SNMP-based Monitoring of a Computing Cluster

Moreno Marzolla
marzolla@dsi.unive.it

Dipartimento di Informatica Università di Venezia
&
INFN Padova/BaBar collaboration
Talk Outline

- The Monitoring Challenge
- Some Existing Tools
- ASC: the Asynchronous SNMP Collector
- Conclusions and Future Work
The Challenge

BaBar’s Data Reprocessing suddenly stops. What happened?

- Client process(es) crashed
- Server process(es) crashed
- The local disk failed
- The CPU melted
- \( /tmp \) overflowed
- None of the above
Requirements

- **Scalable** up to $\approx 150$ machines and more.
- **Easy to configure**: If one has to do simple things the effort required to configure the tool should be minimal.
- **Extensible**: New functionalities should be easy to add as they are needed.
- **General**: Needs to monitor the network switch, tape library, UPS, environmental system...
- **GUI-independent**: Should work as a regular UNIX daemon, yet providing a convenient user interface(s).
Some tools

There are many freely distributable monitoring tools available.

- Many, many others...
What’s wrong?

So, why don’t simply use one of the many programs available?

Because they fail to meet our requirements. In particular:

- Many of them don’t scale well/at all.
- Configuration is highly nontrivial in most cases (lots of different configuration files scattered around...).
- Many of them require special daemons running on the monitored hosts: can’t handle the network switch/tape library/UPS...
The Do-It-Yourself approach

So we decided to build a tool (ASC, the Asynchronous SNMP Collector) from scratch.

- The tool is entirely written in C.
- Use asynchronous (non-blocking) SNMP requests for data collection.

SNMP (Simple Network Management Protocol) is a standard protocol supported by many different pieces of hardware. Even our air conditioning system speaks SNMP...
The Do-It-Yourself approach (2)

- Data are stored in **Round Robin Databases**
  Provide facilities for storing timestamped data with different granularities; facilities for plotting graphs are also provided

- The configuration file is written in **XML**

- ASC embeds a simple **HTTP** interface
  HTML pages are generated by applying an **XSLT** stylesheet to an automatically-generated XML status file
XML Configuration File

```xml
<?xml version="1.0" standalone="no"?>
<!DOCTYPE monitor SYSTEM "monitor.dtd">

<monitor numconnections="20" asclogfile="/monitor/asc.log"
         httpdlogfile="/dev/null" rrddir="/monitor"
         htmldir="/monitor/html" ascverbosity="3" >

<!--
 == Hosts configuration
 -->

<host name="localhost">
  <description>This machine</description>
  <miblist>
    <mib id="cpuUser" name="1.3.6.1.4.1.2021.11.50.0" type="COUNTER">
      <archives>
        <rra cf="AVERAGE" granularity="60" expire="604800"/>
      </archives>
    </mib>
  </miblist>

</host>

</monitor>
```
XML to HTML Output

XML Configuration File

ASC

HTTPD

Monitored Hosts

Status

XSLT Stylesheets

HTML Pages

Workshop sulle problematiche di calcolo e reti nell'INFN – p.10
HTML Interface
Conclusions

ASC is still under development. Are we approaching the goals?

- **Scalability** On the test farm ($\approx 20$ machines) it is working very well. Will it scale tenfold? (Note that applying stylesheets is not cheap...)

- **Easy Configuration** We have a single configuration file, which can optionally be split in different parts and use macros. These are standard features of XML.
Facing a scalability limit
Work(s) in Progress

- Implement Alarms
- Understand SNMP Traps
- Active control of ASC with its WEB interface
- Implement more XSLT stylesheets
- Write the documentation