An SNMP based failure detection service

Matthias Wiesmann, Péter Urbán, Xavier Défago
Overview

• Failure Detection
• SNMP
• SNMP-FD Service
• Conclusion & Future Work
Failure detection

• Basic abstraction for distributed systems
• Needed for solving fundamental problems:
  • Consensus, Atomic broadcast, Atomic commitment
  • Formal definitions: $P$, $S$, $\Omega$, $\phi$, etc.
• Also outside distributed system community
  • System, Network, Cluster management, Middleware, Grid systems
• Common problem $\rightarrow$ common solution
Failure detection service

- Factorize-out failure detection
  - Shared, separate implementation
  - Complex, adaptive policies
- Proposed in the literature
- Implemented by middleware systems, grid systems
- Problem: *no standard*
  - Every system has its own service and interfaces
Standard Needed

- External standard
  - Connect clients of the service to the service
    - Application, Middleware, Monitoring and administration tools
  - Suspicion notifications, configuration
- Internal standard
  - Use standard for heartbeat messages, internal queries
- We need some form of Middleware
Overview

• Failure Detection
• **SNMP**
• SNMP-FD Service
• Conclusion & Future Work
Why SNMP?

- Simple Network Management Protocol
  - Present in many network attached devices *now*
  - Routers, Smart Switches, UPS, Printers…
- Simple Middleware
  - Offers basic features
  - Relatively lightweight
- Many applicable standard defined
  - Monitoring, fault reporting.
SNMP Architecture

- Network Management Station
  - Manages equipment
- Equipment’s agent
  - Exposes Management Information Base (MIB)
  - Sends asynchronous messages (traps) to NMS.

An SNMP based failure detection service
Overview

• Failure Detection
• SNMP
• **SNMP-FD Service**
• Conclusion & Future Work
An SNMP based failure detection service
Service Architecture

- Distributed architecture
- Daemon runs on each host
  - Monitor local processes
  - Exchange heartbeats
  - Export MIB information
- Can interact with equipment
- Similar to other failure detection service architectures
Features

• Uses many standard MIBs
  • Host Resource MIB → Process state
  • Target & Notification MIB → Subscriber description
  • Alarm MIB → Current suspicion description

• Lightweight messaging:
  • Heartbeat are UDP traps.

• Leverages SNMP
  • Configuration, Security, Interoperability with devices
New MIBs

- Heartbeat MIB
  - Describes heartbeats

- Failure detection MIB
  - Describes failure detector configuration

- Described using standard SNA syntax file

- Accessible using standard tools

SNMP table: SnmpFD::kbhbTable

<table>
<thead>
<tr>
<th>index</th>
<th>interval</th>
<th>lastbeat</th>
<th>lasttime</th>
<th>count</th>
<th>lostcount</th>
<th>mean</th>
<th>stddev</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>kt-dhcp23.jaist.ac.jp</td>
<td>100</td>
<td>2678 0:0:04:38.49</td>
<td>2671</td>
<td>0</td>
<td>99.9955</td>
<td>28.3417</td>
<td>0</td>
<td>356</td>
<td></td>
</tr>
</tbody>
</table>

An SNMP based failure detection service
Implementation

• Validate the architecture
• Implemented in Java
  • Open source libraries
  • SNMP4J, SNMP4J Agent, Apache Libraries
• Available for download
  • http://ddsg.jaist.ac.jp//en/projects/snmp-fd/
Implementation

- Validate the architecture
- Implemented in Java
  - Open source libraries
  - SNMP4J, SNMP4J Agent, Apache Libraries
- Available for download
Performance

- Simple failure detector
- Stable performance
  - Consistent with expectations
  - Validates approach
- Issues
  - High latency
  - High CPU load on receiver

An SNMP based failure detection service
Performance

- Simple failure detector
- Stable performance
  - Consistent with expectations
  - Validates approach
- Issues
  - High latency
  - High CPU load on receiver

An SNMP based failure detection service

SNMP4J Issue, code spends 95% of time there.
An SNMP based failure detection service

Overview

• Failure Detection
• SNMP
• SNMP-FD Service
• Conclusion & Future Work
Conclusion

• Failure detection service
• Standard based approach desirable
• No need for new standard → use existing SNMP
• Approach validated by prototype
• Can be extended with more advance FD techniques
  • Hierarchical FD, adaptive FD, etc.
• Can interact with network devices
  • For instance link-down traps
Future work

• Look into performance issues
  • Optimizations needed in the SNMP4J library.
• Port the framework to the latest version of SNMP4J
  • Support for AgentX
  • Implement framework as sub-agent
• Implement more advanced failure detectors
• Handle network equipment data
Questions?
Questions?

Thank you...