Energy Efficient Resource Management and Task Scheduling for IoT Services in Edge Computing Paradigm

Songyuan Li, Jiwei Huang
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China
sylee1416@gmail.com

Outline

- Background
- System Model
- Scheduling Scheme
- Real Data Based Evaluation
- Conclusion and Future Work

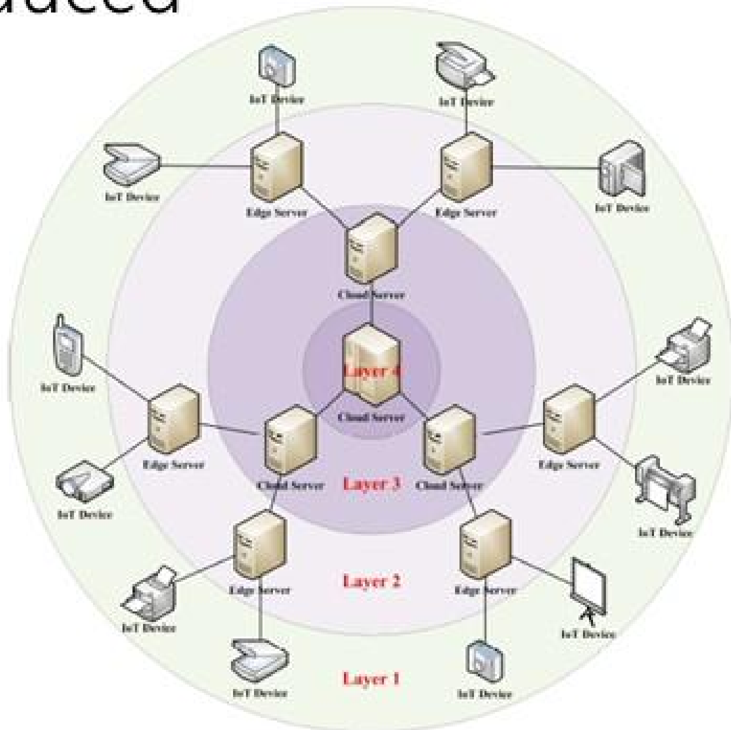
Background

❖Internet of Things (IoT)

- Connect interrelated devices over the network
- Cooperate with each other to reach common goals
- Wide application in several aspects
  - ✓Internet of Vehicles
  - ✓Smart home
  - ✓Transportation
  - ✓...

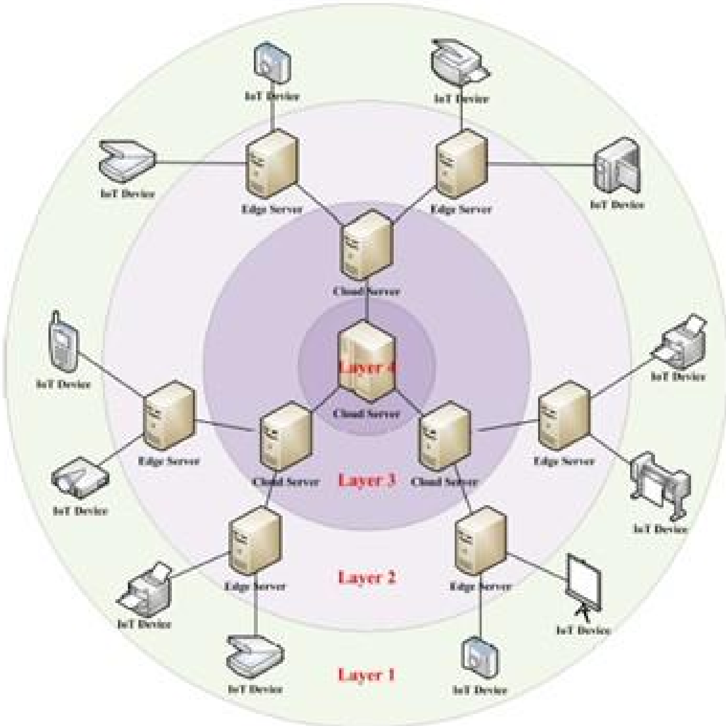
Background

- Edge Computing
  - Novel computing paradigm for IoT service implementation
  - Computation overload distributedly balanced
  - Communication delay reduced



Background

- Energy Efficiency Issues in Edge Computing
  - Resource management according to dynamic workload
  - Task assignment between layers



Background

❖ Our contribution

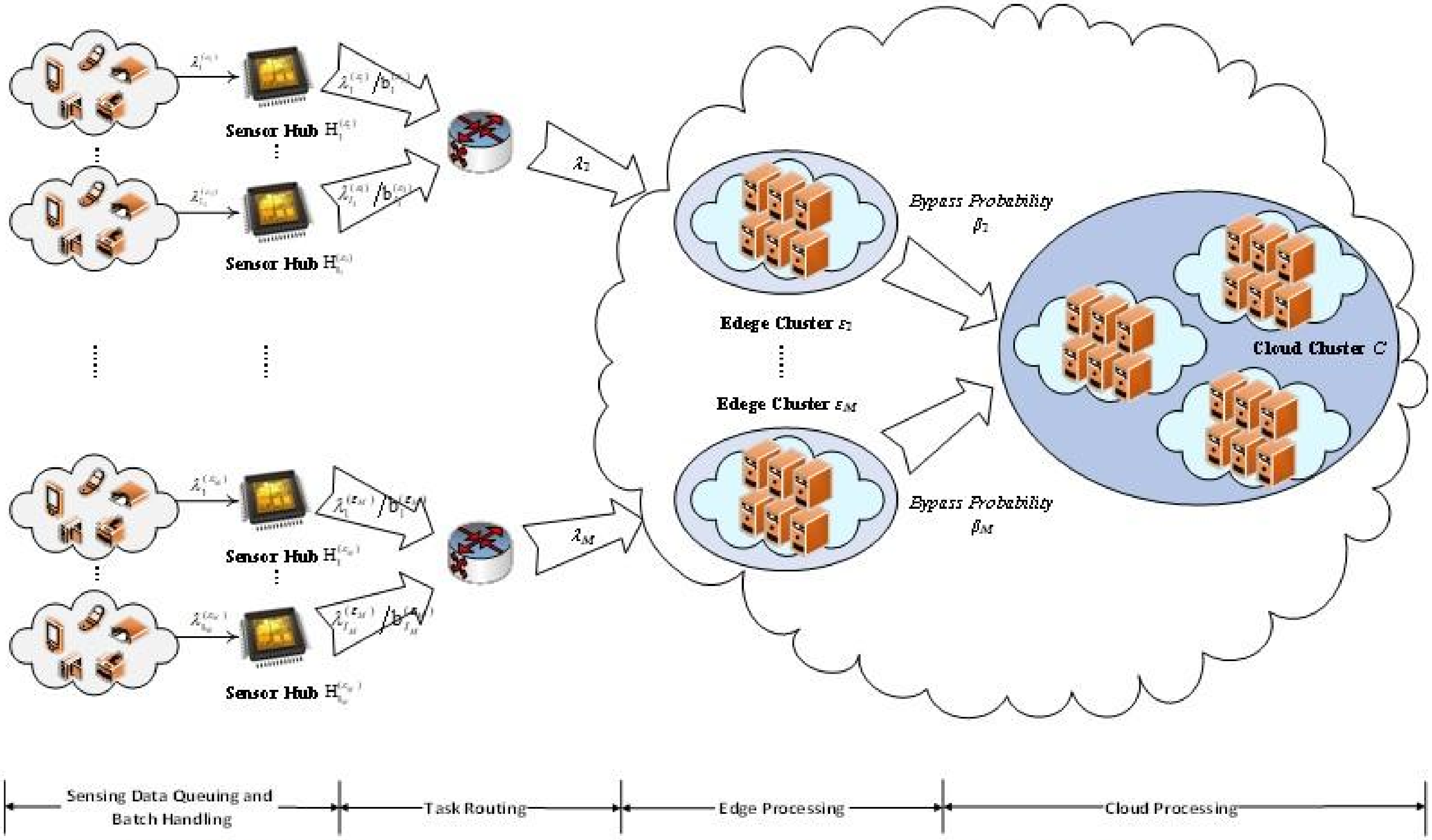
- Dynamic Formulation for IoT Services Under the Edge Computing Paradigm
- Quantitative Analyses for Performance and Energy Costs of IoT Services
- Joint Decision Process of Resource Management and Task Scheduling with High Efficiency
- Empirical Experiments Based on Real-World Data

Outline

- Background
- System Model
- Scheduling Scheme
- Real Data Based Evaluation
- Conclusion and Future Work

System Model

❖Overall Architecture

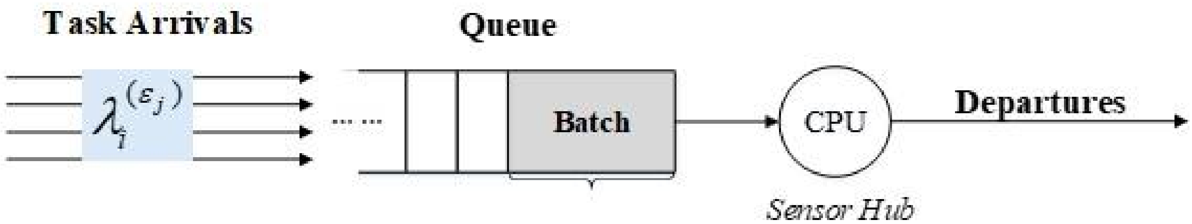


网络与交换技术国家重点实验室
State Key Laboratory of Networking and Switching Technology

北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

System Model

- Sensor Hub $H_i^{\varepsilon_j}$
  - M/M/1 Queue Model with Batch Service
  - Average Queue Length $E[q_i] = \dfrac{\rho_i}{1-\rho_i}$
  - Average Response Time $\mathrm{T}_i^{RS} = \dfrac{b_i \cdot E[q_i]}{\lambda_i}$
  - Power Consumption $\mathrm{P}_i^{\varepsilon_j} = \sigma_i \cdot \left(p_i^{(static)} + \rho_i \cdot p_i^{(dyn)} \right)$

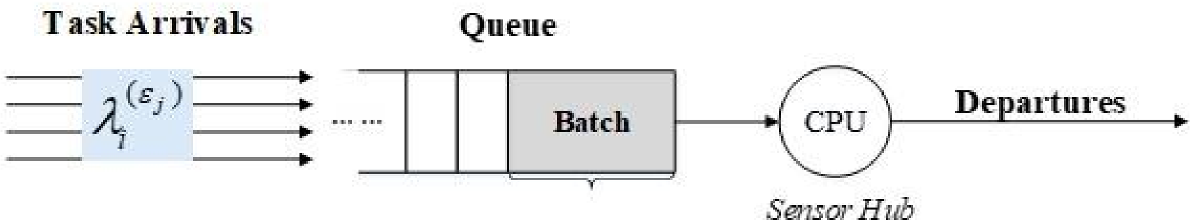


System Model

- Sensor Hub $H_i^{\varepsilon_j}$
  - M/M/1 Queue Model with Batch Service
  - Average Queue Length $E[q_i] = \frac{\rho_i}{1-\rho_i}$
  - Average Response Time $\mathrm{T}_i^{RS} = \frac{b_i \cdot E[q_i]}{\lambda_i}$
  - Power Consumption $\mathrm{P}_i^{\varepsilon_j} = \sigma_i \cdot \left(p_i^{(static)} + \rho_i \cdot p_i^{(dyn)} \right)$

Batch Arrival Rate

$$\hat{\lambda}_i^{(\varepsilon_j)} = \lambda_i^{(\varepsilon_j)} \Big/ b_i^{(\varepsilon_j)}$$



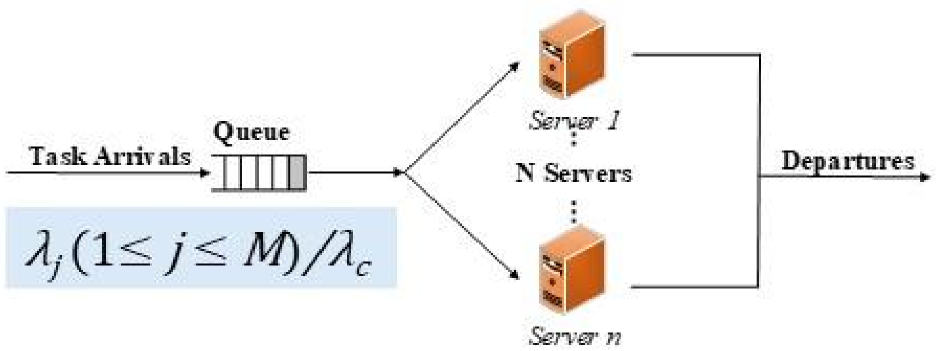
System Model

❖ Edge (ε_j) and Cloud (c) Server Cluster

- M/M/n Queue Model
- Average Queue Length

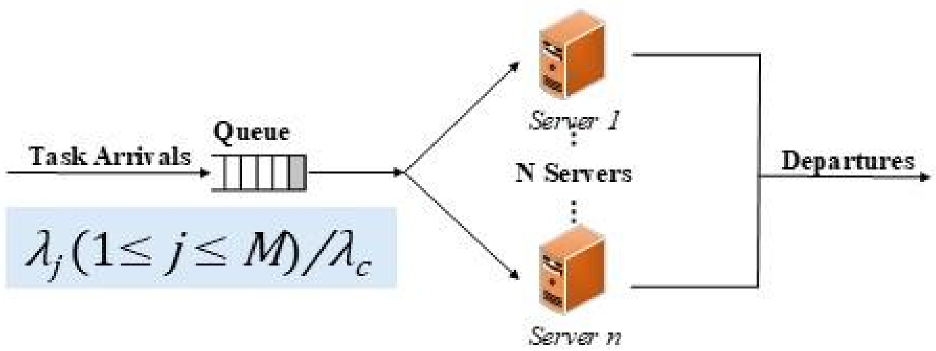
$$E\left[q_j\right]=n_j\rho_j+\frac{\rho_j\left(n_j\rho_j\right)^{n_j}}{n_j!\left(1-\rho_j\right)^2}\left[\sum_{k=0}^{n_j-1}\frac{\left(n_j\rho_j\right)^k}{k!}+\frac{\left(n_j\rho_j\right)^{n_j}}{n_j!\left(1-\rho_j\right)}\right]^{-1}$$

- Average Response Time $T_j^{RS}=\dfrac{E\left[q_j\right]}{\lambda_j}$



System Model

- Edge and Cloud Server Cluster ε_j
  - M/M/n Queue Model
  - Average Queue Length $E[q_j] = n_j\rho_j + \frac{\rho_j (n_j\rho_j)^{n_j}}{n_j!(1-\rho_j)^2}\left[\sum_{k=0}^{n_j-1}\frac{(n_j\rho_j)^k}{k!} + \frac{(n_j\rho_j)^{n_j}}{n_j!(1-\rho_j)}\right]^{-1}$
  - Average Response Time $T_j^{RS} = \frac{E[q_j]}{\lambda_j}$
  - Power Consumption $P_j = \sum_{k=1}^{N_j}\left(\sigma_k \cdot \left(p_k^{(static)} + \rho_k \cdot p_k^{(dyn)}\right)\right)$



$\lambda_j (1 \le j \le M) / \lambda_c$

System Model

❖Cross-Layer Transmission

- Bypass Transmission Probability $\beta_j \left(1 \le j \le M\right)$
- Power Consumption $T_j^{TR} = D_j \cdot t$
- Communication Delay $P_j^{TR} = p^{(dyn)} \cdot \lambda_j \cdot \beta_j + p^{(static)}$

System Model

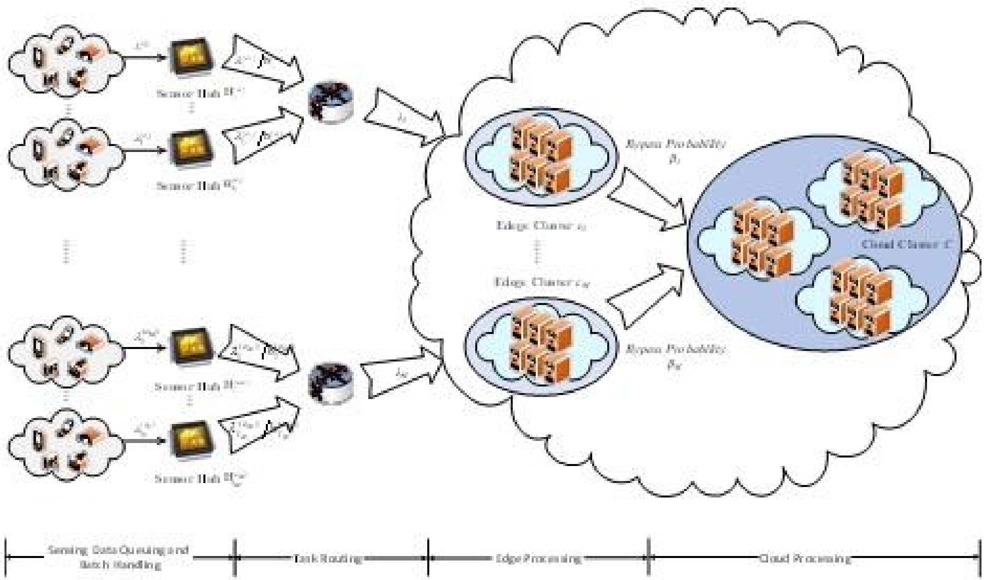
❖ Overall Architecture

- Central Cloud Cluster c
- Edge Clusters $\varepsilon_j \left(1 \le j \le M\right)$
- Sensor Hubs Subordinating to ε_j $H_i^{\varepsilon_j} \left(1 \le i \le I_j\right)$

Arrival Rate λ_c

Arrival Rate λ_j

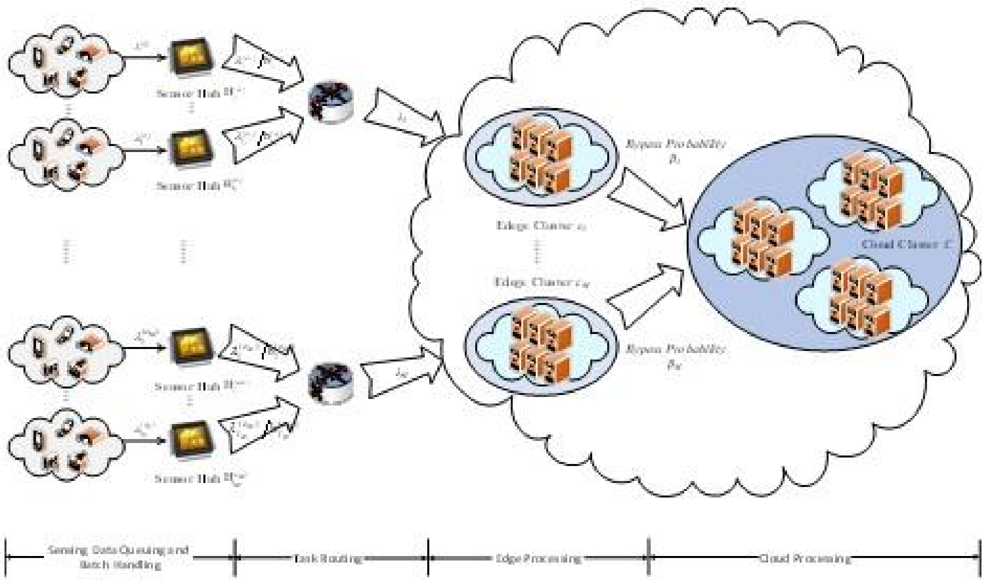
Batch Arrival Rate $\hat{\lambda}_i^{(\varepsilon_j)}$



System Model

❖ Overall Architecture

- Central Cloud Cluster c
- Edge Clusters $\varepsilon_j \left(1 \le j \le M\right)$
- Sensor Hubs Subordinating to ε_j $H_i^{\varepsilon_j} \left(1 \le i \le I_j\right)$



$$\lambda_j = \left(1 - \beta_j\right) \cdot \sum_{i=1}^{I_j} \hat{\lambda}_i^{(\varepsilon_j)}$$

$$\lambda_c = \beta_j \cdot \sum_{i=1}^{I_j} \hat{\lambda}_i^{(\varepsilon_j)}$$

System Model

❖Total System Power Consumption

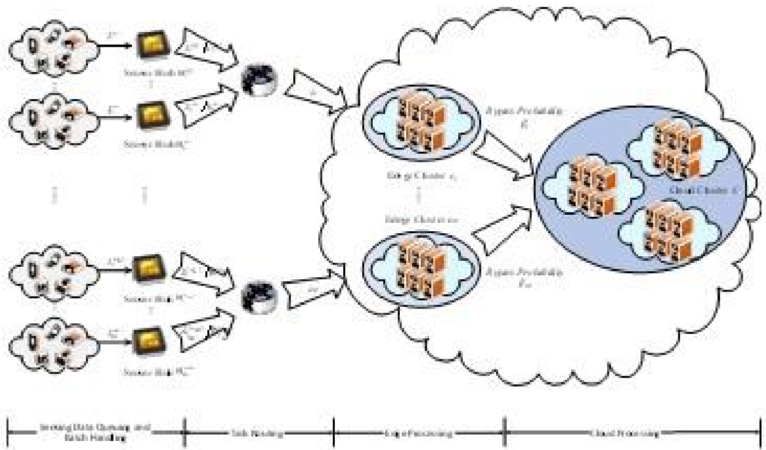
$$P^{sys} = \sum_{j=1}^{M}\left(P_j + P_j^{TR}\right) + P_{\mathrm{c}}$$

❖Total System Response Time

$$T^{sys} = \sum_{j=1}^{M} \lambda_j \cdot \left(\frac{q_j \cdot (1-\beta_j)}{n_j \cdot \mu_j} + \beta_j \cdot \left(T_j^{TR} + T_c\right) \right) \Bigg/ \sum_{j=1}^{M} \lambda_j$$

$$\lambda_j = \left(1-\beta_j\right) \cdot \sum_{i=1}^{I_j} \hat{\lambda}_i^{(\varepsilon_j)}$$

$$\lambda_c = \beta_j \cdot \sum_{i=1}^{I_j} \hat{\lambda}_i^{(\varepsilon_j)}$$



System Model

- Total System Power Consumption

$$P^{sys} = \sum_{j=1}^{M}\left(P_j + P_j^{TR}\right) + P_{\mathrm{c}}$$

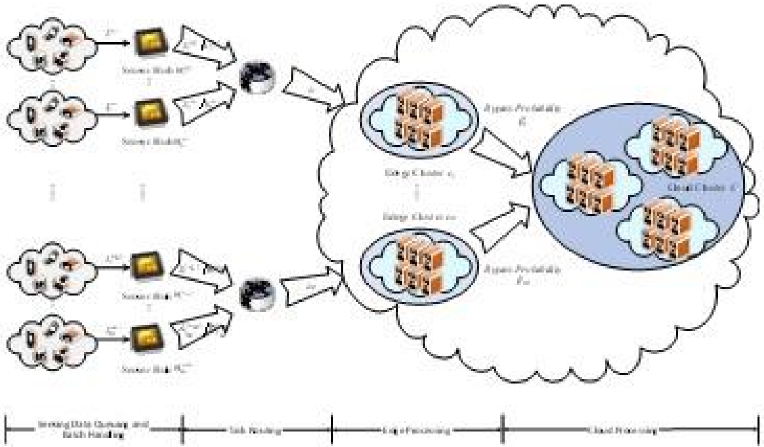
- Total System Response Time

Stationary Response Time at the Cloud Site

$$T^{sys} = \sum_{j=1}^{M}\lambda_j \cdot \left(\frac{q_j \cdot (1-\beta_j)}{n_j \cdot \mu_j} + \boxed{\beta_j \cdot \left(T_j^{TR} + T_c\right)}\right) \Big/ \sum_{j=1}^{M}\lambda_j$$

$$\lambda_j = \left(1-\beta_j\right) \cdot \sum_{i=1}^{I_j} \hat{\lambda}_i^{(\varepsilon_j)}$$

$$\lambda_c = \beta_j \cdot \sum_{i=1}^{I_j} \hat{\lambda}_i^{(\varepsilon_j)}$$



Outline

- Background
- System Model
- Scheduling Scheme
- Real Data Based Evaluation
- Conclusion and Future Work

Scheduling Scheme

Reward Model

- Total System Power Consumption

$$P^{sys} = \sum_{j=1}^{M}\left(P_j + P_j^{TR}\right) + P_c$$

- Total System Response Time

$$T^{sys} = \sum_{j=1}^{M}\lambda_j \cdot \left(\frac{q_j \cdot (1-\beta_j)}{n_j \cdot \mu_j} + \underbrace{\boxed{\beta_j \cdot \left(T_j^{TR} + T_c\right)}}_{\text{Stationary Response Time at the Cloud Site}}\right) \Big/ \sum_{j=1}^{M}\lambda_j$$

Stationary Response Time at the Cloud Site

- Service Level Agreement (SLA)

$$\boxed{T_{SLA}}$$

Upper Bound of the Response Time

Scheduling Scheme

❖ Reward Model

- Total System Power Consumption

$$P^{sys} = \sum_{j=1}^{M} \left(P_j + P_j^{TR} \right) + P_c$$

- Total System Response Time

Stationary Response Time at the Cloud Site

$$T^{sys} = \sum_{j=1}^{M} \lambda_j \cdot \left(\frac{q_j \cdot (1-\beta_j)}{n_j \cdot \mu_j} + \boxed{\beta_j \cdot \left(T_j^{TR} + T_c \right)} \right) \Bigg/ \sum_{j=1}^{M} \lambda_j$$

- Reward Function

$$R = \frac{T_{SLA} - T^{sys}}{T_{SLA}} \Bigg/ \frac{P^{sys}}{P_{\max}}$$

Scheduling Scheme

❖ MDP Formulation

- Decision Epoch $t = \tau, 2\tau, 3\tau, 4\tau, \ldots\ldots$
- State Space

Workload in M edge clusters

$$(q_1, \ldots, q_j, \ldots, q_M)$$

- Action Space
- Reward Function
- State Probability

Scheduling Scheme

MDP Formulation

- Decision Epoch $t = \tau, 2\tau, 3\tau, 4\tau, \dots \dots$
- State Space $(q_1, \dots, q_j, \dots, q_M)$
- Action Space

Cartesian Product for bypass transmission probability n_j and #of power-on machines β_j

$$(\beta_1, \dots, \beta_M) \times (n_1, \dots, n_M, n_c)$$

- Reward Function
- State Probability

Scheduling Scheme

MDP Formulation

- Decision Epoch $t = \tau, 2\tau, 3\tau, 4\tau, \ldots \ldots$
- State Space $(q_1, \ldots, q_j, \ldots, q_M)$
- Action Space $(\beta_1, \ldots, \beta_M) \times (n_1, \ldots, n_M, n_c)$
- Reward Function

$$R = \frac{T_{SLA} - T^{sys}}{T_{SLA}} \bigg/ \frac{P^{sys}}{P_{\max}}$$

- State Transition Probability

MDP Formulation

- State Transition Probability
  - Discrete Time Markov Chain

$$p_{q_j \to q_j+1} = \lambda_j \left(1-\beta_j\right)\left(1-e^{-\sum_{k=1}^{M}(\lambda_k+q_k\mu_k)\cdot\tau}\right) \Bigg/ \sum_{k=1}^{M}(\lambda_k+q_k\mu_k)$$

$$p_{q_j \to q_j+1} = q_j\mu_j \left(1-e^{-\sum_{k=1}^{M}(\lambda_k+q_k\mu_k)\cdot\tau}\right) \Bigg/ \sum_{k=1}^{M}(\lambda_k+q_k\mu_k)$$

$$p_{loop} = e^{-\sum_{k=1}^{M}(\lambda_k+q_k\mu_k)\cdot\tau} + \lambda_j\beta_j\left(1-e^{-\sum_{k=1}^{M}(\lambda_k+q_k\mu_k)\cdot\tau}\right) \Bigg/ \sum_{k=1}^{M}(\lambda_k+q_k\mu_k)$$

Scheduling Scheme

❖OO-Based Solution

- Ordinal Optimization (Edward et al.1997)
- Coarse Model
- Selection Rule

Scheduling Scheme

❖ OO-Based Solution

$$R = \frac{T_{SLA} - T^{sys}}{T_{SLA}} \bigg/ \frac{P^{sys}}{P_{\max}}$$

- Ordinal Optimization
- Coarse Model - Ignore the future earnings and simply focus on the current reward
- Selection Rule

Scheduling Scheme

❖OO-Based Solution

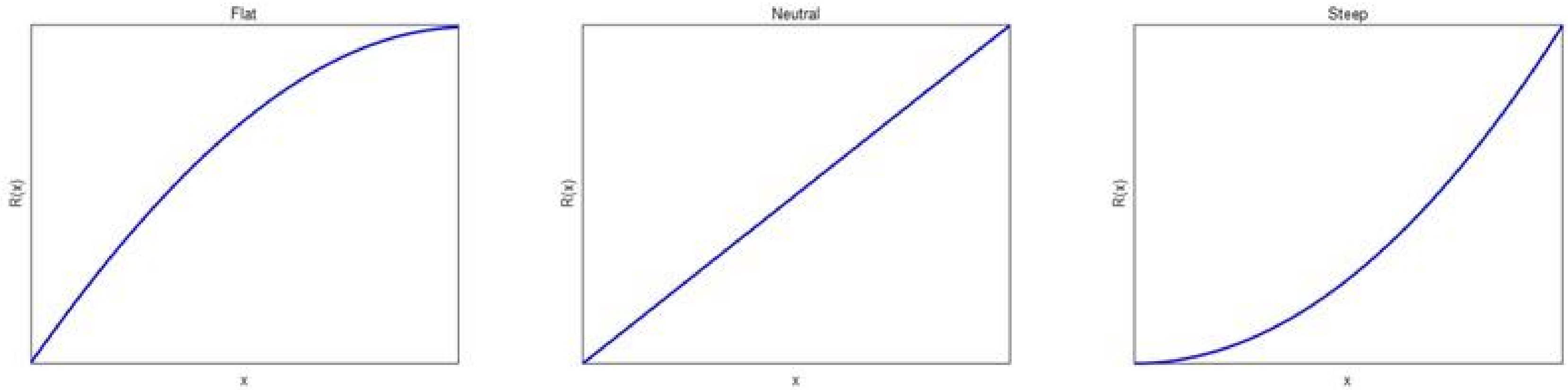
$$R = \frac{T_{SLA} - T^{sys}}{T_{SLA}} \Big/ \frac{P^{sys}}{P_{\max}}$$

- Ordinal Optimization
- Coarse Model - Ignore the future earnings and simply focus on the current reward
- Selection Rule
  - Ordered Performance Curve (OPC)
  - Error Level

Selection Rule

❖ Ordered Performance Curve (OPC)

$$r(x_i) = \frac{r_e^{[i]} - r_e^{[1]}}{r_e^{[N]} - r_e^{[1]}} \qquad x_i = \frac{i-1}{N-1}$$



- Non-decreasing sequence for estimated actions

$$\vec{R}_e = \left(r_e^{[1]}, r_e^{[2]}, \ldots\ldots, r_e^{[N]} \right)$$

Selection Rule

- Error Level

$$error = \max_{1\le k\le N}\left\{1-\frac{\vec{R}_e^{(k)^T}\cdot\vec{R}^{(k)}}{\left\|\vec{R}_e^{(k)}\right\|_2\times\left\|\vec{R}^{(k)}\right\|_2}\right\}$$

  - $0 \le error < 0.5$ (Small Level)
  - $0.5 \le error < 1$ (Medium Level)
  - $error \ge 1$ (Large Level)

Non-decreasing sequence for estimated actions

$$\vec{R}_e^{(k)} = \left(r_e^{[1]}, r_e^{[2]}, \ldots\ldots, r_e^{[N]}\right)$$

Non-decreasing sequence for evaluated actions

$$\vec{R}^{(k)} = \left(r^{[1]}, r^{[2]}, \ldots\ldots, r^{[N]}\right)$$

Scheduling Scheme

❖TSECS Algorithm

Algorithm 1 Task Scheduling of Edge-Cloud System (TSECS)

Input: Decision-making space of scheduling scenarios $\overline{S}$, the number of good enough solutions $\boldsymbol{g}$, alignment level $\boldsymbol{k}$

Output: Determined action $\boldsymbol{a_n}$.

1: Calculate the reward values of all the scheduling scenarios using coarse model

2: Estimate the OPC type based

3: Estimate the normalized error level

4: Calculate the number $\boldsymbol{s}$ of selected scenarios and the theory of OO ensures that $\boldsymbol{s}$ scheduling scenarios contains at least $\boldsymbol{k}$ good enough scheduling scenarios with probability no less than 0.95

5: Use the iterative algorithm for MDP optimization to obtain the determined action $\boldsymbol{a_n}$ within the selected $\boldsymbol{s}$ scheduling scenarios

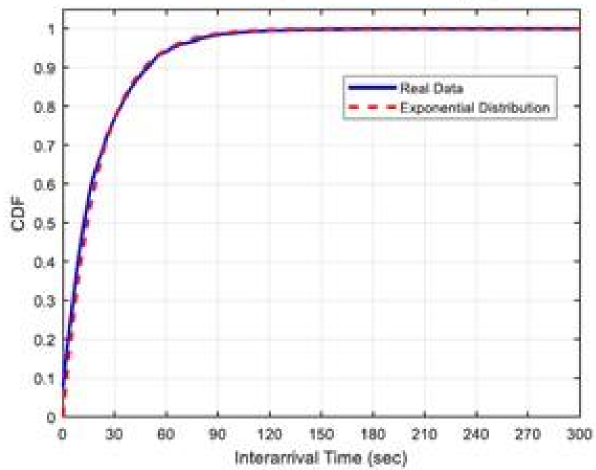
6: Return the determined action $\boldsymbol{a_n}$.

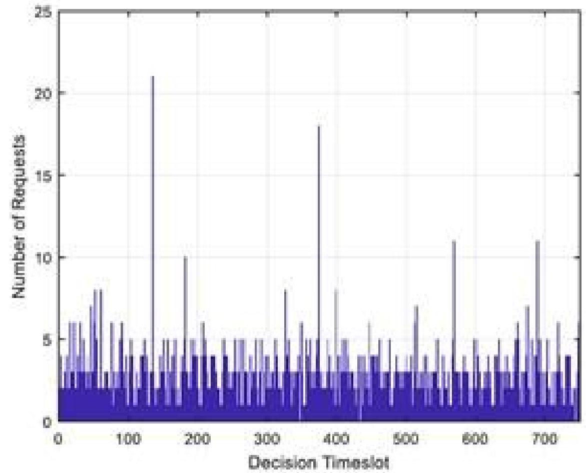
Outline

- ❖Background
- ❖System Model
- ❖Scheduling Algorithm
- ❖Real Data Based Evaluation
- ❖Conclusion and Future Work

Evaluation

- T-Drive Dataset
  - Released by Microsoft Research
  - GPS trajectories of 10,357 taxis within Beijing during a period of one week in 2008
  - 4 fields for each piece of data, including taxi id, timestamp, longitude and latitude

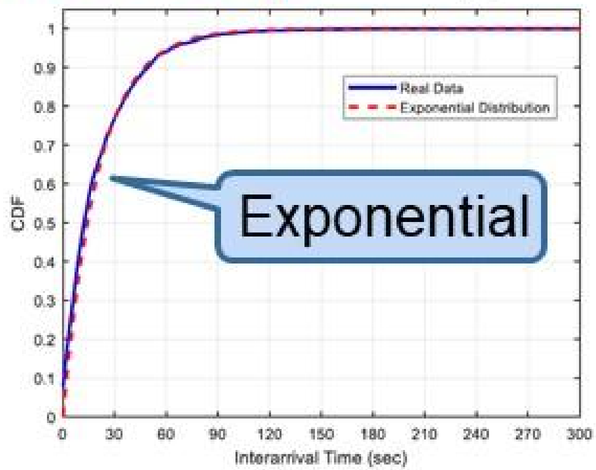


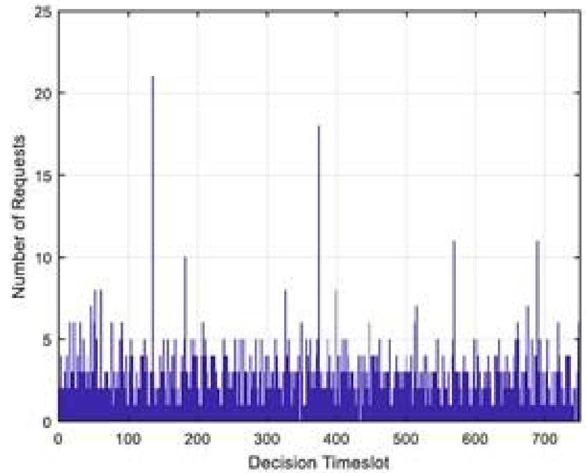


Evaluation

❖T-Drive Dataset

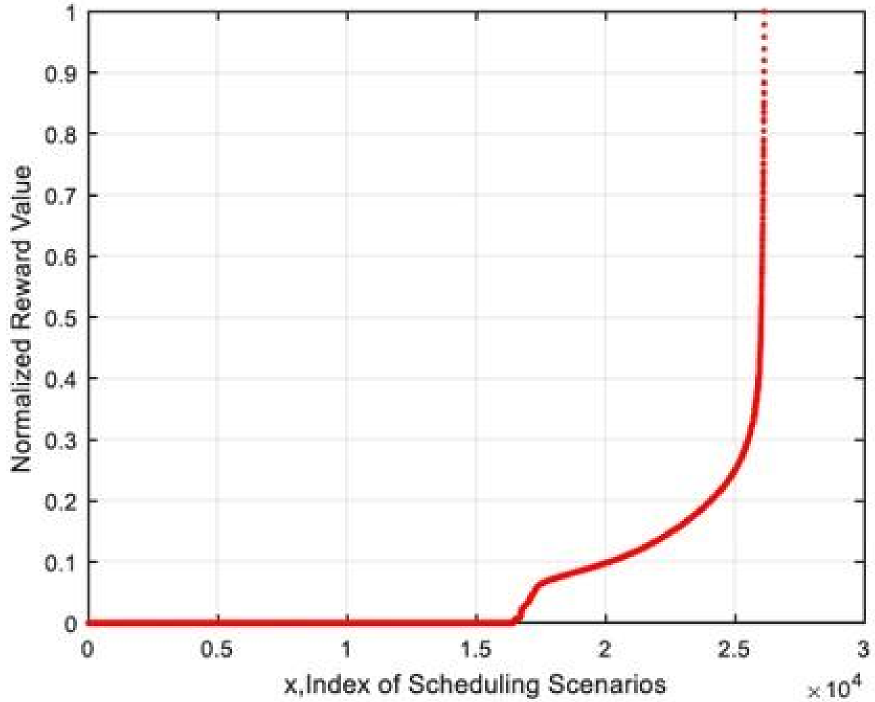
- Released by Microsoft Research
- GPS trajectories of 10,357 taxis within Beijing during a period of one week in 2008
- 4 fields for each piece of data, including taxi id, timestamp, longitude and latitude





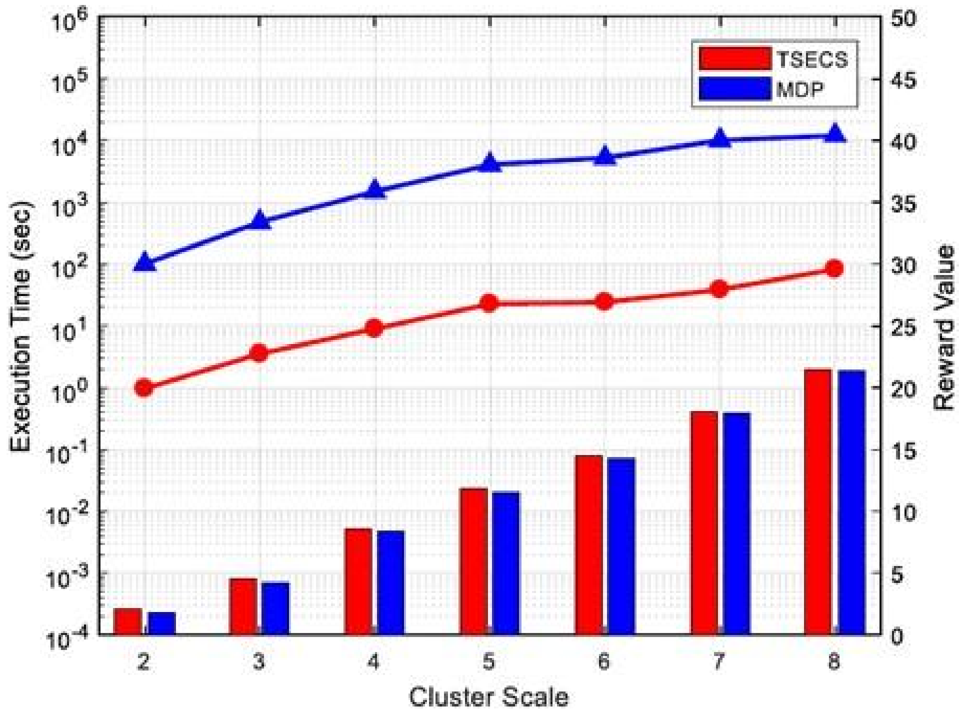
Evaluation

- Normalized Error= 0.5789 (Medium Level)
- Ordered Performance Curves (Steep)



Evaluation

❖TSECS(OO-Assisted) V.S. MDP(Classic)



Evaluation

❖ State-of-the-Art

- Load-Balanced Algorithm

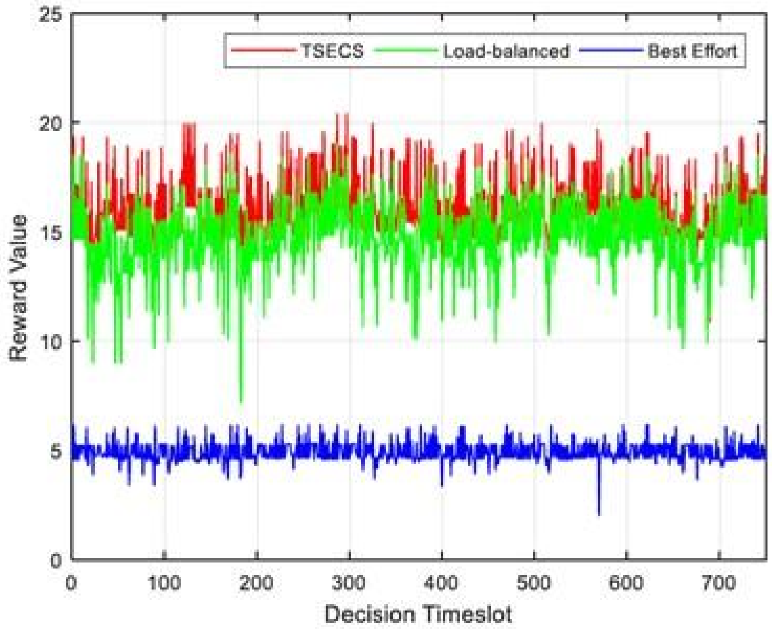
  \- Requests are dispatched to edge or cloud servers based on the serving capacity

- Best Effort Algorithm

  \- All of servers are switched on and keep constantly running

Evaluation

❖State-of-the-Art



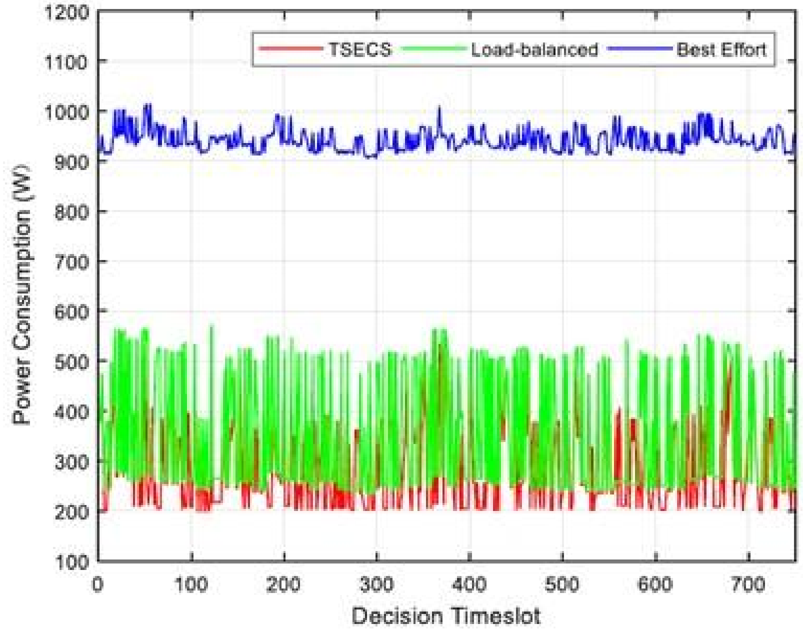


TABLE. AVERAGE RESPONSE TIME OF DIFFERENT ALGORITHMS

T_{SLA} (sec)	Algorithms Response Time (sec)		
	TSECS	Load-balanced	Best Effort
8.00	7.67	7.16	7.59

Outline

- Background
- System Model
- Scheduling Algorithm
- Evaluation
- Conclusion and Future Work

Conclusion

- Modeling approach of energy-aware performance evaluation for an IoT system with edge computing paradigm
- Joint scheme for task scheduling and resource allocations with high efficiency
- Scheduling efficiency improvement with OO techniques applied
- Simulation experiments based on real-world data from IoT and cloud systems

Future Work

- Detailed specifications on models
  - More variety of real-life IoT systems
  - More adaptive to various task arrivals
- Elaborate algorithm design
  - Further theoretical validation of efficiency

Thank you!

Q&A

Songyuan Li
Beijing University of Posts and Telecommunications
Beijing, China
sylee1416@gmail.com