Web Services, Cloud Computing, Mobile Cloud

Satish Srirama
satish.srirama@ut.ee
Outline

• Mobile Web Services
• Cloud Computing
• Mobile Cloud
• Push notification messages
• Announcement about team projects
Mobile Hosts in Enterprise Service Integration

• Web services (WS)
  – “Loosely coupled, standard-based reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols” [Sleeper, 2001]

• Enable enterprise integration

• Supported features
  – Interoperability, heterogeneity, platform neutral etc.

UDDI - Universal Description, Discovery and Integration
WSDL - Web Services Description Language
HyperText Transfer Protocol (HTTP)

HTTP Request

<table>
<thead>
<tr>
<th>method</th>
<th>URL</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/</td>
<td>HTTP/1.1</td>
</tr>
<tr>
<td>Host</td>
<td><a href="http://www.example.com">www.example.com</a></td>
<td></td>
</tr>
</tbody>
</table>

HTTP Response

<table>
<thead>
<tr>
<th>Status line</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP/1.1 200 OK</td>
</tr>
<tr>
<td>Date: Mon, 23 May 2005 22:38:34 GMT</td>
</tr>
<tr>
<td>Content-Type: text/html; charset=UTF-8</td>
</tr>
<tr>
<td>Content-Length: 138</td>
</tr>
<tr>
<td>Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)</td>
</tr>
<tr>
<td>ETag: &quot;3f80f-1b6-3e1cb02b&quot;</td>
</tr>
<tr>
<td>Accept-Ranges: bytes</td>
</tr>
<tr>
<td>Connection: close</td>
</tr>
</tbody>
</table>

Entity body

```html
<html>
<head>
<title>An Example Page</title>
</head>
<body>
<p>Hello World, this is a very simple HTML document.</p>
</body>
</html>
```
XML Web Service Server

SOAP Client Application | UDDI
---|---
Request/Response | Publish

SOAP Messaging | WSDL
---|---
Web Application (.NET, JSP, PHP, Node, Python, ObjC etc.)
HTTP Socket Server
TCP / UDP Socket Server
Host OS

Application Runtime Environment

23/10/2019 Satish Srirama 5/48
Request Message - SOAP vs. REST

List 2.1: SOAP-based request example

```xml
GET / HTTP/1.1
Host: www.example.com
Content-Type: application/soap+xml; charset=UTF-8
Content-Length: (length)

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope" >
  <env:Header>
    <!-- Header information here -->
  </env:Header>
  <env:Body>
    <m:getProductDetail xmlns:m="http://www.example.com/"
      <productID>21</productID>
    </m:getProductDetail>
  </env:Body>
</env:Envelope>
```

**Drawback:** SOAP is heavyweight

List 2.2: REST-based request example

```bash
GET /product/21
Host: www.example.com
Content-Type: application/x-www-form-urlencoded
```

SOAP vs. REST

Reference:
Chang, C. (2013), Service-Oriented Mobile Social Network in Proximity, PhD Dissertation, Monash University, Australia.
Mobile Web Services

- Mobile web services (MWS)
  - Weather, search, maps etc.
- It is also possible to provide services from mobiles
- Mobile Web Service Provisioning
  - Mobile Hosts
- Applications
  - Healthcare
    - Real-time & on-demand sensing
    - Connecting wearable devices
  - Internet of Vehicles
  - Crowd Sensing

[Srirama et al, ICIW 2006; Srirama, 2008]
Limitations with Mobiles

• Longer battery life
  – Battery lasts only for 1-2 hours for continuous computing
  – Modern Android OS tries to limit the background services to save battery

• Same quality of experience as on desktops
  – Weaker CPU and memory
  – Storage capacity

• Still it is a good idea to take the support of external resources
  – For building resource intensive mobile applications
  – Access them as Mobile Web Services
  – Brings in the scope for cloud computing
What is Cloud Computing?

• Computing as a utility
  – Utility services e.g. water, electricity, gas etc
  – Consumers pay based on their usage

1969 – Leonard Kleinrock, ARPANET project
  “As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of ‘computer utilities’, which, like present electric and telephone utilities, will service individual homes and offices across the country”
Clouds - Why Now (not then)?

• Experience with very large datacenters
  – Unprecedented economies of scale
  – Transfer of risk

• Technology factors
  – Pervasive broadband Internet
  – Maturity in Virtualization Technology

• Business factors
  – Minimal capital expenditure
  – Pay-as-you-go billing model
Virtualization

- Virtualization techniques are the basis of the cloud computing
- Virtualization technologies partition hardware and thus provide flexible and scalable computing platforms
- Virtual machine techniques
  - VMware and Xen
  - OpenNebula
  - Amazon EC2
- Recent updates - containers
- Dalvik and ART are virtual machines

Virtualized Stack
Cloud Computing - Characteristics

- Illusion of infinite resources
- No up-front cost
- Fine-grained billing (e.g. hourly)

- **Gartner**: “Cloud computing is a style of computing where massively scalable IT-related capabilities are provided ‘as a service’ across the Internet to multiple external customers”
Cloud Computing - Services

- **Software as a Service – SaaS**
  - A way to access applications hosted on the web through your web browser

- **Platform as a Service – PaaS**
  - Provides a computing platform and a solution stack (e.g. LAMP) as a service

- **Infrastructure as a Service – IaaS**
  - Use of commodity computers, distributed across Internet, to perform parallel processing, distributed storage, indexing and mining of data
  - Virtualization

- **SaaS**
  - Facebook, Flikr, Myspace.com, Google maps API, Gmail

- **PaaS**
  - Google App Engine, Force.com, Hadoop, Azure, Heroku, etc

- **IaaS**
  - Amazon EC2, Rackspace, GoGrid, SciCloud, etc.
Cloud Computing - Themes

• Massively scalable
• On-demand & dynamic
• Only use what you need - Elastic
  – No upfront commitments, use on short term basis
• Accessible via Internet, location independent
• Transparent
  – Complexity concealed from users, virtualized, abstracted
• Service oriented
  – Easy to use SLAs

SLA – Service Level Agreement
Cloud Models

- **Internal (private) cloud**
  - Cloud within an organization

- **Community cloud**
  - Cloud infrastructure jointly owned by several organizations

- **Public cloud**
  - Cloud infrastructure owned by an organization, provided to general public as service

- **Hybrid cloud**
  - Composition of two or more cloud models
Implications of Clouds

- Startups and prototyping
  - Minimize infrastructure risk
  - Lower cost of entry
- Batch jobs
- One-off tasks
  - Washington post, NY Times
- Cost associatively for scientific applications
- Research at scale
- Application software:
  - Cloud & client parts, disconnection tolerance
Economics of Cloud Users

- Many cloud applications have cyclical demand curves
  - Daily, weekly, monthly, ...
- Pay by use instead of provisioning for peak
- Risk of over-provisioning: underutilization
  - Huge sunk cost in infrastructure
- Heavy penalty for under-provisioning
Optimal resource provisioning for auto-scaling enterprise applications

- **Auto-Scaling**
  - Scaling policy -> When to Scale
  - Resource provisioning policy -> How to scale

- **Cloud providers offer various instance types with different processing power and price**
  - Can it be exploited in deciding the resource provisioning policy?
  - Makes the policy to be aware of current deployment configuration

- **Another challenge:** Cloud providers charge the resource usage for fixed time periods
  - E.g. Hourly prices of Amazon cloud

- **Developed an LP based optimization model which considers both the issues** [Srirama and Ostovar, CloudCom 2014]
Scaling enterprise application with the optimization model

Incoming load and scaling curves of Optimization model

Instance type usage curves of Optimization model

Scaling with Amazon AutoScale

[Srirama and Ostovar, CloudCom 2014]
Economics of Cloud Providers

• Cloud Computing providers bring a shift from high reliability/availability servers to commodity servers
  – At least one failure per day in large datacenter
• Why?
  – Significant economic incentives
    • much lower per-server cost
• Caveat: User software has to adapt to failures
  – Very hard problem!
• Solution: Replicate data and computation
  – MapReduce & Distributed File System
Cloud Computing Progress

Armando Fox, 2010
A Manifesto for Future Generation Cloud Computing: Research Directions for the Next Decade

[Buyya and Srirama et al, ACM CSUR 2019]
Mobile Cloud Applications

• Bring the cloud infrastructure to the proximity of the mobile user
• Mobile has significant advantage by going cloud-aware
  – Increased data storage capacity
  – Availability of unlimited processing power
  – PC-like functionality for mobile applications
  – Extended battery life (energy efficiency)
Demo

• Demonstrate the trade-off of using Mobile Cloud
  – Image processing application
Hybrid/heterogeneous Mobile Cloud

[Zhou et al, TSC 2017]
Task Delegation: Mobile Cloud mash-up applications

• Mobiles may have to take advantage of multiple clouds
• Follows traditional SOA (Service Oriented Architecture) model to invoke services
• Typical scenarios
  – Process intensive services
    • Face recognition, sensor mining etc.
• Critical challenges
  – Cloud interoperability is a huge trouble
  – Mobiles need to possess multiple APIs
  – Unavailability of standards and mobile platform specific API
Mobile Cloud Middleware

[Warren et al, IEEE PC 2014]

[Replaced by Firebase cloud messaging]

[Flores et al, MoMM 2011; Flores and Srirama, JSS 2014]
MCM – enables

• Interoperability between different Cloud Services (IaaS, SaaS, PaaS) and Providers (Amazon, Eucalyptus, etc)
• Provides an abstraction layer on top of API
• Composition of different Cloud Services
• Asynchronous communication between the device and MCM
• Means to parallelize the tasks and take advantage of Cloud’s intrinsic characteristics
CroudSTag – Scenario

- CroudSTag takes the pictures/videos from the cloud and tries to recognize people
  - Pictures/Videos are actually taken by the phone
  - Processes the videos
  - Recognizes people using facial recognition technologies
- Reports the user a list of people recognized in the pictures
- The user decides whether to add them or not to the social group
- The people selected by the user receive a message in facebook inviting them to join the social group
CroudSTag [Srirama et al, PCS 2011; SOCA 2012]

- Cloud services used
  - Media storage on Amazon S3
  - Processing videos on Elastic MapReduce
  - face.com to recognize people on facebook
  - Starting social group on facebook
Other applications

• **Zompopo** [Srirama et al, NGMAST 2011]
  – Intelligent calendar, by mining accelerometer sensor data

• **Bakabs** [Paniagua et al, iiWAS-2011]
  – Managing the Cloud resources from mobile

• **Sensor data analysis**
  – Human activity recognition
  – Context aware gaming
  – **MapReduce based sensor data analysis** [Paniagua et al, MobiWIS 2012]
Checking for updates

• Polling

- Polling consumes lot of battery of mobile
- Heavy load on the resources of the server
- Strains the network
Firebase Cloud Messaging

- Free service that allows you to send data from your servers to your users and vice versa
- FCM replaced Google Cloud Messaging (GCM)
  - GCM is now deprecated
- Firebase by platform
  - Goes beyond just messaging
  - Firebase provides the tools to develop high-quality apps, grow your user base, and earn more money
  - Provides services for Authentication, Realtime Database, Performance Monitoring, Cloud Storage, Analytics, AdWords etc.
  - [https://firebase.google.com/docs/](https://firebase.google.com/docs/)
Firebase Cloud Messaging (FCM) - continued

• A cross-platform messaging solution
  – Can now send messages to Android, iOS, etc.
• Handles queuing of messages and delivery to the target application
  – Application does not need to be running in the background for receiving data messages
  – The system will wake up the Android application via Intent broadcast when the message arrives
• FCM also supports sending messages to multiple phones, to devices subscribed to topics etc.
• FCM also supports sending messages to the application server from the mobile
  – Using the Extensible Messaging and Presence Protocol (XMPP)
Types of messages

• Notification messages
  – Handled by the FCM SDK automatically
  – Have a predefined set of user-visible keys
  – Optional data payload of custom key-value pairs
  – When App is in the background,
    • Apps receive the notification payload in the notification tray
    • Apps only handle the data payload when the user taps on the notification
  – When App is in the foreground,
    • App receives a message object with both payloads available

• Data messages
  – Handled by the client app
  – Data messages have only custom key-value pairs

```json
{
  "message":{
    "token":"bk3RNwTe3H0:C12k_HHwgiPoDKClizvDMExUdFQ3P1...",
    "notification":{
      "title":"Portugal vs. Denmark",
      "body":"great match!
    }
  }
}

{
  "message":{
    "token":"bk3RNwTe3H0:C12k_HHwgiPoDKClizvDMExUdFQ3P1...",
    "data":{
      "nick":"Mario",
      "body":"great match!",
      "room":"PortugalVSDenmark"
    }
  }
}
```
Working with FCM – with Registration ID

1. Request for Registration
2. Reply with Registration ID
3. Send Registration ID
4. Send message \{RegId, msg\}
5. Send Message

https://firebase.google.com/docs/cloud-messaging/
Working with FCM – Subscribe to a topic

1. Subscribe to a topic
2. Confirmation
3. Send message \{topic, msg\}
4. Send Message

https://firebase.google.com/docs/cloud-messaging/
Mobile cloud - Binding models

Task Delegation

Code Offloading

[Flores and Srirama, JSS 2014]

[Flores et al, IEEE Communications Mag 2015; Zhou et al, TSC 2017]
Code Offloading - Major Components

• Similar to RMI, RPC etc.
• Mobile
  – Code profiler
  – System profilers
  – Decision engine
• Cloud based surrogate platform

RPC - Remote procedure call
RMI - Remote Method Invocation
Some of the well known frameworks

• MAUI
  – Manual annotations [Cuervo et al., 2010]

• CloneCloud
  – Code profilers & Automated process [Chun et al., 2011]

• ThinkAir
  – Manual annotations and scalability [Kosta et al, INFOCOM 2012]

• mCloud
  – Extended from ThinkAir to incorporate incentive models [Zhou et al, TSC 2017; JSS 2019]
Conclusion

• Mobile Web Services allow interoperability
• Cloud computing offers interesting opportunities in different domains
• Mobile applications benefit a lot by pushing some of the resource intensive tasks to cloud
• Checked Mobile Cloud binding models
• Push notification mechanisms
This week in lab

• Invoking web services from mobile
  – Mobile web service client
• Firebase cloud messaging
Team projects

• Team projects and their requirements will be advertised on 20.11
  – Open ended projects
  – With features learnt in the course e.g. databases, maps, background services etc.

• 3 person projects
  – Already start having an idea with whom you want to team up
  – <3 should have strong reason
    • You will already loose some points at teamwork
Next Lecture

• Working with Android inbuilt sensors and Sensor Framework
• Location based services
References