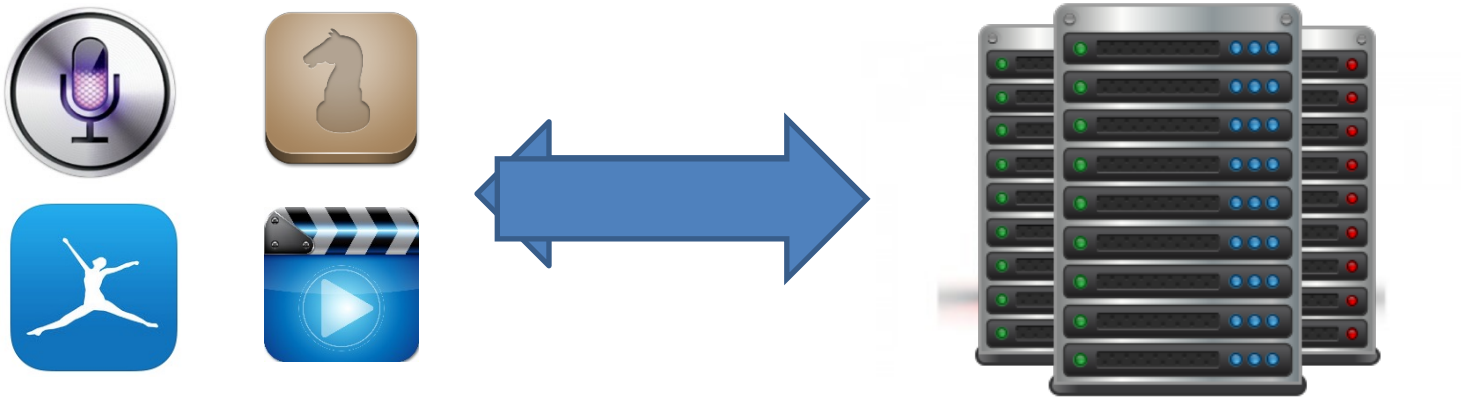
A Hierarchical Edge Cloud Architecture for Mobile Computing

Liang Tong, **Yong Li** and **Wei Gao**

University of Tennessee – Knoxville

Cloud Computing for mobile devices

- Contradiction between limited battery and complex mobile applications



- Mobile Cloud Computing (MCC)
  - Offloading local computations to remote execution
  - Reduced computation delay
  - Increased communication delay

The limits of Cloud Computing

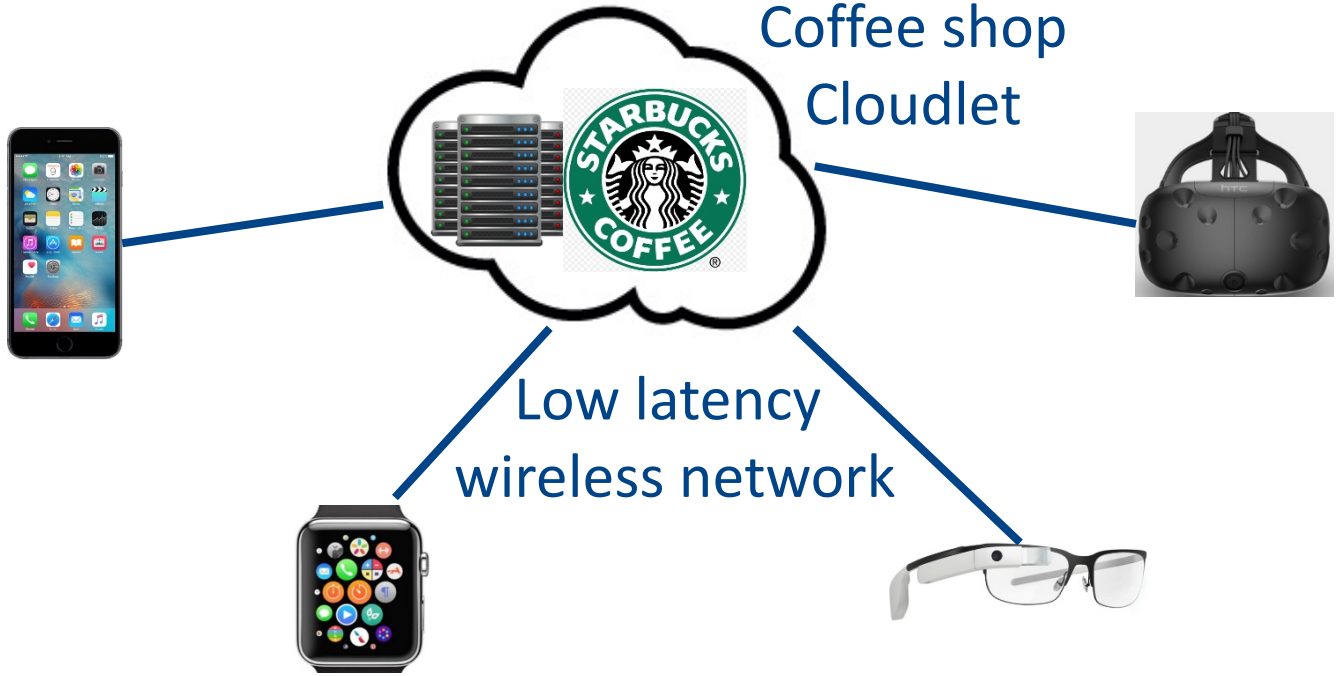
- Network communication latency of MCC
  - Can be up to 400 ms
  - Many mobile apps are delay-sensitive

} Performance degrades!

Round trip cities	Max(ms)	Mean(ms)	Min(ms)
Berkeley-Canberra	174.0	174.7	176.0
Berkeley-Troudheim	197.0	197.0	197.0
Pittsburgh-Hong Kong	217.0	223.1	393.0
Pittsburgh-Seatle	83.0	83.9	84.0
Pittsburgh-Dublin	115.0	115.7	116.0

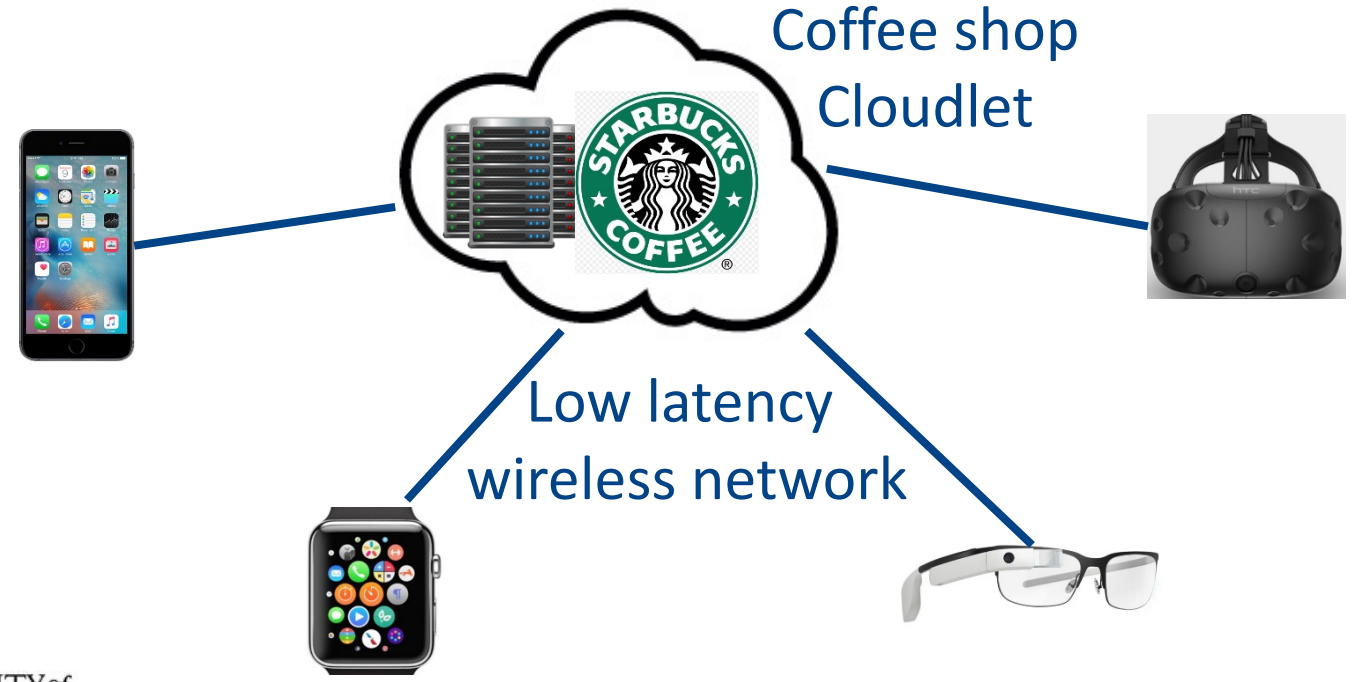
Existing solution

- Small scale cloud servers at the edge
  - Reduce the network latency accessing data center
  - Support user mobility



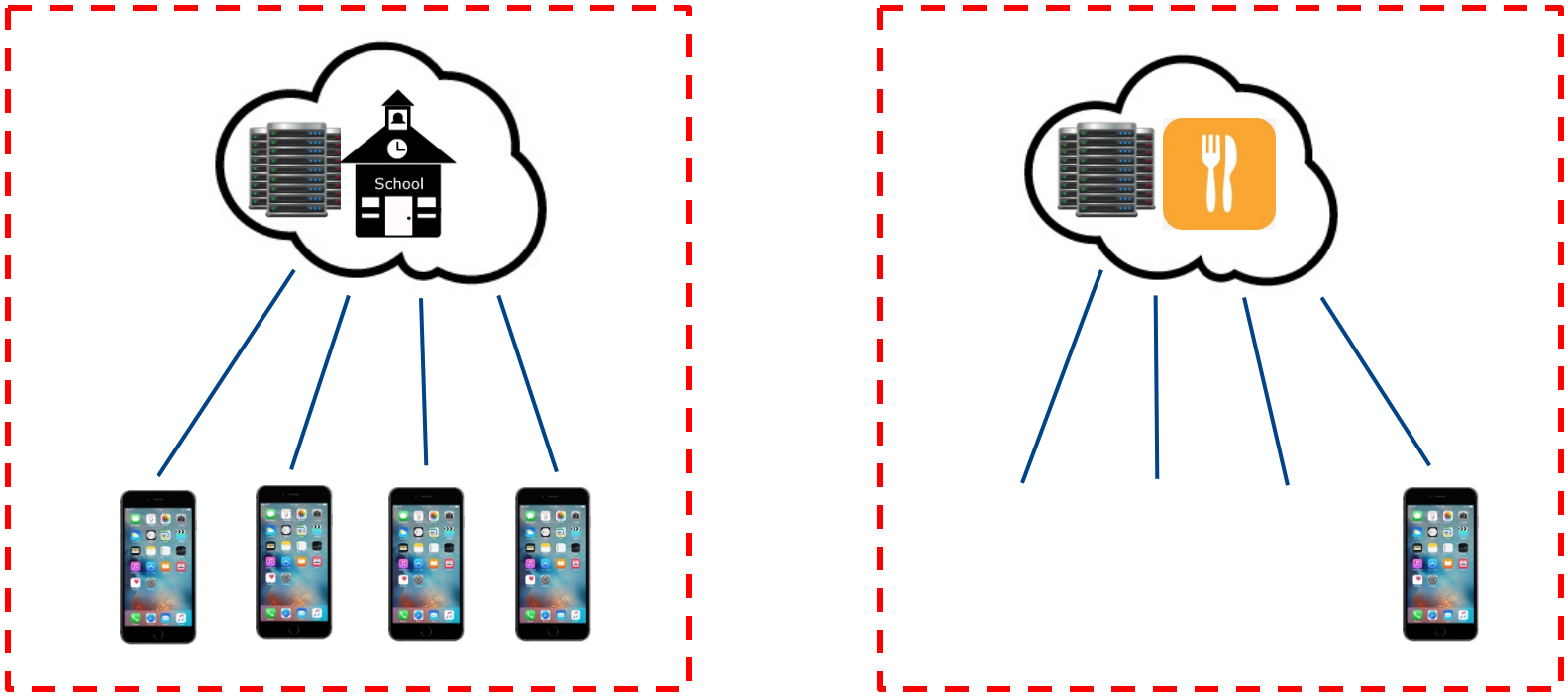
The limits of Cloudlet

- Cloudlet has limited computing resources
  - A large amount of peak load ⇒ latency ↑
  - More capacity?



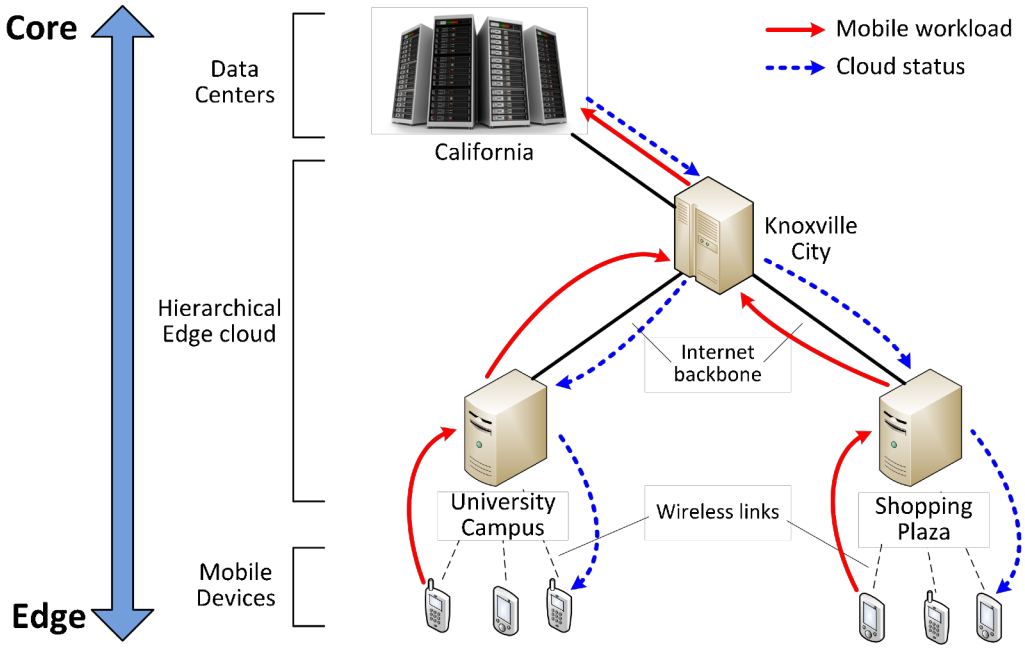
Our solution

- Motivation
  - Peak loads at different edge cloud servers do not appear at the same time



Our solution

- Key idea
  - Hierarchical edge cloud architecture
    - Opportunistically aggregate peak loads
    - Improve the resource utilization

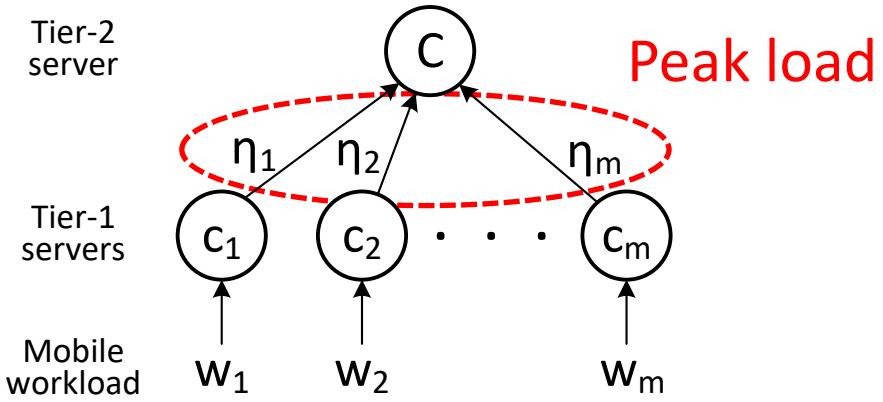


Our solution

- Key problems
  - How to efficiently provision edge cloud capacity?
  - How to appropriately place mobile workload at different tiers of servers?
- Our work
  - Formally study the characteristics of the peak load
  - Analyze the efficiency of capacity provisioning
  - Design a workload placement algorithm to further improve the efficiency of program execution

Formal study of the peak load

- System model
  - m tier-1 server and 1 tier-2 server
  - C: Computational capacity of the tier-2 server
  - c_i and w_i: computational capacity and workload of the i-th tier-1 server
  - When $w_i > c_i$, a workload of $\eta_i = w_i - c_i$ will be offloaded to tier-2.



Formal study of the peak load

- Tier-1 workload model
  - CDF of the peak load
    - Characteristics of workload exceeding c_i
    - $$\mathbf{P}(\eta_i \le x) = \begin{cases} \mathbf{P}(w_i \le x + c_i) & if\ x \ge 0 \\ 0 & \text{otherwise} \end{cases}$$
- Tier-2 workload model
  - Characteristics of tier-2 workloads
    - $$\mathbf{P}\left(\sum_{i=1}^{m} \eta_i \le x\right) = \mathbf{P}\left(\sum_{i=1}^{m-1} \eta_i \le x\right) \times \mathbf{P}(\eta_m = 0) + \int_{0^+}^{x} \mathbf{P}\left(\sum_{i=1}^{m-1} \eta_i \le x - t\right) \mathrm{d}\mathbf{P}(\eta_m \le t)$$
    - Workload of tier-2 server

Formal study of the peak load

- Provisioning of edge cloud capacity
  - Efficiency of resource utilization

Provision C to tier-2

Provision C to tier-1

$$\mathbf{P}\left(\sum_{i=1}^{m} \eta_i \leq C\right) \geq \prod_{i=1}^{m} \mathbf{P}\left(w_i \leq c_i + \alpha_i C\right), \sum_{i=1}^{m} \alpha_i = 1$$

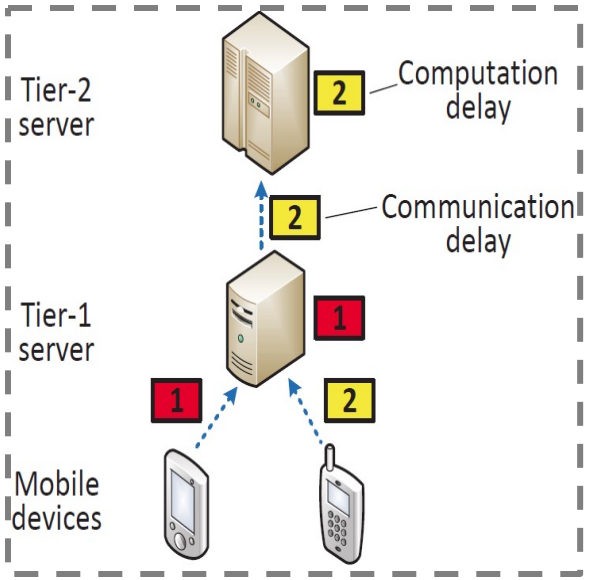
Hierarchical edge cloud

Flat edge cloud

- Insights
  - Hierarchical edge cloud has a higher chance to successfully serve the peak loads with the same capacity provisioned.

Optimal workload placement

- Objective
  - Minimize the total delay of executing all programs
- Our focus
  - Where to place a mobile program
  - How much capacity to each program
- Challenge
  - Computation/communication delay tradeoff
    - delay = computation + communication
    - Higher tiers: less computational delay, but more communication delay



Optimal workload placement

- Problem formulation
  - *m* programs at tier-1, servers in a tree-topology

Computation delay

Communication delay

$$\min f = \sum_{i=1}^{m} \left(\frac{w_i}{\lambda_{i,\gamma_i} c_{\gamma_i}} + (L(\gamma_i) - 1) \frac{s_i}{B_{\gamma_i}} \right),$$

$$\text{s.t.} \sum_{j \in O_j} \lambda_{i,j} = 1, j = 1,2, \ldots, n$$

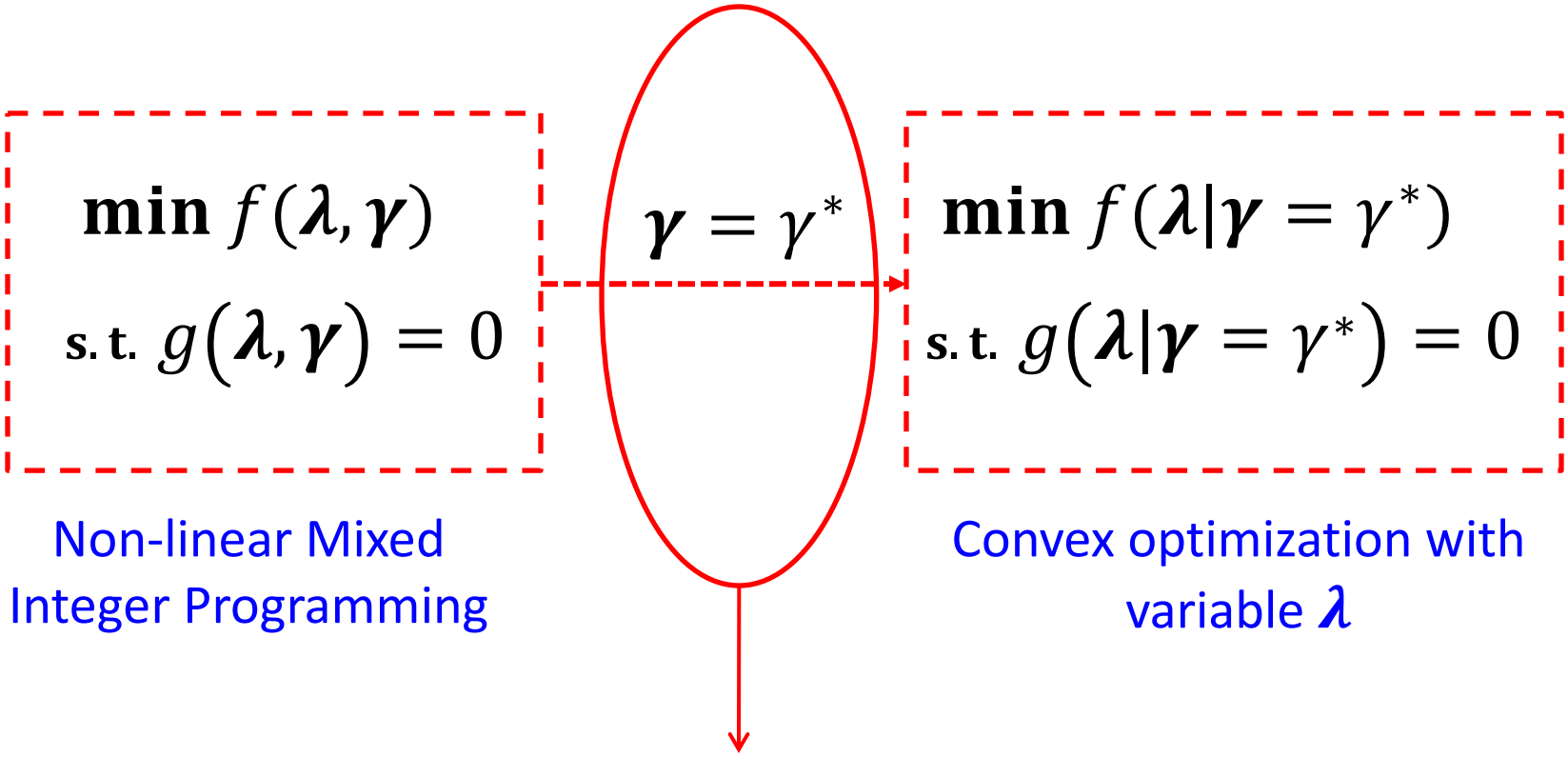
Capacity allocation of server *j* to workload *i*

Placement of workload *i*

  - Nonlinear Mixed Integer Programming
  - Challenge: γ_i and $\lambda_{i,j}$ have interdependency

Optimal workload placement

- Problem transformation



Optimal workload placement

- Solution: Simulated Annealing (SA)
  - Basic idea
    - Local optima avoidance: accepting a new state which has a worse value with an acceptance probability
  - Settings
    - **State**: workload placement vector $\boldsymbol{\gamma}$
    - **Value** $f(\boldsymbol{\gamma})$: optimal value of corresponding convex optimization problem
    - **Acceptance probability**

$$\boldsymbol{P} = exp(-\frac{\Delta d}{T})$$

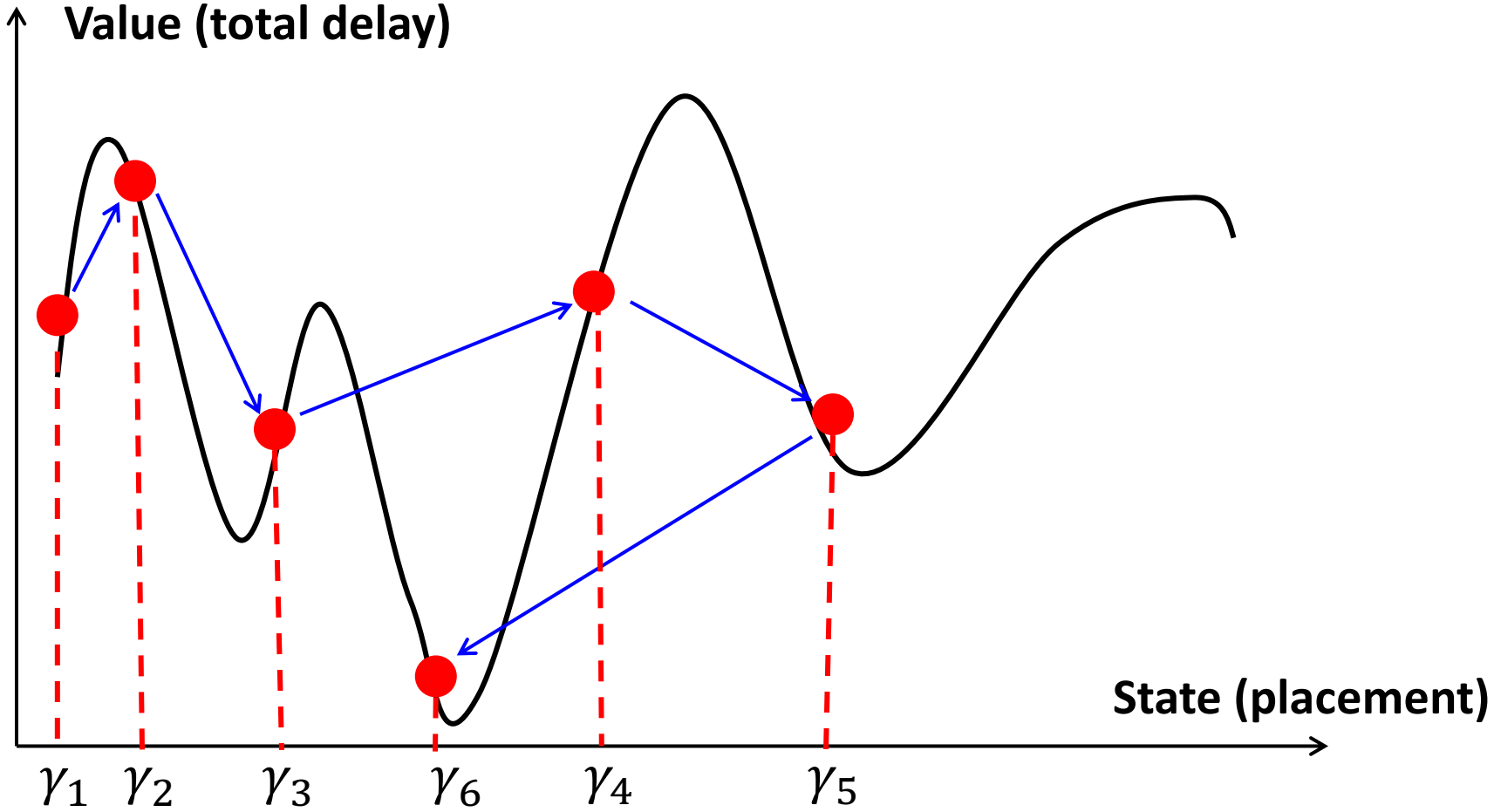
Convergence

↑

Annealing temperature, decreases in each iteration

Optimal workload placement

- Solution: Simulated Annealing



System experimentation

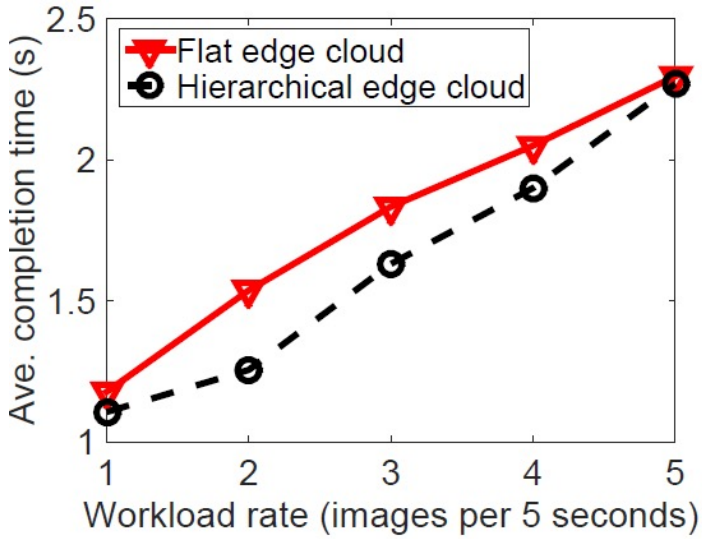
- Comparisons
  - Flat edge cloud
- Evaluation metric
  - Average completion time: indicates computational capacity
- Experiment settings
  - Workload rate
  - Provisioned capacity

Evaluation setup

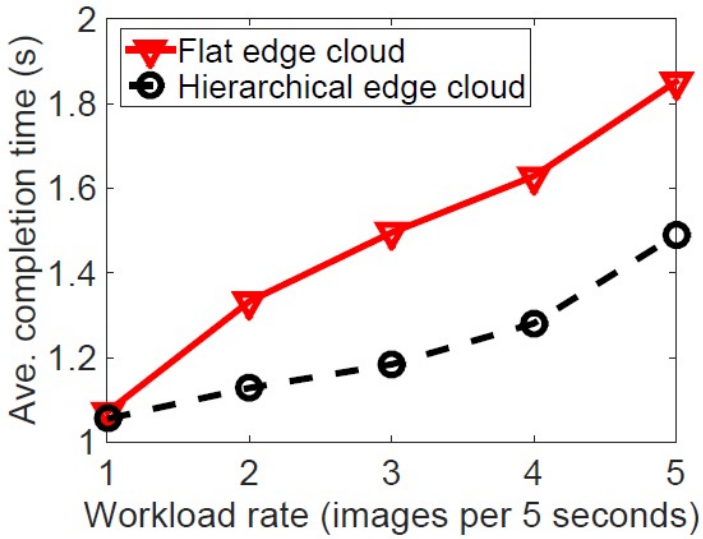
- Evaluation with a computing-intensive application
  - SIFS of images
- Edge cloud topology
  - Flat edge cloud: two tier-1 servers
    - Capacity is equally provisioned to each server
  - Hierarchical edge cloud: two tier-1 and one tier-2 server
    - Capacity is provisioned to the tier-2 server and tier-1 servers
- Experiments
  - 5 minutes with different size of images

Offloading performance

- Maximum capacity: 4 concurrent threads
  - More capacity provisioned, more improvement



(a) Provisioned capacity = 2



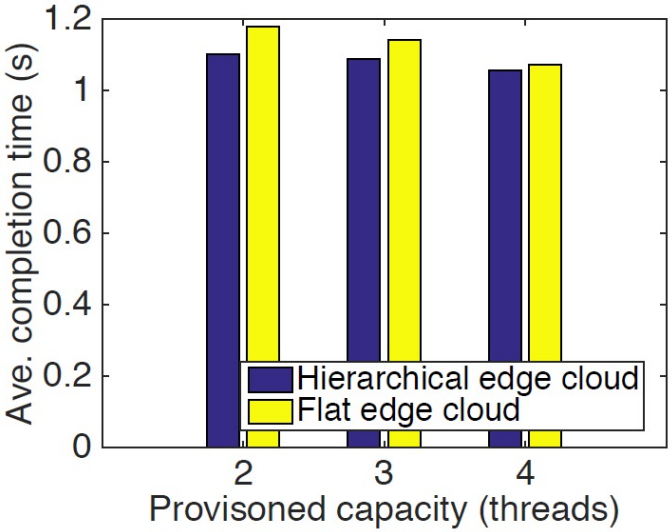
(b) Provisioned capacity = 4

25% completion time saved

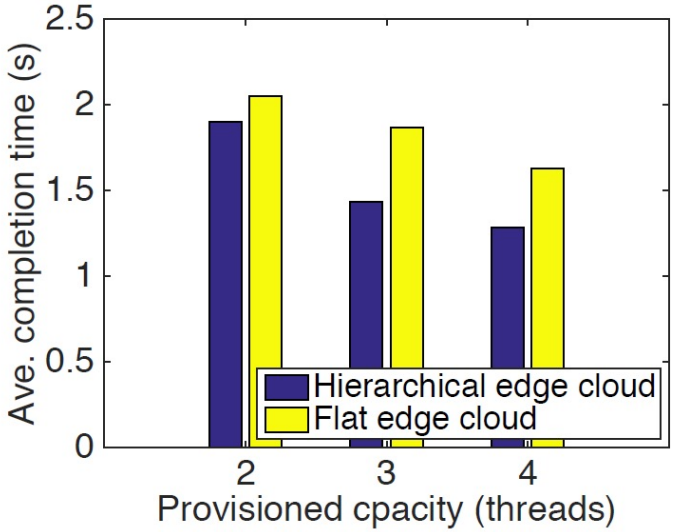
35% completion time saved

Offloading performance

- Maximum capacity: 4 concurrent threads
  - Only limited improvement at low workload



(a) Workload rate = 1



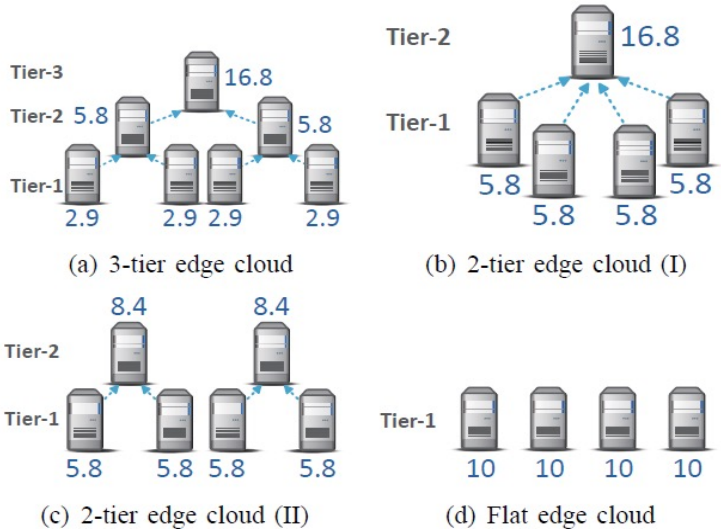
(b) Workload rate = 4

10% completion time saved

25% completion time saved

Simulation experimentation

- Comparisons
  - Four edge clouds with different topologies and capacity provisioning



(a) 3-tier edge cloud (b) 2-tier edge cloud (I)

(c) 2-tier edge cloud (II) (d) Flat edge cloud

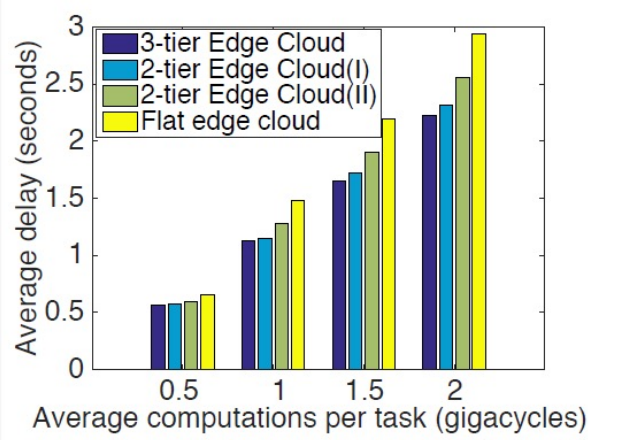
- Evaluation metric
  - Average delay: includes both computation and communication delay

Simulation setup

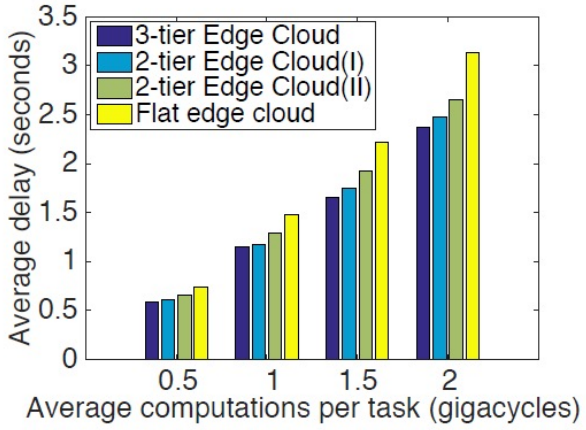
- Evaluation with real trace from Wikipedia
  - Randomly select one segment
- Computational capacity provisioning
  - 40 GHz to be provisioned to each topology
- Network setup
  - Two edge cloud servers are connected via 100 Mbps Ethernet
- Experiments
  - 1000 user requests during each simulation

Effect of computation amounts

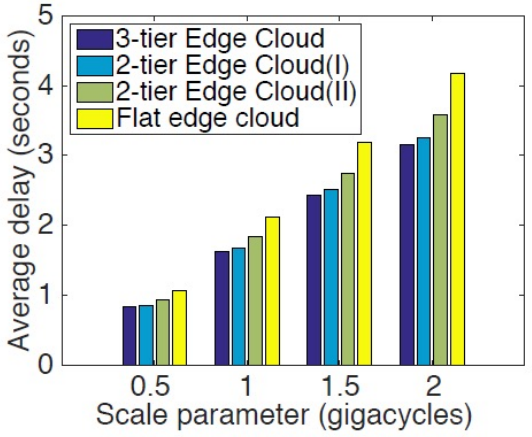
- Workload placement algorithm is used
  - Data size: normal distribution with an average of 5 MB



(a) Uniform distribution (variance = 0.08)



(b) Normal distribution (variance = 0.1)

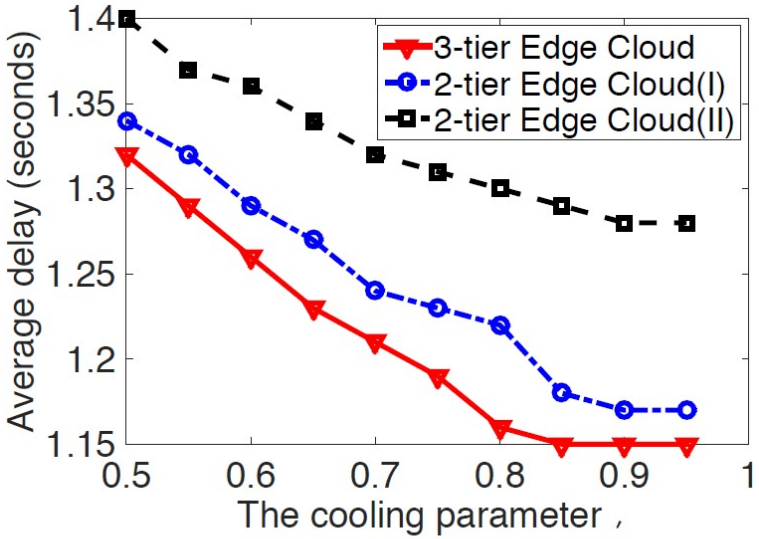


(c) Pareto distribution (shape parameter = 3)

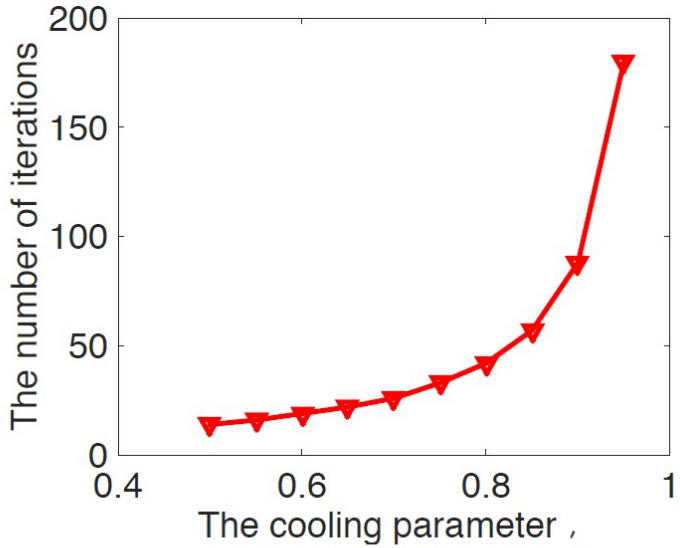
Up to **40%** delay deduction

Effect SA cooling parameter

- Performance when cooling parameter varies
  - Insights: there exists a tradeoff between performance and overhead of workload placement



(a) The average delay



(b) The number of iterations

Summary

- Offloading computations to remote cloud could hurt the performance of mobile apps
  - Long network communication latency
- Cloudlet could not always reduce response time for mobile apps
  - Limited computing resources
- Hierarchical edge cloud improve the efficiency of resource utilization
  - Opportunistically aggregate peak loads

Thank you!

- Questions?

- The paper and slides are also available at:
  http://web.eecs.utk.edu/~weigao