Software Architecture — Project Assignment

Dr. Lei Yang
Email: sely@scut.edu.cn
Tel.: 15622183718

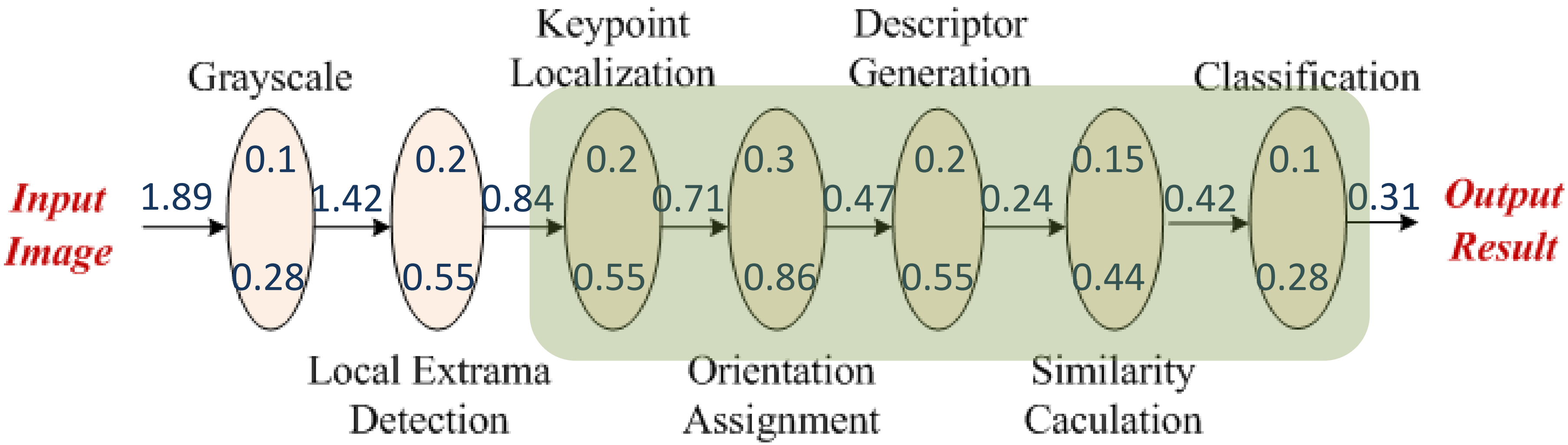
Background

- **Performance**
  - **Scheduling**
- **Allocation Structure**
  - **Module to file**
  - **Module to hardware resource**
  - **Module to human resources**

Computation Partitioning

- **Computation partitioning decomposes application software into a set of modules, and decides which modules are executed locally, and which parts are offloaded onto the remote server or cloud**

1. Computation Partitioning
a simple example

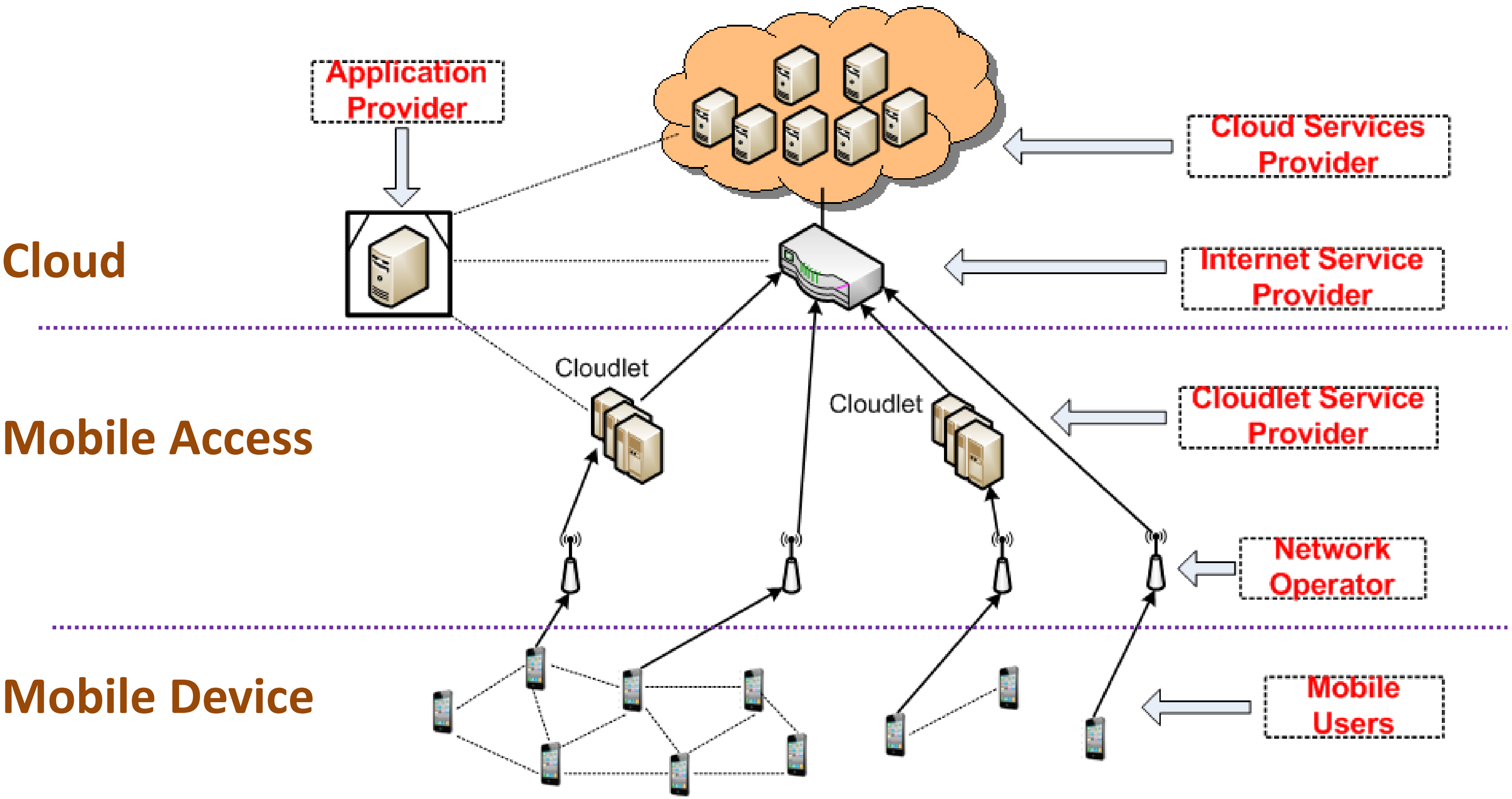


Optimal Partitioning 0.28 + 0.55 + 0.84 + $\underline{0.2 + 0.3 + 0.2 + 0.15 + 0.1}$ + 0.31 = 2.93

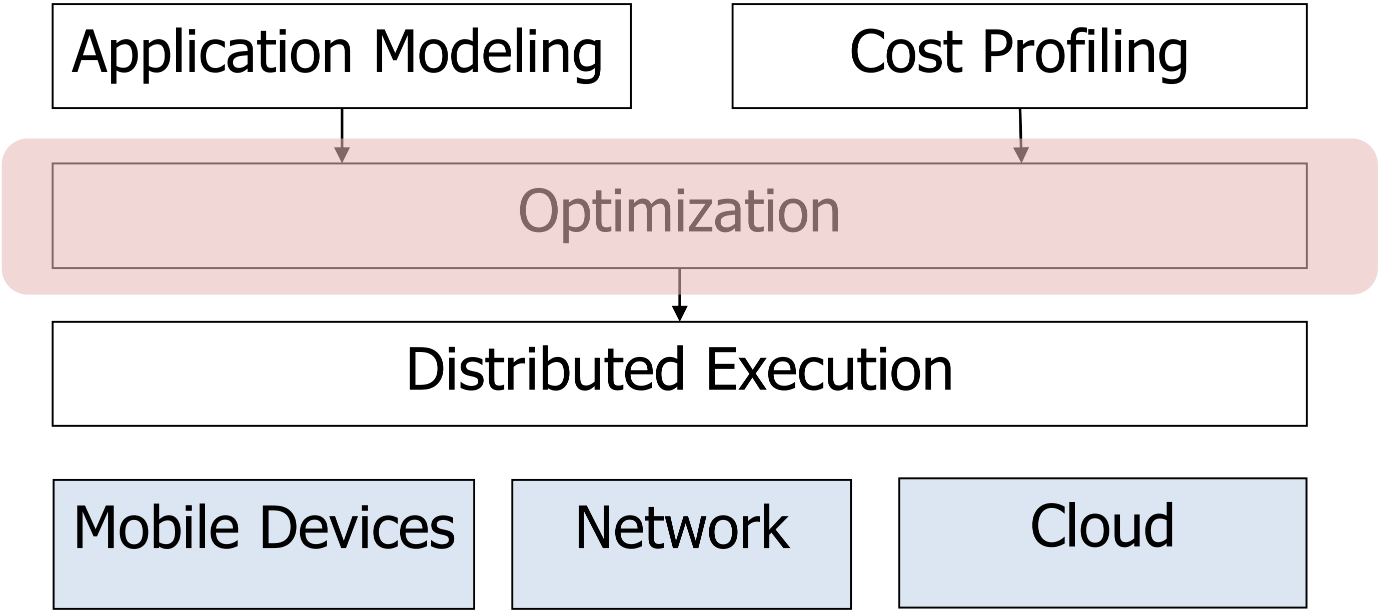
Local Execution: 0.28 + 0.55 + 0.55 + 0.86 + 0.55 + 0.44 + 0.28 = 3.51

Remote Execution: 1.89 + $\underline{0.1 + 0.2 + 0.2 + 0.3 + 0.2 + 0.15 + 0.1}$ + 0.31 = 3.45

MCC System Model



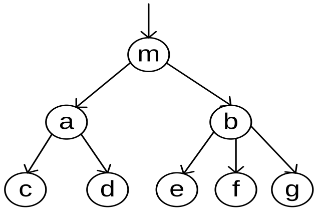
Issues in Computation Partitioning

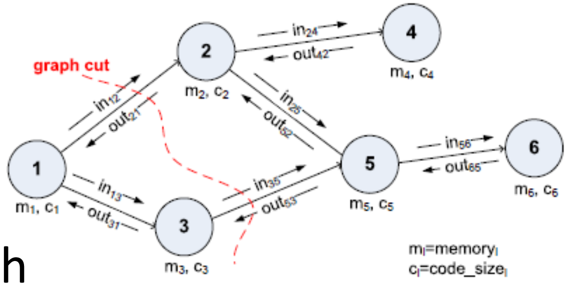


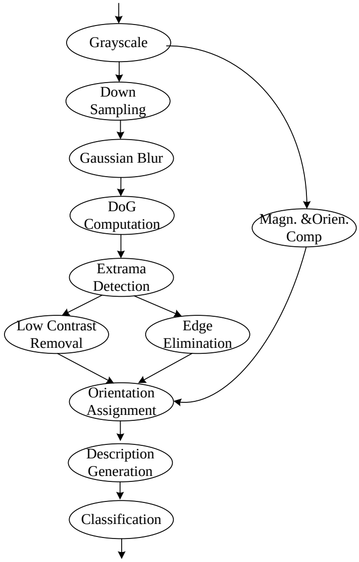
Application Modeling

- How to represent the structure of application: 3 major approaches
  - Procedure calls
    - Application: a set of procedures
    - Function-centric & synchronous
  - Service invocation
    - Application: a service invocation graph
    - Message-centric & asynchronous
  - Dataflow
    - Application: a directed acyclic graph
    - Edge represents the flow of data; Node is the processing function onto the data

```
void m()
{
    a() {
        c();
        d();
    };
    b() {
        e();
        f();
        g();
    };
}
```







Cost Modeling

- Estimate the execution cost of each component in the application and weigh the cost of offloading against the potential gain
  - Execution cost can be measured by one or the weighted summation of the following metrics:
    - execution time (local and remote)
    - energy consumption
    - data transferred over the network
- Profiling is an approach to collecting and estimating the cost of application components
  - Prediction-based profiling
  - Model-based profiling

Optimization

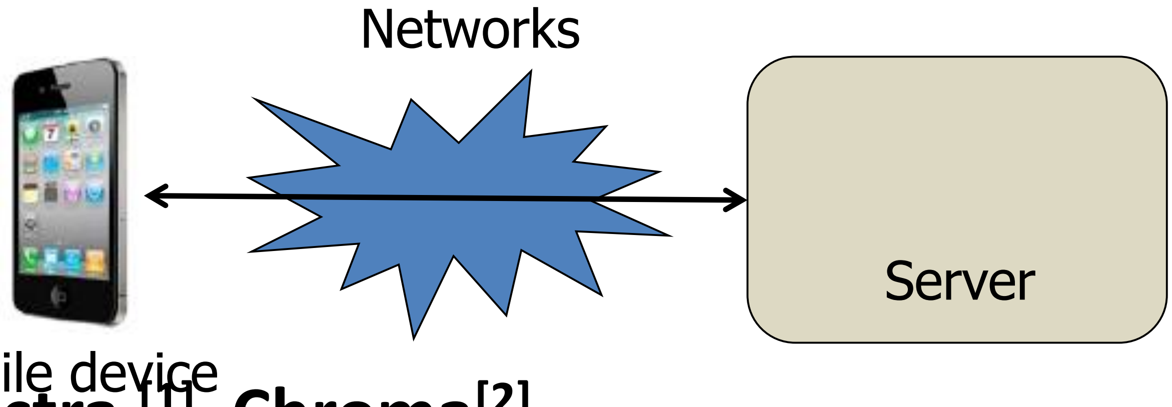
- Obtain optimal partition of the computation - can be solved either *online* or *offline*
  - Online optimization solves the optimization on the fly for each execution of an application.
  - Offline optimization calculates the optimal partitions under different device and network status in offline phase
    - Search the most matched partition given the measurements of the device and network status
    - Avoid the overhead of solving optimization, but need abundant offline test cases
- Optimization can be solved at the *mobile side* or *cloud side*

Distributed Execution

- Execute the partitioned computation components over mobile devices and cloud fabric.
- **Three execution approaches**
  - Client server communication method
  - Virtual machine migration

Client Server Communication

- RPC and RMI
- Require pre-installation on servers, and prone to network disconnection

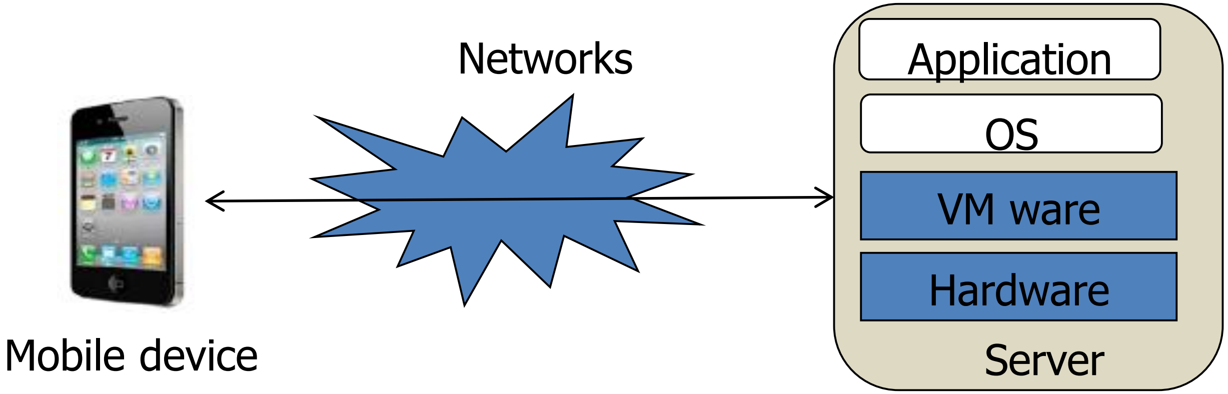


- **e.g., Spectra [1], Chroma[2]**

[1] J.Flinn. Balancing performance, energy, and quality in pervasive computing. ICDCS'02
[2] R. Balan. Tactics-based remote execution for mobile computing. Mobisys'03

Virtual Machine Migration

- Do not need pre-installation on clouds
- Code changes are not required for execution on clouds
- Using VM Migration is heavyweight
  - **e.g., MAUI [1], Cloudlet [2] , CloneCloud[3], ThinkAir [4]**



[1] Maui: making smartphones last longer with code offload. MobiSys'10
[2] Clonecloud: elastic execution between mobile device and cloud.EuroSys'11
[3] The case for VM-basedcloudlets in mobile computing, IEEE Pervasive Computing 2009.
[4] ThinkAir: Dynamic resource allocation and parallel execution in cloud for mobile code offloading. Infocom'12

References

1. Optimizing the performance of dataflow applications in throughput
   - “A framework for partitioning and execution of data stream applications in mobile cloud computing”. *IEEE SIGMETRICS PER 2013*.
2. Optimizing the performance of workflow application in execution time
   - “Run Time Application Repartitioning in Dynamic Mobile Cloud Environments”, *IEEE Trans. On Cloud Computing, 2016*

Project Assignment

- **Part A** Select one mobile application to implement and test its performance on the mobile device. The selected application should be compute-intensive and latency sensitive. Examples include but are not limited to:
  - hand gesture recognition,
  - face recognition,
  - image based object recognition,
  - augmented reality,
  - OCR and etc.

Project Assignment

- **Part B** Please analyze the module structure of the application, and try to partition the modules between the mobile device and a remote server (or cloud). Test the performance of the application under various partitioning, and show via experiments what are the factors and how do they impact the performance of application.
- **Part C** Based on the test results above, try to develop a system/component that supports the dynamic partitioning of the application in the run time.

Score Criterias

- Required to finish at least part A and B.
  - Part A: 60 points; Part B: 90 points; Part C: 100 points.
- Final deliverables for scoring
  - Final Report (60%)
  - Demonstration (40%)

Final Report

- Content of the final report should include:
  - **Title**
  - **Abstract**
  - **Introduction**
  - **[Main Body]**: application; performance metric and measurement; computation partitioning; system design, architecture;
  - **Experiments and results**: state the experiment purposes, environment settings, and results with figures or tables
  - **Conclusions**
  - **References**

Final Report

- ***The module structure*** of the application should be included in your report
- ***Measure the application performance*** under as many settings as possible, i.e., different partitioning, network connections (WiFi or 4G), bandwidth, mobile devices, or input data
- Beyond the experiment results, ***what are the insights*** you want to provide
- If Part C is finished, the ***component-and-connector structure*** the system is required

Demonstrations

- Each group has ***10 minutes*** to demonstrate the system and results
- Design the demonstration procedures, and make sure it ***proceeds smoothly and logically***
  - A checklist indicating what you will demonstrate is required
- Debugging the demonstrations at least ***10 times*** in advance, and make sure ***no failures occur***

Time Schedule

- Send group information to sely@scut.edu.cn on **25/Oct/2018**
  - 所有小组成员姓名和学号，组长的**Email**和手机
- Each group submits a *confirmation report to* my email sely@scut.edu.cn on **8/Nov/2017**. The report shows what application you select to implement, and the module structure of the application source codes.
- Each group submits a *mid-term progress report* by **29/Nov/2017** via emails
- Each group emails the draft of *project report and source code* to me before **25/Dec/2017**.
- *Demonstration* is tentatively arranged on **27/Dec/2017**.
- The *final version of the report* should be submitted within one week after the demonstration **(3/Jan/2017).**