Cricket … the next MRTG
John J. Renwick
30 March, 2001

MRTG is the Multi Router Traffic Grapher tool from Tobias Oetiker which has been available since 1995 http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/ . Cricket is evolving to be the next generation of Graphing tools. http://cricket.sourceforge.net/

Cricket was created due to a need to forecast network growth and plan ahead for expansion. Cricket, like its predecessor MRTG, provides for instantaneous management features such as, is the link up, what is the status of the routers, and what is the traffic load? Also, like MRTG, Cricket features graphical displays of network traffic trends over configurable time periods with comparison to other dates or times. It provides information for long-term analyses of traffic trends on a specific link or an entire network.

The program, while looking much like MRTG, has been enhanced in many ways. It is more configurable, uses a hierarchical configuration tree, gathers data from scripts, SNMP, Perl procedures, etc. It is made up of two modules, a collector and a grapher, the collector does the heart of the work, gathering the data while running out of cron and storing that data in Round Robin Database (RRD) files, which is then fed into the graphing program. The graphing program for the RDD generates the display of that data, which results in graphs similar to those displayed by MRTG.

Cricket has taken its place as the evolutionary successor to MRTG, because in addition to graphing data, it also gathers application and host-based statistics and is able to monitor such events as PVC states, cable modem signal strength, and router CPU load. While it displays much of the same information, the hierarchical configuration tree that it uses and improved code allow it to perform many more tasks much faster.

Quote from NETWORK WORLD FUSION FOCUS: 03/07/01

"It is a redesign of MRTG, which was mentioned last week. One reader mentioned that Cricket is much more scaleable than MRTG. According to one reader, "we have completely replaced our Spectrum reports with those from Cricket.""

The program itself is a high performance, extremely flexible system which monitors trends in time oriented data. The design of Cricket was expressly done to help overburdened network managers visualize and understand and react to the traffic on their networks.

As stated above, Cricket is made up of two modules, a collector and a grapher. The collector runs from cron at 5 minutes intervals, or it can be configured to run more often, or less often depending on the needs of the network manager, however you don't want to set the interval too low as you need to be able to receive and store the data requested in less than the interval set. It stores the data into a database managed by a RRD Tool, more on this tool later. When you want to review the data that has collected, you can simply use a web-browser to view graphs of the data.

Cricket's operation is governed by a set of config files called a config tree. The config tree defines everything Cricket needs to know about the types of data to be collected, where to get the data, and how to get it. The config tree is compact and easy to manage and was designed to minimize redundant information. Think of a config tree as a set of configuration files organized into a tree architecture, the hierarchical structure allows us to be able to use inheritance to reduce repeated information in this configuration. To implement the tree in an easy to understand directory structure we don't allow complicated or tricky concepts such as multiple inheritance.

The rules that are present at the end of the branch is the compilation of all the rules sets in all the branches on a path leading from the trunk of the config tree to the end of the branch. (See Figure 1) Lower rules always override higher rules, Rules that all of the system will share are located in the trunk of the config tree. In this example, the length of the polling interval is set there. At the next level, we set attributes that will be restricted to the current branch. At this level, typically you will find the target type. Finally at the highest level we set things that will vary on a machine basis. For this example, we set the interface or machine name that we are trying to measure here. By using the rules of inheritance, you avoid repeating the rules at the top of the config tree in the trunk of the config tree. This three level config tree in the example is a very simple layout, and in real network operations you will want to develop a more complex tree structure, for instance separating different sites, or operations to be in different branches. There is a sample config tree that comes with Cricket that is just a starting point in time. There are no built-in limits on the shape or the architecture of the config tree, but too complex of a tree can be a configuration management nightmare.

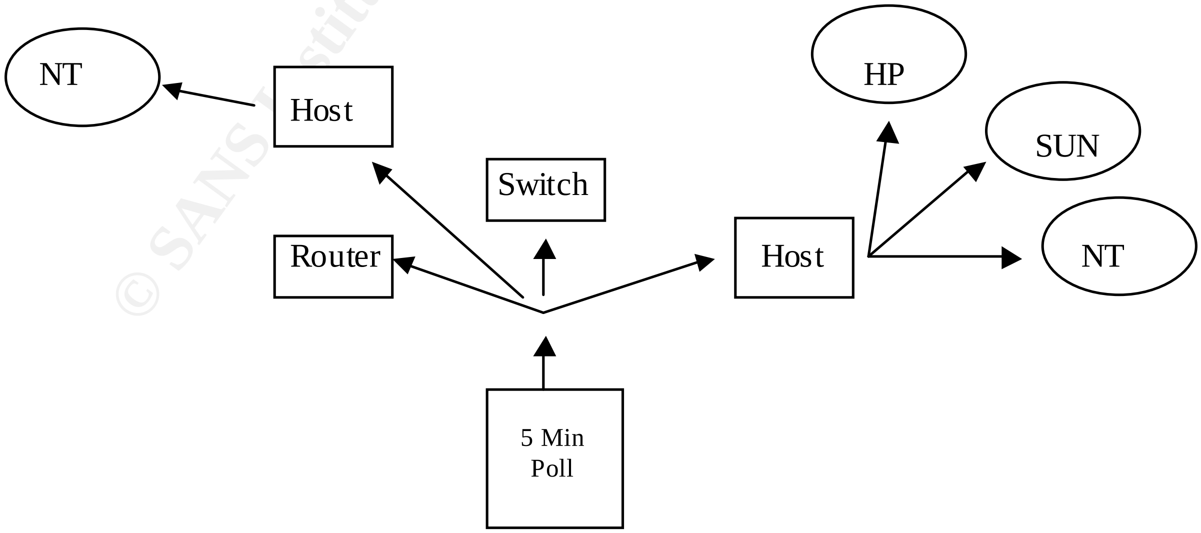

Figure 1
Config Tree

Cricket is written entirely in Perl and was developed on Solaris machines running under Apache. It is known to work on Linux, HP-UX, variants of BSD, and some other operating systems.

RRD Tool is written in C for speed, and comes in both a Perl module and a command line version, which means it can be used interactively or across a pipe from scripts it is what does all the database work for Cricket. RRD achieves its high performance by using binary files to minimize I/O during the common update operation. RRD Tool stores and normalizes the data after the collector retrieves it, and outputs the data into the graphical format when the grapher asks for it. The way that RDDtool accesses the data is in a circular data buffering method that makes the I/O two to three times faster than in MRTG. RDDtool can also keep an arbitrary number of data arrays; each fed at a different rate. For example, you can take 1680 samples taken every 5 minutes for an entire weeks worth of data alongside 720 samples taken every 30 minutes for an entire months worth of data.

If you are familiar with MRTG, you can see the parallel between of RRDtool and of MRTGs graphing and logging features. It is significantly faster and more flexible than MRTG, and that is where it's strength lays. RRD is more flexible than MRTG in at least two aspects. Number one, and probably it's strongest attribute is that it can take data from an arbitrary number of data sources, MRTG was limited to only two datasources, one for input bandwidth and one for output bandwidth. RRD is a system to store and display time-series data which would be, network bandwidth, machine-room temperature, server load average, router loads etc. It stores the data in a very compact non-expanding method, utilizing a fixed size database which grows linearly with the number of monitored devices, not time and it displays the detailed graphs by processing the data to enforce a certain data density. It is used with either shell or Perl scripts or by various frontends that poll the different network devices using a friendly Graphical User Interface (GUI)

RRDtool is available from http://eestaff.ethz.ch/~oetiker/webtools/rrdtool/

It is available for various UNIX and LINX platforms and Windows NT and 2000.

RRDtool does not fully replace MRTG, and there is a very good reason for that as it allows other frontends to be used to interface to it, which allows for a much more robust tool that was never envisioned by MRTG. It provides an excellent basis for building tools, which work much better than MRTG. It is my understanding that MRTG-3 will also utilize the RRDtool, therefore you can get a head start by utilizing the existing frontends written to take advantage of RRDtool, or you can program your own application specific frontends.

Frontends:

The following frontends are available from Tobi Oetiker's webpage:
http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/frontends/

Big Sister

The Big Sister system and network monitor concentrates on detecting failing services and systems, displaying status overviews and alarming. As a means of providing diagnostic information to the system administrator Big Sister also collects trend data and uses RRDtool as a powerful and easy-to-use utility for storing and visualizing collected data.

Big Sister's features include:

- detection of service or system failure via an agent alarming
- display of consolidated and/or detailed status overviews
- collection and visualization of system performance data via RRDTool
- interconnection of different Big Sister sites

http://bigsister.graeff.com/

Ntop

Ntop is a web-based application for both Unix and Win32 that reports information about network traffic, similar to what the popular Unix top command does. Main ntop features include:

- Sort network traffic according to many (IP and non-IP) protocols
- Display traffic statistics
- Show IP traffic distribution among the various protocols
- Analyze IP traffic and sort it according to the source/destination
- Display IP Traffic Subnet matrix (who's talking to who?)
- Identify network security violations
- Ability to browse data from a WAP phone
- Ability to access data from remote using popular languages such as Perl and PHP

By means of the ntop Perl API, it is possible to extract live traffic data from ntop and easily store them into the RRDtool. Thanks to the RRDtool, network administrators can perform advanced and long-term traffic trend analysis statistics. An early prototype of ntop+RRDTool can be found at http://www.ntop.org/RRD/.

http://www.ntop.org/

Remstats

Remstats is a statistics gathering and graphing system. The idea behind remstat is to run little data collection agents on remote machines and poll them from a central server where the data is kept. The data presentation happens on the fly with automatically generated rrdcgi scripts which create the necessary graphs on demand. Graphs can also be created statically if running rrdcgi is not possible.

http://silverlock.dgim.crc.ca/remstats/release/

RRGrapher

RRGrapher is a front-end for RRDtool that allows you to interactively build graphs
of your own design. It allows you the freedom to use any combination of other RRDtool frond-ends such as MRTG, Cricket, and 14all to create and maintain the .rrd files, but generate graphs containing from any of those sources in the same graphs.

Unlike other RRDtool front-ends, RRGrapher doesn't produce ".rrd" files, it is used in conjunction with other popular front-ends, or custom front-ends of your own devising.

RRGrapher's features include:

- A simple installation - RRGrapher is just a single CGI script.
- The ability to create graphs from data sources stored in many ".rrd" files, even those created by any number of different RRDTOOL front-ends.
- "Bookmark-able" URLs for your web browser - RRGrapher generates URLs which contain all the information needed to reproduce your custom graph, allowing you to produce them on-demand in the future.
- Display of an rrdtool command-line that you could use to generate the graph - RRGrapher could help you to learn how to use RRDTOOL, or will at least provide a command line which you can cut-and-paste to generate your graph in "batch" mode, for instance to display on a web page.

http://net.doit.wisc.edu/~plonka/RRGrapher/

NRG

NRG (aka Network Resource Grapher) is the result of work at WiscNet to design and implement a highly automated and scalable MRTG-like system.

http://nrg.hep.wisc.edu/

FwGold

A tool to graphically display Checkpoint Firewall-1® logging statistical data

Snapshots of graphics produced using FwGold can be seen here:
http://rotoni.com/FwGold/example.htm

FwGold's features include:

- Firewall access statistics logging and graphics
- Keep graphics of last day, week, month and year statistical data
- RRDtool based (stores data into non growing round robin databases)
- Completely Perl written (uses RRDs perl module)
- Client/Server structure (the server runs on the fw-1 management module, the client can be anywhere perhaps where a Web server runs, the communication TCP port is configurable)
- Both client and server sides don't need to run as 'root'
- Fully configurable (users can freely define criteria's to match when filtering the FW log and which db and graphics to generate, which graphic's format, colors and labels, which file names etc.)
- Automatic generation of DBs, graphics and html files
- Automatic prevention of spikes in case of counter resets due to server restarts
- Automatic detection of missed data
- Automatic detection of configuration errors
- Default configuration file to generate common firewall statistics (total connections, connections per protocol, total accepted connections per protocol, etc.)
- Detailed installation and configuration instructions

http://rotoni.com/FwGold

Hoth

Hoth is an IP accounting tool exclusively for Linux 2.2, as it relies on the IPChains firewalling code (it will be ported to 2.4 once the kernel is stable). Hoth has the ability to account per source/destination IP, source/destination port, protocol, and interface just like the real ipchains. Furthermore you can stack every accounted data.

Hoth consists of an .ini style like config file, scripts for creating the firewall rules out of the config file and a CGI script to view the accounted data as graphics, which are created realtime.

http://joker.rhwd.de/software/hoth/

NMIS

NMIS is a Network Management System which performs multiple functions from the OSI Network Management Functional Areas, mainly Fault and Performance Management. The idea being to make use of all that polling for performance stats and get fault management for free.

Sample pages are available on the NMIS home page.

NMIS features include:

- The entire network is summarized into a single metric, which indicates reachability, availability and health of all network devices being managed by NMIS.
- Summary page for entire network with reachability, availability, health, response time metrics
- Summary pages of devices including device information, health graph, and interface summary
- Color coded events, status for at a glance interpretation.
- Graphing of Interface, CPU, Memory stats for Cisco Routers and Switches.
- Graphs can be drilled into.
- Graphs produced on the fly.
- Graphs can have varying lengths from 2hours to 1 year
- Interface statistics are returned in Utilization not just bits per second
- Response time graphed and metrics for health and availability generated from statistics collected
- Integrated Fault and Performance Management
- Threshold engine which send alerts on certain thresholds.
- Alert events are issued for device down or interface down
- Event levels are set according to how important the device is
- Events are "State full" including thresholds, meaning that an event is only issued once.
- Notification engine can be expanded to handle any "command line" notification method, including email, paging, signs, speakers, etc
- A list of current events is available and there is an escalation level and time the event has been active.
- Events are logged
- Outage time calculated for each down event
- Planned outages can be put in so alerts are not issued
- Reports for utilization, outages, etc
- Find functions which search based on strings in interface types and descriptions.
- Dynamic handling of if Index changes and difficult SNMP interface handling
- Integrated logging facility to view events and syslog messages.

http://www.sins.com.au/nmis/

Bronc

Bronc aims to be the fastest front-end available for RRDtool. It is written in Perl, using
Mason as a template/component engine and mod_perl to speed things up. Bronc features...

- a SNMP collector, which queries SNMP-capable devices and stores measurements in RRDtool databases
- on-the-fly graphing using BRONC::Grapher, which runs under mod_perl

- a extensible, Perl-based configuration file, which allows for quick configuration of complex graphs

http://bronc.blueaspen.com

Orca
Orca is a tool useful for plotting arbitrary data from text files onto a directory on a Web server.

http://www.gps.caltech.edu/~blair/orca/

SLAMon
SLAMon is a front-end for RRDTool that allows you to calculate working hour and 24-by-7 availability for anything that you're currently monitoring with RRDTool. The graphs and reports can cover any time period you're interested in: daily, monthly, yearly, etc.

Currently SLAMon doesn't directly do any monitoring itself. For that you need another front-end, to create and update RRD files for each resource. The measurements contained in these files are used as input by SLAMon to calculate availability and update the corresponding availability RRDs.

http://slamon.sourceforge.net

References

Allen, Jeff R. "About Cricket" http://cricket.sourceforge.net/

Allen, Jeff R. "Driving by the Rear-View Mirror: Managing a Network with Cricket" First Conference on Network Administration Santa Clara, California, April 7-10, 1999 http://www.usenix.org/publications/library/proceedings/neta99/full_papers/allen/allen_html/index.html

Kramp, Bill "Beyond MRTG" SysAdmin, March 2000, Vol. 9 Issue 3

Oetiker, Tobias "mrtg - What is MRTG" http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html

Rasmussen, Audrey "Your favorite tools, Part 2" Network World Network Systems Management Newsletter, 03/07/01 http://www.nwfusion.com/newsletters/nsm/2001/00477466.html

Wilson, Brian "Apache Server Page Load Monitoring With Cricket" Web Review, 02/25/00 http://www.webreview.com/2000/02_25/designers/02_25_00_2.shtml