RRDtool Tutorial

Tobias Oetiker <tobi@oetiker.ch>

Wednesday, March 13th, 2013

The objective of this tutorial session is to help you get comfortable using RRDtool to write your own monitoring applications.

In order to do some of the exercises you will need information which you do not have yet. There are three main sources from where you can acquire this information.

- The rrdtool web site.
- Or read the rrdtool unix manual pages.
- Ask me!

As you can see, there are no solutions printed on this sheet. I would propose though, that you your sollutions to me. I will then put them together mail them back to everyone who sent in their work.

1 Creating an RRD

1.1 Data Source Types

RRDtool stores and graphs numerical data. In order to do so properly you must provide some information about the nature of the data. RRDtool knows several different types of Data Sources (DS).

EXERCISE

Read the rrdcreate manual page to identify the data source types and write down some examples for each type.

1.2 Data Consolidation Methods

Once you have passed data on to RRDtool it gets stored in a data storage array (Round Robin Archive). Each RRD can contain several RRAs, working at different resolutions and using different data consolidation methods.

EXERCISE

Find the different Data Consolidation methods currently supported in RRDtool. This information can be found in the RRD create manual page.

1.3 Data Validation

Storing invalid data can often be more of a problem than not storing anything at all. In order to help you ensure that only valid data gets into your Round Robin Database, RRDtool allows you to describe some properties of the data you intend to store. This allows RRDtool to throw out invalid input before it even enters the database.

EXERCISE

Identify the parameters to setup these safeguards. This information can be found in the rrdcreate manual page.

1.4 Database Setup

Having solved the exercises up this point you are now ready to setup a Round Robin Database. Use the command line tool
`rrdtool` to work your magic.

EXERCISE

Create an RRD which accepts input from two COUNTER data sources. The data sources provide new data every 300 seconds on average. Allow for a maximal update interval of 600 seconds. The input from both data-sources will always be between zero and 35 million.

The RRD should store the data for 24 hours at 5 minute resolution and for a month at one hour resolution. For the one hour resolution you want to keep both the average and the 5 minute maximum data.

Use `rrdtool info database.rrd` to see the structure of the rrd file you just created.

1.5 Coupling of Data Values

All values stored in a single RRD must be updated synchronously. Also, it is not trivial to add new data-sources to an existing RRD or remove old ones. In most cases it is sensible to create a new RRD for each data source unless you know that they are tightly coupled.

EXERCISE

Think of some data sources which are tightly coupled in the sense that they should be stored into the same RRD and of some which should NOT be stored in the same RRD.

2 RRD Update

2.1 The RRD Perl Interface

The recommended way to interact with an RRD is to use some scripting language and an rrdtool language bindings. There are language bindings for perl, python, ruby, tcl, lua and many other languages available.

EXERCISE

Convert the command line for creating the RRD from the last exercise in the previous section into a perl script.

2.2 The Error Messages in Perl

The RRDs commands do not complain when you call them with invalid arguments. Normally they just get ignored. To catch errors you must actively look for them. This is done with the `RRDs::error` function. The behaviour of other bindings may differ.

EXERCISE

Add error checking to your script and test it by providing the create command with invalid parameters.

2.3 Feeding Data into an RRD

In `/proc/net/snmp` you can find some counters regarding the traffic of your workstation. This pseudo file contains new data each time you read it.

EXERCISE

Use the data from this file to populate the RRD created in the previous exercise. Don't forget to add error checking to the update routine. Make sure you 'fake' the update time by stepping 5 minutes ahead everytime you update.

Use `rrdtool dump file.rrd` what data is stored in the rrd file.

2.4 Data Re-Sampling

RRDtool re-samples any data you feed it to the base interval you set for the database. This means, unless your data arrives exactly at the end of each interval, your original data will not be preserved. People NOT understanding how this works is one of the main toppics on the rrdtool mailinglist.

EXERCISE

Get a grip on data resampling by creating a new rrd file and feeding it some test data to see how it handles the different cases.

3 Creating graphs

Harvesting data and storing in RRDs alone won't help you get a promotion. What really interests people is getting graphs produced from this data.

3.1 Line Graphs

EXERCISE

Use the graph function to create a graph representing the data stored in your RRD. To start, use only `DEF:...` and `LINE1:...` and `--end now+2day` parameters and have RRDtool auto-configure the rest.

3.2 GPRINT Exercise

A RRD graph can also show numerical data.

EXERCISE

Use the `GPRINT` argument to show the maximum 5 minute values of both data sources below the graph.

3.3 A Stack Graph

Lets assume the data in the RRD represents traffic seen on two different web servers which share the load of a busy web site.

EXERCISE

Use the `AREA:` and `:STACK` function to place the data from the first and second data-source on top of each other. This will show the traffic produced by each server on its own as well as the total traffic occurring on your web site.

3.4 Using RPN Math

The network traffic in /proc/net/snmp is in octets passed over the interface. Most people though will expect to see traffic data reported in bits instead of octets.

EXERCISE

Use the `CDEF:` function to multiply your data by 8 before graphing it. This has been discussed in the RRDtool presentation.

3.5 Smoothing

Over the years, the number of functions supported by rrdtools RPN engine has grown considerably. Trending and smoothing functions have been incorporated.

EXERCISE

For data that fluctuates widely, it may help to plot a moving average instead as this will make slow changes more obvious. Sample your the network traffic at 1 second interval and create a graph overlaying the original data with a 1 minute moving average.

3.6 Data Selection

You used the `DEF:` function to pull in data from an rrd file for graphing. The command supports several named parameters to better control the data that is read and to massage it to fit your needs.

EXERCISE

Draw a chart where you overlay the current data with data from a previous interval for comparison.

and while you are at it.

EXERCISE

Instruct DEF to resample your input data at a lower resolution than the original data, to cause a staircase effect.

4 Advanced Exercises

4.1 Alter RRD Parameters

Some parameters of an existing RRD can be changed quite easily using the update command.

EXERCISE

Use the update command to change the name of the two data sources in your RRD. Use dump to very that the changes were successful.

4.2 Web Charting

With browsers improving their abilities, rrdtools native charting abilities are not so important anymore. One might wish to draw his charts in the browser, using the D3.js library for example.

EXERCISE

Use the `xport` function to extract chart data from an rrd file and create a chart using the d3 library.

4.3 Examine an RRD

If you are writing a frontend to RRDtool it might be necessary to find out about the configuration of an existing rrd file. The rrdinfo function helps you with this.

EXERCISE

Use `info command` to fetch config data from an existing rrd and convert it into command line which you could supply to rrdtool create.