COLLEGE OF ENGINEERING
Electrical & Computer Engineering

QoE and Power Efficiency Tradeoff for Fog Computing

Yong Xiao *and Marwan Krunz*

Research Assistant Professor
NSF BWAC Center Manager
Department of Electrical and Computer Engineering
University of Arizona

THE UNIVERSITY OF ARIZONA

Outline

- Introduction
- QoE and Power Efficiency Tradeoff
  - Fog Computing without Cooperation
  - Cooperative Fog Computing
- An ADMM-based Distributed Optimization Algorithm
  - Introduction of ADMM
  - ADMM via Variable Splitting
- Conclusion and Future work

Cloud Computing Challenges

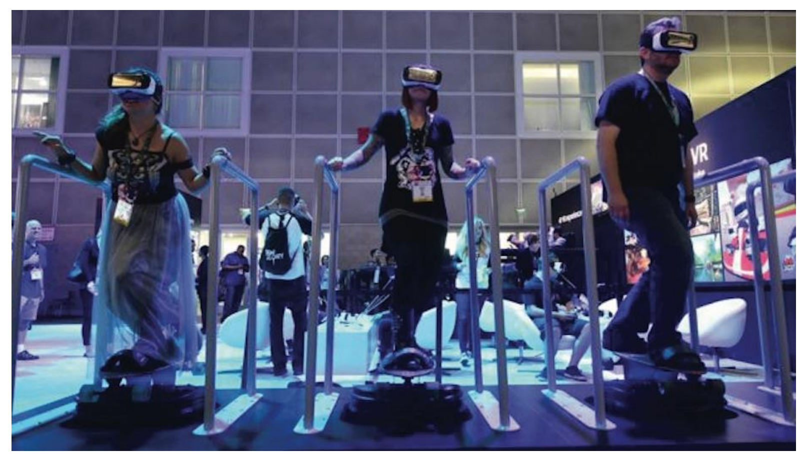
- Global data center IP traffic will grow 3-fold from 2015 to 2020, reaching 15.3 zettabytes by the end of 2020

Latency, Latency, Latency!!!

Big drops in sales and traffic have been found when pages took longer to load

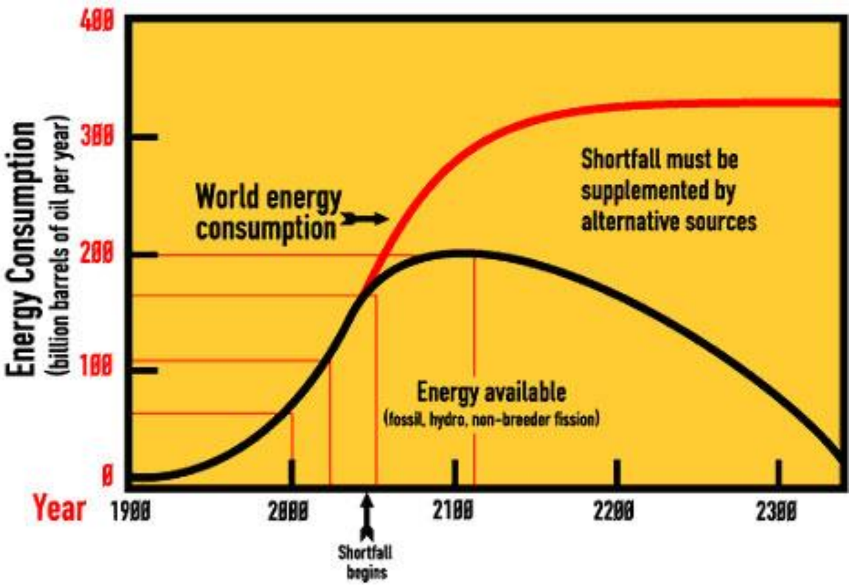
- 0.5s delay will cause a 20% drop in Google's traffic
- 0.1s delay can cause a drop in 1% of Amazon's sales

Many future applications become more sensitive to latency.



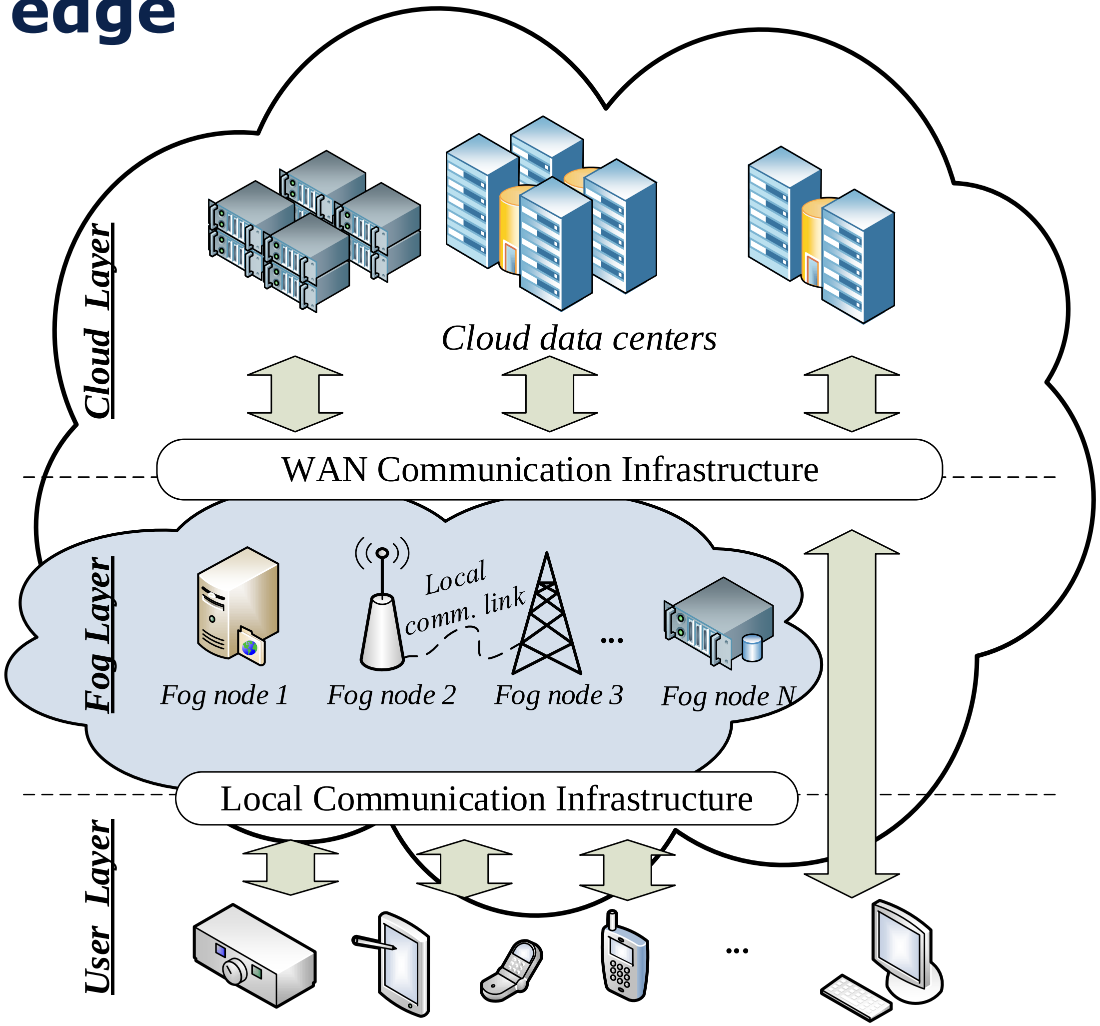
Energy, Energy, Energy!!!

- By the year 2040, world energy consumption would exceed the available energy produced from existing sources



Fog Computing Architecture

Digitization drives data and infrastructure to the edge



Data centers usually located in remote area

Fog nodes are deployed closer to the users

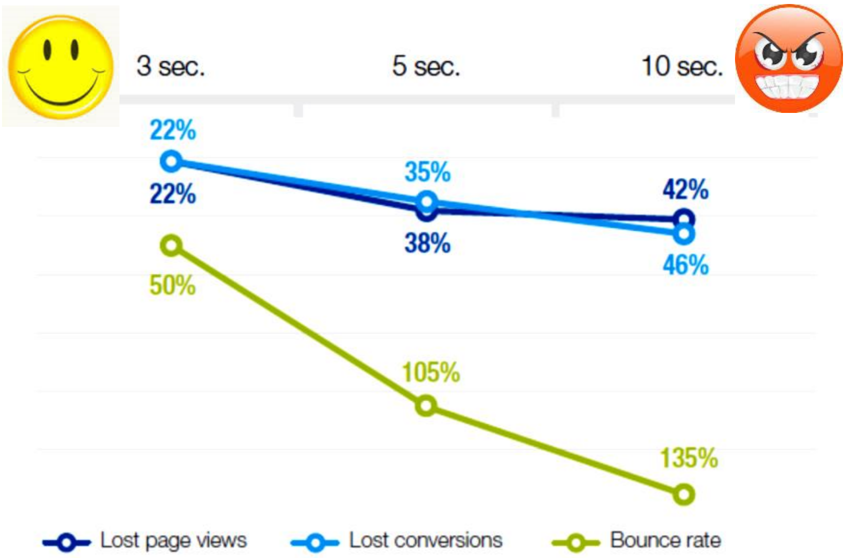
Users desire high QoE services

Key Contributions

- Characterize the fundamental **tradeoff** between **QoE** and **Power Efficiency** for fog computing
- Propose **offload forwarding strategy** for cooperative fog computing
- Propose a new distributed ADMM via **variable splitting approach** to optimize the cooperative fog computing networks

QoE for Fog Computing

- We focus on the QoE of users measured by the average *service response-time* influenced by
  - ❖ Round-trip workload transmission time:
    - ✓ Non-cooperative fog computing
    - ✓ Cooperative fog nodes
  - ❖ Queueing delay.



Response-time Analysis

- No Offloading:

$$R_j^{W1} = \tau_j^u + \tau^c.$$

Upper bound

Workload tx time between fog nodes and cloud

Workload tx time between users and fog nodes

- Full Offloading:

$$R_j^{W2}(\alpha_j) = \tau_j^u + \frac{1}{\mu_j - \lambda_j}.$$

Queueing delay

- Partial Offloading:

$$R_j^{W3}(\alpha_j) = \tau_j^u + \alpha_j \left(\frac{1}{\mu_j - \alpha_j \lambda_j} \right) + (1 - \alpha_j)\tau^c.$$

Portion of offloaded workload

Maximizing QoE

- Response-time minimization problem:
  - ❖ For non-cooperative fog computing: each fog node *j*

Portion of offloaded workload

$$\min_{0 \leq \alpha_j \leq 1} R_j(\alpha_j)$$
$$\text{s.t.} \quad \eta_j(\alpha_j) \leq \bar{\eta}_j.$$

Power efficiency constraint

Power Efficiency

- We define *power efficiency* as the power consumption per unit of offloaded workload by the fog layer:
  - ❖ Total power consumption for each fog node *j*:

$$w_j = e_j \left(w_j^S + w_j^D \alpha_j \lambda_j \right)$$

Static power consumption/leakage power (w_j^S)

Dynamic power consumption ($w_j^D \alpha_j \lambda_j$)

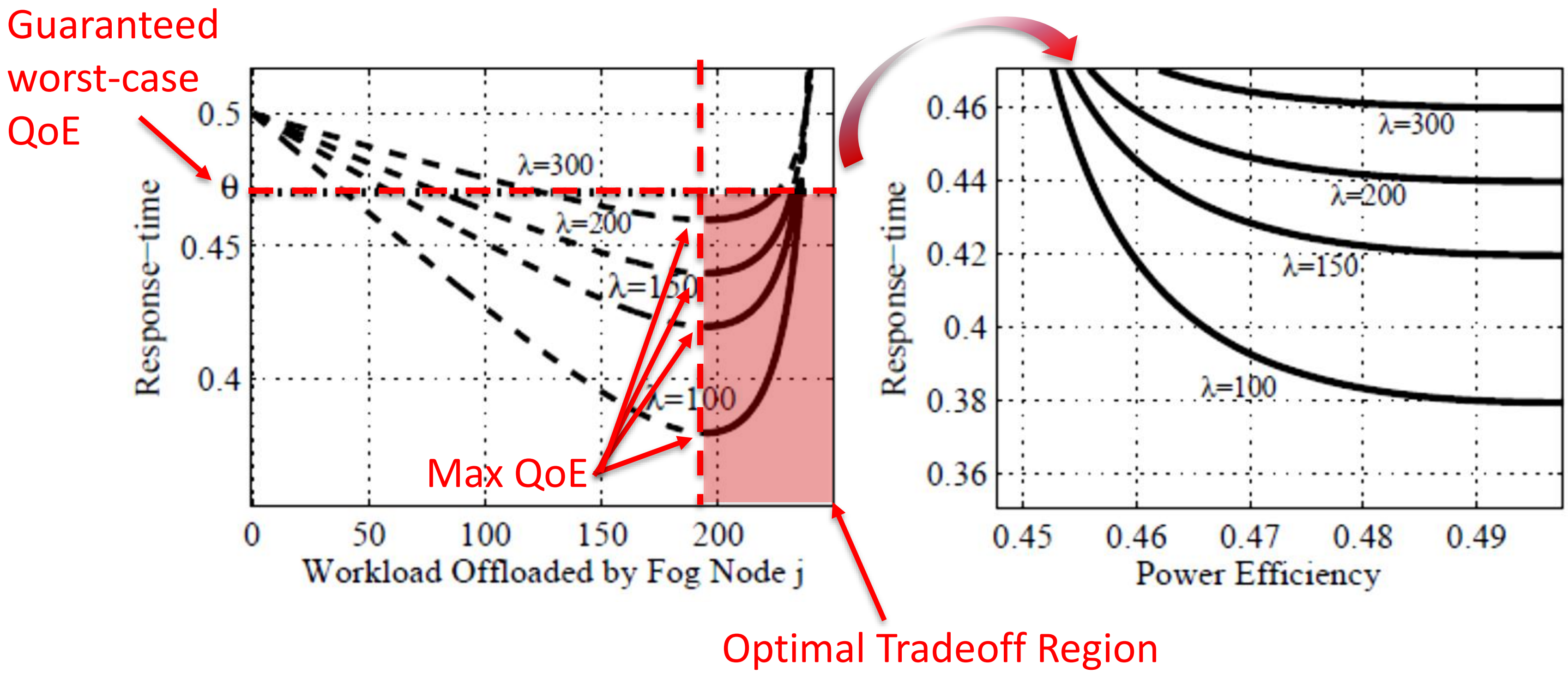
Power usage effectiveness (PUE) (e_j)

  - ❖ Power efficiency:

$$\eta_j(\alpha_j) = \frac{w_j}{\alpha_j \lambda_j} = e_j \left(\frac{w_j^S}{\alpha_j \lambda_j} + w_j^D \right)$$

Workload offloaded by fog node *j* ($\alpha_j \lambda_j$)

QoE and Power Efficiency Tradeoff



Cooperative Fog Computing

- Performance of cooperative fog computing is closely related to the cooperation strategy.
- We propose *offload forwarding* strategy:
  - ❖ Each fog node forwards part of its offloaded workload to others to further improve users' QoE.
  - ❖ Fog nodes can then be divided into
    - ✓ *Requesters*: require help from others.
    - ✓ *Servers*: can help processing workload for others.

Response-time Analysis

- Cooperative fog computing with offload forwarding
  - ❖ Fog node j forwards the offloaded workload to a set of neighboring fog nodes $\mathcal{C}_j$

$$R_j^{C3}\left(\xi_j, \boldsymbol{\varphi}_{j\bullet}\right) = \tau_j^u + \frac{1}{\sum_{i\in\mathcal{F}}\lambda_i}\left[\varphi_{jj}\left(\frac{1}{\mu_j - \varphi_{jj}}\right) + \sum_{i\in\mathcal{C}_j}\varphi_{ji}\left(\tau_{ji} + \frac{1}{\mu_i - \sum\limits_{k\in\mathcal{F}}\varphi_{ki}}\right)\right] + \varphi_{ic}\tau^c,$$

Partition of workload to be forwarded from fog node j to fog node i

Maximizing QoE

- Response-time minimization problem

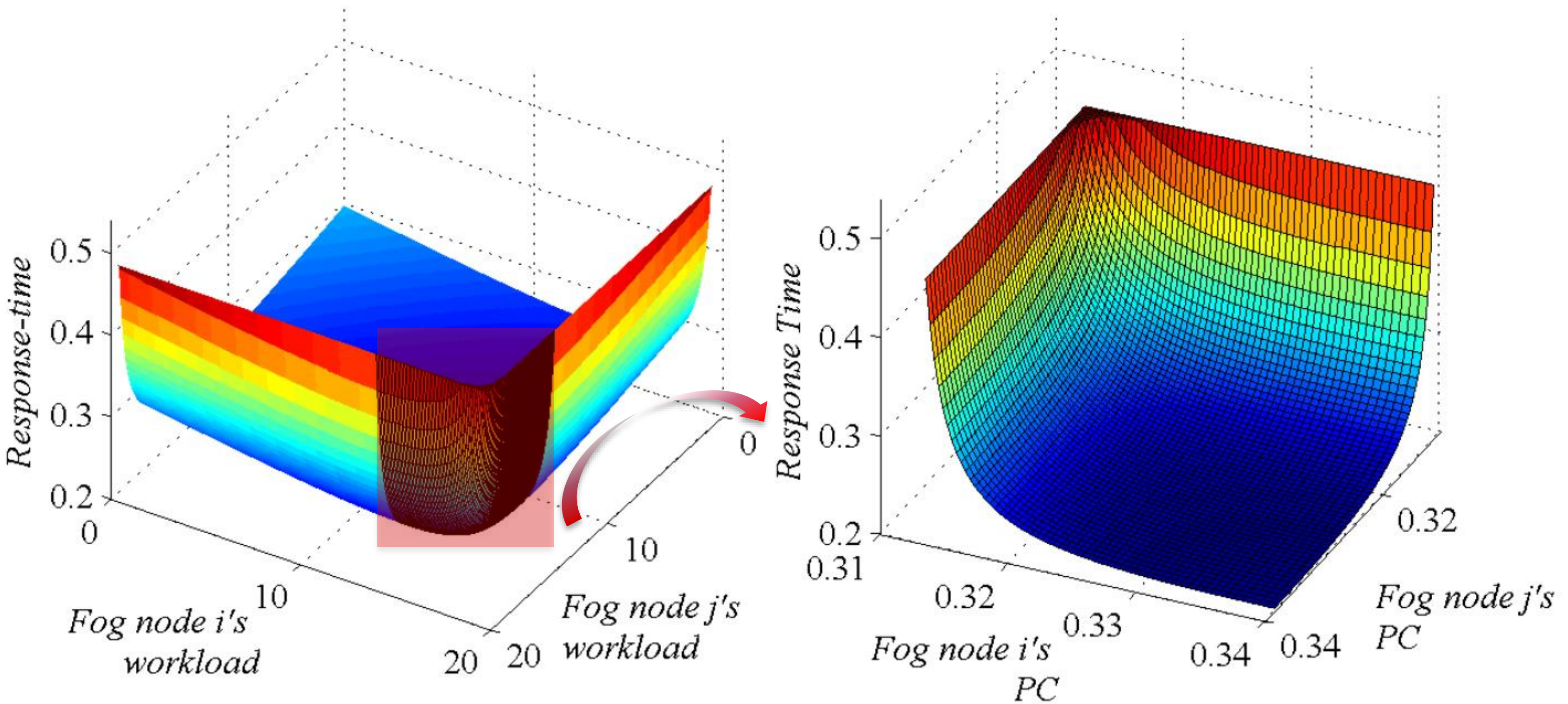
$$\min_{\varphi_{1\bullet},\ldots,\varphi_{N\bullet}} \sum_{j=1}^{N} R_j^{C3}\left(\xi_j, \varphi_{j\bullet}\right)$$

$$\text{s.t.} \sum_{k \in \mathcal{C}_j} \varphi_{jk} + \varphi_{jj} + \varphi_{jc} = \lambda_j,$$

$$\sum_{k \in \mathcal{F}} \varphi_{kj} \leq \min\{\mu_j, \boxed{\chi_j}\}, 0 \leq \varphi_{kj} \leq \lambda_k, \forall k, j \in \mathcal{F}$$

The maximum amount of workload offloaded by fog node j under the power efficiency constraint $\eta_j(\alpha_j) \leq \bar{\eta}_j$.

QoE and Power Efficiency Tradeoff



Outline

- Introduction
- QoE and Power Efficiency Tradeoff
  - Fog Computing without Cooperation
  - Cooperative Fog Computing
- An ADMM-based Distributed Optimization Algorithm
  - Introduction of ADMM
  - ADMM via variable splitting
- Conclusion and Future work

Why Apply ADMM to Optimize Fog Computing

- ADMM approach is suitable to optimize fog computing networks:
  - ❖ Objective function (Users' QoE) is convex;
  - ❖ Distributed optimization for fog nodes;
  - ❖ With equality constraints:
    offloaded + unprocessed workload = workload arrival rate;

Standard ADMM Approach

Optimization Problem

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

ADMM Solution

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_x L_\rho(x, z^k, y^k) \\ z^{k+1} &:= \operatorname*{argmin}_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &:= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

Problems for Applying ADMM to Fog Computing

- Standard ADMM cannot be directly applied because:
  1) *Inequality constraints*: forwarded workload ≤ workload arrival rate;
  2) From two blocks to *multiple blocks*;
  3) *No communication* among fog nodes;
- Objective:
  - ❖ Extending standard ADMM to solve the optimal tradeoff problem

Problem for standard ADMM

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

Our Problem

$$\max_{\varphi_{1\bullet},\ldots,\varphi_{N\bullet}} \sum_{j=1}^{N} R_j^{C3}\left(\xi_j, \varphi_{j\bullet}\right)$$

$$\text{s.t.} \sum_{k \in \mathcal{C}_j} \varphi_{jk} + \varphi_{jj} + \varphi_{jc} = \lambda_j,$$

$$\sum_{k \in \mathcal{F}} \varphi_{kj} \leq \min\{\mu_j, \chi_j\}, 0 \leq \varphi_{kj} \leq \lambda_k,$$

Proposed Distributed Optimization Framework

- A *distributed ADMM via variable splitting* approach:
  1) Introduce indicator functions and auxiliary variables to *remove the inequality constraint*

$$\min_{\boldsymbol{\varphi}_{\bullet 1},\ldots,\boldsymbol{\varphi}_{\bullet N},\boldsymbol{\psi}} \sum_{i\in\mathcal{F}} \left(R_i^{C3}\left(\xi_i, \boldsymbol{\varphi}_{i\bullet}\right) + \mathbf{I}_{\mathcal{G}_i}\left(\boldsymbol{\varphi}_{\bullet i}\right)\right) + \mathbf{I}_{\mathcal{G}_c}\left(\boldsymbol{\psi}\right)$$
$$\text{s.t.} \quad \boldsymbol{\varphi}_{\bullet i} - \boldsymbol{\psi}_i = 0, \forall i \in \mathcal{F}.$$

  2) Convert the original problem with multiple random variables into the form with *two blocks via variable splitting*;

$$\boldsymbol{\varphi}^{t+1} = \arg\min_{\boldsymbol{\varphi}} \mathcal{L}_\rho\left(\boldsymbol{\varphi}_{\bullet 1}, \boldsymbol{\varphi}_{\bullet 2}, \ldots, \boldsymbol{\varphi}_{\bullet N}, \boldsymbol{\psi}^t, \boldsymbol{\Lambda}^t\right)$$

$$\boldsymbol{\psi}^{t+1} = \arg\min_{\boldsymbol{\psi}} \frac{\rho}{2}\|\boldsymbol{\varphi}^{t+1} - \boldsymbol{\psi}^t + \frac{1}{\rho}\boldsymbol{\Lambda}^t\| + \mathbf{I}_{\mathcal{G}_c}\left(\boldsymbol{\psi}\right)$$

$$\boldsymbol{\Lambda}^{t+1} = \boldsymbol{\Lambda}^t - \rho\left(\boldsymbol{\varphi}^{t+1} - \boldsymbol{\psi}^{t+1}\right)$$

Distributed Algorithm

Algorithm 1: Distributed Optimization for Workload Forwarding

Initialization: Each fog node i chooses an initial service vector $\boldsymbol{\varphi}^0_{\bullet i}$ and WFC chooses an initial dual variable $\boldsymbol{\Lambda}^0$.

WHILE t=0, 1, ...

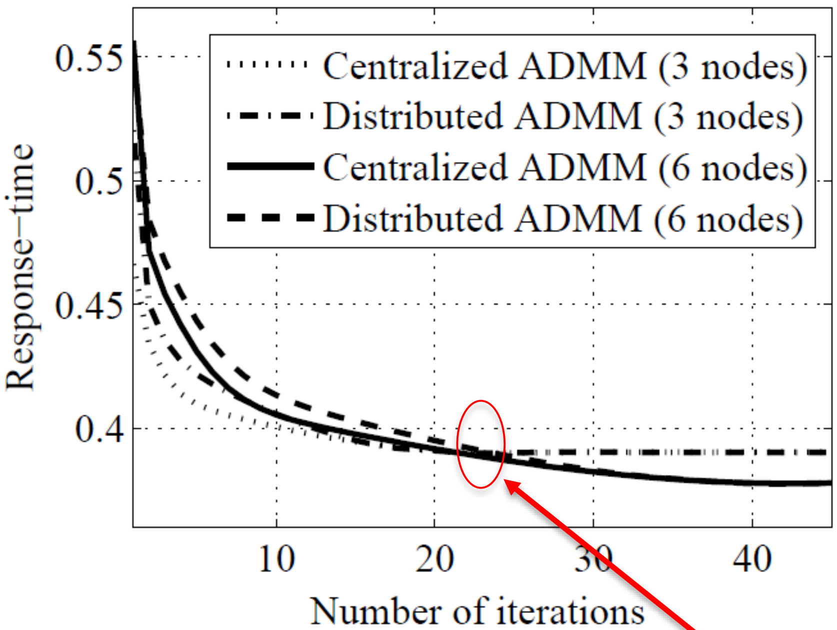
i) *Fog node updating*: Each fog node i calculates $\boldsymbol{\varphi}^{t+1}_{\bullet i}$ by solving ($*$) and then sends the resulting $\boldsymbol{\varphi}^{t+1}_{\bullet i}$ and λ_k to the cloud

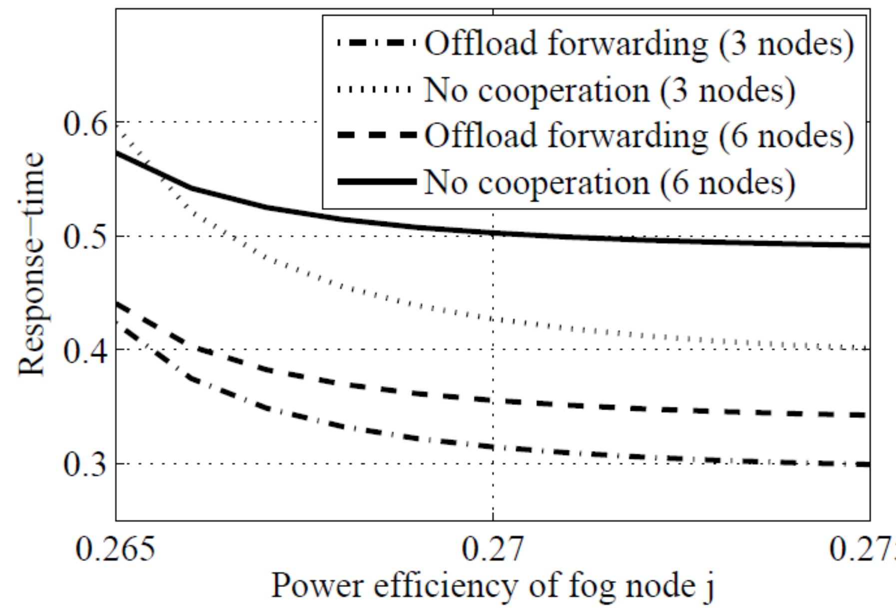
ii) *WFC Updating*: cloud calculates $\boldsymbol{\psi}^{t+1}$ by solving ψ-updating problem in (18).

iii) *Dual Variable Updating* cloud updates dual variables $\boldsymbol{\Lambda}^{t+1} = \boldsymbol{\Lambda}^k - \rho\left(\boldsymbol{\varphi}^{t+1} - \boldsymbol{\psi}^{t+1}\right)$ and sends $\boldsymbol{\varphi}^{t+1}_i$ and Λ^{t+1}_i to fog node i.

ENDWHILE

Simulation results (I)

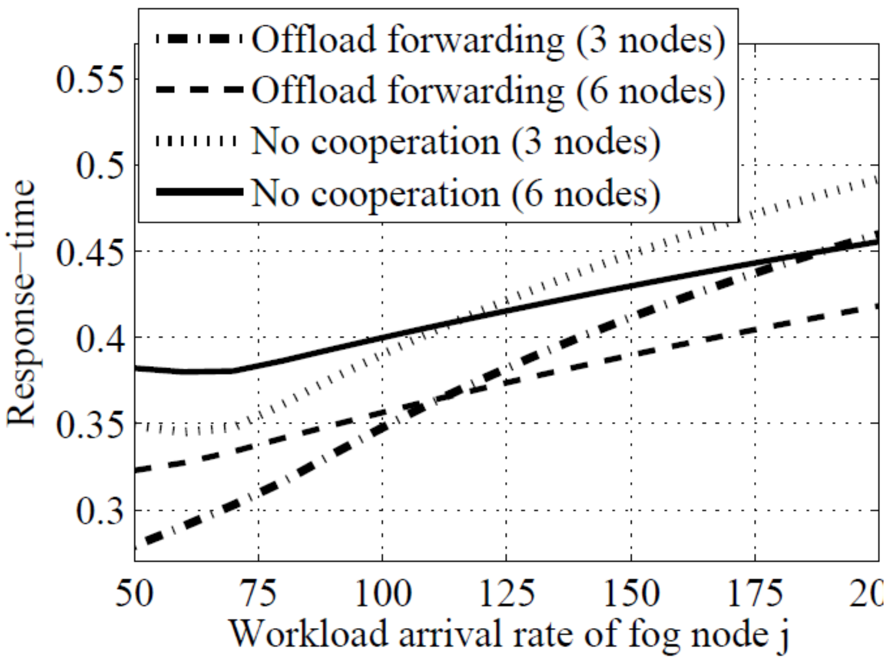


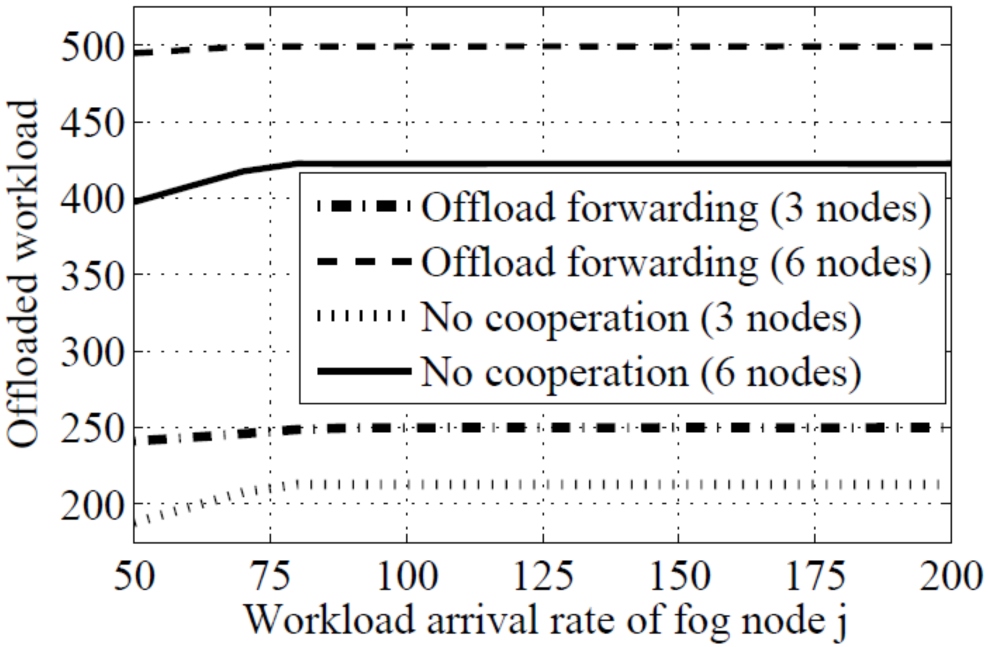


Converge in only 22 iterations

Observation: the number of fog nodes does not affect the convergence speed.

Simulation results (II)





Conclusion

- Characterize the fundamental tradeoff between QoE and Power Efficiency for fog computing
- Propose offload forwarding strategy for cooperative fog computing
- Propose a new distributed ADMM via variable splitting algorithm
- Future work:
  - Extending into stochastic environment
  - Study the QoE and power efficiency tradeoff in more complex fog computing networks, e.g., with other cooperation strategies

E-mail: xyong_2005@yahoo.com
Homepage: https://sites.google.com/site/xyong2007/