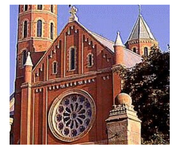
Saint Vincent College

Powderbase

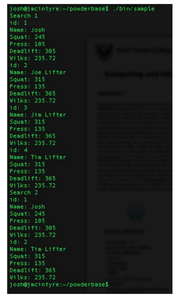
Josh McIntyre and Dr. Cynthia J. Martincic

Computing and Information Science Department, Saint Vincent College, Latrobe, PA 15650



ABSTRACT

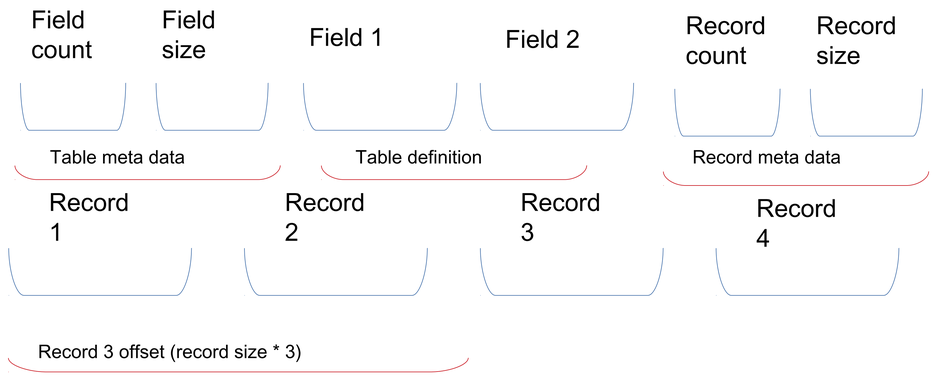
Powderbase is a lightweight, performance-oriented database engine and API for C++ programmers, designed to bridge the gap between database management systems and flat-file storage. The engine seeks to provide a platform that is easier to use and requires a lower footprint than full-fledged relational or document database engines such as MySQL or MongoDB. It also seeks to provide better performance than traditional flat-file storage formats such as XML, JSON, or CSV, especially when storing and retrieving numerical data. Powderbase uses an original binary file format designed to balance the concerns of processor and memory usage. It provides all of the critical operations of a database (create, read, update, and delete) and each algorithm is implemented with performance in mind. Powderbase is under continual development and available for reuse under the BSD free and open source license.



HOW IT WORKS

- Uses a flat-file to reduce memory usage
- Uses a binary file format
- Uses evenly-sized records for quick file access
- Designed with balanced memory and processing performance in mind
- Provides an easy-to-use C++ API

.

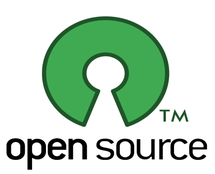


Example: Update Record 3

- Calculate offset to the start of records (table meta + table + record meta)
- Calculate offset to the start of record 3 (record size * record id)
- Overwrite the block of data with the updated record object

DEVELOPMENT

- Language: C++
- Build tools: GNU Make
- Tested platforms:
  - Linux
  - Windows
  - Mac OSX
  - BSD
- License: BSD 2 Clause
- Website: jmcintyre.net/gitlist/powderbase/development



IMPROVEMENTS OVER TRADITIONAL DATABASE MANAGEMENT SYSTEMS

- Flat file simplifies:
  - User access
  - Backup
  - Import/Export
- Reduced feature set; easy to use
- Uses less space on disk
- Better performance in limited environments

IMPROVEMENTS OVER TRADITIONAL FLAT-FILE FORMATS

- No syntax parsing needed; less memory/processor usage
- No need for typecasting for native data types
- Uses less disk space for numerical data

PERFORMANCE METRICS

Test system information:

- CPU: Intel Xeon 2.40 GHz
- Memory: 1.01 GB RAM
- OS: Ubuntu 14.04 LTS 64 bit
- Build: development 804a780
- Records tested: 10,000

Results:

- Database create: 0 ms
- Record insert: 249 ms
- Record update: 224 ms
- Record search: 190 ms
- Record remove: 536 ms
- Database size on disk: 821 KB
- Library size on disk: 674 KB

Saint Vincent College

Powderbase

Josh McIntyre and Dr. Cynthia J. Martincic

Computing and Information Science Department, Saint Vincent College, Latrobe, PA 15650

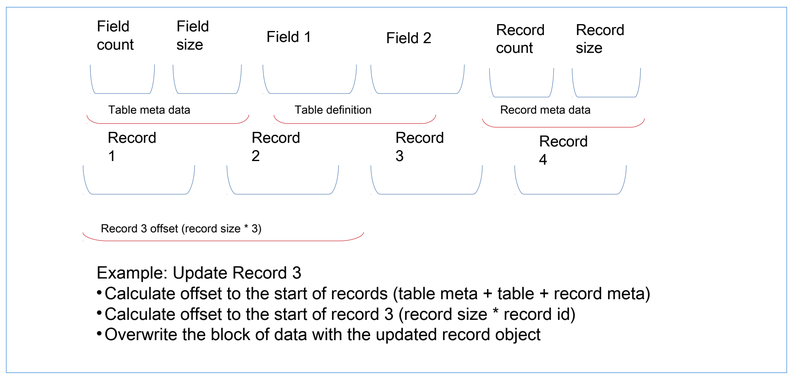
ABSTRACT

Powderbase is a lightweight, performance-oriented database engine and API for C++ programmers, designed to bridge the gap between database management systems and flat-file storage. The engine seeks to provide a platform that is easier to use and requires a lower footprint than full-fledged relational or document database engines such as MySQL or MongoDB. It also seeks to provide better performance than traditional flat-file storage formats such as XML, JSON, or CSV, especially when storing and retrieving numerical data. Powderbase uses an original binary file format designed to balance the concerns of processor and memory usage. It provides all of the critical operations of a database (create, read, update, and delete) and each algorithm is implemented with performance in mind. Powderbase is under continual development and available for reuse under the BSD free and open source license.

HOW IT WORKS

- Uses a flat-file to reduce memory usage
- Uses a binary file format
- Uses evenly-sized records for quick file access
- Designed with balanced memory and processing performance in mind
- Provides an easy-to-use C++ API

.



Example: Update Record 3
- Calculate offset to the start of records (table meta + table + record meta)
- Calculate offset to the start of record 3 (record size * record id)
- Overwrite the block of data with the updated record object

IMPROVEMENTS OVER TRADITIONAL DATABASE MANAGEMENT SYSTEMS

- Flat file simplifies:
  - User access
  - Backup
  - Import/Export
- Reduced feature set; easy to use
- Uses less space on disk
- Better performance in limited environments

IMPROVEMENTS OVER TRADITIONAL FLAT-FILE FORMATS

- No syntax parsing needed; less memory/processor usage
- No need for typecasting for native data types
- Uses less disk space for numerical data

PERFORMANCE METRICS

Test system information:
- CPU: Intel Xeon 2.40 GHz
- Memory: 1.01 GB RAM
- OS: Ubuntu 14.04 LTS 64 bit
- Build: development 804a780
- Records tested: 10,000

Results:
- Database create: 0 ms
- Record insert: 249 ms
- Record update: 224 ms
- Record search: 190 ms
- Record remove: 536 ms
- Database size on disk: 821 KB
- Library size on disk: 674 KB

DEVELOPMENT

- Language: C++
- Build tools: GNU Make
- Tested platforms:
  - Linux
  - Windows
  - Mac OSX
  - BSD
- License: BSD 2 Clause
- Website: jmcintyre.net/gitlist/powderbase/development