RediSearch 1.4 release notes

Requirements
RediSearch v1.4.28 requires:

- Minimum Redis compatibility version (database): 4.0.0
- Minimum Redis Enterprise Software version (cluster): 5.0.0

v1.4.28 (May 2020)
This is a maintenance release for version 1.4.

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Details:

Bugfixes:

- #1218 Potential crash when running without concurrency and using the cursor API.

v1.4.27 (April 2020)

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Details:

Features:

- #1172 Added exists function that can be used on conditional updates (REPLACE PARTIAL) to check if a field exists in the document.

Minor Enhancements:

- #1172 Lazy evaluation of the right side of ‘or’/‘and’ clauses in IF condition.

Bugfixes:

- #1110 Rare GC failure when accessing uninitialized variable.
- #1131 Crash on highlighting a search query where the document no longer exists.

v1.4.26 (March 2020)

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Details:

- Fix rare FORK GC crash which caused by accessing uninitialized variable.
v1.4.25 (March 2020)

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Details:

- Features:
  - #1051 Added support for updating tag fields on document updates with NOINDEX fields.

- Bugfixes:
  - #1051 FORK GC was not updating the unique sum of the numeric index.

v1.4.24 (January 2020)

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Details:

- Bugfixes:
  - #1038 Memory leak on cursor.
  - #1049 Crash on conversion error when freeing other indexed fields.

v1.4.23

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Details:

- Bugfixes:
  - Memory leak when cursor timed out and cursor wasn't consumed.

v1.4.22

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Details:

- Bugfixes:
  - FILTER option was not working correctly with coordinator.
  - Memory leak when cursor was combined with sorted fields.

v1.4.21

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.
Details:

• Bugfixes:
  • #1031 Highlighting crashed when used with NOINDEX fields.

v1.4.20 (January 2020)

Update urgency: Medium This is a maintenance release for version 1.4.

This release improves overall stability and provides fixes for issues found after the previous release.

• Improvements
  • #1020 Performance improvement for reading fields that are not sortable.

• Bugfixes
  • #1022 Illegal memory access by queries during GC run.
  • #1022 Recreating the index with the same name (delete+create) removed the index from cursor list.
  • #1022 Memory leak when performing FT.AGGREGATE on non-existing index.
  • #1022 Potential data corruption during GC run.
  • #1025 Aliasing not working properly with FT.AGGREGATE.

v1.4.19 (December 2019)

Update urgency: Medium This is a maintenance release for version 1.4.

This release improves overall stability and provides fixes for issues found after the previous release.

• Bugfixes
  • #1000 - FT.DEL was not replicated to slave correctly
  • #1004 - Memory leak on TAG array on certain situations
  • #1006 - Unexposed error on conditional update IF that caused the error message to leak

v1.4.18 (November 2019)

Update urgency: Medium This is a maintenance release for version 1.4.

This release improves overall stability and provides fixes for issues found after the previous release.

• Bug fixes
  • #947 Fix short read on FORK GC pipe that could result in a crash and potential data corruption

v1.4.17 (October 2019)

Update urgency: Medium This is a maintenance release for version 1.4.

This release improves overall stability and provides fixes for issues found after the previous release.

• Features:
  • #779 Added to_number() and to_str() functions for ambiguity reasons

• Improvements
  • #891 All memory allocations will now use the redis memory allocator. This means that all the memory will be exposed correctly in the redis INFO MEMORY command.

v1.4.16 (September 2019)
This release improves overall stability and provides fixes for issues found after the previous release.

Main features:

- **#883** Forkgc optimizations - introduce new config parameter FORK_GC_CLEAN_THRESHOLD. RediSearch will only start to clean when the number of not cleaned documents is exceeding this threshold.

Main Fixes:

- **#848** RediSearch will not crash when sorting on fields that don't exist in all documents.
- **#884** Fix wrong results on intersect iterator.

**v1.4.15 (28 August 2019)**

- **#866** - Fix RDB corruption caused by deleting none-existing terms from the suggestion trie.
- When Redis exits, forked processes by FORK Garbage Collection will now be closed accordingly.
- For indices that are not temporary and interleaved: When an index is dropped, the indexer thread is now closed.

**v1.4.14 (20 August 2019)**

- **#851** In interleaved mode (non safemode), documents deleted by concurrent updates, will be ignored.

**v1.4.13 (8 August 2019)**

- **#798** fix issue where phonetic queries return wrong results
- **#820** fix crash on getting a none existing doc from a doc table
- Fix issue with invalid memory read when using tags with " " (space) separator

**v1.4.12 (5 August 2019)**

- **#851** In interleaved mode (non safemode), documents deleted by concurrent updates, will be ignored.
Main features:

- #741 Allow Chinese tokenizing to recognise -escape for punctuations

Main Fixes:

- #739 Fix crash on search error
- #346 Fix issue where fuzzy can not be used with numbers
- #769 Fix rare crash on rdb loading
- #749 On prefix searches, do not expand prefixes to terms which have no documents

v1.4.11 (June 2019)

Update urgency: Medium

This release only add support for aliasing to the previews 1.4.10 release.

- Added functionality
  - #731 Add index aliasing. This allows users to provide (or remove) 'links' to indexes. The commands are FT.ALIASADD, FT.ALIASDEL, and FT.ALIASUPDATE. See ftaliasadd for details.

v1.4.10 (28 May 2019)

Update urgency: Medium

This is a maintenance release for version 1.4.

This release improves overall stability and provides fixes for issues found after the previous release.

Main Fixes:

- Fix memory leak when combining SORT with APPLY on FT.AGGREGATE

v1.4.9 (18 May 2019)

Update urgency: Medium

This is a maintenance release for version 1.4.

This release improves overall stability and provides fixes for issues found after the previous release.

Main Fixes:

- Fork GC will now squash the empty blocks of the inverted indexes.
- Fix invalid memory access when using aggregate with GROUPBY.
- Fix issue where using limit with SORTBY might return duplicate results.

Known Issue:

- Memory leak when combining SORT and APPLY in FT.AGGREGATE. This issue is fixed on 1.4.10. It’s recommended to skip directly to 1.4.10.

v1.4.8 (29 April 2019)
Update urgency: Low
Technical release, no changes nor fixes.

v1.4.7 (29 April 2019)
Update urgency: Medium
This is a maintenance release for version 1.4.
This release improves overall stability and provides fixes for issues found after the previous release.
Main Fixes:
  - Fix issue where Dropping and recreating the same index while querying might cause crashes.

v1.4.6 (8 April 2019)
Update urgency: Medium
This is a maintenance release for version 1.4.
This release improves overall stability and provides fixes for issues found after the previous release.
Main Fixes:
  - Loading a potentially corrupted RDB files generated by versions lower than 1.4
  - Fix issue where REPLACE PARTIAL might not work properly (#621)

v1.4.5 (March 2019)
Update urgency: Low
This is a maintenance release for version 1.4.
This release improves overall stability and provides fixes for issues found after the previous release.
Main Fixes:
  - Fixed issue where FORK GC causing redis rdb fork to fail

v1.4.4 (21 February 2019)
Update urgency: Low
This is a maintenance release for version 1.4.
This release improves overall stability and provides fixes for issues found after the previous release.
Main Fixes:
  - Fixed memory leak on Fork GC
  - Fixed key close after releasing GIL on Fork GC (might cause crashed on rare situations)

v1.4.3 (4 February 2019)
This is a maintenance release for version 1.4. This release improves overall stability and provides fixes for issues found after the previous release.

Main Fixes:

1. Fixed memory leak on mempool destroy
2. Fixed process crash when running with fork GC (preview)
3. Fixed fork GC (preview) deadlock when using Tags
4. Fixed memory leaks on index creation and deletion

v1.4.2 (27 November 2018)

This is a maintenance release for version 1.4. This release improves overall stability and provides fixes to issues found.

v1.4.1 (12 November 2018)

This is a maintenance release for version 1.4, with the next version planned for release being 2.0. This release improves overall stability and focuses on performance improvements of the garbage collector. It also includes:

- New: Runtime Configuration
- Change: Unlimited Autocomplete Results

Garbage collector

RediSearch employs a garbage collector that removes deleted documents from the internal data structures.

In this release the garbage collection mechanism was improved in terms of efficiency, i.e. the amount of memory it reclaims, as well as in terms of performance (i.e. it is faster). The improved mechanism uses forked threads. Additional details can be found in this post: How We Increased Garbage Collection Performance with RediSearch 1.4.1.

The improved collection mechanism is currently *experimental* and is not enabled by default. Enabling the new garbage collection mechanism requires setting the GC_POLICY configuration option to FORK at load time, for example:

```
redis-server --loadmodule ./redisearch.so GC_POLICY FORK
```

Runtime configuration

RediSearch’s configuration is applied via arguments passed to the module at load time. This release introduces the new FT_CONFIG command that allows to retrieve the current configuration as well as change it during runtime.

Unlimited autocomplete results

This version removes the limit of 10 results from FT.SUGGEST - you can set the MAX num as high as needed.
v1.4.0 (August 2018)

**Update urgency:** Medium - mainly due to numerous fixes

This version improves overall stability and performance of RediSearch. It also delivers better support for use cases in which documents are continuously updated. New features:

- Conditional updates
- Schema modification
- Query spelling correction
- Phonetic matching
- Enhancement: More fuzziness in search

Continuous updates

Like most search engines, RediSearch was designed for maintaining append-mostly indices. To update an existing document, the document is actually replaced - that is deleted and added to the index.

Because RediSearch provides realtime indexing and searching, it is sometimes used to search near-realtime data (or the results of its processing). In such cases, the number of indexed documents stays mostly static, but their contents are continuously updated.

To support this use case, RediSearch has been reworked internally to use 64-bit internal document IDs so that an index can sustain high update throughputs without overflowing. Furthermore and as a result, significant effort has been put into improving memory management and garbage collection.

Conditional updates

The IF subcommand has been added to FT.ADD. When used with the existing REPLACE [PARTIAL] subcommand, the document will be updated only if the condition provided evaluates to a truth value, otherwise a special NOADD reply is returned.

Schema modification

The FT.ALTER command has been introduced, and provides the ability to add new fields to the definition of an existing index. The contents of such newly-added fields are indexed only for new or updated documents.

Query spelling correction

Query spelling correction, a.k.a “did you mean”, is now provided via the FT.SPELLCHECK command. It enables generating suggestions for search terms that could be misspelled. For more details see Query Spelling Correction.

Phonetic matching

Phonetic matching, a.k.a “Jon or John?”, is now supported via the PHONETIC text field attribute. The terms in such fields are also indexed by their phonetic equivalents, and search results include these by default. For more details see Phonetic Matching.

More fuzziness in search

The fuzzy match operator, ‘%’, can now be repeated up to three times to specify the Levenshtein distance. That means that the queries %hello%, %%hello%% and %%%hello%%%% will perform fuzzy matching on ‘hello’ for all terms with LD of 1, 2 and 3, respectively.

Updated: December 14, 2021