Introduction to LogTrend

LogTrend

Introduction to LogTrend

by LogTrend

Copyright © 2001 by LogTrend http://www.logtrend.org

This software and all affiliated files are Copyright (C) 2001 by Atrid Systèmes under the terms of the GNU General Public License. A copy of this license entitled "GNU General Public License" is included with the software. The original text can be found on http://www.gnu.org/copyleft/gpl.html

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections , with no Front-Cover Texts , and with no Back-Cover Texts. A copy of the license entitled "GNU Free Documentation License" can be found with this software. The original text can be found on http://www.gnu.org/copyleft/fdl.html

Revision History

Revision 1.1February, 2002

Documentation completed

Revision 1.0November, 2001

First DocBook version

Table of Contents

List of Tables

Chapter 1. LogTrend's tour

LogTrend is a base structure for systems monitoring. Its main goal is to provide an open structure, quite independant from systems monitored. The Agents are in charge of collecting data from any kind of device, as long as the format of data is consistant with LogTrend's formats. Right now, LogTrend comes with agents of very different nature, such as a Linux Agent monitoring a whole system, or a http agent used for monitoring the internals of just one application.

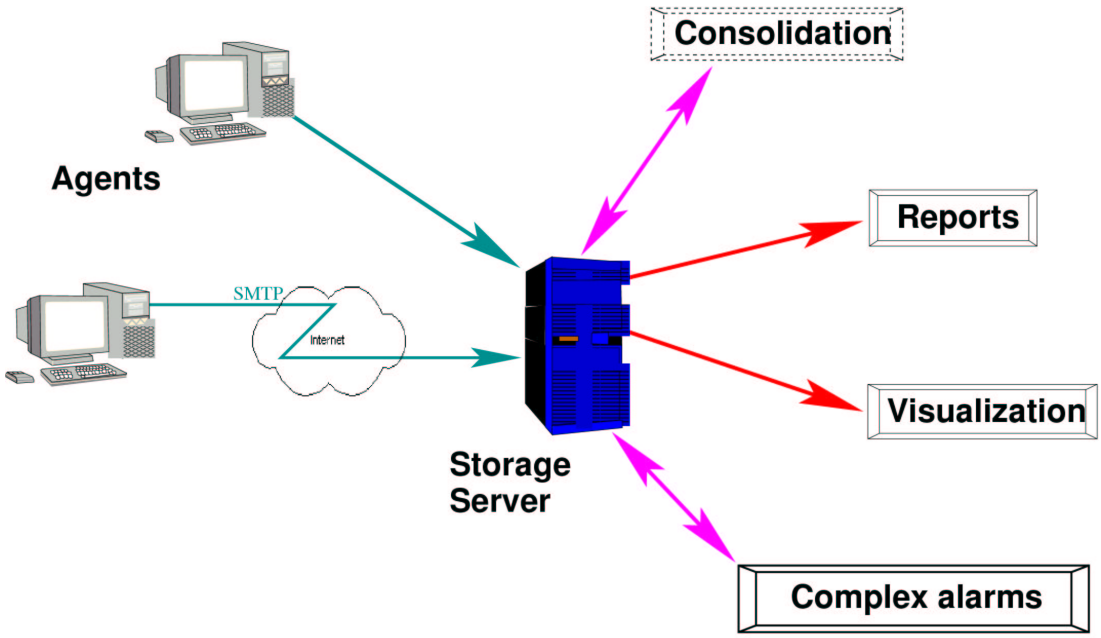
The StorageServer accepts different kind of data : real, integer, date/time or text. Assuming the agent is only using these types of data, it can collects any data; the only constraint is to declare the data before using them. During its life, an agent can modify the nature of data collected; old and new values will be linked using heritage.

LogTrend can be seen as a construction kit, where one can choose which brick he will use and how he will use it. Just configure the agent to monitor the parameters you're interested in, and configure the visualization, using the built-in objects, to create one or several visualization screens meeting your needs. Imagination is the only limit, as you can have different representation of the same data, or create your own screen combining different values, even collected by distnct agents.

The visualization allows you to pick up any data in the database, from any agent and to represent it, either directly or by combination with other data. You can even configure different representation of the same data, using different objects or by modifying the time base for example.

1.1. LogTrend's components

LogTrend is made of different modules, allowing anyone to install what is really needed. Only a few modules are mandatory, others beeing installed only if useful. LogTrend's architecture is depicted on the following schema :



The main components are :

StorageServer

The first of all components; mandatory, as LogTrend won't work without it. It is in charge of collecting data from the Agents and storing them in the database. It is running as a daemon, waiting for TCP connections on the port it is listening to (by default 9999). When an agent send its data, the StorageServer validates the xml syntax, and, if valid, stores the data in the database.

Agents

The agents collect data, and send them to the StorageServer on a regular, configurable, basis. They can operate in connected or deconnected mode, sending data respectively by a socket or by mail. They also can raise alarms which are sent to the Server as a particular type of data. It is also possible for an agent to execute an action itself when an alarm occurs.

You can have as many agents as you want for one StorageServer, and add or remove them on a working Logtrend installation.

Mailbridge

Mailbridge is simply a mail to StorageServer gateway. Called from a .forward (or equivalent, depending on the MTA) it receives data sent by mail by the agents, and resend them to the specified StorageServer.

Complex Alarms

The complex alarms module allows to define more accurate alarms than the agents' ones. In fact, this module can use any value from the database, from any agents. Using logical equations, with ability to define time restrictions, it is possible to create any alarm that may fit your needs. Things like "if CPU usage on server A is more than 90% for more than one hour, and at the same time number of users on server B is more than 10 then raise an alarm" are just...possible !

Another useful function of this module is the ability to monitor agents' activity and raise an alarm if one of them has not sent data for an abnormally long time.

Visualization

This module may appear as the most important, as it is the interface between you and the data. Its XML-based configuration allows almost any representation : declare the agents you'll get data from, and use their data with the provided objects, combining them into screens, and then building the whole visualization when linking the different screens together.

For example, the example Linux_Server class is composed of main screen displaying critical values such as memory, processes, cpu for the last 24 hours. There also are two other screens, displaying the same values, but respectively for the last two hours and for the day before.

Explaining all what can be done with the visualization would be far too long for this presentation, so please refer to the Documentation of the module, and look at the exemples to catch all the possibilities offered.

Reports

Reports is a kind of extension to the visualization whose goal is to provide a synthetic view of critical data for a quite long period, usually one week or one month. Objects and configuration are identical to those used for the visualization. The difference is the output format as reports are generated as a tar.gz archive containing the report in SGML-Docbook format and all necessary files to make it useable. Using sgmltools it's easy to create the report in a variety of format, such as pdf or html.

1.2. What to download ?

LogTrend has been divided in multiple tarballs, as one may not need every single line of code for his needs. This way, just download what you really need. The files available for downloads are listed in the table below, with a short description to help you find your way.

Table 1-1. LogTrend's tarballs

File	Description
LogTrend-ComplexAlarm-version.tar.gz	The complex alarms optional module
LogTrend-Consolidation-version.tar.gz	Consolidation module
LogTrend-Doc-version.tar.gz	Documentation package
LogTrend-FtpAgent-version.tar.gz	A LogTrend's agent for proftpd monitoring
LogTrend-HttpAgent-version.tar.gz	A LogTrend's agent for apache monitoring
LogTrend-LinuxAgent-version.tar.gz	A LogTrend's agent for linux boxes
LogTrend-MailBrige-version.tar.gz	Mailbridge for getting data from distant sources
LogTrend-SNMPAgent-version.tar.gz	A LogTrend's agent for SNMP Monitoring
LogTrend-SimpleAgent-version.tar.gz	A LogTrend's agent for developping purposes
LogTrend-SnortAgent-version.tar.gz	A LogTrend's agent for Snort IDS reports
LogTrend-StorageServer-version.tar.gz	The main part of LogTrend : the server storage
LogTrend-Visu-Apache-version.tar.gz	The Apache's module part of LogTrend's visualization
LogTrend-Visu-Engine-version.tar.gz	The visualization engine

The minimal LogTrend's installation will consist of the StorageServer and one agent. The visualization modules come next in the list as they are the only way (except psql ;) to get a representation of the data. Note that one server can store data for numerous agents.

Agents can send data to the server directly through a TCP socket, or by mail if a direct connection is not possible. In this last case, you will need the mailbridge to be installed on a SMTP server receiving mail for your logtrend's user; the SMTP server must be able to establish a socket to the Storage Server to send the data.

As the number of agents increase, the consolidation module will become more and more necessary, as it will reduce the number of data in the database : older values (configurable) can be averaged or deleted. Consolidation is necessary to keep good performance, not overloading the database with obsolete data.

At last, the complex alarms module will only be needed if you want to establish complex critical situations. Keep in mind that every agent is able to raise its own subset of "simple" alarms, essentially threshold-based ones, so complex alarms modules is not mandatory for getting simple alarms. Nevertheless this module offer the possibility to monitor LogTrend's agents, raising an alarm when an agent doesn't send its data.

1.3. LogTrend's agents

Here is a brief overview of the standard LogTrend's agents.

Linux Agent

The Linux Agent is aimed at monitoring all critical values of a Linux server, including CPU load, Memory, Swap, disk space, number of processes and zombies. This agent can also monitor the presence of some network services (http,ftp,...) and check that any given process is running on the system. The agent is able to raise alarms and can also execute actions like running again the dead process.

SNMP Agent

The SNMP Agent can collect any information from any SNMP-Aware system or device. It can collect SNMP Data as data and SNMP traps as alarms. The agent can collect data from any SNMP Device as you only have to provide the right MIB and select in this MIB which OID to collect as LogTrend's data. Until native agents have been developped for some operating systems, it is possible to collect data from them, as long as they can run a snmp server/service.

Snort Agent

The Snort Agent is a gateway between LogTrend and the Snort IDS (Intrusion Detection System). This agent will classify Snort's messages and generate LogTrend's alarms. Using this agent, one can be informed from LogTrend's visu that an intrusion may have occured, and some automatic actions can even be configured. This agent does not keep you away from getting into Snort's logs, it just does provide a synthetic view of most importants alerts.

HTTP Agent

The HTTP Agent should really be called Apache Agent as it is in charge of collecting data about a running Apache, either locally or remotely. If Apache and the agent run on the same system, the agent will get informations like memory load, number of child processes, number of connections, requests per second,... On a remote server it will only get a download time.

FTP Agent

The FTP Agent should really be called Proftpd Agent as it is in charge of collecting data about a running Proftpd, either locally or remotely. If Proftpd and the agent run on the same system, the agent will get informations like number of connections, bytes sent, globally or URI by URI... On a remote server it will only get a download time. Alarms include too much Login failures notification.

Chapter 2. Quick installation Guide

The installation of each component is detailled in the corresponding documentation, but here is an overview of a whole installation of LogTrend :

i. The first step is to install all Perl Modules needed for LogTrend and a database. Right now only PostgreSQL is supported. Needed Perl Modules are listed in the README file of each package.

ii. Install the StorageServer; for this, get the files, create the database using provided databasecreation.sql script, and configure the Storage Server to match your database configuration. Note that the StorageServer and PostgreSQL do not need to run on the same system.

iii. Declare a new source using the **AddSource** utility. A source must be declared before sending data.

iv. Install an agent on the newly declared source. Configure it to fit your needs. Configuration of the agents is explained in their own documentation.

v. Run the agent in description generation mode (using the `-d` option. This will create a xml file where agent's data are defined.

vi. If the agent cannot open directly open a socket (by default on port 9999), you'll have to install the MailBridge and configure the agent to send its data by Mail. Install the Mailbridge on a SMTP server receiving mail for the configured address, and able to open a socket to the StorageServer.

vii. Send the description of the agent's data to the StorageServer using the **AgentDescriptionToDB** utility. This step is mandatory because the StorageServer needs to know what the data are before accepting them. Now you're up with your first agent !

viii. Repeat steps 3-7 to add more sources/agents.

ix. To get representations of collected data, you will have to install the Visualization module. This module is running as an Apache Module, so you'll have to get Apache running prior to installation. Visualization must next be configured; please refer to its documentation for all options.

x. Install other modules to complete your installation. Consolidation to clean old data in the database, Complex Alarms to create alarms based on logical combinations of data.