THE ANALYSIS AND DEVELOPMENT OF AN APPLICATION SERVER

INESA BUZO
ELINDA KAJO MEÇE
CONTENT

• Objective
• Why?
• Software Requirements
• Design
  – Current architecture
  – Proposed one
• Implementation
  – Development environment & technologies
  – SW structure
• Testing
OBJECTIVE

DESIGN OF A FRAMEWORK FOR BUILDING AN APPLICATION SERVER THAT SUPPORTS A LARGE NUMBER OF REQUESTS BY COMBINING ASPECTS OF THREADS & EVENT-BASED PROGRAMMING MODULES.
WHY?

- Single Page
  - MVC
  - Data comes in portions (JSON format)
    - Lot of requests
- Client’s processing power greater nowadays
SOFTWARE REQUIREMENTS

• SW serves as a framework for building an application server
• SW supports programming in different phases
• Accepts a large number of requests
  – Little response time
• Mechanism that aid SW developers obtain highly concurrent, well-conditioned services
• Uses best of both worlds
  – Event-based
  – Thread-based
DESIGN (CURRENT ARCHITECTURES)

THREAD-BASED

EVENT-BASED
DESIGN (PROPOSED ARCHITECTURE)

PIPELINE

• SEDA architecture
  – Web Server
  – Design in phases
  – Non-blocking request
IMPLEMENTATION
DEVELOPMENT ENVIRONMENT & TECHNOLOGIES

• OS Linux Mint 64 bit
• C++
• STD C++ 14 library
• Threads Library
IMPLEMENTATION

SERVER
- Builds socket server
- Listen to port 30000 for a new request
- Inform the next phase for the new request

REQUEST
- Waits for the request from previous phase
- Reads and formats the request
- Inform the next phase for the new request

PROCESSING
- Processes the request
- Applies the required business logic
- Prepares data structure
TESTING
COMPARISON WITH APACHE

- 1000 requests

<table>
<thead>
<tr>
<th></th>
<th>Project’s web server</th>
<th>Apache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test’s total time</td>
<td>0.204 sec</td>
<td>0.208 sec</td>
</tr>
<tr>
<td>Requests/sec</td>
<td>4911</td>
<td>4810</td>
</tr>
<tr>
<td>Max time of request</td>
<td>20.359 ms</td>
<td>20.7 ms</td>
</tr>
<tr>
<td>Average time for all concurrents</td>
<td>0.204 ms</td>
<td>0.208 ms</td>
</tr>
</tbody>
</table>
**TESTING**

**COMPARISON WITH APACHE**

- 10000 requests

<table>
<thead>
<tr>
<th></th>
<th>Project's web server</th>
<th>Apache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test's total time</td>
<td>0.85 sec</td>
<td>1.3 sec</td>
</tr>
<tr>
<td>Requests/sec</td>
<td>11649</td>
<td>7634</td>
</tr>
<tr>
<td>Max time of request</td>
<td>85.8 ms</td>
<td>13 ms</td>
</tr>
<tr>
<td>Average time for all</td>
<td>0.086 ms</td>
<td>0.131 ms</td>
</tr>
<tr>
<td>concurrents</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Testing

## Comparison With Apache

- **100000 requests**

<table>
<thead>
<tr>
<th></th>
<th>Project's web server</th>
<th>Apache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test’s total time</td>
<td>11.8 sec</td>
<td>19.0 sec</td>
</tr>
<tr>
<td>Requests/sec</td>
<td>6001</td>
<td>5230</td>
</tr>
<tr>
<td>Max time of request</td>
<td>16.3 ms</td>
<td>191.1ms</td>
</tr>
<tr>
<td>Average time for all concurrent</td>
<td>0.164 ms</td>
<td>0.191ms</td>
</tr>
</tbody>
</table>
TESTING
TOTAL TIME OF TEST

Test total time

- **sweb**
- **apache**
TESTING REQUESTS PER SECOND

Sweb
Apache
CONCLUSIONS

• The architecture proposed is easily implemented as a SW using C++

• The framework for building an application server has successfully been developed

• The performance in comparison with current architectures in web servers has improved
  – Multi-threading and pipeline programming combined boost the performance of the application server.
THANK YOU!