Cloud Marketplace and Broker – Revision Notes

- **CloudCmp**
  - **Definition**: Tool to compare cloud providers based on **QoS metrics**.
  - **Use**: Helps in **cloud provider selection**.
- **Fuzzy Provider Selection Mechanism**
  - Uses **fuzzy logic** to handle **imprecise user requirements**.
  - Measures **user satisfaction** with providers.
- **Trust-based Framework**
  - Considers **trustworthiness** and **competence** of providers.

Migration Decider

- Uses a **fuzzy inference engine**.
- **Input**: Fi (fuzzy parameters).
- **Output**: **Degree of SLA Satisfaction**.
- **Decision Rule**: If satisfaction < threshold ⇒ **Migrate**.

Case Study: IaaS & SaaS Marketplace

- **10 providers**, **500 requests**, **1-year simulation**.
- **QoS degradation** follows a **Gaussian distribution**.
- Compared to **conventional (minimum-cost) broker**.

Mobile Cloud Computing (MCC)

- **Definition**

Combination of **cloud computing + mobile computing + wireless networks** to provide **rich computing to mobile users**.

- **Why MCC?**

1. **Speed & Flexibility**
2. **Shared Resources**
3. **Integrated Data**
4. **Multi-device continuity**

Key Components of MCC

- **Profiler**: Monitors app execution (time, power, traffic).
- **Solver**: Decides cloud/mobile task execution.
- **Synchronizer**: Combines split execution results.

Requirements of MCC

- Simple APIs
- Web Interface
- Internet access to cloud apps

Advantages of MCC

1. Extended **battery life**
2. Increased **storage & processing**
3. Better **reliability**
4. **Scalability**, **Multi-tenancy**
5. **Ease of Integration**

Challenges in MCC

Mobile Communication

- Low bandwidth
- Service availability
- Heterogeneity in networks

Computing/Offloading

- Decision on **what/when to offload**
- Requires **optimization & dynamic partitioning**

Code Offloading & Cloudlet

- **Cloudlet**
  - A **trusted, resource-rich** computer near mobile devices.
  - Reduces **latency** and **battery usage**.
- **Offloading Decision Formula:**
  - **Offload** if D/B < C/M and F is high
    (D = data size, B = bandwidth, C = compute cost, M = mobile speed, F = CPU cycles)

Computation Offloading Approaches

1. **Static**: ILP solver at app start
2. **Dynamic**: Adjusts based on run-time conditions
3. **Heuristic & Profiling**-based method

Task Partitioning Problem in MCC

- **Input**: Call graph with energy stats
- **Output**: Two sets (mobile/cloud execution)
- **Goals**:
  1. Minimize energy
  2. Meet execution time
  3. Handle constraints (must-run methods, cost)

📐 **Evaluating MCC Performance**

- **Energy consumption**
- **Time to completion**
- **Monetary cost**
- **Security**

📱 **Applications of MCC**

- **Mobile healthcare**
- **Mobile gaming**
- **M-commerce**
- **M-learning**
- **Assistive tech**

🎯 **MuSIC Framework (Mobility-Aware Service Allocation)**

- Handles **user mobility** in task decomposition.
- Aims to optimize:
  - **Delay**
  - **Power**
  - **Cost**
- Uses **LTW (Location-Time Workflow)** to model mobility.

🚗 **Case Study: Context-Aware Dynamic Parking**

- Problem: **Insufficient parking spaces**
- Uses: **WSNs + Cloud**
- Services:
  - Traditional garages
  - Dynamic street-side
  - Smart reservation via smartphones
- **Traffic cloud** updates & provides live parking info.