Active network support for deployment of Java-based games on mobile platforms

Laurent Lefèvre, Aweni Saroukou

laurent.lefevre@inria.fr

Equipe INRIA RESO - LIP/ ENS Lyon

Support of Région Rhône Alpes

DFMA 2005 Conference, Besançon, Feb 7th
Roadmap

• Active networks
• Games deployment on mobile platforms
• Tamanoir Active Node architecture
• Active network support for games deployment
• Experimental results
• Conclusion and future works
**Introduction**

- Active networks
- From E2E to Hop by Hop
- In search of applications (multimedia, Grid…)
- Collaboration with SME : 3DDL
Active networks

**Smart Packets**
- Contain their own handling instructions
- Network is flexible

**Not-So-Smart Packets**
- All packets treated identically
- Network is rigid, relatively passive

• D. Tennehouse slides

⇒ Active nodes / routers
Different Approaches

Discrete Approach

- Active Services
- ANN
- ANTS
- Messenger

Integrated Approach

- Configurable Node
- CANES
- Active Bridging
- ANCORS
- PLAN
- Smart Packets
- Packet Programming

("out-of-band" code injection)

("in-band" code injection)
Introduction

• Active networks
• From E2E to Hop by Hop
• In search of (killer) applications (multimedia, Grid…)
• Collaboration with SME : 3DDL : development of applications for mobile platforms
Game design for mobile platforms

- Heterogeneous celluolars
- Multiple version of same application / Java
- Development time
- Java2 Micro Edition / J2ME
  - Specific APIs
  - MIDP : Mobile Information Device Profile -
  - Suite MIDDlets (JAD & JAR)
    - JAR : archive of classes of applications
    - JAD : description of JAR (name, version, vendor, profile…)
  - Over The Air provisionning : install, actualize, delete applications on mobiles
Games deployment on mobile nodes
Games deployment on mobile nodes

1: Get URL from WAP Gateway or HTTP Server

2: Pass to JAM

3: Retrieve and Read Java descriptor file (.jad)

4: Download JAR file (.jar)

5: Store JAR file

6: Launch KVM and Application
Without active network support
1a: Enrollment
1b: Downloading game
2: Creation of the temporary directories (user/mobile/game) and copy of the JAD and JAR files of requested game
3: Send SMS via mobile operator
4,8,12: Request for WML, JAD, JAR files
5,9: Request for JAD, JAR files
6,10: JAD, and JAR files
7,11: JAD, and JAR files
Active network support for deployment of mobile games: goals

- Reducing development time: one version of the application
- Modifying data/applications on the fly
- Limiting usage bandwidth and interactions between clients and servers
- Support deployment of games without adding too much latency
Tamanoir Active Network

- Tamanoir: a complete software environment to deploy active routers and services.
- Handle different streams and applications in parallel.
- TCP and UDP support
- Provides a fast and performant Execution Environment.
- Dedicated to high performance networking
• Active services can be deployed on Tamanoir node on various level.
  • programmable network interface card (Myrinet, Network Proc.)
  • kernel space (Netfilter)
  • user space (Java)
  • clustered architecture (LVS)
Tamanoir Architecture: adapted for heterogeneous services

- Resources consuming services: distributed storage, streams transcoding, on the fly compression, cryptography…

- Services deployment / linked with middleware: reliable multicast…

- Middle services: content based routing, QoS…

- Light network services: packet marking, QoS…
Architecture of Tamanoir Active Node
Deployment of services

- Service broker
- Node 2 node
Game deployment with active network support

1: Data base ORION

2: SMS + JAD url (Tamanoir)

3, 8, 11

4, 9, 12

Operator mobile network

Active node TAMANOIR
Sheltered by Operator mobile

5

6

7, 10, 13

Servlet

14: Mobile phone
JVM + WAP

1a, 1b

On the server, hierarchy of files standard, classified only by game type
Game deployment with active network support

1a: Registration, 1b: Download game, 1c: Registration profile
2: Send SMS via operator mobile + URL of the JAD file on Tamanoir
3, 8: Request for JAD file
4: Extraction of the user_agent + identifying user from the URL
5: Request for file Standard JAD + Sending of user_agent, User_ID, Game_ID to the servlet
6: Sending of JAD and JAR files (standard)
9, 10: Adaptation of the JAD content switch user ID and mobile type + Sending
11: Verifying JAD information, if OK request for JAR file
12: Adapt JAR file
13: Sending adapted JAR
14: Verifying + game installing
**Software**

- **ActiveWapS**: active service deployed in Tamanoir node
  - On the fly modification of JAD files
  - On the fly modification of JAR files (pruning, re-archiving…)

- **Servlet**
  - DB accesses (EJBs)
- Stream intercept
- Stream analysis
  - JAR
  - JAD
- JAD / JAR processing
  - Servlet request
  - JAD/JAR adapt
- Data send back to mobile platform
Experiments

- Local platform
- Gbits links
- Mobile networks emulation (with software link emulator NistNet)
- Experiments on various scenario
  - GSM (9,6 Kb/s)
  - GPRS (30 – 128 Kb/s)
  - UMTS (250 Kb/s – 1Mb/s)
- Values:
  - JAD : 0.5 KB
  - JAR : 45 KB
## Supported load

<table>
<thead>
<tr>
<th>N° test</th>
<th>Nb clients</th>
<th>Messages</th>
<th>Tps total</th>
<th>Nb Threads</th>
<th>NbMax threads</th>
<th>Tps/JAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>75</td>
<td>3</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
<td>3049</td>
<td>25</td>
<td>9</td>
<td>121</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td></td>
<td>8570</td>
<td>63</td>
<td>25</td>
<td>136</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>Tamanoir :</td>
<td>22652</td>
<td>126</td>
<td>50</td>
<td>181</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>Tamanoir :</td>
<td>31045</td>
<td>250</td>
<td>100</td>
<td>124</td>
</tr>
<tr>
<td>6</td>
<td>200</td>
<td>OutOfMemoryError</td>
<td>56550</td>
<td>469</td>
<td>200</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>500</td>
<td>OutOfMemoryError</td>
<td>76671</td>
<td>581</td>
<td>474</td>
<td>131</td>
</tr>
</tbody>
</table>

## Performances (with NistNet)

<table>
<thead>
<tr>
<th>N° test</th>
<th>Nb clients</th>
<th>Tps Tamanoir</th>
<th>Tps ancienne version</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0m0:460s</td>
<td>0m0:266s</td>
<td>1.39</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0m0:807s</td>
<td>0m0:687s</td>
<td>1.17</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0m0:989s</td>
<td>0m1:342s</td>
<td>0.73</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0m2:827s</td>
<td>0m2:302s</td>
<td>1.22</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0m6:086s</td>
<td>0m3:446s</td>
<td>1.76</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>0m37:764s</td>
<td>0m18:029s</td>
<td>2.09</td>
</tr>
<tr>
<td>7</td>
<td>75</td>
<td>1m4:660s</td>
<td>0m38:725s</td>
<td>1.36</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>1m50:188s</td>
<td>0m49:226s</td>
<td>2.23</td>
</tr>
<tr>
<td>9</td>
<td>150</td>
<td>java.lang.OutOfMemoryError</td>
<td>4m40.171s</td>
<td>2m15:503s</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>java.lang.OutOfMemoryError</td>
<td>5m15.988s</td>
<td>4m16:406s</td>
</tr>
</tbody>
</table>
Experiments

- “Local hero test”: all clients are on 100Mb network
- After 100 clients: latency increases
Experiments

- With NistNet emulation
- Clients are connected through a “perfect” GSM network
- OK for 100 clients: 45 s for game deployment
Ongoing work

- Continuing network emulation for other scenario
  - Software (Nistnet) and hardware (Gnet) network emulator
- Operational deployment during game campaign
- Cache support in active node

Future works

- Experiments on large scale platform (Grid5000)
- Scalability: cluster-based Tamanoir active node
- Adapting multimedia streams for mobile cellualrs
- More information:
  - http://www.ens-lyon.fr/LIP/RESO/Tamanoir