D-003

放送時刻の範囲データを含むタグ検索に関する一考察

A Consideration of Tag Retrieval Including On-air Time Range Field

金子 豊† 竹内 真也† 黄 民錫† 苗村 昌秀†
Yutaka Kaneko Shinya Takeuchi Minsok Hwang Masahide Naemura

1. まえがき

長期間の放送番組のタイムシフト視聴環境として，気軽にザッピングしながら過去の番組を視聴できる VOD サービスを検討している[1]．ザッピングの一つの方法として，再生中の画面に出演者名などのタグの一覧を表示し，視聴者がタグを選択することで時間方向に再生位置を移動できる視聴システムを開発している．このシステムを実現するには，再生画面に同期して関連するタグを検索できるデータベースが必要となる．

本報告ではタイムシフト視聴の放送番組に関連付けられたタグを高速に検索することを目的として試作したデータベースの構造とその実験結果について述べる．

2 . タグを使った時刻方向のザッピング

開発中の長期間タイムシフト視聴システムでは，放送日時を ID としてフレーム単位でアクセスできる分散ファイルシステムを用いることで，過去の放送番組の任意の放送時刻から視聴することができる[2]．このシステムのザッピング機能として，図１に示すように，再生中の番組に関連するタグ名の一覧を画面に表示し，視聴者はタグ名を選択することで，過去または未来方向（時間的に負または正方向の意味）の同一のタグ名が付与されたタグ位置に移動できる機能の追加を進めている．

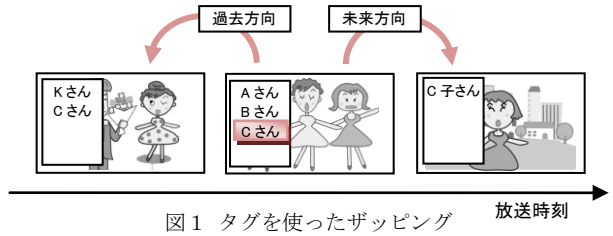

図１ タグを使ったザッピング

3. タグ検索

3.1 タグの検索動作

タグは(name, category, media, st_time, ed_time)からなる．name はタグ名，category はタグの付加データ，media はメディア名(放送チャンネル)，st_time は開始日時，ed_time は終了日時である．タグは st_time から ed_time の範囲の放送時刻に関連付けられる．category は例えば，出演者，監督，字幕など，タグを説明するためのデータ領域として用いているが，ここでは検索対象とはしない．

図２はタグ名 A～E のタグと放送時刻との関連付けの例を示している．現在の再生日時を t とすると，その日時に関連するタグ名は A, B, C である．再生が進み，再生日時が t_1（タグ D の開始日時 st_D）になると，関連するタグ名は，A, B, C, D に変わる．一方，再生時刻 t で視聴者がタグ名 A を選択し，過去方向への移動を指定した場合，再生位置が t_2 に移動し，関連するタグ名は A, E となる．したがって，本システムでは次の２つの検索機能が必要となる．

query①：放送日時 t におけるタグ名一覧の検索
query②：選択されたタグに隣接する過去または未来方向の同一タグ名のタグの検索

query①は図２の時刻 t における縦方向の検索，query②は対象のタグ名における横方向の検索に相当する．

画面に表示するタグ名の一覧は再生時刻 t の変化に応じて更新が必要となる．query①の検索をデータベースに対して定期的に実行することで更新は可能であるがデータベースへ負荷が生じる．query①の検索結果として取得するタグ名の一覧と同時に，そのリストが継続する日時範囲が取得できれば，再生時刻がその日時範囲を超えたときに再検索すれば良くなり，データベースへの負荷を低減できる．タグ名一覧の日時範囲を求めるには，例えば図２の場合，日時 t では ed_E から st_D がその範囲となる．そのため，時刻 t に関連するタグ(A, B, C)の情報からだけでは求めることはできず，時刻 t に最も近いタグの検索が必要となり，検索時間が増加する可能性がある．

3.3 試作データベース

タグ検索を目的とした専用のデータベースを試作した（以下試作 DB と表記）．試作 DB はタグ名ごとに時系列に登録されたタグをソートしてメモリ内に格納する．また，タグ名一覧の検索を高速化するため，あらかじめタグ名一覧を作成する．タグ名一覧はタグの登録時に更新する．図３ではタグ A が登録されている状態で，タグ B，タグ C の順で新たにタグを登録した場合を示している．タグ A とタグ B の時間範囲の重複部分は，[A, B]というタグ名として保管し．さらにタグ C を登録すると，重複箇所は[B, C], [A, B, C], [A, C]というタグ名を時間範囲とともに保管する．

試作 DB ではメモリ上のタグ検索として，主に２分木探索を用い，一部は線形探索を使用した．

3.4 試作 DB へのアクセス方法

文献[2]で使用している分散ファイルシステム（分散 FS）では，映像，音声，字幕などのデータを放送時刻 ID（日付＋90KHz 精度の時刻）を使ってフレーム単位にアクセスできる．試作 DB をこの分散 FS に組み込み，タグ検索をファイルアクセスで行えるようにした．

query①ではファイル名として"/@NHKG/#taglist.tag"を，

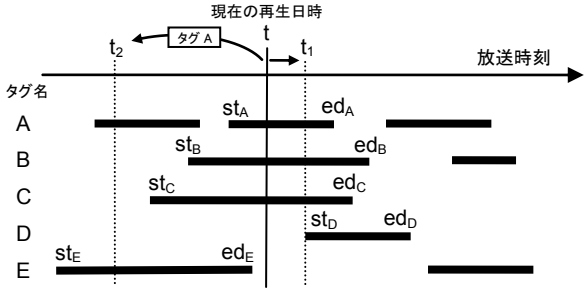

図２ 放送時刻の時間軸に関連付けられたタグの例

†日本放送協会 放送技術研究所, NHK

query②では，例えば検索するタグ名が「A さん」の場合，ファイル名として"/@NHKG/#A さん.tag"とする．これらのファイルを open し，pread のオフセット値として放送時刻 ID を指定することで，その時刻を含むタグ名一覧またはタグ情報を取得できる．ここで，@NHKG はメディア名であり，例えば@NHKG は NHK 総合テレビを示している．

4. 実験結果と考察

タグデータを使い試作 DB の検索時間を測定した．比較として PostgreSQL (以下 PQ と表記)および ElasticSearch (以下 ES と表記)を使用した．PQ のバージョンは 9.2.7，ES のバージョンは 1.1.1 を用いた．実験には 2010 年 10 月～2014 年 4 月の番組情報と字幕データから作成したタグを用いた．タグ名は 68,926 種，タグ数は 11,578,921 である．

データベースはすべて同一の Linux サーバ(Intel Xeon E5-2670(2.6GHz)x2, 128GB メモリ)で起動し，ネットワーク経由でクライアント PC から 100 回の検索を行い平均検索時間を求めた．タグ検索用プログラムは C++を用いて作成し，PQ へのクエリには libpq，ES へのクエリには libcurl，JSON の処理には Jansson を用いた．試作 DB へのアクセスは，WebSocket 経由で分散 FS にアクセスした[2].

タグの日時範囲(st_time, ed_time)は，PQ ではタイムスタンプの範囲型(tsrange)を用い GIST(Generalized Search Tree)インデックスを作成した．ES では個別の Date 型とした．name, media は文字列型とし，ES では完全一致インデックス(not_analyzed)とした．

実験に用いた PQ と ES のクエリを図 4 に示す．query①は複数のクエリを組み合わせて実現した．query①-1 で再生日時を含むタグ名一覧を取得し，query①-2 でその日時範囲を求める．ES ではファセットを使うことで，これらは 1 回のクエリで行う．query①-3 と 4 で，再生時刻に過去方向および未来方向で一番近いタグを検索し，query①-2 で求めたタグ名一覧の日時範囲を修正する．

図 5 に 測定結果（query①+query②の合計時間）を示す．また，表 1 に検索時間の内訳を示す．ES の測定値の括弧内の数値は，ES から報告された検索時間(took 値)である．

PQ, ES ともに登録タグ数に比例して検索時間が増加している．一方，試作 DB ではタグ数による検索時間への影響はなく，ES に比べ 1/10 以上短縮できた．

PQ では query①-1,2 は高速であるが，その他のクエリの検索時間が大幅に遅い．これはインデックスによるところが大きい．現在，query①-1 および 2 で使用した範囲型への演算子@>（範囲を包含する）ではインデックスが使用されているが，query①-3 および 4 で使用している演算子&<（右を超えない），&>（左を超えない）ではインデックスが使われていない．この部分でインデックスを使用するようにできると大幅な速度向上が見込める．

ES の測定結果を見ると，ES から報告される took 値に比べ 1 回のクエリ毎に約 9msec のオーバヘッドが生じている．これは，libcurl 等でのデータ転送や内部処理によるものと考えられる．今回は PQ，ES ともに query①として複数のクエリを組み合わせたが，クエリ回数を減らすことができれば速度の改善が見込める．

5. まとめ

長期間の放送番組に付与された放送時刻の範囲データを含むタグを検索するために試作したデータベースについて述べた．あらかじめ日時によるタグのソート，タグ名一覧の作成，オンメモリで動作させることで検索の高速化が可能であることを実験により示した．既存のデータベースにおいても，これらの処理をフロントエンドとして組み込むことで速度向上が見込める．

参考文献

[1] 竹内，黄，金子，苗村，"時間方向へのザッピングが可能なタイムシフト視聴における操作行動", 映情学会冬大, 13-1, 2013
[2] 金子，黄，竹内，砂崎，"放送時刻でアクセス可能な放送コンテンツのアーカイブシステムの試作", 情処理全大, 2E-2, 2013

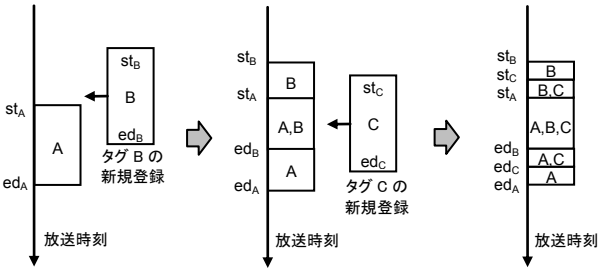

図 3 タグ名一覧の更新動作

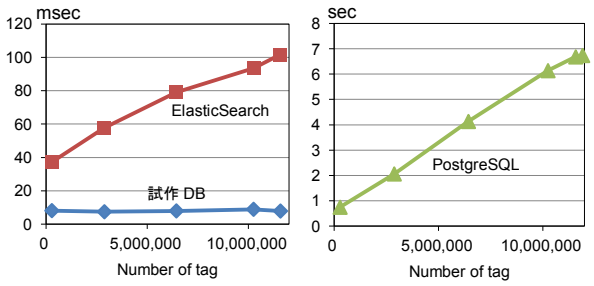

図 5 検索時間

表 1 検索時間(msec)の内訳(タグ数:11,578,921)

	試作 DB	Elastic Search	Postgre SQL
query ①-1	3.2	36.9 (28)	1.7
query ①-2			0.9
query ①-3		8.9 (1)	1680
query ①-4		8.8 (1)	2622
query ②	4.6	46.0 (38)	2378
その他	0.1	1.1	0.2
合計(msec)	7.8	101.6	6684

PostgreSQL に用いたクエリ

query①-1 : SELECT DISTINCT name FROM tags WHERE time @> '2011-11-11 08:10:00'::timestamp AND media='@NHKG';
query①-2 : SELECT max(lower(time)), min(upper(time)) FROM tags WHERE time @> '2011-11-11 08:10:00'::timestamp AND media='@NHKG';
query①-3 : SELECT max(upper(time)) FROM tags WHERE time &< tsrange('(2011-11-11 08:00:00,2011-11-11 08:10:00)') AND media='@NHKG';
query①-4 : SELECT min(lower(time)) FROM tags WHERE time &> tsrange('(2011-11-11 08:10:00,2011-11-11 08:15:00)') AND media='@NHKG';
query② : SELECT time FROM tags WHERE name='Aさん' AND lower(time) > '2011-11-11 08:10:00' AND media='@NHKG' ORDER BY lower(time) offset 0 LIMIT 1;

ElasticSearch に用いたクエリ

query①-1 : { "query":{ "bool":{ "must":[{ "term":{ "media":"@NHKG" } }, { "range":{ "st_time":{ "lte":"2011-11-11T08:10:00" } } }, { "range":{ "ed_time":{ "gt":"2011-11-11T08:10:00" } } }] } }, "facets" : { "st_stat" : { "statistical" : { "field" : "st_time" } }, "ed_stat" : { "statistical" : { "field" : "ed_time" } } }, "from":0, "size":30 }
query①-3 : { "query":{ "bool":{ "must":[{ "term":{ "media":"@NHKG" } }, { "range":{ "ed_time":{ "gt":"2011-11-11T08:00:00", "lt":"2011-11-11T08:10:00" } } }] } }, "facets" : { "st_stat" : { "statistical" : { "field" : "st_time" } }, "ed_stat" : { "statistical" : { "field" : "ed_time" } } }, "from":0, "size":30 }
query①-4 : { "query":{ "bool":{ "must":[{ "term":{ "media":"@NHKG" } }, { "range":{ "st_time":{ "gt":"2011-11-11T08:10:00", "lt":"2011-11-11T08:15:00" } } }] } }, "facets" : { "st_stat" : { "statistical" : { "field" : "st_time" } }, "ed_stat" : { "statistical" : { "field" : "ed_time" } } }, "from":0, "size":30 }
query② : {"filter":{ "and" : [{"range":{ "st_time":{ "gt":"2011-11-11T08:10:00" } } }, { "term":{"media":"@NHKG"}}, { "term":{"name":"Aさん"}}] }, "sort":[{ "st_time":"asc" }], "size":1 }

図 4 PostgreSQL と ElasticSearch のクエリ例