ORACLE®

BERKELEY DB

Factsheet

Shelby Hendrickx

&

Lucas van der Laan

Facts

Berkeley DB is designed for high-performance, transactional data management. The in-process architecture enables speed and reduces complexity.

- BerkeleyDB uses an Embedded Ordered Key-Value store
- BerkeleyDB was acquired by Oracle Corporation in 2006
- BerkeleyDB was owned by Sleepycat Software from 1996 - 2006
- Sleepycat Software was created specifically for BerkeleyDB
- The name BerkeleyDB is given to three products:
    1) Berkeley DB (written in C)
    2) Berkeley DB Java Edition (written in Java)
    3) Berkeley DB XML (written in C++)
- BerkeleyDB implements concurrency by using two-phase locking to permit multiple reader cursors or a single writer cursor to access the database.
- Supports operation on partial records
- Ability to store memory in-memory, on-disk or a combination
- Supports replicability for high system scalability and availability
- Developed in C with API bindings for Java, C++, C#, Python, Ruby, Perl, PHP, etc.
- Uses a B+Tree algorithm to store keys and values in leaves
- B+Tree is sorted to support efficient exact match lookups and range scans
- BerkeleyDB supports licensing under open-source licensing terms and proprietary for commercial use

- BerkeleyDB is fully ACID compliant, but can be fine-tuned to disable certain settings.
- Both keys and values have a maximum size up to 2^32 bytes, but more likely limited by available memory
- Default page size depends on the cluster size of your file system
- Minimum page size is 512 bytes
- Maximum page size is 65.536 bytes
- With minimum page size, the maximum size of the database is 2^41 bytes (2 terabytes)
- With maximum page size, the maximum size of the database is 2^48 bytes (256 terabytes)
- Maximum B+tree depth is 255
- Ranked 88th overall database
- Ranked 14th key-value store
- Selectable isolation levels and durability guarantees, configurable on a per-transaction basis
- Single master, multiple replica model
- Support for nested transactions
- BerkeleyDB saves the key and data as a byte array
- BerkeleyDB allows replication by enabling a group of systems to service the workload
- Bitcoin Core used BerkeleyDB until they switched to LevelDB
- BerkeleyDB uses API calls to handle data in the database, the main API calls are put(also for inserting), get and delete

https://db-engines.com/en/ranking

https://db-engines.com/en/ranking/key-value+store