A hodgepodge of distributed systems: web-based, cloud, IoT, and others

IN5020 autumn 2020
lecturer: Roman Vitenberg

Two main flavors

- Browser-server WWW application
  - Geared towards human interaction
  - Not suitable for automation
    - Automatic restocking from amazon.com
    - Sniping in eBay
- Web services middleware
  - Generic extension of the WWW application
  - Web servers announce and provide services
  - Web server can be a client of another service
Communicating entities

- Client
- Proxy
- Web Server

Web proxy functions

- Protocol translation and conversion
  - Not needed for modern browsers/clients
- Filtering requests and responses
- Logging
- Compression
- Caching
Client-side and proxy caches

- Cache update protocols
  - Pull with if-modified-since GET HTTP header
  - Lease-based propagation
- Not as effective for dynamic content
- Cache replacement policies
  - LRU is commonly used and it performs well
  - A number of specialised policies, e.g., Greedy-Dual

Hierarchical proxy caching

- Commonly deployed throughout the world
- Reduces latency and traffic for popular documents
- Increases latency for other documents
- Higher-level proxies require a lot of storage space

1. Look in local cache
2. Ask next level proxy
3. Look in local cache
4. Forward request to Web server
Cooperative proxy caching

- Deployed on a per-organisation basis
- Can be combined with hierarchical caches

Simple server-side backend

- Several possibilities for request dispatching
  - Round-robin DNS
  - Content-aware dispatcher inspecting HTTP requests
  - TCP-level switch
Content-distribution network (CDN)

- Placement of data/object replicas
  - See the lecture on replication...
- A number of evaluation metrics
  - Latency (real-time and the number of hops)
  - Bandwidth (available and network usage)
  - Financial
- Consistency enforcement
  - See the lecture on replication...
- Redirection of client requests

Redirection of client requests in an Akamai CDN

1. Get base document
2. Document with refs to embedded documents
3. DNS lookups
4.
5. Get embedded documents
6. Get embedded documents (if not already cached)
7. Embedded documents

Regular DNS system

CDN DNS server

Cache

Client

CDN Server

Web Server
Cloud Computing

- **Definition:**
  - Clouds are pay-per-use services offering on-demand compute, network, and storage capacity.

- **Three classes of cloud; viewed as a layered architecture:**
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)

### Types of clouds – (1)

- **IaaS**
  - offering virtualized resources (computation, storage, and communication) on-demand
  - several choices of OS and a customized software stack
  - e.g., Amazon EC2: offering VMs with a software stack customizable like an ordinary server

- **PaaS**
  - offering an environment to create and deploy applications
  - no need to know low-level technical config., e.g., number of processors or amount of memory
  - offering multiple programming models and specialized services, e.g., data access and authentication
  - e.g., Google AppEngine: for developing and hosting Web apps written in Python or Java
**Types of clouds – (2)**

- **SaaS**
  - applications reside on the top of the cloud stack
  - accessed by end users through Web portals
  - desktop apps such as word processing as a service in the Web
  - e.g., Salesforce.com: offering business productivity apps (CRM)

---

**Five essential characteristics of clouds**

- **On-Demand Self-Service**
  - consumers expect on-demand, nearly instant and unilateral access to resource

- **Measured Service**
  - measuring the usage of cloud resources for each individual consumer through its metering capabilities

- **Elasticity**
  - users expect clouds to rapidly provide resources in any quantity at any time

- **Resource Pooling**
  - physical and virtual resources dynamically assigned and reassigned according to consumer demand

- **Broad Network Access**
  - Supporting a range of devices, transport protocols, interfaces, and security technologies for accessing cloud services
Internet of Things

- **Definition:**
  - the worldwide network of interconnected objects uniquely addressable based on standard communication protocols

- IoT Enabling Technologies: RFID and WSNs

---

Characteristics of IoT

- **Interoperable Communication Protocols**
  - IoT devices
  - support a number of interoperable communication protocols
  - communicate with other devices

- **Unique Identity**
  - Each IoT device has a unique identity and a unique identifier (such as an IP address or a URI)

- **Integrated into Information Network**
  - IoT devices: integrated into the information network
  - to communicate and exchange data with other devices and systems

- **Self-Adapting**
  - the capability to dynamically adapt with the changing contexts: operating conditions, user's context, or sensed environment
Physical Design of IoT – Things in IoT

- The “Things” in IoT usually refers to IoT devices
  - unique identities
  - can perform remote sensing, actuating and monitoring
  - can exchange data with other connected devices and applications (directly or indirectly) or collect data from other devices
  - process the data either locally or send the data to centralized server
  - Data analytics systems: resulting in useful information to guide further actions locally or remotely
- Varied in types
  - Wearable Sensors, Smart watches, LED lights, automobiles, and industrial machines

IoT Protocols

<table>
<thead>
<tr>
<th>Layer</th>
<th>Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Layer</td>
<td>HTTP, CoAP, WebSockets, MQTT, XMPP, DDS, AMQP</td>
</tr>
<tr>
<td>Transport Layer</td>
<td>TCP, UCP</td>
</tr>
<tr>
<td>Network Layer</td>
<td>IPv6, IPv4, 6LowPAN</td>
</tr>
<tr>
<td>Link Layer</td>
<td>802.3 - Ethernet, 802.16 - WiMax, 802.15.4 - LR-WPAN, 2G/3G/LTE - Cellular</td>
</tr>
</tbody>
</table>
Logical Design of IoT

- **IoT Functional Blocks**
  - provide the system the capabilities for identification, sensing, actuation, communication, and management
  - Device, Communication, Services, Management, Security, Application

- **IoT Communication Models**
  - Request–Response, Publish/Subscribe, Push-Pull, Exclusive Pair

- **IoT communication APIs**
  - REST–based Communication APIs
  - WebSocket–based Communication APIs

Main Components of IoT

- **Device**: allowing identification, remote sensing, actuating and remote monitoring capabilities
- **Resource**: software components on the IoT device for accessing, processing, and storing sensor information, or controlling actuators connected to the device
- **Controller Service**:
  - sends data from the device to the web service
  - receives commands from the application for controlling the device
- **Database**: either local or in the cloud to store IoT-generated data
- **Web Service**: serves as a link between the IoT device, application, database and analysis components
- **Analysis Component**: analyzing IoT data and generating results
- **Application**: an interface between users and the IoT system to control and monitor its various aspects
**IoT Applications**

- Home
- **Cities**
- Environment
- Energy Systems
- Retail
- Logistics
- Industry
- Agriculture
- **Health & Lifestyle**

**IoT Applications (1)**

- Smart Cities
  - Smart Parking
  - Smart Lighting for Road
  - Smart Road
  - Structural Health Monitoring
  - Surveillance
  - Emergency Response
IoT Applications (2)

- In smart health & lifestyle:
  - Health & Fitness Monitoring
  - Wearable Electronics

[Image: Illustration of a hand holding a phone with a heart rate monitor and a heart on the screen]