

eXtremeDB™

In-Memory Database

組み込み機器への新しいアプローチ 比類なき性能と高い作業効率を実現するデータベース

リアルタイム性能を要求される組み込みシステムのデザインは、その複雑さをますます増大させています。また、より高い性能や信頼性への要求およびより短い開発サイクルに対応するため、開発者は信頼できるソフトウェアコンポーネントの採用に積極的になっています。McObjectのeXtremeDBは、組み込み用データベースエンジンとして性能・信頼性・開発の容易さの全てにおいての最高のメリットを開発者にもたらしめます。

McObjectのデータ管理技術はインテリジェントデバイスの要求を満たします。組み込み用データベースとリアルタイムシステムのエキスパートにより設立されたMcObjectは、ユニークな手法によりインテリジェントデバイスのIQをさらに高めます。

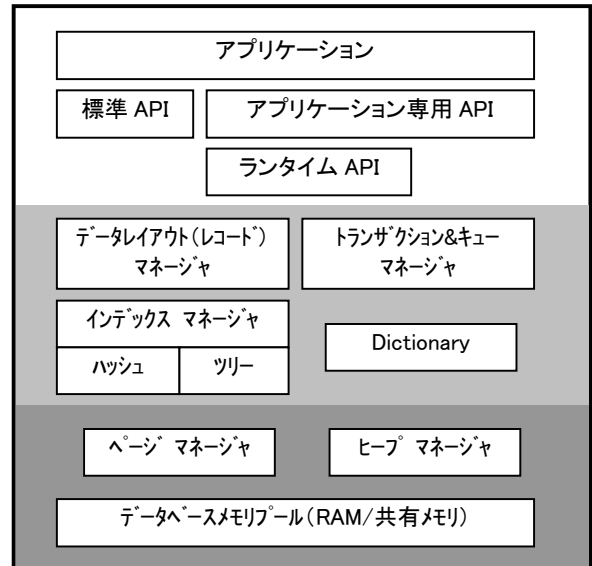
最近の技術の進歩を見ていると、例えばセットトップボックス、通信用スイッチ、コンシューマー製品、産業用制御機器等の組み込みデバイスにおいて洗練されたデータ管理が必要とされていることが明らかです。

既存の技術、つまり業務志向のデータベースの延長では組み込み機器が必要とするリアルタイム性の要求にはついて行けません。McObjectはこれらインテリジェントデバイスの要求を満たすために全く新しい技術を完成させました。

それが、eXtremeDBです。これは組み込みシステム用に最高のパフォーマンスと信頼性をコンパクトな容量で実現するだけでなく、効率のよいプログラミング環境も提供します。

データベースとして必要なACIDもサポートしていますので、トランザクション単位でデータは保証されます。

また、eXtremeDBはW3C SOAP準拠のXML変換機能も標準で備えています。他の機器とのデータ交換にますます重要性を増していくXMLフォーマットのデータとeXtremeDBのデータを自然な形でやりとりします。



採用実績 (一部)

Boeing (apatch), DaimlerChrysler, RadiCo, Homeland Security, F5 Networks, Peiker acoustic, Russia railways, Tyco Thermal Controls, Genesis Microchip,

特徴	利点
少ない要求メモリ - 100KB (CPUとコンパイラに依存)	厳しいリソースに対応可能
In-Memory Database - 全てのデータはメモリに保存	ディスクのオーバーヘッドなし
データへの直接アクセス	データ変換やキャッシュの為のオーバーヘッドなし
高性能	組み込みデバイスの要求に対応
無変換 - アプリケーションがデータをそのまま扱える	不要なオーバーヘッドの排除
データの完全性	高信頼性システムの構築
再帰可能なデータベースエンジン	マルチスレッドでの高性能実現

■ 開発環境

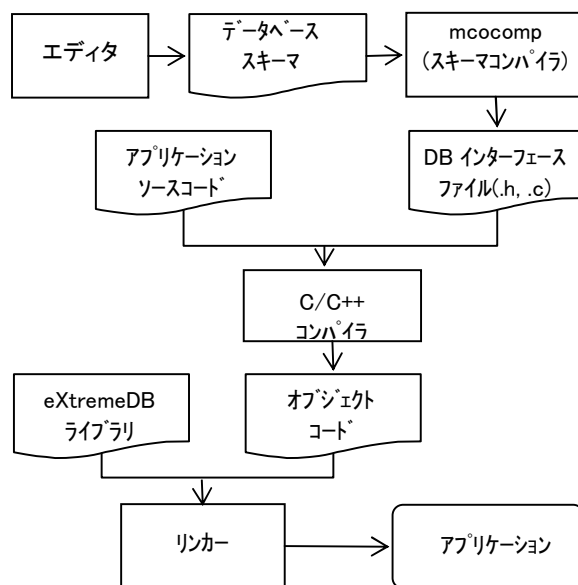
開発者は非常に限られた時間で、信頼性のある、管理しやすい効率のいいコードを作成するためにたゆまぬ努力を行っています。eXtremeDBは開発者がリアルタイムアプリケーションにDBを統合するときに有効ないくつかの特徴を備えています。

eXtremeDBのプロジェクトに特化したAPIは、操作されるデータのタイプを反映していますので、それぞれのデータベースの操作を確実にします。アプリケーションのデバッグにおいては、eXtremeDBは実行時のコードに多くのトラップを用意しています。これは開発やテストの目的または処理速度の最適化の為に無効にすることも可能です。McObjectは、アプリケーションにおけるeXtremeDBの理解をより深めるために、全てのソースコードを提供することも可能です。

さらに、eXtremeDBはクエリの為に極めて有効なインデックスとデータタイプをサポートします。クエリに関しては、完全一致検索の為にハッシュインデックスや、パターン検索の為にツリーインデックス、レンジ操作やソートの機能、オブジェクトIDの参照やデータの直接操作機能も提供します。

また、McObjectのeXtremeDBが採用したデータベースのアーキテクチャでは、ソートや参照用のインデックスにデータの複製をつけたりしませんので、絶対的に少ないメモリ要求にこたえることができます。

eXtremeDBを用いた開発は、通常のプログラム開発と変わりはありません。データの定義、コーディング、ビルド、デバッグのプロセスを経ます。どのフェーズにおいても、eXtremeDBは開発者を強力にサポートします。eXtremeDBは、カーソルの移動やDBのopen/closeといった基本的な操作を行う標準関数用と、データベースのスキーマから得られたデータ操作の2つのAPIを提供します。



■ 開発フロー:

<スキーマ DB定義> eXtremeDBを使用する場合、開発者は最初にデータの定義を行います。そしてスキーマコンパイラを使用してAPIを生成します。その結果、以下の利点が生まれます。

- ・アプリケーション別に最適化
- ・読みやすく管理しやすい
- ・Cコンパイル時のタイプチェック機能活用

<NavigationalAPI コーディング> 次頁の例でもわかるように、Cプログラマにとって何の障壁もないくらいに違和感なくDB管理プログラムの開発が可能です。

<ビルド> eXtremeDBはANSI準拠のコンパイラがあれば、実行するプラットフォームを選びません。開発者が使い慣れた環境で開発を行えます。また、コンパイラがビルド時にデータのタイプなどのエラーをチェックするので、実行されるコードは、非常に信頼性の高いものになります。現実的な問題として、実行時のデータ破壊などは、単純なポインタータイプの不整合により発生することが多くあり、このようなバ

グが発生したときのデバッグに予想以上の工数が必要になることも決して珍しくありません。です。VOIDタイプを多用するインターフェースはこのような危険が伴いますが、eXtremeDBではVOIDデータは使用しませんので、もし不整合が発生するコードを書いたとしても、コンパイラが厳密にチェックしてくれます。

<デバッグ> もし、アプリケーションが誤って間違ったトランザクションやオブジェクト操作を実行しようとしたら、初期設定のeXtremeDBは致命的な例外を通知してプログラムの実行を止めます。ほとんどのケースでは、アプリケーションの開発者はソースコードの間違いを見つけるために呼び出したスタックを調べます。eXtremeDBのランタイムはトラップや矛盾の確認を行います。その場合、CPUサイクルやメモリが余分に必要になります。アプリケーションがデバッグされ矛盾がなくなったときには、開発者は、トラップや内部のチェック機能を削除し、貴重なCPUサイクルをとりもどしたeXtremeDBランタイムのリリースバージョンを生成することができます。

スキーマ定義例:

```
#define uint4 unsigned<4>
#define uint2 unsigned<2>
#include "common.h"
declare database meteo;
struct Datetime {
    uint2 year;
    uint2 month;
    uint2 day;
    uint2 hour;
    time timestamp;
};
class Data {
    uint4    key;        /* unique key */
    Datetime dt;
    uint2    temp;      /* temperature */
    float    pres;      /* pressure    */
    uint2    humid;    /* humidity    */
    hash <key> IKey[100000];
    tree <dt.year, dt.month, dt.day, dt.timestamp> Idt;
};
```

データ操作例:

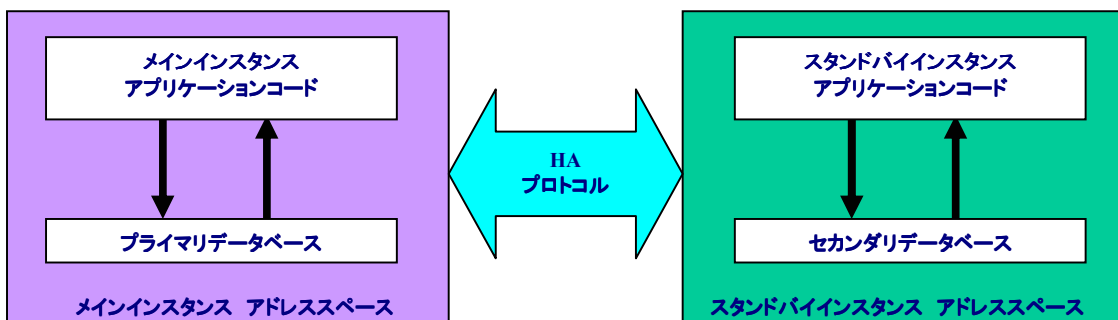
```
Data data; // handle to class Data defined in
schema
Datetime dt; // Datetime structure defined in schema
rc = mco_trans_start ( db, MCO_READ_WRITE,
MCO_TRANS_FOREGROUND, &t );
rc = Data_new( t,&data );
// put fields:
Data_pres_put(&data, get_pressure() );
Data_humid_put(&data, rand2(50, 70) );
Datetime_year_put ( &dt, year );
Datetime_month_put( &dt, month );
Datetime_day_put ( &dt, day );
Datetime_hour_put( &dt, hour );
Datetime_timestamp_put( &dt, get_clock() );
rc = mco_trans_commit (t);
```

■ **信頼性拡張オプション:**

eXtremeDBはデータベースの信頼性をさらに高めるためのオプションを用意しています。

＜ハイアベラビリティ＞

このオプションにより、複数のデータベースを同期させることが出来るようになります。同一性を保つべきデータベースは別のボードやケースに入っても、高速の通信が保たれていれば問題ありません。データベース間の通信手段には依存しませんので、LAN接続された別ユニットや高速バスでつながったボード間でも同一DBを維持できます。また、同期には、時間を認識した2フェーズコミットプロトコルを採用していますので、メイン・スタンドバイの両DBがどちらも同じようにデータの更新に成功または失敗することを保障します。



＜トランザクションログ＞

このオプションにより、二次記憶装置へのデータ更新をスムーズに実現します。トランザクションの発生時に、変更のあった部分のみをDBの更新とは別に記録します。この情報は非常に小さなものですので、DB全体のバックアップに比べると遥かに高速に処理します。システムの空き時間や定期的なサイクルで、二次記憶上のDBデータを更新(チェックポイント)し、異常発生時には、DBとトランザクションログで最新の状態に修復します。

対応プラットフォーム:

- Linux, QNX 6, LynxOS
- Windows Embedded Platforms
- VxWorks 5.4/5.5, Quadros RTXC
- Solaris, HP-UX, Windows 98/Me/NT/2000/XP
- eXtremeDBのソースコードは全てのプラットフォームに対応可能です。

データベースの仕様:

- 1データベースの最大オブジェクト数:
2,147,483,647
- 1データベースの最大クラス数: 32,767
- 1データベースの最大インデックス数: 32,767
- 1クラスの最大フィールド/ベクター数: 32,767
- 1インデックスの最大フィールド数: 32,767
- 1ベクターの最大エレメント数: 32,767
- 必要メモリ容量: 約 50KB
- 最大同時オープンデータベース数: 16
- 1データベースの最大同時接続数: 64

複雑なデータタイプと効率的なクエリ:

- 構造体、アレイ、ベクター、BLOBのサポート
- 正確な検索の為にハッシュを含むクエリ
- パターンマッチ、レンジ検索、ソートのクエリに対応したツリーインデックスのサポート
- Voluntary インデックスのサポート
- オブジェクト認識によるデータへの直接操作
- オブジェクト認識の為に自動ID生成
- 使用メモリ容量の最小化のための最適化
- 同期・非同期イベント通知

サポートデータタイプ:

- 1, 2, 4, 8ビット 符号付整数 (signed)
- 1, 2, 4, 8ビット符号なし整数 (unsigned)
- 列挙型 (enum)
- 4ビット実数 (float)、8ビット実数 (double)
- 日付 (date)、時間 (time)
- 固定長文字 (char)、可変長文字 (string)
- 固定長アレイ (array)
- 可変長ベクター (vector)
- 構造体 (structure)
- 自動ID (auto-increment)
- BLOB (Binary Large Object)
- ユーザー定義のオブジェクトIDとリファレンス

関連製品

■ SPEED Kit

AMD-Alchemy Au1500 プロセッサ(MIPS32コア)搭載の小型コントローラとeXtremeDB開発キットのバンドルHW:

AMD-Alchemy Au1500-400MHz CPU, 64MB RAM, 16MB FLASH,
interface: Ethernet x2, serial x2, USB 1.1-Host, Audio, CompactFlash

SW:

Linux kernel 2.4.21 & Linux Toolchain, MIPS cross gdb 5.3, gdbserver, Debug monitor, Busybox 1.00 等
eXtremeDB 開発キット - Shared Memory edition for MIPS Linux, DLL Compiler (x86 host)

■ eXtremeWS

コンパクトで高性能な組込み向けWebサーバーソフトウェア

- ・ソースコード提供でロイヤリティフリー(プラットフォーム、開発者、製品単位)
- ・フットプリント 15K - 35K
- ・HTTP 1.1 (RFC2616) compliant, HTTP 1.0、インプロセスCGI対応、セッション対応、コンテンツコンパイラー
- ・通信層から独立
- ・確認済みOS: WIndows, Embedded Linux and Linux, Lynx OS, QNX