映像配信サービス向け
メモリ常駐型メタデータ管理システムの設計と実現

柿沼弘員[†1] 堀口恭太郎[†2] 深津真二[†3]
大橋盛徳[†3] 阿久津明人[†3] 鈴木英夫[†3]

本稿では，IPTV やモバイルマルチメディア放送などの映像配信サービスにおいて，従来の TV-Anytime に準拠したコンテンツメタデータに加え，近年の多くの Web サービスが利用しているようなユーザがコンテンツに対して付加するレビューやレーティング情報（CGM メタデータ）を扱えるように拡張し，それらのメタデータの統合的な管理を実現するメタデータ管理システムを提案する．本提案システムは，メモリ常駐型のリレーショナルデータベースを用い，コンテンツメタデータと新たに定義する CGM メタデータの要素を組み合わせた複合検索の高速処理や，膨大に増えるメタデータに対応するための管理機構，サービス事業者から定期的に更新されるコンテンツメタデータとユーザから不定期且つ高頻度に投稿される CGM メタデータの即時反映を実現する．また，各種メタデータの検索性能と登録性能の検証を提案システムと一般的なデータベースとの比較を通じて行い，本提案システムの有効性を確認した結果を報告する．

A Design and Implementation of In-memory Metadata Management System for the Video Distribution Service

HIROKAZU KAKINUMA[†1] KYOTARO HORIGUCHI[†2] SHINJI FUKATSU[†3]
SHIGENORI OHASHI[†3] AKIHITO AKUTSU[†3] HIDEO SUZUKI[†3]

1. はじめに

IPTV やモバイルマルチメディア放送（MM 放送）サービス[1]では，サービス事業者からコンテンツの情報（タイトルやジャンル，出演者，価格など）を記述したメタデータが登録され，ユーザは STB（Set Top Box）や対応端末を通じてメタデータを検索し，コンテンツを取得する．

一方，近年の Web サービスに目を向けると，大規模なユーザの生成するデータの集合をコンテンツとして取り込んだ CGM（Consumer Generated Media）サービスが主流となっている[2]．例えば，Amazon や YouTube，Netflix，Hulu などのコンテンツ流通サービスでは，ユーザの付加する感想やタグ，レーティングなどの情報を基準にコンテンツを選び，視聴，購入する．他にも，goo 映画や Yahoo!映画などのコンテンツレビュー情報だけを扱うサービスも一般的になっている．

また，番組視聴とソーシャルメディアを組み合わせた NHK の実証実験“teleda”[3][4]の結果として，視聴数上位の番組では，他者の行動（口コミや話題性）をきっかけに番組の視聴に到達したユーザが過半数を占め，VOD サービスに SNS 要素を取り入れることで幅広い視聴行動を生み出す可能性があることが示されている．

ここで，IPTV や MM 放送などで扱われるメタデータ[5]は，コンテンツ情報の表示，コンテンツの検索や購入・視聴を実現することを目的に“コンテンツメタデータ”として TV-Anytime Forum 等で規格化されており[6][7][8]，サービス事業者が専門家の論評をレビューとして登録することは想定されているが，一般ユーザによる自由なレビューやコメント，タグ，レーティングなどを登録するための要素（本稿では“CGM メタデータ”と呼ぶ）については検討されていない．また，従来のメタデータ管理システムでは，サービス事業者からの定期的なメタデータ登録に対応するだけで十分であったが，CGM メタデータを扱う場合には，多数のユーザからの不定期且つ高頻度なメタデータ登録にも対応するアーキテクチャであることが必要になる．

そこで，我々は，映像配信サービス分野における CGM メタデータを新たに定義し，コンテンツメタデータと CGM メタデータという性質の異なるメタデータを統合的に扱え，各メタデータの高頻度な登録と反映を実現するメモリ常駐型メタデータ管理システムを設計，実装した．本稿では，本システムの設計方針，システム構成及び各種制御方式を示すとともに，性能評価を行った結果を示す．

†1 株式会社 NTT ぷらら（元：NTT サービスエボリューション研究所）
NTT Plala Inc.
†2 日本電信電話株式会社 NTT ソフトウェアイノベーションセンタ
NTT Software Innovation Center
†3 日本電信電話株式会社 NTT サービスエボリューション研究所
NTT Service Evolution Laboratories

具体的には,2 章で関連研究,3 章で新たに定義する CGM メタデータの詳細，4 章で提案システムの設計方針，5 章で提案システムの構成と各種制御方式を述べ，6 章でプロトタイプの性能評価結果を述べる．

2. 関連研究

映像配信サービスにおける TV-Anytime に準拠したメタデータのように，1 件あたりに含まれる要素数が多く，複雑な階層構造を持つようなデータの自由な検索を実現するアーキテクチャとして，リレーショナルデータベース管理システム（RDBMS）がある．例えば，商用の RDBMS として Oracle Database が，オープンソースの RDBMS として PostgreSQL があるが，これらの DB はディスクへのアクセスになるため，例えば，1 件あたりのデータサイズが数キロバイトのメタデータの検索では，1000 件返却した場合の応答時間は数百ミリ秒オーダーになる．

一方，メモリキャッシュを利用した Key-Value ストア[9]ではメモリアクセスになるため，同条件の検索応答時間は数十ミリ秒オーダーまで高速化するが，Key-Value ストアのシステムでは，複数テーブルを結合（JOIN）し，各テーブルから任意に選択した複数カラムをキーとするような複雑な検索には対応できない．

また，メモリ常駐型 XML データベースとして，Karearea[10]があるが，トランザクションの保証や高頻度なデータ登録，スケールアウトには現状対応しておらず，今回想定する映像配信サービス向けのメタデータを管理するシステムとしては適していない．

他にも，メモリ常駐型 DB システムとして，RENA[11]があり，メモリセル型記憶構造，改良拡張ハッシングなどの特徴を持ち，テーブルを前テーブルと後テーブルの２つに大きく分け，検索及び登録の対象とするテーブルを使い分けることで，サービスを中断させずにメモリのメンテナンスを実現している．

しかしながら，ネットワークサービス（特に高度電話サービス）に特化して開発されたシステムであるため，単純なデータベースアクセスを対象に最適化（例えば，検索，更新,削除操作は条件指定による 1 レコードアクセスのみ）されており，今回想定する映像配信サービス向けのメタデータを管理するシステムとしては適していない．

3. CGM メタデータ

映像配信サービス分野における CGM メタデータを新たに定義するために，ユーザの付加するレビューやレーティング情報を活用している既存の Web サービス（Amazon, YouTube，iTunes Store，Netflix，Hulu などの物販・映像配信系サービスと，goo 映画，Yahoo!映画，映画.com などのコンテンツレビューに特化したサービスなど計 20 のサービス）において，ユーザによる付加や編集が可能な情報を調査した．

その結果，コンテンツやユーザとの関連性（誰の何に対する評価か）や出現数（1 つのコンテンツに対していくつ存在するか）の観点から，6 つのカテゴリに分類し，CGM メタデータを定義した．コンテンツ，ユーザ，メタデータ（コンテンツメタデータと CGM メタデータ）の対応関係を図 1 に示し,各カテゴリに含まれる要素の概要を述べる．

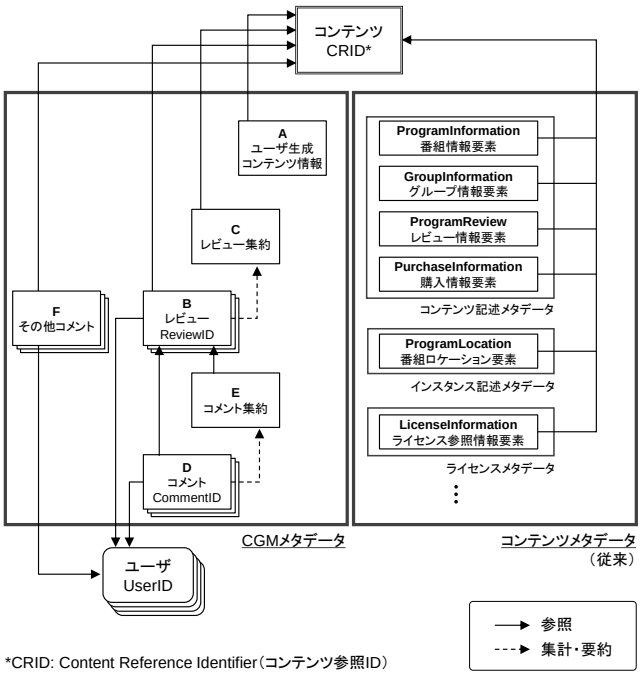

図 1　コンテンツに対する，ユーザとメタデータ（コンテンツメタデータ，CGM メタデータ）の参照関係

Figure 1　Reference relationship between contents, user and metadata (Contents metadata and CGM metadata).

(A) ユーザ生成コンテンツ情報

ユーザが制作したコンテンツの情報．従来のコンテンツメタデータと同様で，タイトル，概要文，制作日時などを記載する．

(B) レビュー

コンテンツに対するユーザ毎のレビュー情報．具体的には，ユーザ名，レビューテキスト，タグ，評価点数，投稿日時などを記載する．

(C) レビュー集約

コンテンツに対する各ユーザのレビュー（B）をとりまとめたデータ．具体的には，総レビュー数，評価点数の平均，評価点数の分散などを記載する．

(D) コメント

各コンテンツのレビュー（B）に対するユーザ毎のコメント情報．具体的には，ユーザ名，コメントテキスト，投票（“参考になった”など），投稿日時などを記載する．

(E) コメント集約

レビューに対する各ユーザのコメント（D）をとりまと

めたデータ．具体的には，総コメント数，投票数（“参考になった数”など）の合計などを記載する．

(F)その他コメント

コンテンツに対してトピックを設けて議論する形式をとる掲示板での投稿，チャットでの発言，ミニブログでの投稿（Twitter でのツイート）など．具体的には，ユーザ名，投稿テキスト，投稿日時などを記載する．

4. メタデータ管理システムの設計方針

4.1 要求条件

今回想定するサービスにおいて，コンテンツメタデータと CGM メタデータという性質の異なる 2 種類のメタデータを統合的に扱うメタデータ管理システムを考える場合，以下の要求条件を満足することが必要になる．

(1) 複雑な検索に対する高速なレスポンス

コンテンツメタデータと CGM メタデータの複合検索（例えば，“ジャンルが洋画でユーザの評価点数の平均が 4 点以上のコンテンツをタイトルの昇順で検索し，コンテンツ参照 ID と出演者を返却する”というような検索要求）に対する応答を高速に行える必要がある．

(2) データ量増加への対応

コンテンツメタデータに比べ，CGM メタデータの量はこれまで以上に膨大になるため，メタデータ量の増加に応じてシステムが柔軟に対応できる必要がある．

(3) 登録と検索頻度に適した管理方式

サービス事業者から定期的に大量のコンテンツメタデータが登録されるパターンと，多数のユーザからの不定期且つ高頻度に CGM メタデータが登録されるパターンの両方に対応する必要がある．また，メタデータの登録中もサービスを停止させることなく検索要求を常時受け付けられるように設計する必要がある．

4.2 設計方針

4.1 の要求条件を満たすために，以下の考え方に基づき設計を行った．

(1) メモリ常駐型 DBMS

検索性能の向上のため，DB をメモリ常駐型とし，メタデータの正規化とメモリアドレスのポイントを工夫する結合方式により，リレーショナルな検索も高速に実現できる方式を取り入れた．

(2) スケールアウト構成とデータの簡素化

増加する多種のメタデータへの対応として，登録側では，各メタデータの持つコンテンツ参照 ID 単位で複数 DB に振り分け，検索側では，全 DB に対して検索要求の一斉送信と集約を行うことで処理を分散できるような構成を採用し，管理側では，検索処理に遅延が発生することを極力避けたいため，データの圧縮を工夫し，登録処理が低負荷になるような中間データを生成保持する方式を採用した．

(3) メモリ面切り替えによるデータ更新方式

メモリ上のデータの格納領域を 2 つに切り分けることで，検索と登録の同時実行を可能とし，定期的な更新と不定期な更新の両方に対応できる更新方式を取り入れた．

5. メタデータ管理システムの詳細

5.1 システム構成

提案するメモリ常駐型メタデータ管理システムのシステム構成概要を図 2 に示す．

本システムに入力されるデータとしては，サービス事業者から提供されるコンテンツメタデータと，ユーザから投稿される CGM メタデータがある．コンテンツメタデータは低頻度（定期的）に更新され，CGM メタデータは高頻度（不定期）に更新されるような登録パターンを想定している．ここで，入力されたメタデータは，バイナリ形式データ（詳細は 5.2 節参照）に変換し，登録用の非アクティブなメモリ領域（バックエンドメモリ）に登録する．その後，検索用のアクティブなメモリ領域（フロントエンドメモリ）と領域の切り替え（メモリ面切り替え）を行うことで，登録結果を検索側に反映し，メタデータの更新を実施する．

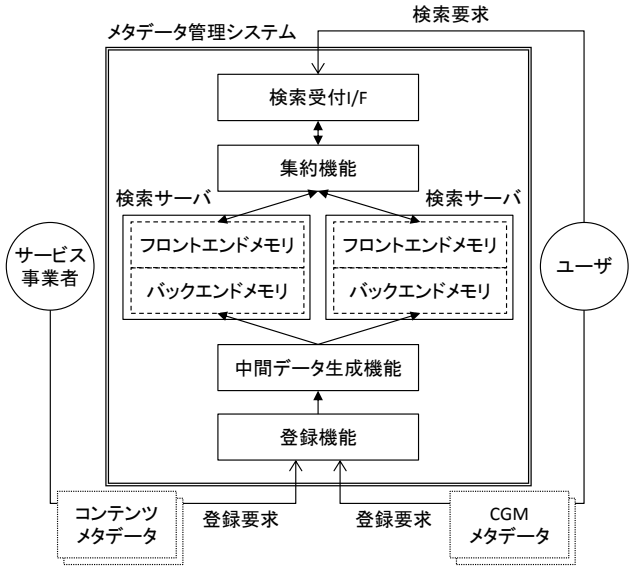

図 2　システム構成概要

Figure 2　Architecture of metadata management system.

5.2 スケールアウト構成とデータの簡素化

図 2 に示すように，本システムは，メモリ上にデータが展開される検索サーバを増設し，コンテンツ参照 ID 単位でメタデータを水平分割して（例えば，コンテンツ参照 ID が 0000001 から 0100000 までのメタデータは対象の 1 つの検索サーバに，コンテンツ参照 ID が 0100001 から 0200000 までのメタデータはもう一方の検索サーバに分割して）格納することで，メタデータ量の増加に対応できるアーキテクチャとしている．

このとき，メタデータの検索側では，集約機能が各検索

サーバへの検索要求の一斉送信と，各検索サーバからの検索結果の集計とフィルタリングを実施することで，システムのスケールアウトに対応する．

一方，メタデータの登録側では，中間データ生成機能が別プロセスで XML 形式のメタデータから登録用データイメージを生成し，それをメモリ上に登録する方式をとることで，低負荷な更新処理を実現する．ここで生成する登録用データイメージのことをロードデータと呼ぶ(図 3 参照)．

また，ロードデータを検索サーバのメモリ上に格納する際は，データサイズの圧縮，検索コスト軽減のため，データ格納時のカラム内のデータ型の種類を必要最低限（文字列，数値，バイナリブロックのみ）にすることで，データ構造の簡素化を行っている．

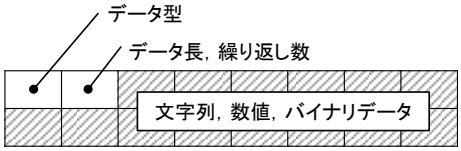

図 3　登録用データイメージ（ロードデータ）の書式

Figure 3　Format of an intermediate data for registration (Load data).

5.3 メモリ面切り替えによる更新方式

本システムでは，検索サーバの管理するメモリ領域を，検索要求のみ受け付けるフロントエンドメモリと，登録要求のみ受け付けるバックエンドメモリの 2 領域に論理的に分割することで，メタデータの登録時のロックが検索処理を妨げることを回避し，メタデータの高速な検索と登録の同時実行（ANSI/ISO SQL 規格[12]において定義されている READ COMMITTED 分離レベルのトランザクション）を実現する．この際，前述したメタデータの登録パターンに合わせて，一括更新方式と随時更新方式の 2 種類の方式を提案する．更に，随時更新と一括更新を併用する場合の方式を説明する．

(i) 一括更新方式

メモリ上の全メタデータを一括で更新する方式であり，メモリ上のデータの総入れ替えをすることで，登録したメタデータの一括更新を実現する．

具体的な処理フローは，図 4 に示すように，(1)バックエンドメモリに対して全メタデータを追加する．このとき，検索処理はフロントエンドメモリに対して行われるため，検索と登録の同時実行が可能である．(2)その後，バックエンドメモリへの書き込みが終わったタイミングでメモリ面切り替えを行い，(3)新たにバックエンドメモリになったメモリ領域のデータをクリアし，更新を完了する．

この一括更新方式では，新たにバックエンドメモリになったメモリ領域のデータを全件クリアするため，メモリの断片化解消の効果も得られる．

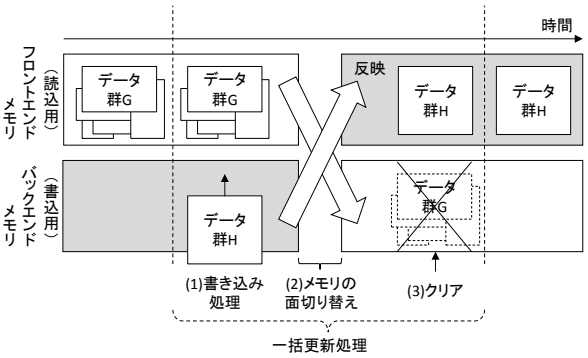

図 4　一括更新方式によるメタデータ登録フロー

Figure 4　Metadata registration flow by the batch update method.

(ii) 随時更新方式

メモリ上のメタデータを随時更新する方式であり，メモリ上のデータを維持しつつ，メタデータの追加，更新，削除を実現する．

具体的な処理フローは，図 5 に示すように，1 回の登録要求（追加，更新，削除）につき，(1)バックエンドメモリへの書き込み処理（追加，更新，削除）を行い，(2)メモリ面切り替えによりそれをフロントエンドメモリ側に反映する．(3)切り替え後のバックエンドメモリに再度書き込み処理を行うことで，(4)メモリの両面が同期された状態になり，1 回の登録処理が完了する．

この随時更新方式では，書き込み処理が読み込み処理をブロックすることなく，1 件単位でメタデータの登録と検索への反映を実現できる．

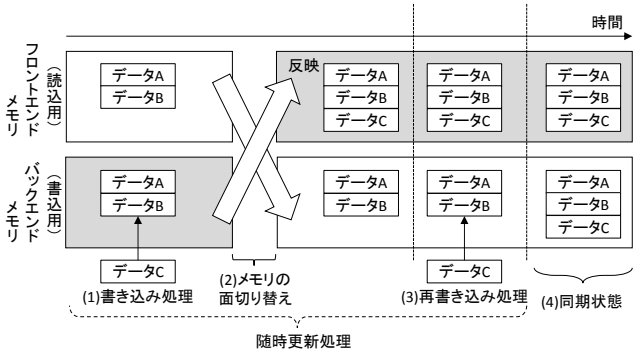

図 5　随時更新方式によるメタデータ登録フロー

Figure 5　Metadata registration flow by the frequent update method.

(iii) 随時更新と一括更新の併用

随時更新と一括更新を併用する場合，フロントエンドメモリとバックエンドメモリのデータが基本的に同期状態になっている必要があり，そのためには，一括更新を次のようなフローで行いながら，(ii)の随時更新方式と同様の処理を行うことで随時更新と一括更新の併用を実現する．

具体的な処理フローは，図 6 に示すように，(1)バックエンドメモリを全件クリアした後，(2)バックエンドメモリに対して全メタデータの追加を行う．このとき，検索処理はフロントエンドメモリに対して行われるため，検索と登録の同時実行が可能である．(3)その後，バックエンドメモリの更新が終わったタイミングでメモリ面切り替えを行い，(4)新たにバックエンドメモリになったメモリ領域のデータのクリアと，(5)全メタデータの追加を再度行うことで，(6)両面同期状態で更新を完了する．

この更新方式では，バックエンドメモリになったメモリ領域のデータを全件クリアするため，メモリの断片化解消の効果も得られる．

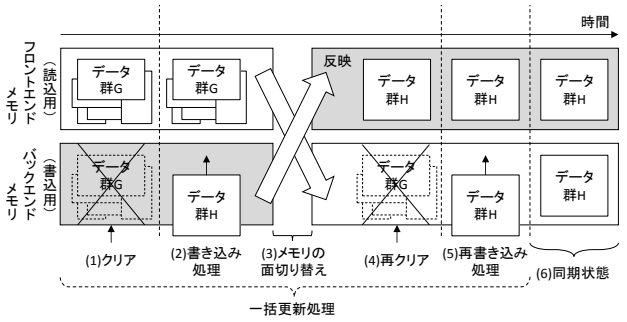

図 6　随時更新方式と併用する場合の，一括更新方式によるメタデータ登録フロー

Figure 6　Metadata registration flow by the batch update method. (When used in conjunction with the frequent update method.)

5.4 メタデータの正規化と事前結合

本システムでは，任意の複数カラムをキーとする検索及びコンテンツメタデータと CGM メタデータの複合検索を効率的に実現するために，メタデータの要素を，主となる master テーブルと従となる non-master テーブルに正規化し，メモリ上に展開する．

具体的には，図 7 に示すように，番組情報を記載する master テーブルに従属する形で，購入情報やライセンス参照情報などを記載する non-master テーブルに正規化する．また，CGM メタデータは non-master テーブルとして，対応する番組情報に対応付ける．これにより，データの管理が容易になると共に，あわせてよく検索される要素群がまとめて一つのテーブル内に保持されるため，検索の高速化が実現できる．

更には，各 non-master テーブルのレコードと master テーブルのレコードとの対応関係を，non-master テーブルのレコードの親となる master テーブルのレコードのメモリ上のアドレスへのポインタで保持すること（事前結合）で，master テーブルと non-master テーブル間の複合検索，すなわち，コンテンツメタデータと CGM メタデータの複合検索の高速化を実現している．

他にも，メモリ上に登録するメタデータの検索される多数の要素に対して，それぞれインデックスを設定しておくことを前提としているため，テーブル間の結合方式として，コンテンツ参照 ID をキーとしたソートマージ結合を採用することで，並列実行可能で高速な検索を実現している．

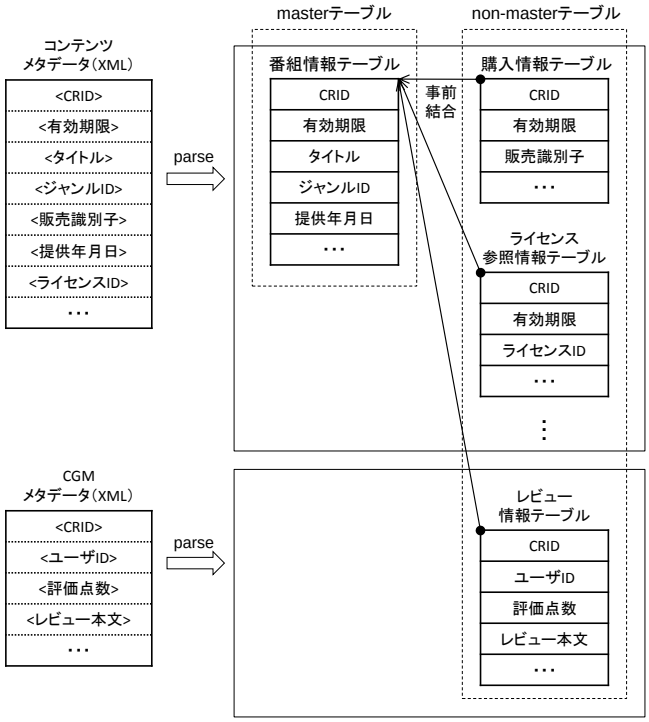

図 7　メタデータの正規化と事前結合

Figure 7　Normalization of metadata and pre-join of records.

6. 性能評価

6.1 登録性能

提案システムの随時更新方式を用いた場合のメタデータ登録（追加，削除，更新）性能について，ディスクにデータを格納する市販 RDBMS との比較を行った. 具体的には，DB に一定のメタデータ(2 万件)が格納されている状態で，実際のサービスで登録が想定されるメタデータとして十分大きいサイズ（約 20KB）のメタデータを登録要求として送信し，その応答時間を計測した．

提案システムでの応答時間に対する各システムでの応答時間の相対値を図 8 に示す．なお，図の相対値は提案システムのメタデータ追加時の応答時間を基準としている．

実験の結果より，追加の場合は，5.2 節で述べたロードデータの生成処理と，5.3 節で述べたバックエンドメモリへの 2 度の書き込み処理を行っているが，メモリへのデータ格納はロードデータを載せる軽量な処理であるため，結果的に市販 RDBMS の追加処理と同程度の性能が出ている．また，削除の場合についても，対象のレコードに削除フラグを付与する軽量の処理であるため，メモリ常駐型で検索が高速に実施できる提案システムでは高速に完了している．

更新の場合については，上記追加処理と削除処理を連続で実施しているが，追加に比べて削除は高速であるため，更新性能は追加処理にかかる時間に起因するところが大きく，市販 RDBMS と同程度の性能となっている．

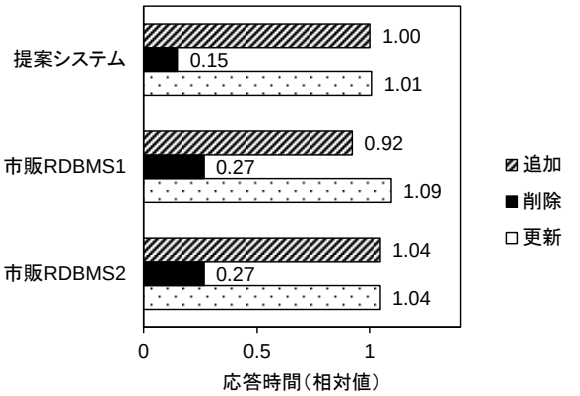

図 8　登録応答時間
Figure 8　Response time for registration.

また，一括データ登録時の応答時間を提案システムとディスクにデータを格納する市販 RDBMS3 とで比較する実験を追加で実施した. その結果を図 9 に示す. 具体的には，平均的なメタデータのサイズを目指した約 10KB のメタデータ 1 万件を登録する場合の応答時間を比較した．なお，図の事前処理とは XML 形式のメタデータを，提案システムではロードデータに, 市販 RDBMS3 では登録クエリに変換するための処理にかかった時間を示しており，登録処理とは，それらの中間データを各 DB に登録する処理にかかった時間を示している．

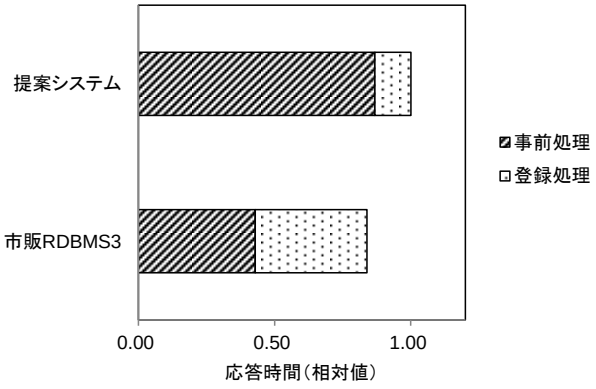

図 9　一括データ登録処理応答時間
Figure 9　Response time for registration by the batch update.

結果より，提案システムは DB に登録する前の XML 形式のメタデータをロードデータに変換する処理に時間がかかっていることが確認できた．これは，市販 RDBMS3 を含む一般的な RDBMS への登録については事前処理として XML をパースし，登録クエリを作成するだけであるが，提案システムでは，XML のパースに加えて，5.2 節に示したように登録後のメタデータ検索を高速化するための登録用データイメージを作成しつつデータの簡素化処理を行っているためである．

6.2 検索性能

提案システムのメタデータ検索性能については，一般的な RDBMS と，それらにメモリキャッシュを組み合わせた場合との比較を行った．具体的には，DB に一定のメタデータ（10 万件）が格納されている状態で, 1000 件ヒットし，1000 件返却する 1 種類の検索クエリをクライアント側から常時 10 多重（検索クライアントが 10 スレッド並列に，要求送信から応答受信までのサイクルを間断なく繰り返している状態）で一定時間送信し，その平均応答時間を計測した．提案システムでの検索応答時間に対する各システムでの検索応答時間の相対値を図 10 に示す.

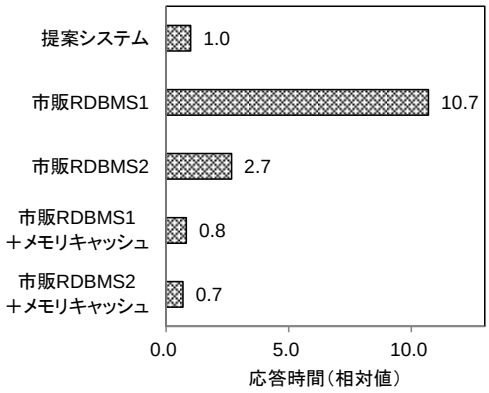

図 10　検索処理応答時間
Figure 10　Response time for search operations.

実験の結果，提案システムは高速に検索が実行できているが，メモリキャッシュを用いた場合はさらに高い性能を示す結果となった．これは，メモリキャッシュ利用時は，初回にキャッシュした検索結果を取得し続けており, 実質，毎回リレーショナルな検索は行われておらず，検索クエリをキーとしてメモリ上に用意された検索結果を取得し，そのまま返却する処理を行っているためである．

また，検索のパターンを変えた実験として，複数テーブルの要素を組み合わせた検索（複合検索）において，組み合わせるテーブル数を増減させた場合の検索応答時間を提案システムと市販 RDBMS3 とで比較する実験を追加で実施した．その結果を図 11 に示す．具体的には，DB に一定のメタデータ（1 万件）が 7 種類のテーブルに格納されている状態で，それらのテーブルの要素を組み合わせて 100 件ヒットし，100 件返却する複合検索を行った．検索クエリとしては，JOIN するテーブルの数が n の場合は，master テーブルとなるメインのテーブルから 7-n の要素を，それ以外の JOIN する n 個のテーブルからそれぞれ 1 つの要素を選択し，各要素を AND 条件で並べたものをキーとし，

コンテンツ参照 ID，コンテンツタイトル，コンテンツ概要の組み合わせをコンテンツ参照 ID の昇順でソートして返却する検索クエリを作成した．なお，市販 RDBMS3 では，検索キーとなる要素全てにインデックスの設定を行った．

実験の結果，提案システムでは JOIN するテーブル数を増加させていっても市販 RDBMS3 ほどの比で検索応答時間が遅くなっていくことはないということがわかる．これは，提案システムでは 5.4 節に示す事前結合を行っているためで，これにより毎回のテーブル間の結合処理の高速化を実現している．

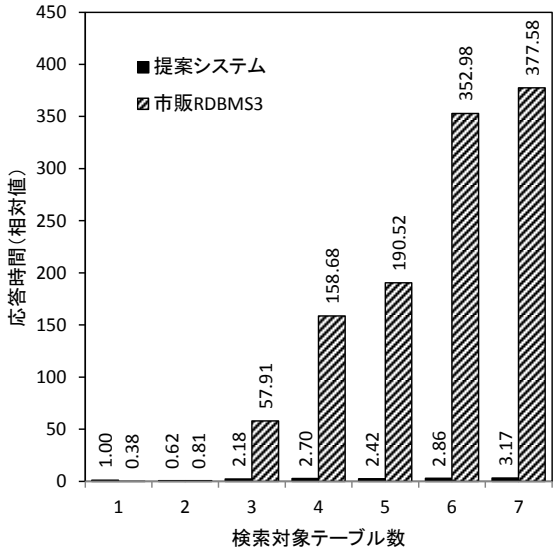

図 11　複合検索処理応答時間

Figure 11　Response time for complex search operations.

これらの結果より，メモリキャッシュは検索クエリのパターンがある程度決まっている場合には有効であるが，今回想定するサービスのように，検索クエリのバリエーションが多数存在し，複雑な検索が頻繁に行われることを想定する場合，キャッシュに存在していない検索クエリを後段の DB に問い合わせる処理が行われたときに，検索性能が大幅に低下する可能性があることがわかる．

7. まとめと今後の予定

本稿では，ユーザのレビューやレーティング情報を利用している既存の Web サービスの調査から，CGM メタデータとして 6 つのカテゴリに分類し，レビュー，レビュー集約，コメント，コメント集約のメタデータを新たに定義した．

また，コンテンツメタデータと CGM メタデータに対応したメタデータ管理システムとして，高頻度なメモリ面切り替えによるメタデータの即時登録・反映，master テーブルと non-master テーブルの事前結合による複合検索の高速化を特徴に持つメモリ常駐型メタデータ管理システムを提案した．そして，提案システムの性能を市販の RDBMS やメモリキャッシュと比較し，高頻度なメタデータの登録と，高速な検索が実現できることを確認した．また，その際，中間データであるロードデータの生成処理，事前結合処理など DB への登録の事前処理が有効に機能していることを示した．

今後の予定としては，市販のインメモリの RDB との性能比較や，本システムの実サービス使用に向けた運用機能（トランザクション管理，バックアップ，リカバリなど）の追加検討，検証実験を行い，提案手法の機能拡張を図る．

参考文献

1) 前橋佳林，日比野壮，深津真二，小野朗，堀井統之，“モバキャスにおけるコンテンツ情報管理方法の検討，”信学技報，Vol.111，No.203，pp.27-31，Sep. 2011.

2) 稲葉真純，長野伸一，長健太，溝口祐美子，川村隆浩，“CGM 分析技術の現状と課題 - メタデータ，オントロジーの応用可能性について -，”人工知能学会研究会資料，SIG-SWO-A603-06，Mar. 2007.

3) Masaru Miyazaki, Narichika Hamaguchi, and Hiroshi Fujisawa, "User Behavior in the “teleda” Social TV Service - Achieving a Public Forum on the Network -," Proceedings of the IEEE International Symposium on Multimedia (ISM2011), 2011, pp.345-350

4) 小川浩司，宮崎勝，馬場秋継，大竹剛，“ソーシャル機能による番組との出会いの創出 ： 番組レビューSNS サイト"teleda"の実証実験から，”放送研究と調査，pp.2-16, Sep. 2012

5) Broadcast and On-line Services: ("TV-Anytime"), Part 3: Metadata, Sub-part 1: Phase 1 - Metadata schemas, ETSI TS 102 822-3-1 V1.7.1, Nov. 2011.

6) “サーバ型放送における符号化，伝送及び蓄積制御方式，”ARIB 標準規格，ARIB STD-B38 2.1 版，Mar. 2011.

7) “セグメント連結伝送方式による地上マルチメディア放送運用規定，”ARIB 技術資料，ARIB TR-B33 1.2 版，Dec. 2011.

8) IPTVFJ STD-0006 CDN スコープ サービスアプローチ仕様 1.3 版，Jul. 2010.

9) R. Cattell. “Scalable SQL and NoSQL Data Stores,” SIGMOD Record, 39(4):12–27, 2010.

10) 川口浩司，沼崎慎吾，柳下精一郎，土井憲雄，“インメモリ XML データベースの開発，”情報処理学会研究報告，2003-DD-40, pp.9-16 (2003)

11) 中村仁之輔，芳西崇，梅本佳宏，小林伸幸，佐藤文明，水野忠則“ネットワークサービス向けメモリ常駐型リレーショナル DBMS の設計と実現，”情報処理学会論文誌 データベース，Vol.43, No.SIG 5(TOD 14)，pp.134-144，Jun. 2002.

12) ANSI X3.135-1992, American National Standard for Information Systems - Database Language - SQL, Nov. 1992.