LevelDB のちょっとした話

@天下一 InfluxDB 勉強会 (LT)

nhiroki

2014/06/27

この LT について

- LevelDB を使って得られた知見をランダムに喋ります (C++ binding, version 1.17)
- LevelDB を使って何か作りたい人向けです

- InfluxDB の話はほとんどないです
- 代わりにブラウザのストレージである IndexedDB (Chrome 実装) の話が出てきます

LevelDB の特徴（おさらい）

- Key と value は任意のバイト列
- データは key でソートされる
  - デフォルトは byte-wise な辞書順
- ランダムアクセス or 双方向イテレータによるアクセス

これらを踏まえた Key の設計

- なるべくシーケンシャルアクセスしたい
  - [参考] LevelDB Benchmarks
- アクセスしたい順にデータが並ぶように key を作る

Key coding の方針

時系列データ (InfluxDB)

- id と timestamp と seq で key を生成

順序を持たないデータ (IndexedDB)

- 1 LevelDB インスタンスに複数 DB/ObjectStore を格納
- DB/ObjectStore 単位にデータが並ぶよう key を生成
- 例) [database id][store id][1][key] => [value]

[参考] indexed_db_leveldb_coding.cc

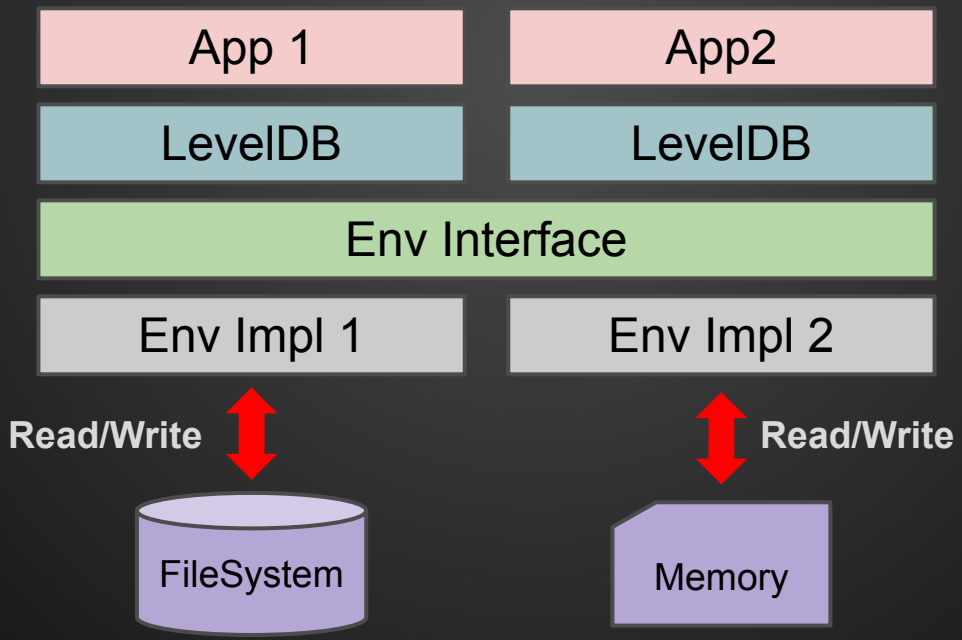
辞書順以外の順序を持つデータ

- 独自の比較関数 (comparator) を作る

Env - Platform Interface

OS (特に FS) の機能にアクセスするためのインターフェース

DB open 時に指定可能, FS へのアクセス前後に色々できる



Env - Platform Interface

IndexedDB (Chrome) は独自 Env を使用

- Corruption 対策
- メトリクスの取得
- ファイル書き換え時にディレクトリに fsync

[参考] env_chromium.h

InfluxDB ではデフォルトの Env を使用

MemEnv が便利

- On-memory で DB を作成
- LevelDB に標準で付属
  - leveldb/helpers/memenv/memenv.h
- 使い道
  - テストの高速化
  - IndexedDB (Chrome) のシークレットモード

```
Env* env = NewMemEnv(Env::Default());
Options options;
options.create_if_missing = true;
options.env = env;
DB* db = NULL;
DB::Open(options, path, &db);
```

ステータスコード

例) 破損した DB ファイルを open/read しました。
返ってくるステータスは次のうちどれでしょうか？

(1) Ok,
(2) NotFound,
(3) Corruption
(4) NotSupported,
(5) InvalidArgument
(6) IOError

Read 時に InvalidArgument が返ってきた時は, 基本的には Corruption 扱いで大丈夫 ??

ステータスの確認

Status はチェック用のメンバ関数で確認

```
class Status {
 public:
  // チェック関数は　4 種類
  bool Ok() const;
  bool IsNotFound() const;
  bool IsCorruption() const;
  bool IsIOError() const;
```

```
 private:
  enum Code {
    kOk, kNotFound, kCorruption,
    kIOError, kNotSupported,
    kInvalidArgument,
  }; // ステータスは　6 種類
  Code code_;
};
```

~~IsNotSupported() と IsInvalidArgument() がないが、kNotSupported は使われていないので、上記 4 つでなければ InvalidArgument~~ (バッドノウハウ, Issue)

まとめ

- データやアクセスパターンを考慮して key を作る
- 読み書きで特別な操作をしたければ Env を使う
- テストでは MemEnv を使うといいかも
- ステータスコードは注意してチェック