The SmartRoom Infrastructure: Service Runtime Reliability

Ivan Galov, Dmitry Korzun

Petrozavodsk State University
Department of Computer Science

This project is supported by grant KA179 of Karelia ENPI — joint program of the European Union, Russian Federation and the Republic of Finland

14th FRUCT conference
November 12, 2013, Helsinki, Finland
Smart Room at PetrSU

- holding conferences, meetings, and lectures
- personalized interaction with room participant
- participating using mobile devices
SmartRoom services

- conference services
- sensor services
- activity tracking services
- discussion services
- ...

SmartRoom is being deployed at PetrSU IT-park, 104 aud.
SIB is deployed either locally or on a remote machine

- Conference-service: conference runtime management
- Presentation-service: slide show of current speaker
- Agenda-service: visual activity agenda
- Future: Meeting-service, Lecture-service, ...
Infrastructure KP deployment options

1. the same server computer with the SIB
2. dedicated computers to attach specific devices
3. additional server computers (non-SIB)

infrastructure also includes KPs for the system administration
Services reliability

- Infrastructure element failure: crash of Smart-M3 processes or SmartRoom services
- Subscription query: it is not guaranteed that the application receives every subscription notification

Possible solutions:

- Subscription application-level control: KP performs additional timer-based checks for the service subscription
- Restart/Reconnection: A service restores its runtime and connection with the smart space
- Infrastructure: online services and content management
Restart/Reconnection

- Restart: application is shut down and launched again
- Reconnection: application is still running but connection to SIB is re-established

Problems when it is needed:

- SIB failure
  - data are saved (Virtuoso)
  - data are lost
- WiFi access point failure
- particular service failure
Online Services

- main goal: increase reliability of service provision
- operate permanently and continuously on server computers
- provide the service in 24/7 mode
- Upstart event-based init daemon handles the services
  - automatic respawning: whatever happens with the service, it will be relaunched again
  - services chaining: one service is automatically launched when another service is started

```
redsibd → sib-tcp → Smart Room initialization →
conference-service → other services
```
Online Services example

<table>
<thead>
<tr>
<th>Service</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>redsibd</td>
<td>start on startup</td>
</tr>
<tr>
<td>sib-tcp</td>
<td>start on started redsibd</td>
</tr>
<tr>
<td></td>
<td>stop on stopping redsibd</td>
</tr>
<tr>
<td>conference-service</td>
<td>start on started sib-tcp</td>
</tr>
<tr>
<td></td>
<td>stop on stopping sib-tcp</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

```
start on started sib-tcp
stop on stopping sib-tcp

env SERVICE=/usr/local/bin/conference-service

respawn
exec $SERVICE

post-start script
    python /usr/local/bin/register.py
end script
```
Content Management

- Smart space is a semantic hub: relating multi-source information without duplicating all the data in the smart space
- Content-service stores and shares information among others
- Every service or client can upload some data (video, audio, etc.)
- Content-service saves data and provides download links
Content-service links

- direct links
  - global — on external resources
  - local — on resources stored by the service

Problem: dynamic changing of content-service IP for local links

- relative links — relative path to resources stored on content-service

Content-service IP is stored and updated in service space
Conclusion

■ Mechanisms for service reliability
  ▶ Service delivery: subscription control, reconnection
  ▶ Infrastructure: online services, content management

■ Basic online services controlled by Upstart init daemon: SIB processes, Conference-service

■ Content-service is still in development stage

■ Open source code:
  http://sourceforge.net/projects/smartroom

■ Welcome to our demo on 13.11.2013

Thank you for attention!