

# 超高速読み込み専用DB Mimdb(MasterInMemoryDatabase)

# Webアプリのあれこれ

熟練Webアプリケーション開発者ならば「当然」の話なのですが、Webアプリケーションの約半分の処理時間が「データベース処理」であることを。

データベースは何故遅いのでしょうか？何処が遅いのでしょうか？遅いと言われている部分はどこにあるのでしょうか？何が遅いのでしょうか？

1. トランザクション書き込み
2. 集計処理
3. ディスクI/O
4. プロセス間通信
5. SQLパーサー

実は単純なselect文は、それ自体は遅くないのですが、この場合3, 4, 5の処理が発生するわけで、これが遅さの原因です。

# Webアプリのあれこれ (つづき)

Webアプリの性質にもよりますが、一般的なホームページタイプのWebアプリの場合、トランザクション書き込みを行う機会は、あまり有りません。

どちらかと言えば、読み込み処理が大半を占めるはずですが、それでも、Webアプリケーションの処理時間の約半分を、データベースI/Oが占めるわけで、これを「超高速」にすれば、その分「超高速」なWebアプリケーションができるかも知れません。

前頁に書いたように、単純なSelect文でも「ディスクI/O」「プロセス間通信」「SQLパーサー」これらが、速度を落としているわけで、SQLパーサーはSQL文を使う以上、しかたのないことですが仮に「ディスクI/O」「プロセス間通信」を除外できれば、かなりの高速動作が可能となるはずですが。

これを考慮したのが、mimdb (MasterInMemoryDatabase) です。

# Mimdb(MasterInMemoryDatabase)

Mimdbは、読み込み専用のJava専用データベースです。  
利用方法として、Embedded (組み込み)方式を採用しています。  
これによって「プロセス間通信」が一切発生しないことから、  
爆速なデータベースを構築することが可能となります。

また、名前の通り、Mimdbはインメモリです。

昨今OSの64Bit化と、それに伴うメモリ容量の増加、価格低下により、予算内で、数十ギガのメモリをサーバーに積むことも可能となったことにより、インメモリーデータベースの開発が活発なものとなっています。

まだ、分野としては、始まったばかりの技術ではありますが、最低でも既存DBの10倍以上の速度が見込めることから、Webアプリケーションに採用する事例も増えてきています。

Mimdbも同様に既存DBと比べると、大体10倍～100倍の処理速度差を見込めます。

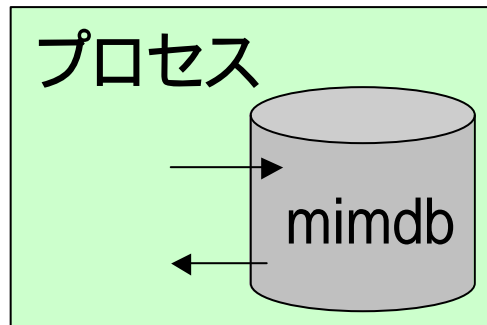
# Mimdbのあれこれ

Mimdbは、読み込み専用のインメモリ組み込みデータベースです。  
名前の通り、利用用途としては、書き込み頻度の無いマスター  
テーブル用のデータベースです。  
100%PureJavaで動作します。

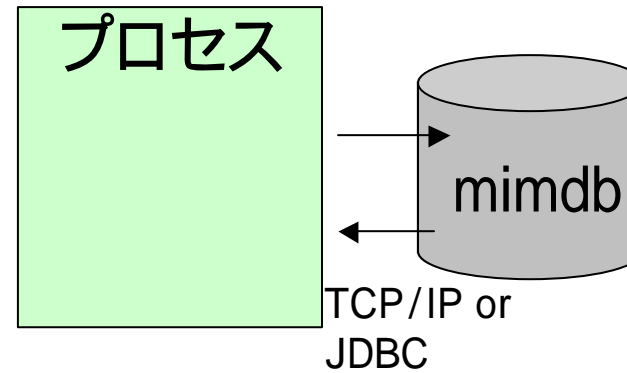
JDBCにも完全標準ではありませんが、対応はしています。  
また、組み込み方式以外でも、サーバー起動モードも対応しており、  
速度は落ちますが、込み入った検索でも、安定した速度で返却  
されるので、たとえば、郵便番号情報などの検索で、Javascriptなど  
から検索する場合などは、利用用途として、確立できます。

読み込み専用のデータ情報は、CSVファイルか、JDBC経由での  
他のデータベースサーバーから、メモリ上にインメモリデータベース  
を複製することが可能です。

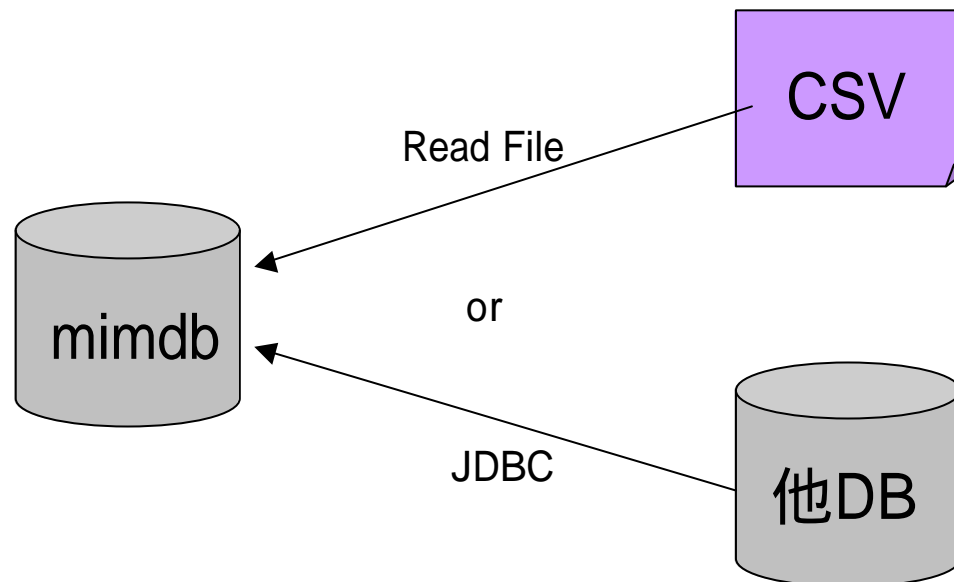
# Mimdbのあれこれ (つづき)



Embeddedモード



サーバーモード

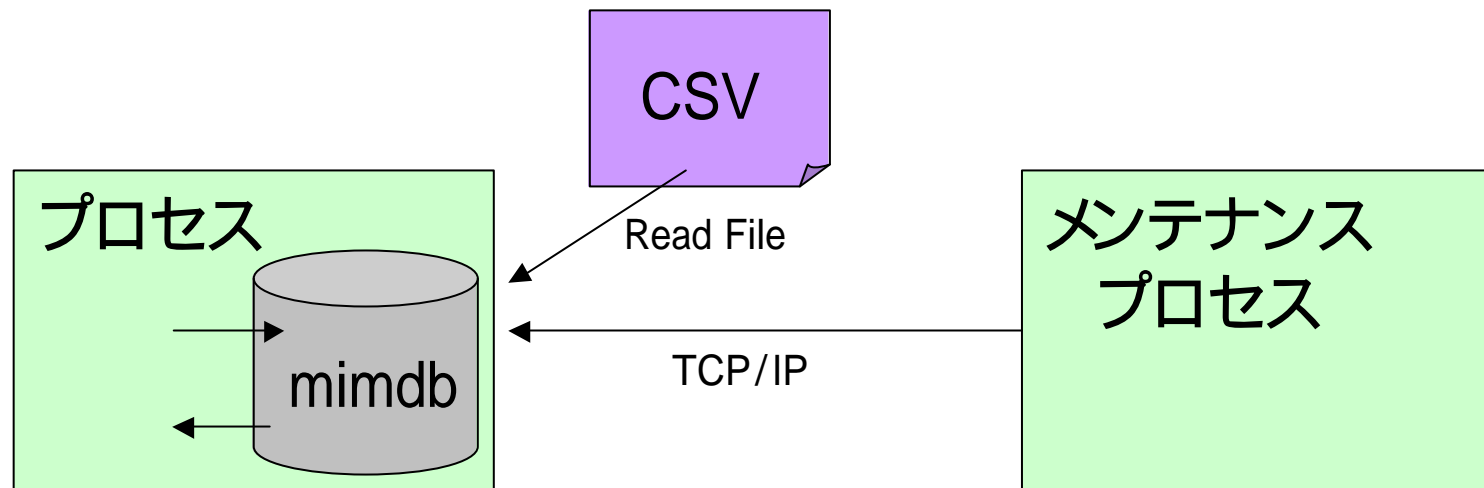


Mimdbインデックス作成処理

# Mimdbのメンテナンス

Mimdbには、メンテナンスモードがサポートされています。  
一々、Mimdbのインデックス張替えを行う毎に、組み込んでいるシステムの再起動をするのは、現実的では有りません。

メンテナンスモードを利用することで、現在の読み込み済みのデータベース内容の閲覧やバージョン番号、データ再読み込みを行い、インデックスの張替えなどが、このモードで行うことができます



# Mimdbのベンチマーク

以下に、MimdbとPostgresとの比較ベンチマーク結果を示します。

単純クエリー 検索10万回の結果 テーブル行数約3000行

```
SQL=select id,name from xxx_master where id=1 ;  
Postgres::25828msec  
mimdb:313msec 82倍
```

単純クエリー 検索10万回の結果 郵便局での郵便番号全件[KEN\_ALL.csv]データ(11万件)

```
SQL=select * from zip_master where id='1000000';  
Postgres: 26859msec  
mimdb:531msec 51倍
```

between検索10万回の結果 テーブル行数約3000行

```
Postgres=select id,name from xxxx_master where id BETWEEN 1000 and 1100  
time:50890msec  
  
mimdb=select id,name from xxxx_master where id between(1000,1100) ;  
time:781msec 65倍
```

単純計算で、PostgresqlよりMimdbは51倍から82倍の速度差。



# Mimdbの欠点

- ・Insert, Updateなど書き込み処理が行えない。
- ・テーブル結合が出来ない
- ・Viewができない。
- ・インデックス作成に時間がかかる (郵便番号11万件で4秒)
- ・集計処理 (Sum、Min、Max、Avg)ができない。
- ・Group Byができない。
- ・IN定義が他のSQL文と違う
- ・Betwee定義が他のSQL文と違う

# さいごに

Mimdbはオープンソースです。  
Apache2.0ライセンスで提供されています。

以下のサイトからダウンロード可能です。  
(現在バージョンは1.02)

<http://sourceforge.jp/projects/seabass/downloads/60299/mimdb-1.02.zip/>

または、googleなどの検索エンジンで[mimdb]と検索すれば  
sourceForge.jpのページが出てくるので、そこで最新版を  
ダウンロードしてください。