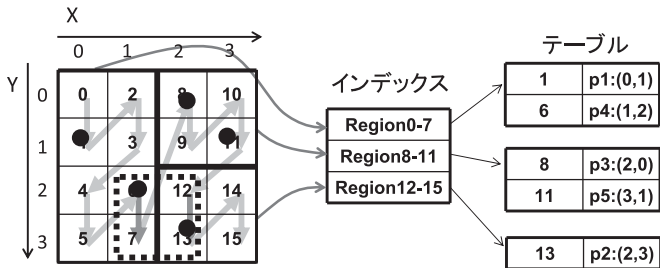


図5 MD-HBaseにおける空間分割とインデックス

順で分割空間を並べてテーブルへの参照を保持し、それをインデックスとします。

4.2 MD-HBaseにおける多次元範囲検索

MD-HBaseにおける多次元範囲検索を、先ほどの図3と同じ範囲を検索する例で説明します。ここでは、図5で示すように空間分割され、インデックスが構成されていたとします。先ほどの例と同じく、検索領域内のキーの最小値と最大値を求めます。先ほどは、キーの最小値と最大値の間にあるレコードをそのまま走査しましたが、MD-HBaseの場合は、まず、インデックスを見て、キーの最小値と最大値の範囲内にある分割空間のリストを得ます。この例では、3つの分割空間がヒットします。それぞれの分割空間の各次元の境界が分かっているので、検索領域と重なっているかを検査します。もし、分割空間と検索領域とが重ならないということは、その分割空間内にあるすべてのレコードは、検索領域内に存在しないと云えます。このため、その部分空間に関連付けられたテーブルに対するレコードの走査をスキップすることができます。この例では、Region0-7とRegion12-15は検索領域と重なっていますが、Region8-11は重なっていません。つまり、Region8-11に対するテーブルの走査を、安全にスキップすることができます。

この例では1つのテーブルの走査をスキップできたにすぎませんが、特にデータが密に集まっている領域周辺を検索したり、次元数が大きかったりすると、このスキップの効果が性能向上に寄与します。実際、車の移動を模擬したデータセットに対する多次元範囲検索の性能評価実験では、MD-HBaseは、単に空間充填曲線だけを適用したものに比べて約10倍の高速化を達成しました(図6)。なお、図6は道路地図の上を動く

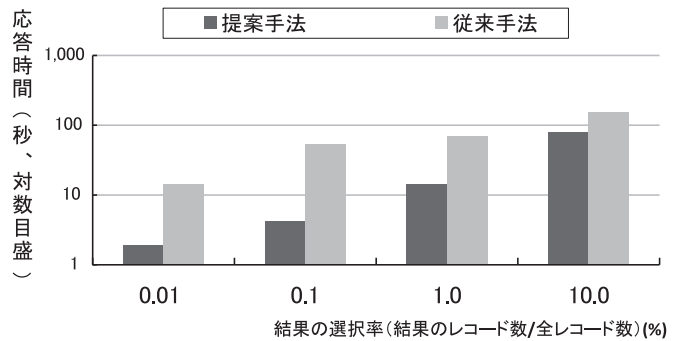


図6 多次元範囲検索評価結果

車を模擬した、4億点の生成データに対する範囲検索の評価結果です。横軸は検索範囲の大きさに相当します。

5. まとめ

本稿では、大量データを扱う時代の要件にあった、データサイズに対してスケーラブルで、効率的な多次元範囲検索を実現するKVSであるMD-HBaseを紹介しました。本研究成果は、バッチ処理を効率化するアプリケーション実行基盤である「WebOTX Batch Server」の一部として製品化予定です。なお、MD-HBaseはカリフォルニア大学サンタバーバラ校のDivyakant Agrawal教授、Amr El Abbadi教授、Sudipto Das博士との共同研究成果です。

参考文献

- 1) Shoji Nishimura, et. al, "MD-HBase: A Scalable Multi-dimensional Data Infrastructure for Location Aware Services," International Conference on Mobile Data Management 2011
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6068416>
- 2) Apache HBase Project
<http://hbase.apache.org/>

執筆者プロフィール

西村 祥治
中央研究所
クラウドシステム研究所
主任
情報処理学会会員
ACM会員

NEC 技報のご案内

NEC 技報の論文をご覧くださいありがとうございます。
ご興味がありましたら、関連する他の論文もご一読ください。

NEC技報WEBサイトはこちら

NEC技報(日本語)

NEC Technical Journal(英語)

Vol.65 No.2 ビッグデータ活用を支える 基盤技術・ソリューション特集

ビッグデータ活用を支える基盤技術・ソリューション特集よせて
ビッグデータを価値に変えるNECのITインフラ

◇ 特集論文

データ管理/処理基盤

超高速データ分析プラットフォーム [InfoFrame DWH Appliance]
SDN 技術で通信フローを制御する [UNIVERGE PF シリーズ]
大量データをリアルタイムに処理する [InfoFrame Table Access Method]
大量データを高速に処理する [InfoFrame DataBooster]
ビッグデータの活用最適なスケールアウト型新データベース [InfoFrame Relational Store]
高い信頼性と拡張性を実現した Express5800/ スケーラブル HA サーバ
大規模データ処理に対する OSS Hadoop の活用
大容量・高信頼グリッドストレージ iStorage HS シリーズ (HYDRAStor)

データ分析基盤

ファイルサーバのデータ整理・活用を支援する [Information Assessment System]
超大規模バイオメトリック認証システムとその実現
WebSAMの分析技術と応用例～インバリエント分析の特長と適用領域～

データ収集基盤

スマートな社会を実現する M2M とビッグデータ
微小な振動を検知する超高感度振動センサ技術開発とその応用

ビッグデータ処理を支える先進技術

多次元範囲検索を可能とするキーバリューストア [MD-HBase]
高倍率・高精細を実現する事例ベースの学習型超解像方式
ビッグデータ活用のためのテキスト分析技術
ビッグデータ時代の最先端データマイニング
ジオタグ付きデータをクラウドでスケラブルに処理するジオフェンシングシステム
柔軟性と高性能を備えたビッグデータ・ストリーム分析プラットフォーム [Blockmon] とその使用事例

◇ 普通論文

地デジ TV を活用した「まちづくりコミュニティ形成支援システム」

◇ NEC Information

NEWS

スケールアウト型新データベース [InfoFrame Relational Store] が 2 つの賞を受賞



Vol.65 No.2
(2012年9月)

特集TOP