Write Operation Performance

Indexes

After every insert, update, or delete operation, MongoDB must update *every* index associated with the collection in addition to the data itself. Therefore, every index on a collection adds some amount of overhead for the performance of write operations. [6]

In general, the performance gains that indexes provide for *read operations* are worth the insertion penalty. However, in order to optimize write performance when possible, be careful when creating new indexes and evaluate the existing indexes to ensure that your queries actually use these indexes.

For indexes and queries, see *Query Optimization* (page 65). For more information on indexes, see *Indexes* (page 481) and *Indexing Strategies* (page 546).

Document Growth and the MMAPv1 Storage Engine

Some update operations can increase the size of the document; for instance, if an update adds a new field to the document.

For the MMAPv1 storage engine, if an update operation causes a document to exceed the currently allocated *record size*, MongoDB relocates the document on disk with enough contiguous space to hold the document. Updates that require relocations take longer than updates that do not, particularly if the collection has indexes. If a collection has indexes, MongoDB must update all index entries. Thus, for a collection with many indexes, the move will impact the write throughput.

Changed in version 3.0.0: By default, MongoDB uses *Power of 2 Sized Allocations* (page 91) to add *padding automatically* (page 91) for the MMAPv1 storage engine. The *Power of 2 Sized Allocations* (page 91) ensures that MongoDB allocates document space in sizes that are powers of 2, which helps ensure that MongoDB can efficiently reuse free space created by document deletion or relocation as well as reduce the occurrences of reallocations in many cases.

Although *Power of 2 Sized Allocations* (page 91) minimizes the occurrence of re-allocation, it does not eliminate document re-allocation.

See *Storage* (page 89) for more information.

Storage Performance

Hardware The capability of the storage system creates some important physical limits for the performance of MongoDB's write operations. Many unique factors related to the storage system of the drive affect write performance, including random access patterns, disk caches, disk readahead and RAID configurations.

Solid state drives (SSDs) can outperform spinning hard disks (HDDs) by 100 times or more for random workloads.

See

Production Notes (page 197) for recommendations regarding additional hardware and configuration options.

Journaling MongoDB uses *write ahead logging* to an on-disk *journal* to guarantee *write operation* (page 73) durability and to provide crash resiliency. Before applying a change to the data files, MongoDB writes the change operation to the journal.

[6] For inserts and updates to un-indexed fields, the overhead for *sparse indexes* (page 505) is less than for non-sparse indexes. Also for non-sparse indexes, updates that do not change the record size have less indexing overhead.