LINUX System Monitor

Design

v. 0.1, 6/17/09, Dan Graham

This document provides a rough design for Linux System Monitor, *which monitors Linux server assets..*

Table of Contents

Introduction

The purpose of the system is to provide a remote web-based graphical monitoring tool for system parameters, such as, memory, disk, cpu, and network load. The system should also provide for Alerts in the form of email sent when triggers occur. The system can use a MySQL database back-end.

General Architecture

1. Each Linux server asset can have “probes” installed that, periodically (5-10 minutes), polls for system statistics and writes them to XML files in a specified local location.
2. The central database server has an ETL process running, which performs the following steps (every 5-10 minutes):

Sftp XML files from remote servers → load into database → transform into summary tables → load email alert table

3. The central server, after ETL, will scan the email-alert table and send the alerts.

The front-end could consist of the following:

4. Read-only dashboard display, served through Apache. This display can use a module, such as GD::Graph to display the charts, via CGI.

Probe Design

The probes can create XML files in the following categories:

A. Network Load

```
<netusage>
  <fromIP>10.47.82.47</fromIP>
  <toIP>localhost</toIP>
  <bytes>12355413</bytes>
 </netusage>
```

It is possible that the “trafshow” utility, could be used to generate this data. Or, if necessary, a custom C++ program could be written using BPF (Berkeley Packet Filter) and LPF (Linux Packet Filter) libraries.

B. CPU

```
<cpu>
   <pid>
     1234
    </pid>
    <command>
          qdserv
     </command>
      <owner>
         lbxprod
      </owner>
    <percent>
    92
   </percent>
 </cpu>
```

Various commands such as systat, top, or ps could generate this data.

Note: Some of these commands generate Curses formatted displays, which would have to be parsed.

C. Memory

```
<mem>
      <pid>
         12345
      </pid>
     <command>
          ssh
      </command>
      <owner>
         degraha
       </owner>
       <percent>
          30
```

```
        </percent>
 </mem>
```

D. Disk

```
<disk>
    <filesystem>
       /dev/amrd0s1g
    </filesystem>
     <mount>
         /usr
      </mount>
      <capacity>
          19
       </capacity>
</disk>
```

df –k command could be used.

Central Database

The ETL process would first load data into the MySQL tables. The transform step could summarize data from the XML files/tables, such as total network traffic, etc.

MySQL triggers could be used to insert data into the email_alert table. Complex conditions can be used in the trigger.

Finally, an outside process would read from the email_alert table, and send out the emails.

GUI dashboard design

A Java/Swing applet, served through Apache, is another design possibility for the front-end. The JFreeChart library could generate the charts. But, it is, probably, more common practice to use a Perl library such as GD::Graph to serve up GIF file graphs through CGI. AJAX technology can be used to update the charts, dynamically, without reloading the page.

Some ideas for the GUI are as follows:

1. Tabbed sites, with a combo-box (drop-down) to select servers within a site.
2. selectable (scrollable) times (past to present) – possibly with a slider.
3. selectable time playback speed – possibly with a slider.
4. Pie charts showing composition of 5 highest consumers.
5. Moving bar charts showing totals.

Notes

In order to synchronize the transfer of XML files, they can be sftp'ed from a directory that only contains symbolic links to the XML files. This will prevent an SFTP of a partially written file.

The probe footprint should be as small as possible, with polling occurring only once every 5 minutes. However, for network load, the trafshow (or similar) utility must run continuously, in order to get bandwidth usage totals, since a single snapshot view is not

possible. Although, the footprint, for LPF/BPF monitoring, is not, necessarily, large; and needs to be determined. (The trafshow utility, when run interactively from a terminal, shows a lot more activity than normal, because of a feedback loop of tty/ssh activity). Another possibility would be to gather network usage totals for 5 minutes, then sleep for 5 minutes, giving a 5-minute sample, every 10 minutes. This would reduce the footprint.