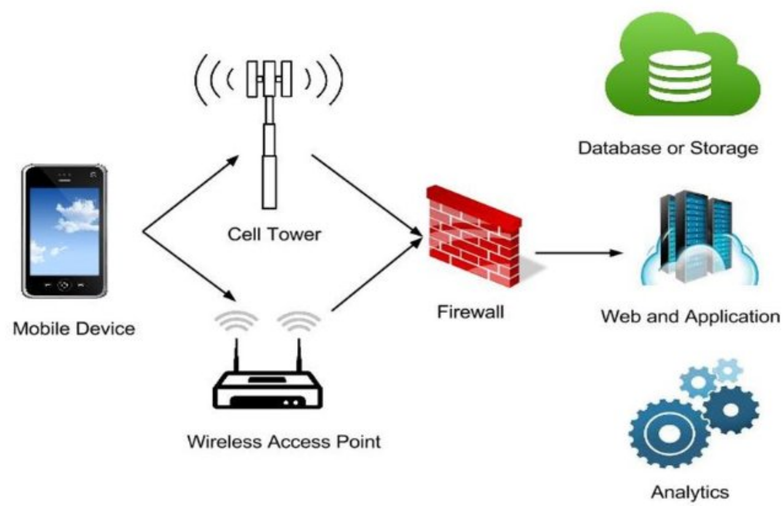
Maximizing the User's Benefit in Mobile Cloud Computing

Ning Wang and Jie Wu
Temple University

Mobile Cloud Computing

- **Concept of Mobile Cloud Computing (MCC):**
  - Offload the computation complexity task from mobile devices to the cloud
  - Combination of cloud computing, mobile computing, and wireless networks to bring rich computational resources to mobile users



Mobile Cloud Computing

- **Different computation abilities**
  - Cloud (e.g., Amazon EC2, Azure):
    - ❑ Rich computation resource --- short processing delay
    - ❑ Omitted in most scenarios
  - Mobile devices:
    - ❑ Limited computation resource --- long processing delay
      - ❑ E.g., natural language processing, face recognition

- **Cost of cloud computing**
  - extra offloading/transmission delay
    - Cellular networks (low bandwidth, high energy consumption)
    - Wi-Fi (high bandwidth, low energy consumption)

Mobile Cloud Computing

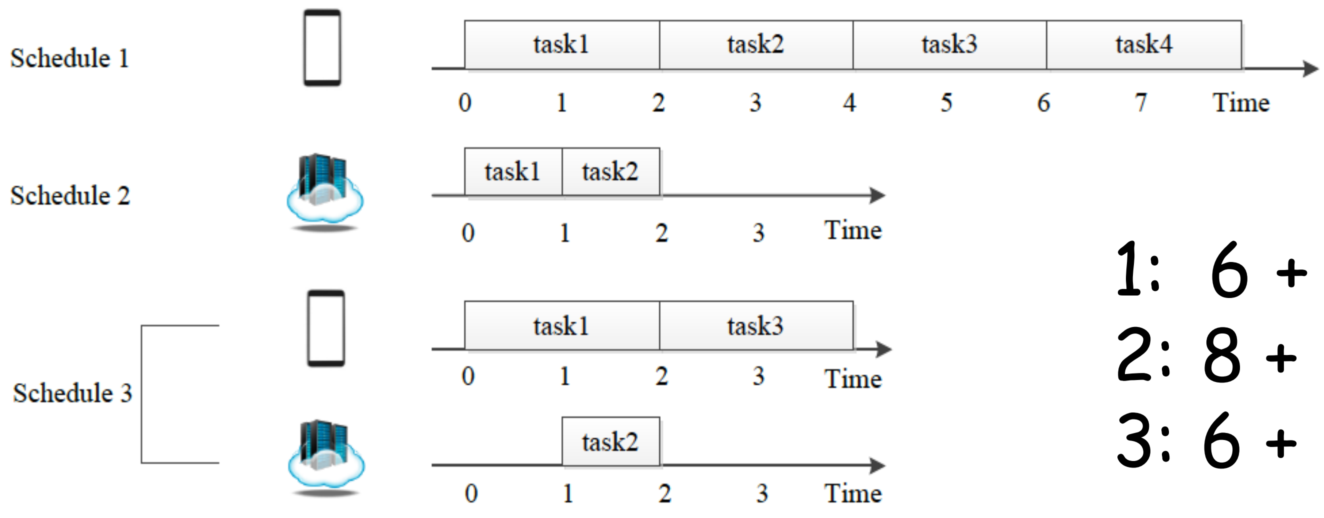
- **Task finishing time**
  - Task utility decay (reward for finishing the task)
    - E.g. hot news spreading, Siri
      - Linear decay model in this paper, $U_t = U_0 - wt$
- **Trade-off in the MCC:**
  - Smaller task-finishing time
  - More energy consumption is possible
- **Challenge**
  - Maximize the gained utility with the given energy budget

Problem Formulation

- **Network Model**
  - Limited battery of the mobile device
  - Non-preemptive task
    - Cannot be interrupted once it is scheduled
- **Scheduling problem**
  - Given the arrival time, local processing time, transmission time of each task, and the battery constraint of the smartphone, find a scheduling so that the utility gain is the maximum
  - NP - complete

Challenges

- A motivational example
  - Every second, a new task arrives, $U_t = 10 - 2t$
  - Each task consumes 1 energy unit in local or 2 units to transmit
  - The energy budget is 4 units



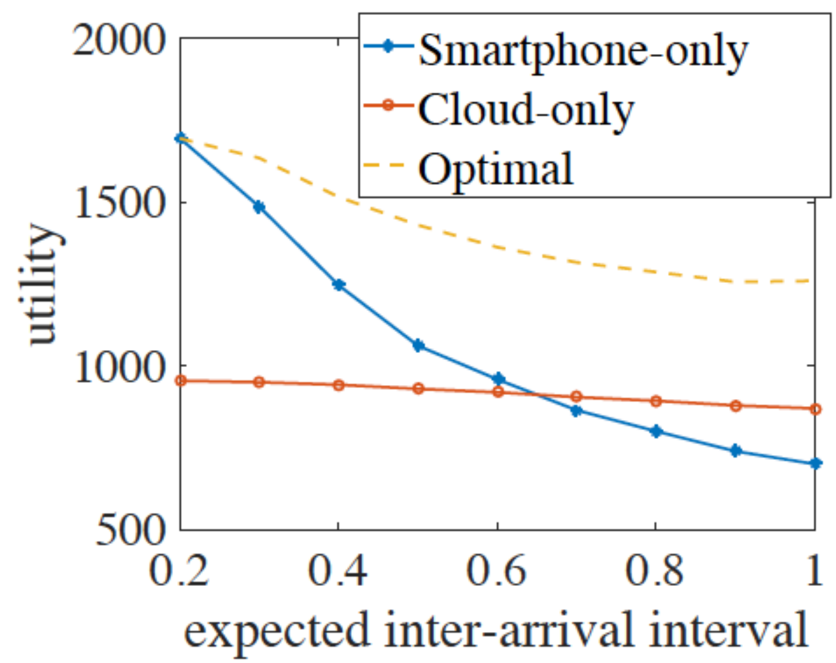
1: 6 + 4 + 2 + 0 = 12

2: 8 + 8 = 16

3: 6 + 8 + 6 = 20

Challenges

- How can we optimize the mobile device and the cloud scheduling jointly?
  - Traditional approaches will lead to a bad performance in certain scenarios



Solution

- Simple greedy scheduling does not have a performance bound
  - Cost effective
    - Unbalanced task assignment
  - Balance the finishing time
    - Low energy usage efficiency

- We propose an approximation algorithm with a good performance bound!

Solution

- LP Rounding algorithm
  - Assume tasks can be interrupted
    - ❑ y_{ijt} , task i is scheduled at j at time t
    - ❑ Problem reduces into the linear programming

$$\max \quad \sum_{j\in J} U_j - \sum_{j \in J} w_j (f_j - a_j)$$

$$\text{s.t.} \sum_{i=1}^{2} \sum_{t=a_i}^{T} \frac{y_{ijt}}{p_{ij}} = 1, \qquad \forall j,$$

— Task must be completely finished

$$\sum_{j\in J} y_{ijt} \le 1, \qquad \forall i \& t,$$

— Processing or transmitting one task a time

$$f_j \ge \sum_{i=1}^{2} \sum_{t=a_i}^{T} \left(\frac{y_{ijt}}{p_{ij}} \left(\frac{2t+1+p_{ij}}{p_{ij}}\right)\right) \qquad \forall j,$$

$$f_j \ge \sum_{i=1}^{2} \sum_{t=a_i}^{T} y_{ijt} \qquad \forall j,$$

— Processing time constraint

$$\sum_{i=1}^{2} \sum_{j\in J} e_{ij} \frac{y_{ijt}}{p_{ij}} \le B$$

— Energy constraint

$$y_{ijt} \ge 0, \qquad \forall i, j, \& t.$$

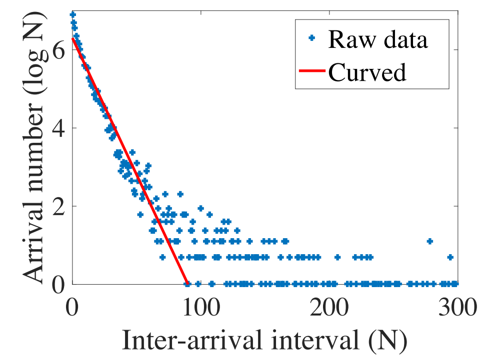
Solution

- LP Rounding algorithm
  - Calculate the LP rounding result
  - Assign the task i to j at time t with a probability based on y_{ijt}.
  - Each job is assigned at the earliest feasible time

Theorem: the LP rounding algorithm has a 2-approximation ratio on expectation

Experimental Setting

- Task arrival time
  - Average data size is 2 MB, 1 to 3 MB.
  - Follows the updates in Sina Weibo trace
    - Power-law task arrival time

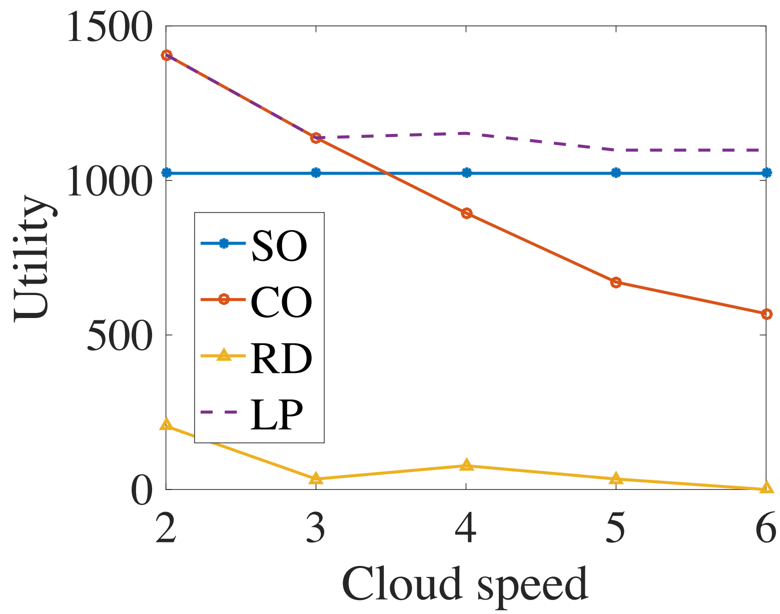


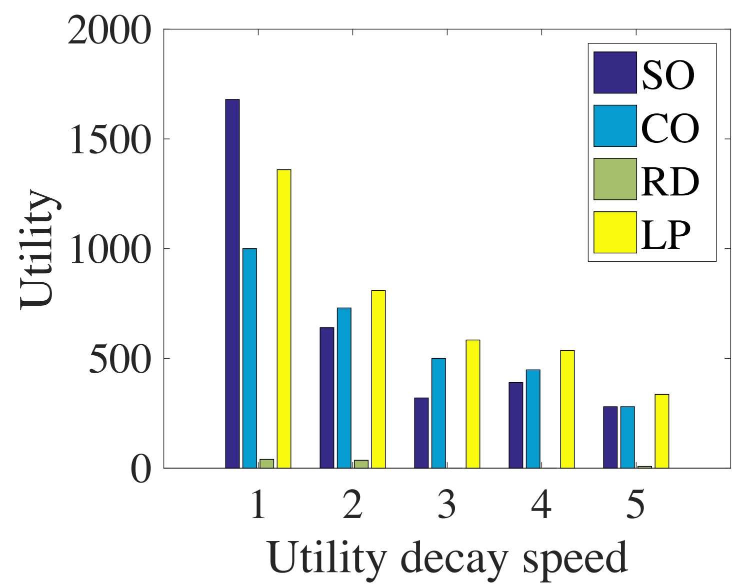
- Smartphone
  - Processing time/energy is 1 MBps, 500 mW.
  - Transmission bandwidth/energy 2 MBps, 2000 mW.
  - 100 mWh energy budget

Algorithm Comparison

- Algorithms:
  - Cloud-only (CO) algorithm, which only utilizes the cloud for computation, referred to as the AllServer algorithm in [5]
  - Smartphone-only (SO) algorithm, which only utilizes the smartphone for computation, and is referred to as the AllMobile algorithm in [5]
  - Random (RD) algorithm, which randomly assigns the new task to a device
  - Proposed LP rounding algorithm

Simulation Result





SO and CO algorithms only achieve a good performance in certain scenarios. The proposed LP algorithm achieves a good performance or the best performance in a wide scenario. The rounding algorithm leads to a bad performance.

Conclusion and Future Work

- We investigate mobile cloud offloading in a general scenario, where both mobile and cloud are used to maximized the gained utility
- An LP rounding algorithm is proposed with the performance bound
- Future Work
  - Online scheduling, real testbeds.

Thank you!

Ning Wang
ning.wang@temple.edu