Interactive Real-time Interface for Smart Health Monitoring and Analysis

mS151161, Achraf Ben Ahmed
Supervised by Prof. Abderezek Ben Abdallah
Adaptive Systems Laboratory,
Master of Computer Science and Engineering
Graduate School of the University of Aizu, Japan

Background (1/2)
- Multicore SoCs became an opportunity to satisfy the high requirement of data and computation intensive applications such as bio-medical data processing.
- We previously proposed and developed a novel embedded health monitoring platform based on various efficient HW and SW techniques (BANSMOM*)
- The system can be easily adapted to different subjects or different signals of interest.

Background (2/2)

Research Motivation
- Existing interfaces drawbacks:
  - Manually-intensive workflow for data acquisition, formatting, and visualization.
  - No information related to the node (name, location...)
  - No recording time
  - ECG wave is illustrated only with number format
- No support for secure RT visualization
- Need for a more robust system for collection, visualization, and analysis of physiological data.

Research Goal
- Design and evaluation of an Interactive Real Time Interface for monitoring purpose:
  - Real-Time support Algorithm for data visualization
    - Efficient data recording algorithm
    - Smart record management
    - Multiple-Nodes support scheme
    - User friendly
    - Data adaptation to current BANSMOM hardware

Outline
- Background
- Motivation
- Research goal
- Interactive real time interface features
- Evaluation
- Conclusion and future work
The IRTI was developed following the MVC pattern:
- Model, View, Controller
- All requests are independent since they are managed separately by the creation of an application instance

Multiple Nodes can be managed in parallel without a negative impact on the overall performance.

**IRIT General Structure**

**IRI Development Phases**

**IRTI Workflow**
- We assume for 1 peak
  - Insertion frequency $I_f$
  - BANSMOM processing time $TE_{Tp} = 0.06$ second/peak
  - Database update $DB_{up} = 0.07$ second/peak
  - Reading frequency $R_f$ (by Ajax call)

$$R_f = I_f = TE_{Tp} + DB_{up}$$
- A buffering mechanism approach is used in the DB level to cover any additional delay

**BANSMOM ECG Processing Distribution**

$$TE_T = R_T + PPD_T + ST$$

Sample No 16265, 10 seconds, 14 periods, 84 peaks

$$TE_T = 3.608 + 2.144 + 0.001 = 5.753s$$

$$TE_{Tp} = 0.06s$$

**ECG Data Capturing and Storing Phase**

$TE_T$: Total execution time (10 seconds sample data)
$R_T$: Lead reading time
$PPD_T$: PPD processing time
$ST$: Storing Time

$TE_T = R_T + PPD_T + ST$

If DB empty?
Yes
Capture data [day, $i$, $j$]
Store data

No
Check DB [day-$3$] abnormality

If data is normal
Yes
Delete DB [day-$3$]

No
day = day-$3$
Development & Evaluation Methodology

- Development Environment
  - PHP 5.3.13
  - MySQL 5.6
  - Apache 2.2.22
  - YII MVC framework 1.1
  - Netbeans IDE 7.1

- Evaluation Parameters
  - Developed tool
  - Security and vulnerability issues
  - Real time visualization
  - Code Complexity

Developed IRTI

Node List View

Old Records Visualization View
**Security and Vulnerability Evaluation**

- Scan parameters: (Netsparker web vulnerability and security testing)
  - Remote Code Evaluation, SQL Injection, Cross-site Scripting, HTTP Header Injection

**Vulnerability & security Evaluation**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>09:25</td>
</tr>
<tr>
<td>Total request</td>
<td>34313</td>
</tr>
<tr>
<td>Average speed</td>
<td>16.28 req/sec</td>
</tr>
<tr>
<td>Nbr of issues found</td>
<td>20</td>
</tr>
<tr>
<td>Informational issues</td>
<td>20</td>
</tr>
</tbody>
</table>

Although the proposed IRTI SW tool has required features with RT capability, the total code is small and only about 700 lines.

**Code Complexity Evaluation**

<table>
<thead>
<tr>
<th>Files (.php)</th>
<th>Models</th>
<th>Views</th>
<th>controllers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ecg,EcgHistory,EcgInfo</td>
<td>70</td>
<td>450</td>
<td>200</td>
</tr>
<tr>
<td>Ecgformmodel,Ecgform</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responsible,user</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sensorinfo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Index,Contact,Login</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>List,Info,live</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EcgController</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Future Work**

- Currently my developed SW tool supports only ECG data RT monitoring.
- Further study and test should performed with other biomedical data.
- Efficient Compression and Encryption Algorithms should be investigated for more secure communication.

**Conclusion**

- Design and evaluation of a Interactive Real Time Interface for BANS MOM monitoring system.
- IRTI tool is scalable and can display RT data
- When evaluation its security and vulnerability, IRTI has zero critical issue.
- IRTI code is small and user friendly. The total code is only about 700 lines.

Thank you for your listening
Backup Slides

---

MySQL configuration: my.ini

```php
key_buffer = 1G
max_allowed_packet = 1M
table_cache = 64
sort_buffer_size = 16M
read_buffer_length = 8K
read_buffer_size = 256K
read_rnd_buffer_size = 512K
innodb_buffer_pool_size = 16M
innodb_additional_mem_pool_size = 2M
innodb_log_file_size = 5M
innodb_log_buffer_size = 8M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50
```

---

Placement of electrodes

- **1 lead** refer to the tracing of the voltage difference between two of the electrodes.
- For example, "lead I" is the voltage between the right arm electrode and the left arm electrode. "lead II" is the voltage between the right arm and the foot.