NAME

rrdtune − Modify some basic properties of a Round Robin Database

SYNOPSIS

rrdtool tune *filename* [**−−heartbeat**|**−h** *ds-name*:*heartbeat*] [**−−minimum**|**−i** *ds-name*:*min*] [**−−maximum**|**−a** *ds-name*:*max*] [**−−data−source−type**|**−d** *ds-name*:*DST*] [**−−data−source−rename**|**−r** *old-name*:*new-name*] [**−−deltapos** *scale-value*] [**−−deltaneg** *scale-value*] [**−−failure−threshold** *failure-threshold*] [**−−window−length** *window-length*] [**−−alpha** *adaption-parameter*] [**−−beta** *adaption-parameter*] [**−−gamma** *adaption-parameter*] [**−−gamma−deviation** *adaption-parameter*] [**−−smoothing−window** *fraction-of-season*] [**−−smoothing−window−deviation** *fraction-of-season*] [**−−aberrant−reset** *ds-name*] [**−−daemon**|**−D** *address*] [**DEL:***ds-name*] [**DS:***ds-spec*] [**DELRRA:***index*] [**RRA:***rra-spec*] [**RRA#***index*:[+−=]<number]>

DESCRIPTION

The tune option allows you to alter some of the basic configuration values stored in the header area of a Round Robin Database (**RRD**).

One application of the **tune** function is to relax the validation rules on an **RRD**. This allows to fill a new **RRD** with data available in larger intervals than what you would normally want to permit. Be very careful with tune operations for COMPUTE data sources. Setting the *min*, *max*, and *heartbeat* for a COMPUTE data source without changing the data source type to a non-COMPUTE **DST** WILL corrupt the data source header in the **RRD**.

A second application of the **tune** function is to set or alter parameters used by the specialized function **RRAs** for aberrant behavior detection.

Still another application is to add or remove data sources (DS) or add / remove or alter some aspects of round-robin archives (RRA). These operations are not really done in-place, but rather generate a new RRD file internally and move it over the original file. Data is kept intact during theses operations. For even more in-depth modifications you may review the **−−source** and **−−template** options of the **create** function which allow you to combine multiple RRD files into a new one and which is even more clever in what data it is able to keep or ‘‘regenerate’’.

filename The name of the **RRD** you want to tune.

−−heartbeat|**−h** *ds-name*:*heartbeat*
: modify the *heartbeat* of a data source. By setting this to a high value the RRD will accept things like one value per day.

−−minimum|**−i** *ds-name*:*min*
: alter the minimum value acceptable as input from the data source. Setting *min* to ’U’ will disable this limit.

−−maximum|**−a** *ds-name*:*max*
: alter the maximum value acceptable as input from the data source. Setting *max* to ’U’ will disable this limit.

−−data−source−type|**−d** *ds-name*:*DST*
: alter the type **DST** of a data source.

−−data−source−rename|**−r** *old-name*:*new-name*
: rename a data source.

−−deltapos *scale-value*
: Alter the deviation scaling factor for the upper bound of the confidence band used internally to calculate violations for the FAILURES **RRA**. The default value is 2. Note that this parameter is not related to graphing confidence bounds which must be specified as a CDEF argument to generate a graph with confidence bounds. The graph scale factor need not to agree with the value used internally by the FAILURES **RRA**.

−−deltaneg *scale-value*
Alter the deviation scaling factor for the lower bound of the confidence band used internally to calculate violations for the FAILURES **RRA**. The default value is 2. As with **−−deltapos**, this argument is unrelated to the scale factor chosen when graphing confidence bounds.

−−failure−threshold *failure-threshold*
Alter the number of confidence bound violations that constitute a failure for purposes of the FAILURES **RRA**. This must be an integer less than or equal to the window length of the FAILURES **RRA**. This restriction is not verified by the tune option, so one can reset failure-threshold and window-length simultaneously. Setting this option will reset the count of violations to 0.

−−window−length *window-length*
Alter the number of time points in the temporal window for determining failures. This must be an integer greater than or equal to the window length of the FAILURES **RRA** and less than or equal to 28. Setting this option will reset the count of violations to 0.

−−alpha *adaption-parameter*
Alter the intercept adaptation parameter for the Holt-Winters forecasting algorithm. This parameter must be between 0 and 1.

−−beta *adaption-parameter*
Alter the slope adaptation parameter for the Holt-Winters forecasting algorithm. This parameter must be between 0 and 1.

−−gamma *adaption-parameter*
Alter the seasonal coefficient adaptation parameter for the SEASONAL **RRA**. This parameter must be between 0 and 1.

−−gamma−deviation *adaption-parameter*
Alter the seasonal deviation adaptation parameter for the DEVSEASONAL **RRA**. This parameter must be between 0 and 1.

−−smoothing−window *fraction-of-season*
Alter the size of the smoothing window for the SEASONAL **RRA**. This must be between 0 and 1.

−−smoothing−window−deviation *fraction-of-season*
Alter the size of the smoothing window for the DEVSEASONAL **RRA**. This must be between 0 and 1.

−−aberrant−reset *ds-name*
This option causes the aberrant behavior detection algorithm to reset for the specified data source; that is, forget all it is has learnt so far. Specifically, for the HWPREDICT or MHWPREDICT **RRA**, it sets the intercept and slope coefficients to unknown. For the SEASONAL **RRA**, it sets all seasonal coefficients to unknown. For the DEVSEASONAL **RRA**, it sets all seasonal deviation coefficients to unknown. For the FAILURES **RRA**, it erases the violation history. Note that reset does not erase past predictions (the values of the HWPREDICT or MHWPREDICT **RRA**), predicted deviations (the values of the DEVPREDICT **RRA**), or failure history (the values of the FAILURES **RRA**). This option will function even if not all the listed **RRAs** are present.

Due to the implementation of this option, there is an indirect impact on other data sources in the RRD. A smoothing algorithm is applied to SEASONAL and DEVSEASONAL values on a periodic basis. During bootstrap initialization this smoothing is deferred. For efficiency, the implementation of smoothing is not data source specific. This means that utilizing reset for one data source will delay running the smoothing algorithm for all data sources in the file. This is unlikely to have serious consequences, unless the data being collected for the non-reset data sources is unusually volatile during the reinitialization period of the reset data source.

Use of this tuning option is advised when the behavior of the data source time series changes in a drastic and permanent manner.

−−daemon|**−D** *address*

NOTE: Because the **−d** (small letter 'd') option was already taken, this function (unlike most other) uses the capital letter 'D' for the one-letter option to name the cache daemon.

If given, **RRDTool** will try to connect to the caching daemon rrdcached at *address* and will fail if the connection cannot be established. If the connection is successfully established the data for the *filename* will be flushed before performing the copy/modify operation. Afterwards the *filename* will be forgotten by the cache daemon, so that the next access using the caching daemon will read the proper structure.

This sequence of operations is designed to achieve a consistent overall result with respect to RRD internal file consistency when using one of the **DS** or **RRA** changing operations (that is: the resulting file should always be a valid RRD file, regardless of concurrent updates through the caching daemon). Regarding data consistency such guarantees are not made: Without external synchronisation concurrent updates may be lost.

For a list of accepted formats, see the **−l** option in the rrdcached manual.

DEL:*ds-name*

Every data source named with a DEL specification will be removed. The resulting RRD will miss both the definition and the data for that data source. Multiple DEL specifications are permitted.

DS:*ds-spec*

For every such data source definition (for the exact syntax see the **create** command), a new data source will be added to the RRD. Multiple DS specifications are permitted.

DELRRA:*index*

Removes the RRA with index *index*. The index is zero-based, that is the very first RRA has index 0.

RRA:*rra-spec*

For every such archive definition (for the exact syntax see the **create** command), a new RRA will be added to the output RRD. Multiple RRA specifications are permitted.

RRA#*index*:[+−=]<number>

Adds/removes or sets the given number of rows for the RRA with index <index>. The index is zero-based, that is the very first RRA has index 0.

EXAMPLE 1

```
rrdtool tune data.rrd −h in:100000 −h out:100000 −h through:100000
```

Set the minimum required heartbeat for data sources 'in', 'out' and 'through' to 10'000 seconds which is a little over one day in data.rrd. This would allow to feed old data from MRTG−2.0 right into RRDtool without generating *UNKNOWN* entries.

EXAMPLE 2

```
rrdtool tune monitor.rrd −−window−length 5 −−failure−threshold 3
```

If the FAILURES **RRA** is implicitly created, the default window-length is 9 and the default failure-threshold is 7. This command now defines a failure as 3 or more violations in a temporal window of 5 time points.

EXAMPLE 3

```
rrdtool tune some.rrd DEL:a RRA#0:+10
```

Delete the data source named **a** and extend the very first archive by 10 rows. This will in fact replace the input RRD with a new RRD keeping all existing data. For most practical use cases this is identical to a real in-place modification.

AUTHORS

Tobias Oetiker <tobi@oetiker.ch>, Peter Stamfest <peter@stamfest.at>