Remote Health Monitoring for an Embedded System

Authors: Puneet Gupta, Kundan Kumar, Vishnu H Prasad
Background
Scope, Requirements & Challenges
Health of an embedded system refers to a **state of well-being**

- No unexpected behavior (exceptions or errors)
- No crashes or hangs of system software
- System performance at expected levels
- Resource utilization within permissible bounds
- Handling user inputs in the expected manner

**health**  [helth]  

*noun*

1. the general condition of the body or mind with reference to soundness and vigor: *good health; poor health.*
2. soundness of body or mind; freedom from disease or ailment: *to have one's health; to lose one's health.*
3. a polite or complimentary wish for a person's health, *happiness,* etc., especially as a toast: *We drank a health to our guest of honor.*
4. vigor; vitality: *economic health.*
Applications

- Any **networked** embedded system
  - Enterprise, military, government, defense, infrastructure or industrial systems

- Ittiam focus: embedded multimedia systems
  - Video Surveillance Systems
  - Health Care Systems
  - Telepresence and Video Conferencing Devices
  - Home Networks
  - Enterprise Streaming Systems, Digital Signage
  - Digital Broadcast Systems
Need for Health Monitoring

- Need for inspection of system behavior on the field
  - QoS guarantee
  - System reliability requirements
  - Caps on resource utilization
  - Enabling troubleshooting in deployed units

- Evolution of designs and software frameworks
  - Feedback for implementation
  - Inputs for optimization

- Failure-proof operation for mission critical applications
  - System state based notifications and alarms
  - Information to enable corrective action(s)
  - Fail-safe operation on the field
But Why Monitor Remotely?

_remote refers to off-the-device operation
- Non-intrusive in nature
- Minimal impact on the device operation during monitoring
- Many a times the device might be physically inaccessible

Networked devices offer an easy mechanism to monitor
- Socket based interactions with remote monitoring system
- Standard ways available in OS for enabling tracing and communication

No dependency on the embedded device’s UI capabilities

Zero overhead post processing of the monitored data
- Operations may be computation intensive
- May require rich rendering of graphs and plots
Monitoring Requirements

- Network-based platform agnostic framework to observe system behavior
  - Statistics, events, alerts and notifications, exceptions (errors, warnings), debug information, system traces
- Low cost in terms of processing resources needed on the embedded device
- Ability to post process received system data – plot, draw more evolved statistics and observe trends
- Present data in user friendly format(s)
- Allow rule-based extensions to the core functionality
- User authentication for monitoring client
Challenges

- Controlled volume of data transmitted
- Minimal memory footprint
- Minimal processor loading overhead
- Priority of execution, background mode
- Platform agnostic
- Application agnostic
- Consistent framework for all use cases
- Portable on any standard OS
- Usability of monitored info
- Minimal network overhead
- Bandwidth requirements
- Richness of statistics, logs and associated details
**Server client model**

- **Server**: embedded device to be monitored (source of information)
- **Client**: monitoring end point (interpreter of information)

**Networked architecture**

- Multicast or unicast modes
- Multiple clients for each server

**Based on text based low-bandwidth data flow from server to client(s)**

- Assumes syntax compatibility on server & client
Data Flow Diagram

Embedded System (Server) → Logs → Network → Logs → RHM Client Tool

- GUI
- Log Parser
- Analytics Engine
Modes of Monitoring

- **Offline**
  - Monitoring information is archived while the device is in operation

- **Live**
  - Information is monitored while the device is in operation

- **Active**
  - Information is requested actively from the monitoring end

- **Passive**
  - Monitoring end receives whatever information is sent from the device
What to Monitor?

Statistics

- Resource Usage
- Network Statistics
- Media Statistics

Exception Occurrences

- Errors, Warnings

Events and Alarms

- Limits / Thresholds Based
- Occurrence Based

System Logs

- System state
- General purpose Information
Technology Features

**Server End Features**
- Framework for logging information and reporting statistics
- Framework for logging exceptions and key system events
- Standardization of the format for text based logs
- User authentication
- Information filtering based on user permissions
- Data transmission (TCP / UDP, unicast / multicast)

**Client End Features**
- Desktop tool to monitor with user log in
- Filtering of incoming data
- Long term and short term data representations
- Rule based alarm generation framework
- Archiving of monitored information
Server: Logging Framework (1)

- Foundation of the RHM functionality
- Comprehensive and system-wide logging mechanism
  - Independent of the logger module used
  - Independent of the platform
- Information rich logs
  - Generated in a standardized format
    - RHM server & client need to align on this
  - Categorized in a scalable manner
    - Statistics, System Logs, Exceptions, Events, etc.
  - Information on source of the log
  - Details on the time of generation of the log
- Designed for multi-threaded systems
- Extends to multi-process systems
Server: Logging Framework (2)

- **System Logs**
  - Periodic state checks
  - Watchdog output
  - User access (command or configuration) response time
  - etc.

- **Exception Reporting**
  - Synchronous or asynchronous occurrences
    - Errors
    - Warnings
    - etc.

- **Event Reporting**
  - Asynchronous occurrences
    - Notifications
    - Alerts
    - etc.
Statistics Reporting

Resource Usage
- Thread stack usage, load
- Heap memory usage, DDR usage
- etc.

Network Statistics
- Total packet transmitted/received
- Packet loss incurred
- etc.

Media Statistics
- Number of frames encoded, dropped
- End to end latency of operation
- etc.

Exception Statistics
- Number of errors (of each type) encountered
- Tally of special events reported
- etc.
## Monitored Data: Illustration

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Log Type</th>
<th>Timestamp</th>
<th>Source ID</th>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIDEO-PHONE</td>
<td>Error</td>
<td>[322045525]</td>
<td>[0x3001]</td>
<td>VFR: Out of memory</td>
<td></td>
</tr>
<tr>
<td>DVR[Info]</td>
<td>[323674356]</td>
<td>[0x7020]</td>
<td>EXTRACT: Reached GOP boundary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SERVER[Event]</td>
<td>[342552769]</td>
<td>[0x1031]</td>
<td>Camera Disconnected</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SERVER[Warning]</td>
<td>[342045769]</td>
<td>[0x4050]</td>
<td>ENC: Frame skips observed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TXCODER[Stat]</td>
<td>[329834209]</td>
<td>[0xA127]</td>
<td>DEC: Avg Frame Rate = 29.97</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Server – Client Communication

- Extended functionality
  - Secure logging mode
    - Encryption of logs (SSL/TLS)
  - User authentication
    - Multi-client access to the same device
    - Multicast
Client: Log Parsing

- Receives logs over network
  - 1 client caters only to 1 server
  - Multicast / unicast modes
  - Queue based implementation to ensure no-loss-reception

- Parsing Framework
  - Support for real time parsing of incoming logs
  - Configurable parsing level
  - Tag based parsing
  - Pre-defined syntax
  - Source of information for the analytics engine
Client: Log Parsing (Exceptions/Events)

High
- Emergency
- Alert
- Critical

Medium
- Error
- Warning

Low
- Notice
- Information
Define predefined static limits and raise alarm when limits are met
- Statistics based
- Log parsing based
- Exception count based

Machine learning algorithms
- Derived statistics from reported statistics
- Finding patterns in statistics, events or exceptions
Illustration 2

REMOTE HEALTH MONITORING TOOL

Enter Parameter: AVG BR
Scaling Factor: 1000000

: Segment 0
: Segment 1
: Segment 2
: Segment 3

Scaled Values

0 10
0 50

Quit
Client: Analytics Engine

- Archiving and smart Indexing of system logs
  - Offline mode
  - Keyword based log summary and search tags

Filtering
Pattern Searches
Data Tracking
Data Post Processing
Usage Scenarios

- Remote Troubleshooting
- Automated Alarms
- Non-intrusive Debugging
- Critical Event Monitoring
- Performance Monitoring
- Design Optimization
Usage Scenario 1 – Remote Troubleshooting

- Debugging for field deployed or off-site systems
  - Remote customer support, development asset
  - RHM delivered as a standard diagnostic feature

- Case Study:

  - Digital Video Transcoder
    - Converts from MPEG2 TS streams to H264 RTP streams
    - Multi-channel transcode

  - Monitoring Mode Example
    - Live, Active
      - Processing latency, system loading
      - Occurrence of decode and encode exceptions
    - Analysis to help user understand any quality issues
    - Tuning configurations for better performance
    - Inputs for debug activity needed to resolve bigger issues
Usage Scenario 2 – Automated Alarms

Events when manual intervention or corrective action is needed

- Surveillance equipment, Infrastructure devices
- May require constant monitoring over a long period
- RHM integrated as a standard product feature

Case Study:

Security DVR

- Live video feed (16 channels) recording and streaming to a station
- Operations under harsh climate conditions

Monitoring Mode Example

- Live, Passive
  - Data pertaining to temperature of key hardware components
  - Events on camera and network connectivity
- Tuning criteria for events like camera tampering
- Aid for operator / network administrator
Usage Scenario 3 - Non-intrusive debugging

- Debugging without affecting system performance
  - Minimize impact of monitoring on resource consumption
  - Design and optimization aid for developers

Case Study:

- Digital Video Broadcast System
  - DVB server streaming to network media player on TV
  - High quality compression and low latency streaming

Monitoring Mode Example

- Offline, Passive
  - Number of frame skips by encoder
  - Peak and average bit rate statistics
  - Server latency

- Analysis to determine optimal settings of encoder
- Tune amount of buffering in the media pipeline
Usage Scenario 4 – Critical Event Monitoring

General purpose functionality for devices
- Keep track of events of interest from the device
- RHM integrated as a standard product feature

Case Study:
- Battery powered unmanned surveillance drone
  - Streaming live feed from reconnaissance mission to a client PC over wireless network
  - Periodic snapshots recorded to memory card

Monitoring Mode Example
- Live, Active
  - Total distance traveled
  - Level of disk usage
- Decision on the continuity of operation
- Adjusting the resolution of captured images
Devices with long term deployments and sustained usage
- Infrastructure devices: need to continuously monitor state of operation
- RHM integrated as a standard product feature

Case Study:

**Entertainment Server**
- File streaming to multiple users (high density) in various formats

**Monitoring Mode Example**
- Live, Active
  - Number of users connected
  - Bandwidth available
  - Overall CPU load
- Adaptively update bit-rate or choice of video /audio codec
- Tune the size of network packets or socket buffers
Usage Scenario 6 – Design Optimization

Next generation designs based on the performance of currently deployed systems

- May be critical for design evolution: performance differentiator
- RHM used as a design optimization aid for developers

Case Study:

- Video Communication Equipment
  - 4-way conference with audio and video, low latency operation

Monitoring Mode Example

- Offline, Passive
  - System loading
  - Overhead for thread switching
  - Time taken for IPC communication

- Analysis of collected statistics to update thread model
- Assessment of IPC communication architecture
- Optimization of system memory map
Ittiam’s Solutions with RHM Embedded

A Synopsys
The entire suite of video networking and communication systems:
- Video surveillance systems (servers, DVRs, NVRs)
- Enterprise streaming server, transcoder and client systems
- In flight entertainment systems
- Health care systems
- Video communication systems

RHM Server ported on Linux, x86, Android
RHM Client ported on x86 Linux, Windows
Premium feature for premium customer deliverables

Key Advantages:
- Low cost implementation in terms of resource usage
- Runs on any standard PC
- Platform portable UI developed in QT
Performance Metrics (Server)
- TI’s DM6467 Platform
- Memory Footprint: <100KB (TBD)
- CPU Overhead: <2% (TBD)

Performance Metrics (Client)
- Intel x86 (i5, Windows XP / Windows 7 / Linux)
- Memory Footprint: <10KB (TBD)
- CPU Overhead: <5% (TBD)
About The Authors

- **Puneet Gupta**
  - Principal Engineer, Video Communication & Networking
  - puneet.gupta@ittiam.com

- **Kundan Kumar**
  - Engineer, Video Communication & Networking
  - kundan.kumar@ittiam.com

- **Vishnu H. Prasad**
  - Engineer, Video Communication & Networking
  - vishnu.prasad@ittiam.com

- **Ittiam Website** ([www.ittiam.com](http://www.ittiam.com))
Thanks