

組み込み機器のデータ管理に適した軽量データベース TinyBrace™

TinyBrace™ Lightweight Database for Embedded Devices

金松 基孝

山地 圭

■KANEMATSU Mototaka

■YAMAJI Kei

近年、モバイル機器やデジタル家電製品は、ストレージ容量の増加に伴って様々なデータを管理することが可能になり、高度な機能を提供できるようになった。しかし、これらの組み込み機器では、ハードウェアやソフトウェアの制約が大きくサーバなどで利用されるデータベース (DB) を動作させることが難しいことから、軽量DBの導入が求められている。

このような製品の高機能化に対応させることを目的に、東芝は、OSS (オープンソースソフトウェア) のDB管理システムをベースに、組み込み機器で動作させるため機能強化した軽量DB “TinyBrace™”を開発した。この技術により、製品開発コストを抑えるとともに、短期間でのデータ管理モジュールの開発を可能にした。

With the increasing storage capacity of mobile devices and digital consumer products, the management of various types of data has become possible and advanced functions have become available. These trends have given rise to the need for a lightweight database in order to create products with advanced functions. However, the severe hardware constraints placed on embedded devices make it difficult to operate popular databases.

Toshiba has developed TinyBrace™, a lightweight database that is based on an open source software database management system. TinyBrace™ lowers product development costs and allows a data control module to be developed within a short period of time.

1 まえがき

近年、モバイル機器やデジタル家電製品の多機能化と多機能化に伴い、機器内部で管理するデータが大容量化してきている。また、高機能化によってアプリケーション相互の関係が複雑になり、同一データに複数のアプリケーションからアクセスすることが増えてきている。

これまでのデータ管理はファイルでの管理を採用していることが多かったが、データの大容量化や複雑化に対応していくためには、データ管理部分の開発効率化や検索性能の向上を実現できる、DBの導入が不可欠になってきている。一方で、ライセンス費用が少ない、低コストなDBが求められている。

東芝は、このようなニーズに応えるため、OSS (オープンソースソフトウェア) のリレーショナルDB管理システム (RDBMS) であるSQLite^(注1)をベースとして、モバイル機器やデジタル家電製品などの組み込み機器向けに機能を強化させた軽量DB TinyBrace™を開発した。ここでは、TinyBrace™で強化した機能とその効果について述べる。

2 SQLiteの概要

当社は、市販されている軽量RDBMSの11製品とOSS軽量RDBMSの9製品をベンチマークして、機能及び性能を調

(注1) SQLiteは、Hipp, Wyrick & Company, Inc. の商標。

査した結果、今回、SQLiteをTinyBrace™開発のベースとして選択した。

SQLiteは、著作権を放棄したパブリックドメインとして公開されており、コンパクトでありながら高速なRDBMSである^{(1), (2)}。一般的なクライアントサーバ型のRDBMSと異なり、DBサーバプロセスを持たないことから、プログラムサイズが約300 Kバイトとコンパクトである。SQLiteを使用するプログラムはSQLiteライブラリとリンクすることで、SQLiteの機能を使用することができる。プログラムはシンプルな関数呼出しを通してSQLiteの機能を使用する。また関数呼出しは、クライアントサーバ型のプロセス間通信より効率的であるので、DBにアクセス中の待ち時間を短縮することが可能であり、SQLiteは高速なRDBMSを実現している。

このように、SQLiteはコンパクトかつ高速なRDBMSであることから、モバイル機器やデジタル家電製品など組み込み機器に適した軽量DBである。しかし、組み込み機器にSQLiteをインストールするだけで簡単に動作するわけではなく、製品から求められている機能も備わっていないなど問題点がある。当社はこれらの問題を解決するため、以下に述べる機能強化のための開発を行った。

3 組み込み機器へ適用するための機能強化

組み込み機器はハードウェアやソフトウェアの制約が大きく、DBを組み込み機器で動作させるための機能強化が必要になる。

ここでは、組込み機器へ適用するための機能強化技術について述べる。

3.1 組込みOSへの対応

SQLiteがサポートするOS（オペレーティングシステム）はLinux^(注2)やWindows^{®(注3)}などがあるが、組込みOSはサポートされていない。組込みOSで動作させるためには、コンパイラの違いを考慮して実装する必要があることに加えて、タスク切替えやファイルシステムがLinuxやWindows[®]のような汎用OSと異なることを考慮する必要がある。

組込みOSのタスク切替えは、LinuxやWindows[®]のようなタイムシェアリング系のOSとは異なり、一定時間単位で、タスク処理を順番に実行させるラウンドロビン方式を採用していることが多い。ラウンドロビン方式では一定時間内で終了しなかったタスクは一度処理を中断し、待ち行列の最後に回され、これを繰り返す仕組みとなっている。SQLiteの関数は処理が完了するまで呼出し元が待たされる同期呼出し関数であるため、ラウンドロビン方式には向いていない。そこでTinyBraceTMでは、RDBMSを別のタスクで動作させるように非同期化することで、DB処理の実行権を制御できるようにして組込みOS上での動作を可能にした。

組込みOSのファイルシステムはLinuxやWindows[®]などと大差ないが、ファイル操作API（Application Programming Interface）と、ドライブやパーティションの扱いに違いがあり、それらを考慮して実装を行った。

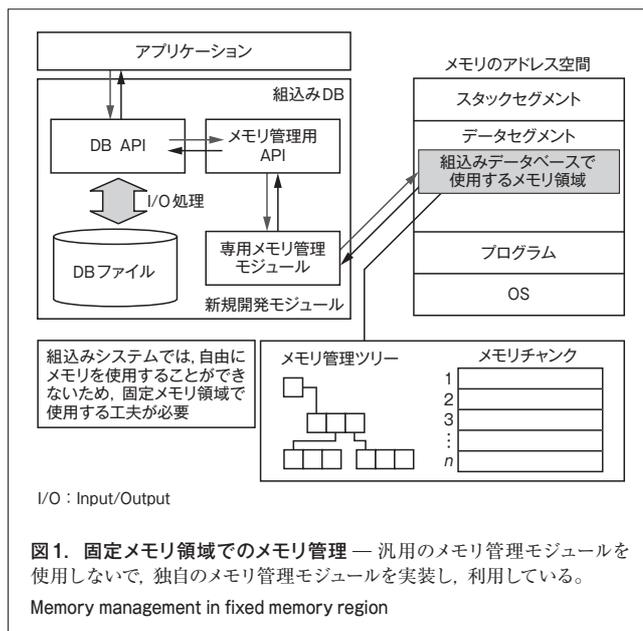
3.2 使用メモリ領域の固定化

メモリ領域の動的割当てと解放を繰り返すと、一般にメモリの空き領域が歯抜け状態になるフラグメンテーション（断片化）が発生し、処理速度の低下につながる。組込みOSにもメモリ管理機能があるが、フラグメンテーションを解消する機能（ガベージコレクション：GC）は備わっていない場合もあり、リアルタイム性を損なわないために、組込み機器では標準ライブラリの動的なメモリ割当てと解放を避ける傾向にある。

TinyBraceTMでは、あらかじめ専用のメモリモジュールを用意し、割り当てた領域のメモリを使い回すことで対応した（図1）。このモジュールは、SQLiteのメモリ割当て特性を解析し、いくつかの固定長サイズのメモリチャンクを用意することで、効率的なメモリ割当てを実現している。また、組込み機器ではメモリ使用量が限られていることが多いため、限られたメモリで動作するような実装を施している。

3.3 DBファイル領域の固定化

組込みOSのファイルシステムではフラグメンテーションを解消する機能があっても、機器の仕様上、機能を使用できない



場合が多い。DBに格納するデータの増加に伴ってファイルサイズも増加していくが、連続したファイル領域（ページの集合）として格納されない可能性がある。このDBファイルの断片化が性能低下につながるおそれがあることから、フラグメンテーションを起こさない工夫が必要である。

TinyBraceTMでは、このページ処理を改良し、直接ストレージ領域にページの確保と解放を行う方法ではなく、あらかじめ予想される最大ページ数を確保したDBファイルを作成し、その中でページの確保と解放を管理する機構を提供している。

4 軽量DBの機能強化

組込み機器へDBを適用するための機能強化のほかに、TinyBraceTMを搭載することで魅力ある製品にするための機能強化も必要である。ここでは、軽量DBの機能を強化するための技術について述べる。

4.1 索引型全文検索機能

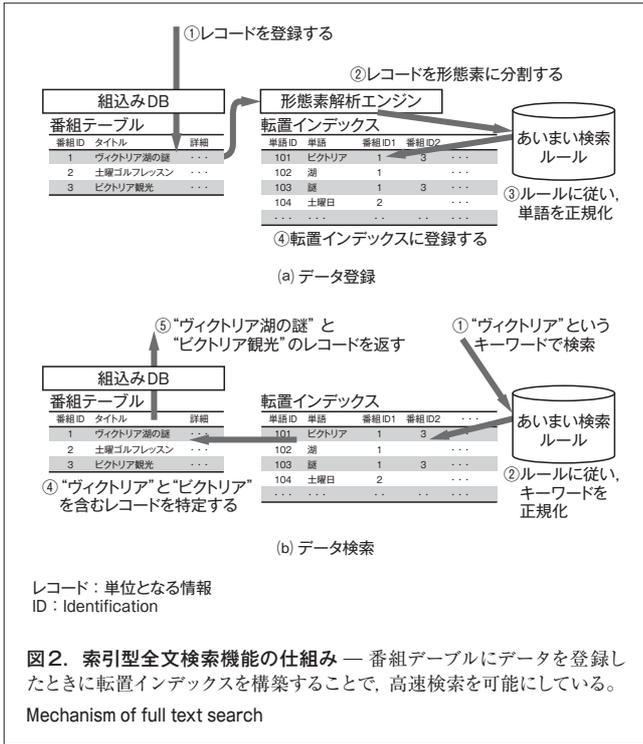
デジタル家電製品では、番組コンテンツに関する番組・録画情報などが機器内で管理されるようになったが、コンテンツ数が増えるにつれて検索時間が遅くなるといった問題があった。これは、コンテンツが格納されたテキストファイルの内容から検索対象となる文字列を順次探し出す、grep型と呼ばれる全文検索手法を用いていたためである。

TinyBraceTMでは、従来よりも高速で検索を可能にするため、索引型全文検索モジュールを開発した。この全文検索モジュールを組み込んだRDBMSで番組コンテンツ情報を管理することにより、高速な検索を実現した。

索引型全文検索の基本的な仕組みを図2に示す。この全文検索モジュールは番組テーブルにデータを追加したときに転

(注2) Linuxは、Linus Torvalds氏の米国及びその他の国における登録商標。

(注3) Windowsは、Microsoft Corporationの米国及びその他の国における商標。



置インデックスを構築することで高速検索を可能にしている。この転置インデックスを構築する手法には、辞書を用いて、データを形態素という単語単位に分割してインデックスを構築する形態素解析手法と、 N 文字 (N : 任意の整数) ずつに分割してインデックスを構築する N -gram 化手法が存在する。

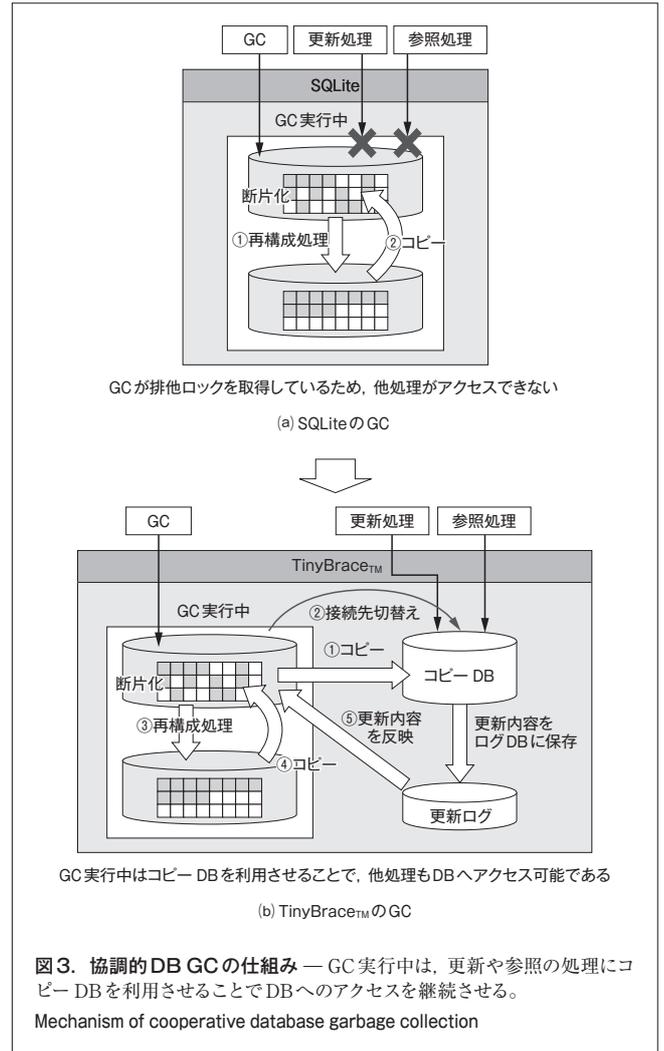
TinyBrace™では、字句解析部を手法ごとに独立させることで、製品の要求仕様や実行環境に応じて容易に選択できるようにした。5万件のコンテンツ情報を検索して表示するまで、従来は最大約30秒掛かる場合があったが、この機能により約1秒まで短縮することができた。

しかし、索引型全文検索機能を持つDBは、データ登録時に転置インデックスを構築するためメモリ使用量が多く組み込み機器で使用するのが難しい。そこで、助詞や句読点など、文章に類出するが検索語として意味がない語句をインデックス対応から除外することで、メモリ使用量を約50%削減した。

また、インデックス情報をルールベースで正規化することにより、かたかな異表記や同音異字など日本語特有の表現にも対応した全文検索を可能にした。例えば、“ビクトリア”で検索しても“ヴィクトリア”を含んでいる文章も検索することができる。

4.2 協調的DB GC機能

一般のRDBMSと異なり、組み込み機器ではDB管理者を介したメンテナンス処理が不可能である。また、24時間動作する組み込み機器もあり、メンテナンスによる動作停止は避けなければならない。したがって、DB管理者を不要にするアーキテクチャで稼働停止させない実装をする必要がある。例えば、DBへのデータの追加や、削除、更新が繰り返されると、格納



領域の断片化が進みDBの格納効率が低下する。メンテナンスとして、RDBMSには格納領域の断片化を解消するためにDB GC機能がある。GCを実行すると、実行中はDBが占有され利用できなくなり、機能を制限されてしまう。また、組み込み機器ではDB管理者が存在しないため、GCを一定周期で動作させる必要がある。

TinyBrace™では、GC実行中でもDBを利用できる協調的DB GC機能を開発した。処理の流れを図3に示す。

GC開始時に、まずメインDBのコピーを実施する。コピー中に更新処理が行われた場合、更新されたページだけをコピーし直すことでコピー側との同期を保っている。コピー終了後、メインDBのGCを実行する。アプリケーションのDB接続をコピーしたDBに変えて、DBへのアクセスを継続させる。更新処理は更新ログに記録し、GC終了後、更新ログをメインDBに適用することで、更新内容を反映させる。

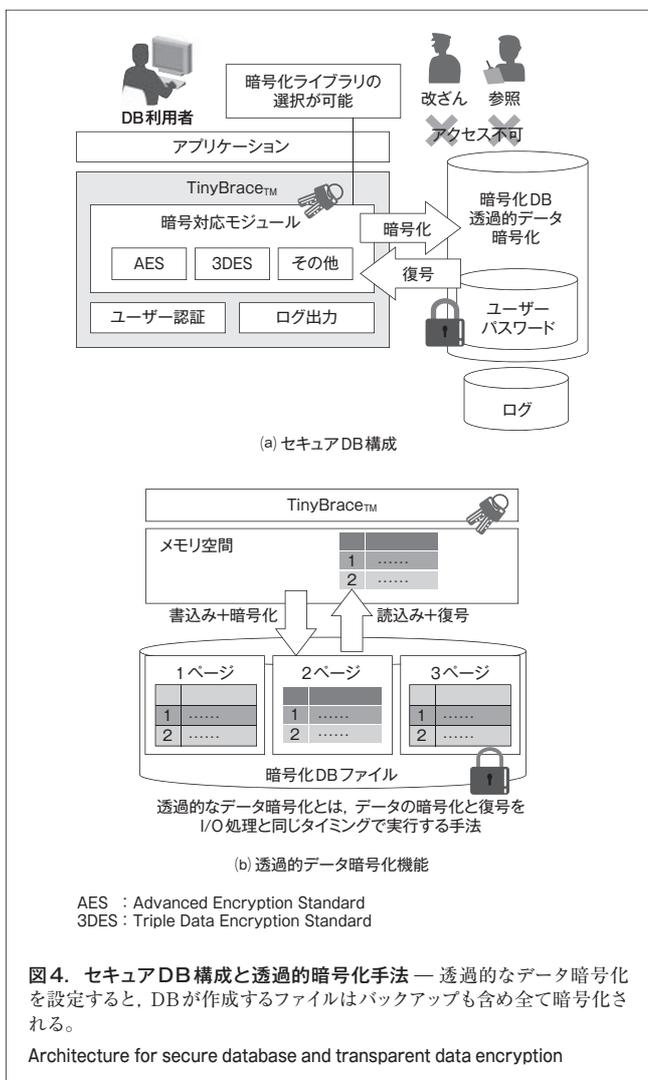
この機能により、メンテナンスの一つであるGCを、DB管理者なしで一定時間で実行することができ、実行中であってもデータの登録や参照が可能であることから、製品動作に制限

を設定しなくてもよい。

4.3 セキュリティ機能

DBはOSレベルからみるとファイルの集合体であり、データファイルを盗まれて解析されると内部の情報がのぞき見られてしまう可能性がある。SQLiteで作成されたデータファイルはバイナリエディタなどのツールを使うとデータの中身を容易に推定でき、情報漏えいやデータ改ざんなどの危険性がある。データファイルを暗号化することで、不正なアクセスによるデータ盗用及び情報漏えいを防止でき、重要なデータを守ることができる。ファイルシステム全体を暗号化する方法でも重要なデータを守れるが、ユーザー認証後は復号したデータファイルとして利用できるためデータ改ざんなどが可能であり、データファイル単位の暗号化も必要になる。

TinyBrace™の暗号化及び復号には透過的なデータ暗号化



(注4) DBファイルへの書き込み及び読み込みと同じタイミングで暗号化及び復号を行う手法。

(注5) TPC-Bは、Transaction Processing Performance Councilの商標。

手法^(注4)を採用した(図4)。この透過的なデータ暗号化はデータファイル単位での暗号化が可能のため、アプリケーションコードを変更せずに暗号化されたDBにアクセスできるというメリットがある。DBに対して透過的なデータ暗号化を設定すると、DBが作成するファイルは全て暗号化される。また、そのDBのバックアップも暗号化された状態になる。データファイルの暗号化と復号はページI/O (Input/Output) 時にリアルタイムに実行される。

関連する複数の処理をまとめて処理する際のトランザクション性能を測定するベンチマークの一つであるTPC-B^(注5)ベンチマーク⁽³⁾を用いて、通常のDBと暗号化したDBの性能を比較した結果、暗号化による性能劣化を最大8%に抑えることができた。

また、暗号モジュールはTinyBrace™から独立しており、暗号化ライブラリの置換えが容易なため、要件に応じて暗号化ライブラリを選択することができる。

5 あとがき

ここでは、モバイル機器やデジタル家電製品などの組み込み機器でも動作可能な軽量DB TinyBrace™を、OSSのRDBMSをベースに開発したことと、魅力ある製品にするために行った機能強化について述べた。TinyBrace™は、既にいくつかの当社製品に搭載されている。また、開発で得られた知見及びバグ情報については、SQLiteコミュニティへフィードバックしている。

今後は、組み込み機器だけでなく、社会インフラ製品のデータ管理においても活用されるように、TinyBrace™のいっそうの機能強化を進めていく。

文献

- (1) SQLite. "Features Of SQLite". < <http://www.sqlite.org/features.html> >, (accessed 2012-04-15).
- (2) SQLite. "About SQLite". < <http://www.sqlite.org/about.html> >, (accessed 2012-04-15).
- (3) TPC. "TPC-B". < <http://www.tpc.org/tpcb/default.asp> >, (accessed 2012-04-15).



金松 基孝 KANEMATSU Mototaka

ソフトウェア技術センター 先端ソフトウェア開発担当専事。
データベース管理システムの開発に従事。
Corporate Software Engineering Center



山地 圭 YAMAJI Kei

ソフトウェア技術センター 先端ソフトウェア開発担当専務。
データベース管理システムの開発に従事。
Corporate Software Engineering Center