RRDtool Tutorial

Tobias Oetiker <tobi@caida.org>

September, 29th 1999

The goal of this 3 hour tutorial session is to help you get comfortable using RRDtool so you will be able 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:

- Read the RRDtool Manual on the web site or the unix manual pages.
  `https://www.caida.org/Tools/RRDtool`
- In `cider:/home/tobi/rrdtool` there is a complete rrdtool installation. With all the documentation in ASCII and some examples.
- Ask one of the tutors in the room.

Please be aware that there is no magic to learning how to use RRDtool, only now you have the chance to do it without getting distracted. Keep in mind that there are probably too many exercises to solve in the time available . . .

1 Creating an RRD

1.1 Data Source Types

RRDtool stores and graphs data you feed it. In order to do so properly you must provide some information about the nature of the data. RRDtool currently knows 3 types of Data Sources (DS).

EXERCISE

Read the RRDcreate manual page to identify the 3 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 the 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 RRD create manual page.

1.4 Data Base Setup

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

EXERCISE

> Create an RRD which accepts input from two GAUGE 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.

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 fastest way to interact with an RRD is to write a perl script which uses the RRD shared module interface. You can find documentation on this in the RRDs manual page. On `cider.caida.org` the perl module is stored in `/home/tobi/rrdtool/lib/perl`. In order to access it you must add

```
use lib '/home/tobi/rrdtool/lib/perl';
use RRDs;
```

to the very beginning of your script.

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.

EXERCISE

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

2.3 Feeding Data Into an RRD

In `/home/tobi/rrdtool/data.txt` you can find a text file with sample data[1]. The format is `time:rate-1:rate-2`. Note that the data in this file starts about two months back. So you want to use `--start 19990801` when creating the RRD for this data. To look at your graph, put it into `public_html` and call `http://cider.caida.org/~guest?/????.gif`.

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. When you get things to work, try recreating the database and add data-broken.txt instead of the original file.

3 Creating graphs

3.1 Line 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.

EXERCISE

> Use the RRDs::graph function to create a graph representing the data stored in your RRD. To start, use only `DEF:...` and `LINE1:...` and `--end 19990930_15:00` 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.

[1]This is real traffic data from the transatlantic link of the Swiss University Network to the US.

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

When acquiring traffic data from a router you are likely to get the number of 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.

4 Graphing On The Fly

4.1 Simple RRDcgi Use

The most time consuming operation in RRDtool is creating the graphs. So when setting up a complex monitor you want to avoid generating graphs when ever you can. One approach to this problem is to generate graphs on-the-fly. The `rrdcgi` tool helps you do this with a very simple script interpreter. Check the corresponding manual page.

Find `rrdcgi` in `/home/tobi/rrdtool/bin`...

EXERCISE

> Write a `rrdcgi` input file which generates the stacked graph you did in the last example. Check the effect of the `--lazy` option. You can test your input file by putting it into `public_html/xyz.cgi` and then calling
> `http://cider.caida.org/~guest??/xyz.cgi`

4.2 Interactive RRDcgi

The second example on the `rrdcgi` manual page shows how to access `FORM` parameters.

EXERCISE

> Use this to give the user of your page the option to select whether to generate a graph for the last day (`--start end-24h`) or for the last week (`--start end-7d`).

5 Mixed Features

5.1 Dump and Restore

An RRD is stored in native binary format. When you want to transport an RRD from one hardware/OS combination to another one you must use RRDtool dump and restore.

EXERCISE

> Use `dump` to convert your RRD into XML format and have a look at it. You should be able recognize several of the elements of the RRD. Dumping an RRD can also be useful when you are debugging.

5.2 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.

5.3 95th Percentile Calculator

A lot of internet providers are billing their customer using a method called 95th percentile. It works like this: They measure your network traffic in 5 minute intervals and keeps the data for a whole month. At the end of the month the data gets sorted and the top 5% of the 5 minute averages get thrown away. The remaining data is used to calculate the amount of traffic for the month or which period you are billing by.

The advantage of this method is, that if you get high traffic for a day once a month due to a network attack or some other incident, your bill will not sky-rocket.

EXERCISE

> Use `RRDs::fetch` and sort to build such a 95th percentile calculator. An example for the use of `RRDs::fetch` is in the manual page.