RedisTimeSeries 1.2 release notes

Requirements
RedisTimeSeries v1.2.7 requires:

- Minimum Redis compatibility version (database): 5.0.0

v1.2.7 (June 2020)
This is a maintenance release for version 1.2.

Details:

- Bugfixes:
  - #414 Crash when a query had an empty label (foo,)

v1.2.6 (May 2020)
This is a maintenance release for version 1.2.

Details:

- Minor enhancements:
  - #403 Support for multi-value filtering in TS.MGET and TS.MRANGE.
- Bugfixes:
  - #378 Using snprintf to ensure the same precision of floating-point value replies.
  - #374 TS.RANGE crashed when COUNT argument was missing.
  - #395 Check minimum compatible Redis version at module load time.

v1.2.5 (March 2020)

Headlines:

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

Details:

- Bugfixes:
  - #358 Wrong behaviour in TS.RANGE due to shifting left.
  - #353 Crash where the name of a time-series was already taken due to auto-compaction.

v1.2.3 (February 2020)

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

Details:

- Bugfixes:
  - #338 reverting #319. Aggregation should happen on deterministic time buckets.

**v1.2 GA (1.2.2 - January 2020)**

This is the general availability (GA) release of RedisTimeSeries 1.2 (1.2.2).

Headlines:

- Compression added which can reduce memory up to 98% and improve read performance up to 50%.
- Stable ingestion time independent of the number of the data points on a time-series.
- Reviewed API with performance improvements and removed ambiguity.
- Extended client support

(we will blog about this release soon including performance improvements results and the link here)

Full details:

- Added functionality
  - #261 Samples are compressed using Double Delta compression which results in cost savings and faster query times.
  - Based on the Gorilla paper.
    - In theory, this can save space up to 98%. (2 bits per sample in stead of 128).
    - In practice, a memory reduction of 5-8x is common but depends on the use case.
  - Initial benchmarks show 94% memory savings and performance improvements in reads up to XX%.
  - UNCOMPRESSED option in TS.CREATE.

- API changes / Enhancements
  - #241 Overwriting the last sample with the same timestamp is not allowed.
  - #242 revised TS.INCRBY/DECRBY
    - Returns a timestamp. The behaviour is now aligned with TS.ADD.
    - The RESET functionality was removed. RESET contradicted the rewriting of the last sample (#241). Alternatively, you can reconstruct similar behaviour by
      - TS.ADD ts * 1 + sum aggregation
      - TS.INCRBY ts 1 + range aggregation
  - #317 Aligning response on empty series of TS.GET with TS.RANGE.
  - #285 #318 Changed default behaviour of TS.MRANGE and TS.MGET to no longer returns the labels of each time-series in order reduce network traffic. Optional WITHLABELS argument added.
  - #319 TS.RANGE and TS.MRANGE aggregation starting from requested timestamp.

- Performance improvements
  - #237 Downsampling after time window is closed vs. downsampling with each sample.
  - #285 #318 Optional WITHLABELS argument added. This feature improves read performance drastically.

- Minor Enhancements
  - #230 TS.INFO now includes total samples, memory usage, first time stamp, ...
  - #230 MEMORY calculates series memory footprint.

- Bugfixes since 1.0.3
- #204 Module initialization params changed to 64 bits.
- #266 Memory leak in the aggregator context.
- #260 Better error messages.
- #259 #257 #219 Miscellaneous.
- #320 Delete the existing key prior to restoring it.
- #323 Empty first sample on aggregation.

⚠️ Note: The version inside Redis will be 10202 or 1.2.2 in semantic versioning.

Updated: December 14, 2021