

時区間データ構造の性能評価

川村 雄介† 三好 涼介† 三浦 孝夫†

† 法政大学工学部情報電気電子工学科

1 前書き

時区間データに効果的なデータモデルとその実現構造を提案する。データの有効区間の表現方法として、時点データ(timestamp data)方式と時区間(time interval)データ方式がある[1]。後者の手法の表現力は大きい、実現が複雑で工夫を要する。時区間データの表現方法として、Append Only(AO) tree[1]などの1次元データ構造がある。しかし、時間をキーとするだけでは特定できないことが多く、検索操作を介する必要がある。2次元座標表示範囲検索に対応するデータ構造として、SR-tree[2]などの空間索引構造がある。しかし、これらでは対象領域の重なりが生じ、時として検索・更新効率を低下させる。

本論文では、時区間データを2次元座標上に表し、データ操作を効率的に範囲検索に対応させる手法を示す。範囲検索を効果的に扱うため、筆者らは拡張可能グリッドファイル(Extended Grid File, EGF)[3]を提案しているが、本稿では時区間データについても効果的であることを述べる。2章では、時制データのモデル化と実現のためのアイデアを、3章では、時区間データ検索の実現手法について述べる。4章で実験により良好な性能を得ることを示し、5章で結びとする。

2 時区間データと時点データ

2.1 データ表現と操作

はじめに、時制データベース D が時点データ表現されている場合(D_1)と時区間データ表現されている場合(D_2)で、格納方式と操作を比較する。データ d が時間 t で有効である条件は、 D_1 では $|t - t'| < \epsilon$ となる時間 t' に対して $d_{t'}$ が D_1 に格納されていなければならない、有効時間が長いほどデータ量が増加する。一方 D_2 では $d_{[s,e]}$ が $s \leq t \leq e$ を充足すればよいので、一定のデータ量で済み、かつ近似量を必要としない。

時制データベースのためのデータ操作を論じる。時間 t でデータを検索するには、 D_1 では許容誤差 ϵ に対して $\{d_{t'} \in D_1 | |t - t'| < \epsilon\}$ で表すのに対し、 D_2 では $\{d_{[s,e]} \in D_2 | s < t < e\}$ と区間検索で表される。

キー K を有するデータ d の開始時間(終了時間)を検索するには、それぞれ $\text{Min}\{s | d \in D_1, d.\text{key} = k\}$ 、 $\{s | d_{[s,e]} \in D_2, d.\text{key} = k\}$ と表される。前者は全件検索に相当する。

時区間 $[t_1, t_2]$ と交わるデータをすべて検索するためにはそれぞれ $\{d_t \in D_1 | t_1 \leq t \leq t_2\}$ 、 $\{d_{[s,e]} \in D_2 | [s,e] \cap [t_1, t_2] \neq \emptyset\}$ と表される。また時区間 $[t_1, t_2]$ に含まれるデータをすべて検索するためにはそれぞれ $\{d \in D_1 | \text{Min}(\{t | d_t \in D_1\}) \geq t_1, \text{Max}(\{t | d_t \in D_1\}) \leq t_2\}$ 、 $\{d_{[s,e]} \in D_2 | t_1 \leq s, e \leq t_2\}$ となる。 $[t_1, t_2]$ を含むデータの場合も同様である。いずれも D_1 では全件検索になっている。

"Evaluating Data Structure suitable for Time Intervals":
Yusuke Kawamura†, Ryosuke Miyoshi†, Takao Miura† :
†Hosei University, Dept. of Elec. and Elec. Eng.
Kajino-cho 3-7-2, Koganei, Tokyo, JAPAN

2.2 時区間データの格納表現

時区間データの表現方法として、素朴には1次元データ構造があげられる。例えば、AO-treeは、B+-treeとISAMを組み合わせており、有効期間の開始時間を検索キーとし、値に終了時間と付帯情報を入れる。区間検索をする場合、開始時間で検索し、検索結果の値から終了時間を取得し、質問区間に有効か調べる。しかしこの手法では、開始時間で検索をしないと終了時間がわからない為、開始時間が質問区間に有効なデータを全て検索しなければならない。

2次元座標表示では、SR-treeや拡張可能グリッドファイル(EGF)が代表的である。SR-treeは、R木を基盤とした空間索引構造であり、R*-tree、SS-treeを上回るCPU時間、ディスクアクセス回数の性能を持っている。EGFは、グリッド検索の参照回数問題に拡張ハッシュを適用し、グリッド空間内のデータ絞込みにMBRを適用した高速な範囲検索を可能にするデータ構造である。これらのデータ構造では、有効期間の開始時間と終了時間を座標軸に取ると、時区間データは、2次元空間の点で表現できる。区間検索の質問範囲も図1の区間 $[t_1, t_2] = [3, 6]$ の例で示すように2次元座標上の決まった範囲で表現できる。このため、範囲検索で区間検索ができる。

- (1) 区間 $[t_1, t_2]$ に含まれる時区間データ
- (2) 区間 $[t_1, t_2]$ を含む時区間データ
- (3) 区間 $[t_1, t_2]$ と開始時間で交わる時区間データ
- (4) 区間 $[t_1, t_2]$ と終了時間で交わる時区間データ

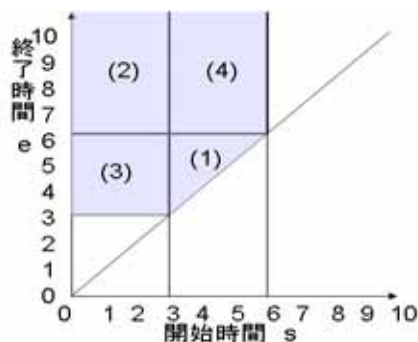


図1: 区間 $[t_1, t_2]$ の検索範囲

3 時区間データ検索の実現

AO-tree操作では、1次元データ構造のため、検索キー(開始時間)だけで時区間データが有効か判断できないため、まず検索キー(開始時間)が検索の条件を満たすデータを検索し、葉の中に入っている終了時間を取り出す。この終了時間が検索の条件を満たしているか調べる。

SR-tree操作では、rootから葉へとたどり、検索範囲と交わる包囲長方形を見つける。次にその包囲長方形のデータが、検索範囲内にあるか調べる。

EGF操作では、検索範囲の開始時間と終了時間を2進数表示にする。その検索範囲の上位 n 桁(本論文では2桁)を取得

する。その n 桁でディレクトリより該当するグリッド領域を見つける。検索範囲と交わるグリッド領域内にある最小包囲長方形を求め、その包囲長方形のデータが、検索範囲内にあるか調べる。

区間 $[t1, t2]=[3, 6]$ で例示する。図2にAO-treeの操作を示す。

- (1) 開始時間 $s \geq 3$ の節の葉を見る
- (2) 1の終了時間 $e \leq 6$ か調べる
- (3) 右隣の節へ移動
- (4) 開始時間 $s < 6$ の間2,3を繰り返す

図5にSR-treeの操作を示す。

- (1) $3 \leq s < e \leq 6$ と交わる包囲長方形を検索
- (2) 該当包囲長方形の中で検索範囲に含まれるかどうかを調べる

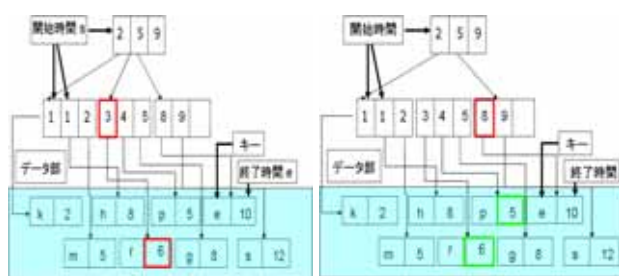


図2: AO-tree の場合

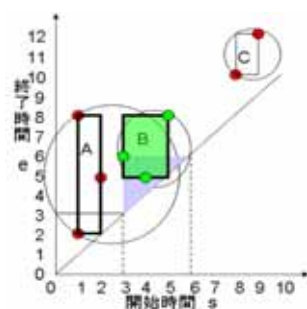


図3: SR-tree の場合

図4にEGF操作を示す。

- (1) 質問範囲を2進数表示する
- (2) 質問範囲の上位2桁を取得(00,01)
- (3) $00... \leq s \leq 01... \text{ 且つ } 00... \leq e \leq 01...$ のディレクトリを検索
- (4) 検索範囲と重なるグリッドファイルの最小包囲長方形を検索
- (5) 該当包囲長方形の中で検索範囲内か調べる

4 実験

4.1 実験環境

本稿で行う実験は、FreeBSD4.6.2-RELEASEを用い、Pentium4, 1.8GHz, 256MByte メモリ環境を使用する。時区データは131855件である。これは、各レコードが24バイト長(2つの5バイトlong integer型項目(開始時間、終了時間各々5バイト)、14バイトcharacter型(キー3バイト、値7バイト含む))の1989年から2003年の間にNSF賞を獲得した研究が資金援助を受けた期間を表す。時間値は1989からの通算日数で表現するよう変換した。キーには、NFS Org(3バイトcharacter型)、値には、Award Number(7バイトcharacter型)を使用する。また、質問区間100件はデータの領域内で乱数より生成し、質問キー100件は、NFS Orgよりランダムに取得する。K=15, B=24Kbyteとして実験を行う。2次元座標表示範囲検索の性能調査の対象として、1次元データ構造ではAO-treeを、2次元範囲検索ではSR-treeを用いる。

4.2 実験結果

実験は、条件を満たすデータ検索をし、質問区間 $= [t1, t2]$ に対して次に示すような各操作のアクセス回数と実行時間を計測し、実験結果を表2,3に示す。

- (1) 援助を受けた期間が含まれる検索
- (2) $t1$ 以前から $t2$ 以降までを期限とする検索
- (3) 援助を受けた期間が交差する検索
- (4) 質問点 $t1$ に援助を受けている検索
- (5) 質問点 $t1$ に援助を受けていてキーが指定したNFS Orgである検索

	実験1	実験2	実験3	実験4	実験5
AO-Tree	4914694	4357376	9265313	4914694	485858
SR-Tree	7027	20443	39072	22711	22711
EGF	1676	3980	8376	4674	4674

表2 ファイルアクセス回数

	実験1	実験2	実験3	実験4	実験5
AO-Tree	51.51	46.13	91.07	51.32	10.51
SR-Tree	13.38	50.56	109.63	50.18	50.64
EGF	11.85	8.63	22.82	15.98	15.7

表3 実行時間

4.3 考察

実験(1)から実験(4)までの実験結果について考察する。EGFのアクセス回数は、AO-treeとの比較でどの条件でも、良好な結果が得られた。また実行時間は3割程度の結果が得られた。AO-treeは検索結果で有効データか判断しなければならないため、大量にファイルアクセスする必要がある。また、範囲検索では、1度のファイルアクセスで範囲内のデータを取得できるが、AO-treeでは、1件のデータしか取得できないため、大量にファイルアクセスをする必要がある。

EGFはSR-treeとの比較でどの条件でも、ファイルアクセス回数が23パーセント、実行時間は89パーセント程度の結果が得られた。SR-treeの場合、MBRの重複によるオーバーヘッドアクセスが生じるためによるものであろう。他方EGFではハッシュ構造とMBRを用いているため、1read操作で当該ディレクトリを発見し、ファイルアクセス回数、実行時間共にSR-treeに勝る事ができたと考えられる。

実験5では、EGFは、アクセス回数でAO-treeの0.96パーセントで完了しているが、実行時間で149パーセントという遅い状況を生む。これはAO-treeの検索キーをNFS Orgにしたため実行時間で劣ってしまったと考えられる。時区データでの検索よりもキーデータでの検索の方が、対象データの絞り込みが可能であろう。

5 結論

本論文では、2次元座標によるEGFによる範囲検索の有効性をAO-treeとの実験により評価し、良好な結果を得ることを示した。EGFの範囲検索の性能を評価するため、SR-treeとの実験により、良好な結果を得ることを性能を示した。

参考文献

- [1] Vassilis J. Tsotras: "A Comparison of Access Methods for Time Evolving Data", Computing Surveys June 1999.
- [2] 片山紀生・佐藤真一: "SR-Tree: 高次元店データに対する最近某検索のためのインデックス構造の提案", 信学論D-I, Vol. J80-D-I, No.8, pp.703.717, Aug.1997.
- [3] 三好 涼介・三浦 孝夫: "拡張可能グリッドファイルによる空間データの検索", 電子情報通信学会論文誌(D1), March 2004.