Percival and Lancelot Installation and Operation Manual

Sasha Mikheev <sasha@avalon-net.co.il>,
Samuel Levin <samuel@avalon-net.co.il>,
Artemy Kovalev <artemy@avalon-net.co.il>
Eli Yukelzon <ilja@avalon-net.co.il>

2003/03/10

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Percival is a Network Monitoring and Capacity Planning front-end to the excellent RRDtool software. It is based on our experience of providing customized network monitoring solutions to the ISPs, banks and large enterprises in Israel.

We found that existing commercial tools are too expensive and ill designed for this task while existing RRD frontends lack essential features such as user profiles, simple configuration model, customizable GUI, performance, reports etc.

We started with stock Cricket and with time we essentially rewrote it to address its shortcomings.

We also had to add number of bug fixes and improvements to the RRDtool. Percival is a subset of our Lancelot Monitoring Framework that can be released under the GPL.

Many chapters of this guide are relevant for both Percival and Lancelot. There are several chapters such as **??**, **??** and others that are valid only for Lancelot. Since Percival is a subset of the Lancelot, we decided that it is better to have one manual for both systems.

Percival Features:

- WEB User Interface
- Themable user interface
- Support of MIB2
- Cisco, Linux and Windows
- Configuration is stored in hierarchical database
- Database supports on the fly editing, links and multiple users
- Reports: top utilized interfaces, errors, discards etc
- Totals: average, sum, many small graphs on one page
- Drill-down on all graphs
- Moving average smoothing
- Percentile 95%

- User profiles. Each user can see only his part of the configuration tree
- Each device polled only once even if it appears in multiple profiles
- Consistent CLI tools for the system configuration and maintenance
- Modified RRDtool is used for the data storage
- Designed to hold around 500 interfaces with 120 seconds polling frequency
- OS: Linux, Solaris(must built from source)

Percival is written in Perl so it should be pretty portable. The officially supported platforms are Linux and Solaris.

Lancelot is a superset of the Percival, thus it has all Percival features plus some more.

Lancelot Features:

- Major performance improvements. Lancelot is two orders of magnitude faster then Percival. It is possible to collect performance data about 30000 interfaces on the single server
- asynchronous data collection
- Many more supported network devices
- Alerts and Notifications
- Dashboards
- SLA on infrastructure
- CISCO SAA support. One way and two way round trip time, jitter. HTTP response time
- CISCO Quality of Service monitoring
- NetFlow Analysis
- Service monitoring module (POP3, HTTP, NNTP etc)
- Improved Web User Interface
- Fully compatible with Percival
- Merlin — MS Windows NT/2000/XP based remote GUI configuration module

Chapter 2

Installation

2.1 Installation from RPM

It is best to install Percival from RPM. Percival RPM can be installed on RedHat, Mandrake and Turbolinux. RPMS can be obtained from SourceForge http://percival.sourceforge.net. The installation is as simple as doing

```
rpm -ihv percival-1-1.i386.rpm
```

After successful completion of the command you have Percival up and running.

2.1.1 Verifying Installation

Connect to `http://<yourhost>:8181.` You should see system login screen. The system comes with user guest and password guest. It is a good idea to change them.

Then do following commands:

```
su - avalon
cd /usr/local/percival/bin
./overlord.pl check
```

Command will produce following output:

```
konfigd - ok
kollector - ok
thaw - ok
querymaker - ok
```

Another check is to rebuild database:

```
./kompile
```

It will produce something like this:

```
[10-Mar-2003 17:11:29 :8336] Starting compile: Percival version 1.1.1
built on rothut.avalon-net.co.il at Sun Mar  9 16:05:50 IST 2003 by root
features: light gpl

[10-Mar-2003 17:11:31 :8336] Processed 62 nodes (in 30 files) in 4 seconds.
```

2.2 Installing from Binaries

Binary installation is a little more tricky. Percival must be installed in the `/usr/local/percival`. It expects to find perl and all supporting packages in `/usr/local/avalon`. Other configurations are not supported at the moment. It is possible to relocate Percival and its supporting packages but you should be prepared to write some scripts.

First, you have to download following tar-balls:

- Percival-Perl.tar.gz
- Percival-Apache.tar.gz
- Percival-RRD.tar.gz
- Percival-Source.tar.gz

I am assuming you have put all tar-balls in `/tmp`. Next step is to open them:

```
cd /
gunzip -c Percival-Perl.tar.gz | tar -xvf -
gunzip -c Percival-Apache.tar.gz | tar -xvf -
gunzip -c Percival-RRD.tar.gz | tar -xvf -
gunzip -c Percival-Source.tar.gz | tar -xvf -
```

The last step is to add Percival to OS start up scripts. This is OS dependent. The example of such script is located in `/usr/local/percival/bin/lancelotd`

2.3 Building from Sources

Percival builds are done auto-magically from the unified source base. Still it should be perfectly possible to build Percival from its stand alone components. Percival expects to find its supporting packages at `/usr/local/avalon`. First, you will need Perl and following list of packages:

- perl-5.6.1
- Net-Telnet-3.02
- Net-Radius-1.43
- Statistics-Descriptive-2.4
- AppConfig-1.52
- Apache-Admin-Config-0.15
- CGI-FastTemplate-1.09
- CGI.pm-2.78
- Cflow-1.025
- Color-Object-0.1_02

- Compress-Zlib-1.14
- DB_File-1.76
- Digest-MD5-2.12
- File-Tail-0.98
- HTML-Parser-3.23
- HTML-Tagset-3.03
- IO-stringy-1.220
- MIME-Base64-2.12
- MIME-tools-5.410
- MailTools-1.15
- Net-DNS-0.12
- Net-Patricia-1.010
- NetServer-Generic-1.03
- Parse-Syslog-0.03
- RadiusPerl-0.05
- SNMP_Session-0.83
- Storable-1.0.11
- Template-Toolkit-2.02
- Time-HiRes-01.20
- TimeDate-1.10
- URI-1.11
- XML-Parser-2.30
- libnet-1.0703
- libwww-perl-5.50
- IO-Tty-1.02
- Expect-1.15
- expat-1.95.2
- zlib-1.1.3
- db-4.0.14

- BerkeleyDB-0.17

Then you need to have apache and mod_perl installed. We use following packages:

- apache_1.3.27
- mod_perl-1.26

After you have done with apache you have to build rrdtool that comes with Percival. *Percival will not work with the standard rrdtool.* You have to download rrdtool-1.0.28avalon.tar.gz from the Percival site and install it.

Finally, you have to untar Percival-Source.tar.gz

Chapter 3

Percival and Lancelot Operation Basics

Despite its very simple look, Percival is a complex system. In this chapter you will learn how to operate Percival from command line. You will learn how to stop/start the system, what daemons(services) should be running, what each of them do. You will also learn how to manage network elements using `konfne` command and basic concepts of the Percival configuration database. In this chapter we try to keep things as single as possible, advanced concepts will be handled in the separate chapter.

All Percival commands a located in `/usr/local/percival/bin`. In the next sections we assume that you are working as user avalon you and you have this directory in your `PATH` variable or have done `cd` there. We also assume that Percival was installed from RPM on RedHat like Linux distribution.

It is very important to operate Percival as user avalon and not as root!!! You can switch to avalon by typing `su - avalon`

3.1 Starting/Stopping Percival

As root execute

```
/etc/rc.d/init.d/lancelotd start
```

to start the system

```
/etc/rc.d/init.d/lancelotd stop
```

to stop the system. [1]

3.2 Percival Daemons (Services) and Commands

There are several daemons that are required for Percival normal operation. The daemons are responsible for data collection, detection of not responding IPs, report generation, web user interface and managing Percival configuration by remote client applications such as Merlin. Every daemon except the web server must be managed by `overlord.pl` command.

There are following Percival daemons:

[1]The full system is called Lancelot. We did not rename all occurrences of Lancelot to Percival because we want to preserve upward compatibility between systems.

kollector does all data collection.

thawne.pl detects not responding network devices.

querymaker.pl generates all reports

konfigd provides configuration API to the external clients

httpd Apache web server. Needed for Percival Web interface.

Apache is controlled with `apachectl` command. The command is located in `/usr/local/avalon/bin`. *Apache must be managed as user root.*

`apachectl start` starts apache

`apachectl stop` stops apache.

Every other daemon is controlled with overlord.

3.2.1 Controlling Daemons with Overlord

Overlord was developed because amount of Percival daemons[2] was increasing rapidly. It was clear that some master daemon is needed to rule them all. You need to know following basic overlord commands:

`overlord.pl ping` tells status of each configured daemon

`overlord.pl check` tells status of each configured daemon. Restart the daemon if it is not running

`overlord.pl list` lists options for each configured daemon

`overlord.pl tail <name>` shows last lines of the daemon log

`overlord.pl --help` shows all available options and short help

`overlord.pl modify <name> <param=value>...` tweaks daemon options. If you installed from RPM system has very reasonable default values. You should not change them unless you really know what you are doing.

3.2.2 Common Command Line Options

Every Percival process be it a command or daemon understand some common options. These are:

`--help` show help message and exit

`--version` show version information and exit

`--loglevel <level>` controls process output. Level can be either Debug, Info, Warn or Error. Default level is Info. Debug is the most verbose and must not be used in production.

`--logfile <file>` controls where to send process output. By default all output goes to STDOUT

[2]Lancelot has even more daemons

Following options are accepted by any process but the process may ignore them. After all it makes no sense to run device configurator command (konfne) as daemon.

`--daemon` if specified process becomes a daemon(service)

`--interval <seconds>` controls how often daemon should work. For example kollector is run every 120s

Next options are available in Lancelot only. In Percival they will generate an error message.

`--cached` turns on Perl based in memory cache. The cache is optimized to be very memory friendly and produces nice speed up. The option is **obsolete**. *This option is only available in Lancelot.*

`--hdb` use an alternative implementation of the configuration database (HDB). HDB database is optimized for speed and provides order of magnitude (and in some cases even more) speedup in comparison with Percival implementation. It is also *fully backward compatible* with the old database on the API level. *This option is only available in Lancelot.* HDB database is always in use since Lancelot version 1.1.6 or later.

`--nohdb` use Percival configuration database. Beware that Percival configuration database is *very* slow. The option is available since Lancelot version 1.1.6 or later.

3.2.3 Data Collection Service (kollector)

The `kollector` daemon is responsible for gathering performance data and storing them in the database. It has following options:

`--server` run collector in the server mode. You need collector in this mode for replication and for real time graphing services. *Option is only available in Lancelot. Option is only available in Lancelot.*

`--port <port>` listen for connections on given port. Defaults to 1919 if not specified. *Option is only available in Lancelot.*

`--host <ip|name>` listen for connection on given ip address or host name. By default collector listens on 0.0.0.0 for incoming connections. *Option is only available in Lancelot.*

`--allowed <ip regex>` only allow connections from ip addresses which match a given regular expression. If not given, then connection are allowed from any ip address. *Option is only available in Lancelot.*

`--spooler` store data on disk without updating the database. The data can be used for replication or for very fast alert and report computation. *Option is only available in Lancelot.*

`--localcopy` works with the `spooler` option. If given, data are stored in the database. *Option is only available in Lancelot.*

`--client` specifies that collector should take data from remote servers. Remote servers must be given in the “servers” dictionary at the configuration root. Targets that reside on remote server must have `client = true` in their definition. Configuration synchronization between client and server should be done with out band mechanism, such as CVS, rsync or rdist. *Option is only available in Lancelot.*

`--mtargets` process collectible totals.

`--count <n>` do N iterations and exit. Useful for troubleshooting.

`--autotune` automatically rearrange RRD files if number of data sources in the file does not match number of data sources in the target definition. *Option is only available in Lancelot.*

`--lwupdate <pwrite|mmap|fullmmap>` use high performance RRD update mode. In this mode RRD file is kept open all the time and its meta data are stored in memory. The option should be used for the very large number of RRD files. *It is mandatory that you consult Avalon Net support before enabling this option. Option is only available in Lancelot.*

`--nomap` do no do interface instance mapping in collector. Instead, collector relies on `thawne.pl` to do it.

`--ioslaves <n>` use asynchronous data collection with N slaves. This option can provide a great speedup for SNMP data collection, assuming the system have enough CPU and memory. Recommended values are between 5 and 20. It is also recommended to have one collector for SNMP devices and other for slow devices such as Netflow report and others. *Option is only available in Lancelot.*

3.2.4 Network Element Status Monitor (thawne.pl)

The service is responsible for keeping status of every network element. Collector will skip those elements which are down or not responding to SNMP. Thus if device goes down, it will not stop timely data collection from the rest of devices.

It has following options:

`--address <email>` send notifications about device status changes to the given email address. Multiple addresses are coma separated. It is possible to override addresses which were set via this option by adding `thaw-notification-email` tag to the device.

`--from <email>` set address of notification sender. Default is `lancelot@localhost.localdomain`.

`--subject <text>` specifies notification subject. Default is `Device(s) changed state`.

`--freeze <ip1,...,ipN` Force freeze of all devices with given IP addresses.

`--unfreeze <ip1,...,ipN` Force unfreeze of all devices with given IP addresses.

`--slowstart` Once device is responding on SNMP queries, wait one iteration before marking it as "up". We observed incositant SNMP counters behaviour just after reboot on several Cisco routers. This option provides a work around for the problem.

`--map` prepare target mappings which can be used by the collector and other services. Since target mapping can be a rather time consuming process, the option can be used to achieve collection significant speedup.

`--report` create SNMP round trip time text report for all configured devices.

Format of `thawne.pl` notifications is specified by the `alerts/thawne-alert.tpl` template file in the current skin or in the `_default` skin.

3.2.5 Report Preparation Service

.

Percival reports are prepared by the `querymaker.pl`. The service scans specified sub trees or entire configuration, looking for reports.

It has following options:

`--profile <profile>` user profile to check for reports.

For the large configurations, it is recommended to give exact path to reports, so `querymaker.pl` will not have to traverse entire tree.

3.2.6 Remote Configuration Service (konfigd)

Percival configuration can be managed from remote via special configuration protocol. This service is provided by the `konfigd`.

It has following options:

`--port <port>` listen for connections on given port. Defaults to 1921 if not specified.

`--host <ip|name>` listen for connection on given ip address or host name. By default collector listens on 0.0.0.0 for incoming connections.

`--allowed <ip regex>` only allow connections from ip addresses which match a given regular expression. If not given, then connection are allowed from any ip address.

`--nozip` do not use LZW compression for the protocol stream.

`--links` if given, database links are expanded and not shown as a separate target.

3.2.7 Commands

Percival only has four commands and you already know everything you need about overlord. It leaves three others. One, `target.pl`, is used for debugging of the configuration database and out of the scope of this chapter. The other, `konfne` [3], is how you add, update or remove network devices from Percival. In upcoming chapters we will deal with it a lot. The last one, `kompile`, produces binary database from Percival text based configuration database and downloads device configuration files from device modules. You need to issue this command every time you edit configuration database by hand or if you suspect that Percival database is corrupted.

3.3 Managing Devices

By the time you get to this section you don't want to read anything about processes, commands, daemons and options. You want to add your new Cisco router to the Percival **NOW.** This is how you do it:

```
konfne -af --ip <ip> --community <secret> cfg /Tree/Routers/dummyname
```

[3] It is a shortcut of Konfigure Network Element

Thats all. Now you see the device once you login into the Percival as user guest.

Percival configuration is based on the concept of the "device". Device can be a router, computer, switch or other network element. In this case we call it "real" device. There is another class of device such as reports, summary graphs(totals), profile etc... These devices can be created based only on information in the database. We call such devices "Virtual".

3.3.1 Configuration Database Basics

Percival keeps its configuration information in the hierarchical(tree) database. The database is completely text based. It makes it very easy to backup and you can modify it using standard Unix CLI tools. Each user has its own view of the database.[4]

Percival converts textual database into binary format. Lancelot uses alternative implementation of the database , called HDB, that works directly over text files. `kompile` converts database into binary form usable by Percival and `konfne`. `konfne` works on *both binary and text database.*Thus you don't need to worry about keeping database copies in sync with each other.

The database is located in
`/usr/local/percival/etc/lancelot-config`
directory. Directory in the filesytem represents directory in the database. However one file can have several database child nodes. Each database entry can have many properties in form of `attribute=value` pairs. Node attributes can be inherited. Database node may have a reference (link) to another node. It work pretty much as symbolic links in Unix or shortcuts in Windows.

Percival and Lancelot come with several pre-configured database entries:

Defaults has system wide settings such as skin, location of RRD files and others

SysProfiles has default system profile. The only way to change administrator user name or password is to edit this file.

daemons contains settings for all system services. `overlord.pl` is the preferred way to manage it.

profiles contains definitions of Percival users. Percival comes with preconfigured with guest account with the root at `/Tree`. `konfne` is the preferred way to manage users.

/Devices directory contains all currently configured network devices. It also contains instructions how devices should be processed.

/Tree is root of preconfigured guest profile.

NOTE: remember to run kompile if you changed database manually.

3.3.2 `kompile`

Command

The command converts ASCII configuration database into the binary format and to notify running services and Apache web server about configuration change. It also causes all device modules to update its configuration. Run `kompile` every time you manually change configuration files.

It has no options.

[4]In Unix terms every user has its own `chroot` 'ed environment

3.3.3 `konfne` Basics

`konfne` will be your main tool for managing Percival. When you configure new or already existing devices several things happen:

- device might be SNMP scanned
- device global configuration is placed under `/Devices` according to element classes. For example Router may have interfaces and chassis. Interfaces are placed under `/Devices/Interfaces/<routername>/` and chassis configuration is placed under `/Devices/Routers/`. Some devices, such as report, may not have global configuration.
- If global configuration already exists, it is updated.
- Links and other needed device elements are created in the specified path

`konfne` has several standard options:

`--devlist` shows all available devices

`--autotype or -a` guess device type auto magically

`--ip or -i` device ip address or host name

`--community or -c` device SNMP community. If not specified defaults to public

`--fetchname or -f` fetch device name from sysName. *Everything after the last / in the path is replaced by the fetched name*

`--device or -d` configure a particular device. This options is incompatible with the automatic device discovery.

`--recursively or -r` apply command to all devices in specified subtree. Usually used for automatic reconfiguration of already configured devices. For example, `konfne -r /Tree` will reconfigure whole guest profile.

`--tag or -t <attribute>=<value>` apply device specific parameters. Each device may have specific configuration options.

konfne has following basic commands:

`help` show help for specified device or path. To get help for profile configuration you can do:

```
konfne help Devices::Virtual::Profile
```

or

```
konfne help /Tree
```

`cfg` will configure new device or update already configured device

`del` deletes profile visible device configuration. Device is still collected.

DELETE deletes profile visible device configuration and delete global device configuration. *There is no concept of device usage count. So if you have device configured in other profiles it will stop working there.* Already collected data are not removed.

DEMOLISH deletes device configuration from profile, from the database and removes all collected data.

probe check if device is responding to SNMP

3.3.4 Managing User Profiles

Percival supports concept of user. Each user must have different profile. For example, you can have one profile with the access to all of your routers. On the other hand your customer profile will give access only to specific router interface. Profile creation is governed by several simple rules:

- Nested profiles are not allowed
- All devices, except folders, must be created under profile
- Profiles with the same root are not allowed
- Profile name must be unique

Profile has three basic parameters:

1. Profile name. In this document it is also referred as user name.
2. Profile password
3. Profile root. The closes analogy to profile root is user home directory in Unix.

Device `Devices::Virtual::Profile` provides all necessary profile management.
Profile has following device specific options:

profile specifies profile name

auth specifies profile authentication mode. Only local mode, which is a default, is supported in the Percival

editable if option is present and equal to true profile user can use Merlin to manage profile.

alt-legend can be either true or false. If present and is true then graph legend is displayed under the graph in MRTG like style.

su-allowed user of this profile can switch to another profile without performing an authentication. This is mostly useful for large installation when you want to have 'master' account.

show-jrtg enable Java Real Time Grapher for this profile. This option is only available in Lancelot.

Example of creating new profile foobar with the root at `/MyProfiles/FoobarTree`:

```
konfne --device Devices::Virtual::Profile -t profile=foobar
        -t password=secret -t 'alt-legend=true'
        cfg /MyProfiles/FoobarTree
```

3.3.5 Automatic Configuration of Network Devices

Percival has ability to automatically detect type of the network devices and invoke correct device module. Auto-detection works in many cases and is the easiest way to add new equipment. The downside of auto-detection is that you can not pass device specific options to the `konfne`. *The auto-detection will not work for virtual devices.*

This is how you auto detect device:

```
konfne --autotype --ip <ip> --community <secret>
        cfg /Tree/Routers/myrouter
```

3.3.6 Standard Device Options

Every Percival device must support following standard options:

`display-name` if specified it overrides device name specified in the path. Unlike path it may have embedded HTML tags and spaces.

3.3.7 Configuring Generic MIB2 Device

Almost any SNMP manageable equipment implements MIB2. Percival uses MIB2 to obtain network interface statistics. If there is no specific Percival device for your equipment you can use `Devices::Routers::Generic` to obtain traffic statistics.

Options supported by `Devices::Routers::Generic` must be supported by any other device dealing with network interfaces. It should be noted that any device that has network interfaces is expected to provide reasonable default values. Following options are supported:

`if.namedonly` configure only named interfaces. That is interfaces which have description set in ifAlias.

`if.config-counters64` if on, try to use 64 bit high performance counters (ifHCInOctets, ifHCOutOctets) for the high speed interfaces. Device checks if interface can really return high speed counters. In our experience there are a lot of problems with 64 bit counters on CISCO routers. Care must be taken when invoking this option. If option value is `force`, interface is configured with 64 bit counters without performing any sanity checks.

`if.config-counters64-speed` 64 bit counters can only be used if interface speed is greater then specified threshold. Speed is given in megabits. Default speed value is 100M.

`if.name-source` controls what is used for interface name. Possible values are:

ifDescr a textual string containing information about the interface

ifAlias this object is an "alias" name for the interface as specified by a network manager, and provides a non- volatile "handle" for the interface

ifName the textual name of the interface. The value of this object should be the name of the interface as assigned by the local device and should be suitable for use in commands entered at the device's "console"

ifPhysAddress the interface's address at its protocol sub-layer. For example, for an 802.x interface, this object normally contains a MAC address. The interface's media-specific MIB must define the bit and byte ordering and the format of the value of this object. For interfaces which do not have such an address (e.g., a serial line), this object should contain an octet string of zero length

ifIndex a unique value, greater than zero, for each interface

by default ifDescr is used to get interface names. Some devices may have identical ifDescr but different ifName. In this case this option should be set to true.

`if.id-source` the option controls how to find ifIndex for the interface. In most cases ifDescr can be used. Possible values are: ifDescr, ifAlias, ifName and ifPhysAddress

`if.types` list of symbolic or numeric interface types that should be configured. Interface will not be configured if this option was specified and interface type does not match. Interface types are assigned by the Internet Assigned Numbers Authority (IANA).

`if.match-regexp` only configure interfaces with names that match given regexp.

`if.keep-removed` do not remove interface from configuration if it does not present on router anymore. Instead the interface is marked as "frozen". It will have word frozen added to the description and its default graphs will display will end at the time the interface was "frozen". This feature is useful to keep graph of old lines.

`if.hide-removed` hide "frozen" interfaces if true.

`if.use-ip` show interface ip name in the summary. Possible values are off, on and resolve. If option value is resolve then konfne will use DNS to resolve ip. *Note, that DNS resolving may take significant amount of time and may slow configuration and reconfiguration process.*

3.3.8 Configuring CISCO Equipment

It is well known fact that majority of the network equipment is manufactured by CISCO. Percival and Lancelot have very good support for the CISCO routers and switches, including advanced features such as SAA, Netflow and Quality of Service monitoring.[5]

Cisco Routers

CISCO routers are configured with the `Devices::Routers::Cisco` device. The devices has following options:

`setup-pptp-session` normally PPTP sessions are ignored unless the value of this option is true.

`ppp-names` normally interfaces with ifType ppp are not configured. This option accepts a regular expression. If the expression match interface name as given in ifDescr and interface type is ppp then interface will be configured.

`config-virtual` normally interfaces with the world "virtual" in ifDescr are skipped unless this option is true.

[5]Percival does not support SAA, Netflow and Quality of Service

`telnet-login` user on the router for doing login. This is needed for configuring either BGP or Pings.

`telnet-password` password of the user that was specified with previous option

`pings` configure pings from Cisco router. The option accepts coma separated list of ips or host names.

`setup-cflow` mark that the router is exporting NetFlow data.

Cisco IOS Switches

CISCO IOS switches are configured with `Devices::Switches::IOS`. The device does not have any specific options.

Cisco Catalyst Switches

CISCO Catalyst switches are configured with `Devices::Switches::Catalyst`. Device has following options:

`setup-cflow` mark that the switch is exporting NetFlow data.

`configure-slots` make separate folder for each slot. It is recommended to use this option for very large switches.

3.3.9 Configuring Linux

We support UCD-SNMP or NET-SNMP agents on Linux. We have encountered problems with the packaged SNMP agent on RedHat 7.3. You can download our build of NET-SNMP that fixed that problem from percival site on SourceForge.

Linux computers are configured with `Devices::Computers::Linux`. There are no device specific options. Linux device supports monitoring of CPU load average, memory and disk usage in addition to the interface monitoring.

3.3.10 Configuring Windows 2000

Percival can configure Windows2000 with Host MIB or with Compaq Insight Manager MIB. The correct MIB is auto-detected. Windows 2000 computers are configured with `Devices::Computers::Win2000`.

Device supports monitoring of CPU, memory and disk usage. The device has following options:

`process-watchdog` gather service uptime statistics. Accepts coma separated list of services.

3.3.11 Configuring Windows NT

NT has very basic SNMP support. To get advanced statistics you must install SNMP4C from www.wtcs.org `http://www.wtcs.org`. Windows NT computers are configured with `Devices::Computers::WinNT`. There are no device specific options.

3.3.12 Configuring Reports

Reports in Percival provides you with high level system summary. Using reports you will be able quickly determine problems in your network and zoom to the problem area to view detailed statistics. Reports are configured with `Devices::Virtual::Report`. Following options are supported:

`type` report type. There are several builtin reports:

- **`utilization`** compares network interface utilization over the period of time. Utilization is computed as $traffic/bandwidth$where bandwidth value is take from ifSpeed of the interface. Results will not be valid if your interface speed is set wrong.
- **`errors`** sort interfaces by error count over the period of time. Presence of errors on the interface usually indicates hardware problems.
- **`discards`** sort interfaces by discarded packets over the period of time. Packets are discarded when router queue is getting long. Presence of discards indicates routing problems or lack of bandwidth.
- **`overloaded`** show interfaces that are consistently utilized with over 70% of capacity. Report shows percentage of *time interface was over utilized.*
- **`idle`** *inverse of previous report.*
- **`cisco_router_cpu`** *cpu utilization of Cisco routers.*

`limit` how many results to show. Must be positive number.

`desc` detailed report description. HTML tags may be used here.

`archive` how to process data. Can be either AVERAGE or MAX.

`sort` sort report in either ascendant or descendant order. Can be either asc or desc.

`range` range of report in seconds.

`subtrees` specifies on what devices to report. Subtrees specification *is absolute to the configuration root.* Multiple subtrees are separated by comas.

3.3.13 Configuring Totals

Percival has ability to combine several graphs into one. This is useful when you want to see average utilization of some several interfaces, or your total international traffic or to see all graphs on one page. Device `Devices::Virtual::Total` provides this functionality. It can be configured with following options:

`long-desc` defines long description of the total. HTML tags and spaces are allowed.

`subtrees` list of subtrees to search for report targets.

`regexp` match target name based on given regexp. Must be used with the subtree tag.

`selection` coma separated list of targets.

`type` report type. Can be one of the following:

> **`report`** show small graphs for every interface or other target on one page.
>
> **`sum`** show graph that sums all information.
>
> **`contrib`** show stack graph for all interfaces
>
> **`avg`** show graph that averages all information.

`static` if `true` then total data are stored in the permanent RRD file and not computed on the fly. For example, ISP may want to preserve history of its traffic to the outside world. The usual total are not suitable here, because ISP periodically changes (or upgrades) its backbone. *The parameter is only valid for totals of type* `sum` *or* `avg`*.*

Percival is smart enough to figure out how to aggregate information from different devices. The details of this process are out of the scope of this manual.

This example will place graph with "sum" of all Ethernets from folder `/Tree/ServerFarm` (actually all interfaces which names match /eth/) in the folder `/Tree/Totals/` under the name total_01:

```
konfne cfg -d Devices::Virtual::Total /Tree/Totals/total_01
-t display-name="Ethernets" -t type=sum -t subtrees=/Tree/ServerFarm
-t regexp="eth"
```

3.3.14 Creating Shortcuts

In many cases, there is a need to view only some elements from other device. For example, ISP network team may have one profile that has all routers and other equipment. However ISP customer only needs to view his own interfaces on ISP routers. But interface is a part of the router device and device configuration can not be changed in another profile (or place) without affecting already configured devices.

Ability to create link to any element of the configuration tree is needed to solve this problem. Such ability is provided by `Devices::Virtual::Link`. The device has following options:

`linkto` full path to the configuration element.

Linking is only possible to the element that in turn has a link to some target in `/Devices`. Also it must be a leaf node and not a directory node.

3.4 Logfiles

Percival daemons are supposed to write their log files under the `/usr/local/percival/var/lancelot-logs/` directory. While it is possible to change that using `--logfile` option and overlord.pl, it is p probably best to stick to the convention. Overlord has command `rotate` that can be used to rotate logs periodically. In particularly log of `kollector` can grow quite large.

Logs format is:

```
[dd-Mon-yyyy hh:mm:ss :pid] free text
```

Log of every kollector measurement is written in following format:

```
[time] Retrieved data for <path>(inst)[error] :
  <ds1>@timestamp[low_bound-upper_bound],<ds2>,...,<dsn>
```

where

time log time stamp

path location of the element in configuration database

inst element instance as determined by mapping. Can be empty.

error provides a precision estimation based on error in time measurement and database sampling interval.

timestamp in milliseconds. Actual time that goes to database

low_bound in fractional seconds. Shows when measurement was started.

upper_bound in fractional seconds. Shows when measurement was completed.

ds1..dsn datasourcses. Things like ifInOctets, ifOutOctets etc... Value of `U` indicates that data source was not collected for some reason. Possible reasons are network failure, device failure and others.

3.5 Backup Procedures

As a bare minimum you need to backup configuration and data directories. They are located in
`/usr/local/percival/etc/lancelot-config`
and in
`/usr/local/percival/var/lancelot-data`
Since the Percival keeps both configuration and data in files there is no need to in any special agent. Backup can be done with standard Unix tools like tar, cpio, or dump.

To perform a full restore do:

1. stop the system
2. restore data and configuration
3. use kompile to rebuild database
4. start the system

3.6 Upgrading the system

Before you do upgrade make sure to backup your data first. Upgrading from RPM is quite easy. First stop the system:

```
/etc/rc.d/init.d/lancelotd stop
```

Remove installed RPM:

```
rpm -e percival
```

this will not remove your configuration and data. You will see some messages about directory not being empty. Install new RPM:

```
rpm -ihv precival-1-1.x.i386.rpm
```

Rebuild configuration database:

```
kompile
```

Restart the system:

```
/etc/rc.d/init.d/lancelotd stop
/etc/rc.d/init.d/lancelotd start
```

Thats all.

Chapter 4

Architecture of Configuration Database

In this chapter we present detailed description of Percival configuration database. Lancelot has same database structure as Percival but much better implementation of the database. The material in this chapter applies both for Percival and Lancelot unless explicitly specified.

As you have already learned in previous chapter, Percival stores its configuration in hierarchical database. Major database features are:

hierarchical database objects form a hierarchy. Child objects inherit properties of their parents. Child objects can override inherited properties, or introduce new properties of their own. In practice, it provides a facility to make sweeping configuration changes or updates to global policies by changing just a few lines. One can think of the database as of LDAP with the addition of attribute inheritance.

links database nodes can refer(link) to another node. It allows implementation of multiple inheritance of attributes where user specific settings override system wide defaults. It also allows to keep only one copy of the data per network device, even if device is used by several users.

text based configuration data are stored as text files in the file system, though for certain tasks, where efficiency is required, database may be converted into relational or binary form. Storing configuration data in human readable format provides for easy integration with the existing environments.

memory resident depending on the database implementation, Lancelot Engine uses either highly sophisticated caching or special data structures in order to keep the configuration database in the main memory, which ensures faster execution of common queries. [1]

optimized for read only access most of the time configuration database stays unchanged, which allows the use of an architecture that is tailored for more efficient read access. [2]

access protocol there is a client server protocol that allows to read and modify database content from remote machines. We provide Java and C++. libraries that can be used for writing of remote clients. For example, MERLIN tool uses this protocol.

[1]Percival database is disk based

[2]Percival has no database optimizations whatsoever

4.1 Database Organization

The configuration database consists of several tree structures called *dictionaries*. Each dictionary consists of nodes called *targets*. Each target can have an arbitrary number of *tag/value* [3] pairs. Leaf tree nodes are stored in files. An arbitrary number of such nodes may be stored in one file. Non-leaf tree nodes are represented by directories in the file system. Thus, it is possible to get a general idea of a configuration simply by navigating through the file system with standard tools of the operating system.

When database is being opened all files in the same directory are sorted alphabetically and then scanned. It is common convention to name files which contain general definitions as `Defaults` and files that contain number of leaf nodes as `targets`.

There are two kinds of dictionaries: dictionaries with associated tag/value pairs and dictionaries with values only. The following dictionaries are defined:

target this is a tag/value dictionary. This particular dictionary is a "special" one, because all other dictionaries complement this one. Information about all currently configured elements is stored in this dictionary. Nodes in this dictionary can have special tag that links to another node in target dictionary. *Other dictionaries do no support links*. Nodes in this dictionary are frequently called targets.

targettype this is a tag/value dictionary. The dictionary defines the type of a node in the target dictionary. It contains information about what data are collected collected and how data should be grouped by the user interface.

datasource this is a tag/value dictionary. It describes how the data are collected.

graph this is a tag/value dictionary. It describes how a datasource should be presented to the user. In most cases datasource is presented as a graph. *graph dictionary must have the same name as the existing datasource dictionary*.

graph this is a tag/value dictionary. It provides means of extending the Percival GUI with dynamically built reports.

map this is a tag/value dictionary. It controls OID instance mapping.

profile this is a tag/value dictionary. It controls authentication and access to the system, as well as user preferences. *Nodes in this dictionary must be placed at the database root folder*.

servers this is a tag/value dictionary. It controls distributed data collection capabilities.

oid this is a simple dictionary. It maps symbolic names to the numeric OIDs.

rra this is a simple dictionary. It defines parameters of time series storage database, be it RRDtool or other back-end.

The following dictionaries are deprecated: **color**, **html**, **event**, **device**, **alert**.

[3] tag is synonym to the "attribute" or "property"

4.2 File Formats

As mentioned in previous section, the configuration database is stored as a set of text files with directories representing non-leaf nodes. Leaf nodes are stored in files; each file can store several leaf nodes.

Tag/value dictionary node are defined as:

```
<dictionary name>   <instance name>
        <tag> = <value>
        <tag> = "<value>"

<dictionary name> <instance name>
        ...
```

For example:

```
target serial0_1
        interface-name = Serial0/1
        short-desc = "My interface \" at new router"
```

Simple dictionaries are defined as:

```
<dictionary name> <instance name> <value>
```

For example:

```
OID ifInOctets  1.2.3.4.5.6.7.8.9
```

Names of dictionaries are *case insensitive* and stored as lower case in the database. Same is true for names of tags. On the other hand, tag *values* are case sensitive and stored as is. Double quotation marks (") must be used when tag values contain spaces or line breaks. When a tag value contains a quote mark, the quote mark must be preceded by a backslash (\).

# character is treated as a comment and everything after it is ignored.

Line feed is used as separator in tag/value dictionaries. Simple dictionaries always contain just one line.

Tag values can contain references to other tags. For example:

```
target --default--
        router = 10.0.0.1

target serial0_1
        interface-name = Serial0/1
        short-desc = "My interface \" at new router at %router%"
```

In this example, value of the `router` tag is substituted to the `short-desc` instead of occurrence of `%router%` macro.

4.3 Inheritance of Attribute Values

The rules that govern data inheritance for tag lookup are as follows:

1. lookup tag at the current target. If the tag is found, no other steps are performed. Tag value is returned.
2. lookup tag at the nearest target named `--default--`. First, look at the same level. Then, look at the previous tree level until the tag is found or we are at the root node. There are three possibilities:
   (a) tag is found but has value `--forget--`, the search is aborted.
   (b) tag is found, no further steps are performed. Tag value is returned.
   (c) tag is not found. The lookup is terminated. Empty value is returned.

In simple terms this means that tags defined in `--default--` nodes are inherited by all other nodes.

If path to a node contains links (and if the database was opened with links enabled), then the node can be represented as an array of nodes (one node per link) with real path. Nodes are placed in the array starting from the path end. The lookup procedure is performed for each node in array, until the tag is resolved.

4.4 Displaying Node Attribute Values

Percival comes with the `target.pl` command that can be used to display attributes of the database node. The command requires several parameters:

target path path that uniquely identifies database node. The node can be either leaf or directory. The parameter is required for command to work.

dictionary valid dictionary name. If not specified, `target` dictionary is assumed.

dictionary name name of the dictionary instance. For example, `ifInOctets` is a valid name for the `graph` dictionary. The parameter is not needed for the target dictionary.

These options are accepted by the command:

`--show-children` shows all children nodes of the given database node.

In the following example we get all tags of the directory node `/Tree/Routers/rothut`:

```
target.pl --show-children /Tree/Routers/rothut
```

Command output can be something like this:

```
target: /Tree/Routers/rothut
        --heritage-- = :--default--:--default--:--default--
        --theme-- = gpl
        auto-base = /usr/local/percival/etc/lancelot-config
        auto-is-dir = true
        auto-real-name = /Tree/Routers/rothut
        auto-root = /../..
        auto-target-name = rothut
        auto-target-name-linked = rothut
        auto-target-name-unlinked = rothut
        auto-target-path = /Tree/Routers
        auto-target-path-linked = /Tree/Routers
        auto-target-path-unlinked = /Tree/Routers
        community = bigsecret
        datadir = /usr/local/percival/var/lancelot-data//Tree/Routers
        device = Devices::Computers::Linux
        ip = rothut
        password = guest
        profile = guest
        rrd-datafile = /usr/local/percival/var/lancelot-data//Tree/Routers/rothut.rrd
        tree-background = ccccff

children:
        /Tree/Routers/rothut/cpu-lnk
        /Tree/Routers/rothut/eth0-lnk
        /Tree/Routers/rothut/memory-lnk
        /Tree/Routers/rothut/storage1-lnk
        /Tree/Routers/rothut/storage3-lnk
        /Tree/Routers/rothut/storage4-lnk
        /Tree/Routers/rothut/storage5-lnk
```

4.5 Devices

Device is a group of database entries that are managed as one unit. It represents a model of a real world device, such as router, switch etc. Or it can be an arbitrary group of nodes in the database.

For example, device for the CISCO router provides database objects that describe how data are collected, stored and presented. It also provides life cycle management for the configured device instances.

Devices are managed by konfne.