TTL Collections TTL collections remove expired data from a collection, using a special index and a background thread that deletes expired documents every minute. These collections are useful as an alternative to *capped collections* in some cases, such as for data warehousing and caching cases, including: machine generated event data, logs, and session information that needs to persist in a database for only a limited period of time.

For more information, see the *Expire Data from Collections by Setting TTL* (page 211) tutorial.

Concurrency Improvements MongoDB 2.2 increases the server's capacity for concurrent operations with the following improvements:

1. DB Level Locking[827]
2. Improved Yielding on Page Faults[828]
3. Improved Page Fault Detection on Windows[829]

To reflect these changes, MongoDB now provides changed and improved reporting for concurrency and use. See *locks*, recordStats[830], `db.currentOp()`, `mongotop`, and `mongostat`.

Improved Data Center Awareness with Tag Aware Sharding MongoDB 2.2 adds additional support for geographic distribution or other custom partitioning for sharded collections in *clusters*. By using this "tag aware" sharding, you can automatically ensure that data in a sharded database system is always on specific shards. For example, with tag aware sharding, you can ensure that data is closest to the application servers that use that data most frequently.

Shard tagging controls data location, and is complementary but separate from replica set tagging, which controls *read preference* (page 588) and *write concern* (page 78). For example, shard tagging can pin all "USA" data to one or more logical shards, while replica set tagging can control which `mongod` instances (e.g. "`production`" or "`reporting`") the application uses to service requests.

See the documentation for the following helpers in the `mongo` shell that support tagged sharding configuration:

- `sh.addShardTag()`
- `sh.addTagRange()`
- `sh.removeShardTag()`

Also, see *Tag Aware Sharding* (page 682) and *Manage Shard Tags* (page 730).

Fully Supported Read Preference Semantics All MongoDB clients and drivers now support full *read preferences* (page 588), including consistent support for a full range of *read preference modes* (page 657) and *tag sets* (page 590). This support extends to the `mongos` and applies identically to single replica sets and to the replica sets for each shard in a *sharded cluster*.

Additional read preference support now exists in the `mongo` shell using the `readPref()` cursor method.

Compatibility Changes

Authentication Changes MongoDB 2.2 provides more reliable and robust support for authentication clients, including drivers and `mongos` instances.

If your cluster runs with authentication:

- For all drivers, use the latest release of your driver and check its release notes.

[827] https://jira.mongodb.org/browse/SERVER-4328
[828] https://jira.mongodb.org/browse/SERVER-3357
[829] https://jira.mongodb.org/browse/SERVER-4538
[830] http://docs.mongodb.org/v2.2/reference/server-status