Adaptive Computation Offloading for Mobile Edge Computing Environment

Houssemeddine MAZOUZI

Direction

Nadjib ACHIR, Khaled BOUSSETTA

L2TI, Institut Galilée, Université Paris 13

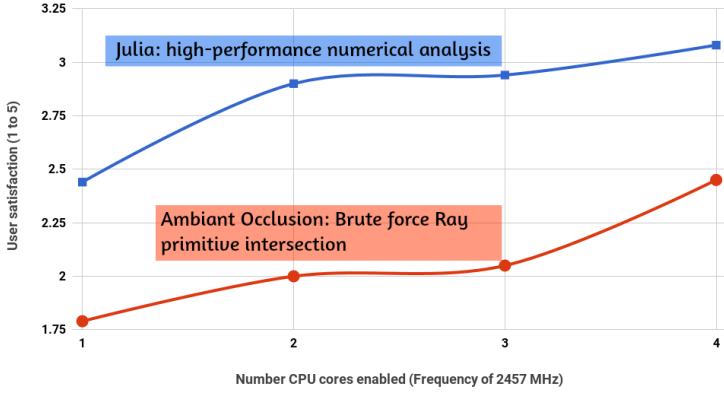
Journée MAGI Calcul scientifique 3 juillet 2018

Outline

Nowadays Mobile Devices

What is the problem?

$\Rightarrow$ how to extend the capacity of mobile device?



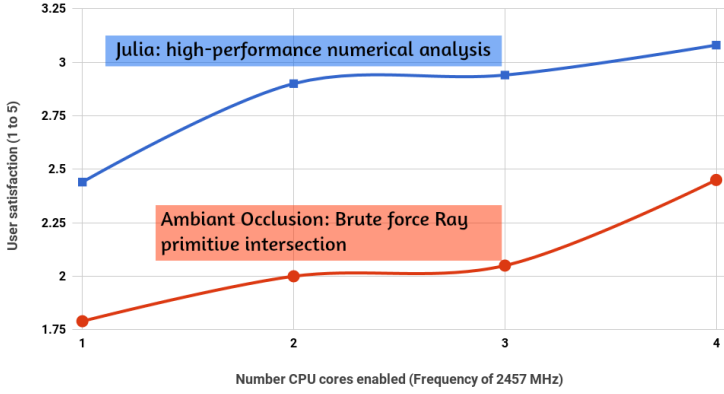
User satisfaction on Galaxy S5. Rating system: (1) Very Dissatisfactory (5) Very Satisfactory [1]

.

[1] M. Halpern, Y. Zhu, and V. J. Reddi, "Mobile cpu's rise to power: Quantifying the impact of generational mobile cpu design trends on performance, energy, and user satisfaction", in *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*, IEEE, 2016, pp. 64–76

What is the problem?

⇒ how to extend the capacity of mobile device?

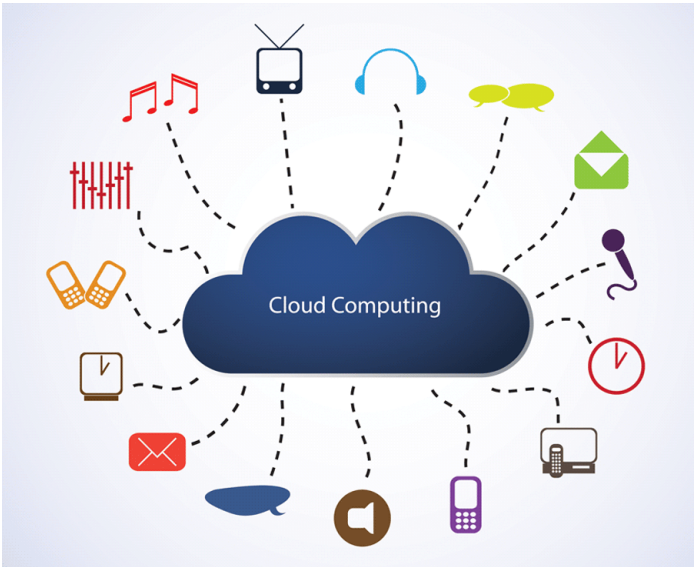


User satisfaction on Galaxy S5. Rating system: (1) Very Dissatisfactory (5) Very Satisfactory [1]

.

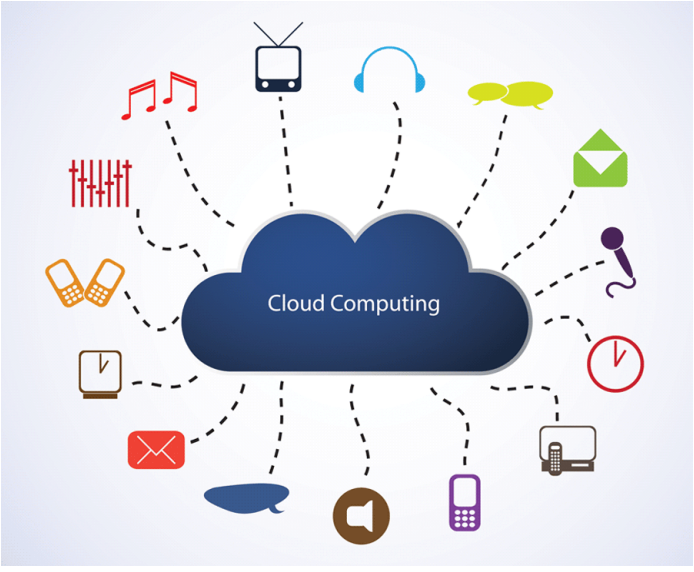
[1] M. Halpern, Y. Zhu, and V. J. Reddi, "Mobile cpu's rise to power: Quantifying the impact of generational mobile cpu design trends on performance, energy, and user satisfaction", in *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*, IEEE, 2016, pp. 64–76

The new emerging computing paradigm: Mobile Cloud Computing



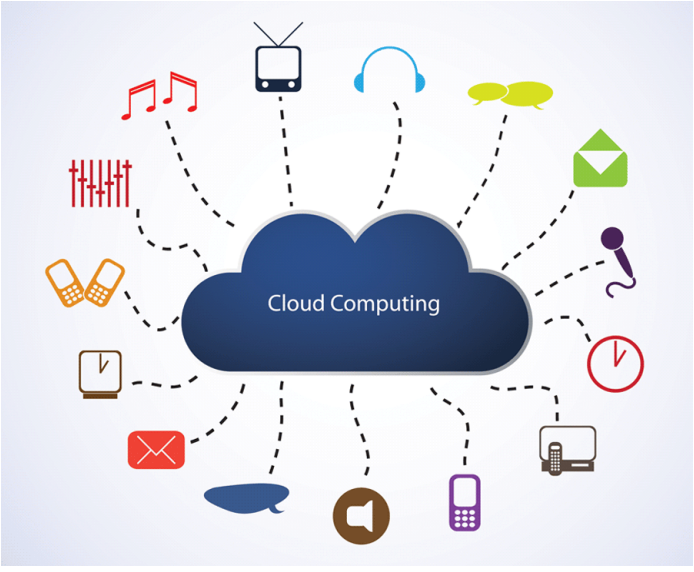
- End-to-end network latency to the closest AWS data center using wired network 20-30 ms, up to 50-150 ms on 4G mobile network.
- Ambiant occlusion requires end-to-end delays **under 20 ms** !!!!!
- What the hell !! Even the cloud is not enough!

The new emerging computing paradigm: Mobile Cloud Computing



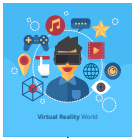
- End-to-end network latency to the closest AWS data center using wired network 20-30 ms, up to 50-150 ms on 4G mobile network.
- Ambiant occlusion requires end-to-end delays **under 20 ms** !!!!!
- **What the hell !! Even the cloud is not enough!**

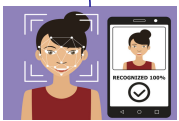
The new emerging computing paradigm: Mobile Cloud Computing

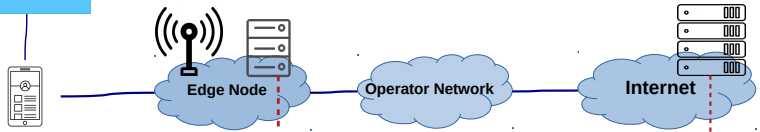


- End-to-end network latency to the closest AWS data center using wired network 20-30 ms, up to 50-150 ms on 4G mobile network.
- Ambiant occlusion requires end-to-end delays **under 20 ms** !!!!!
- **What the hell !! Even the cloud is not enough!**

The new emerging computing paradigm: extension







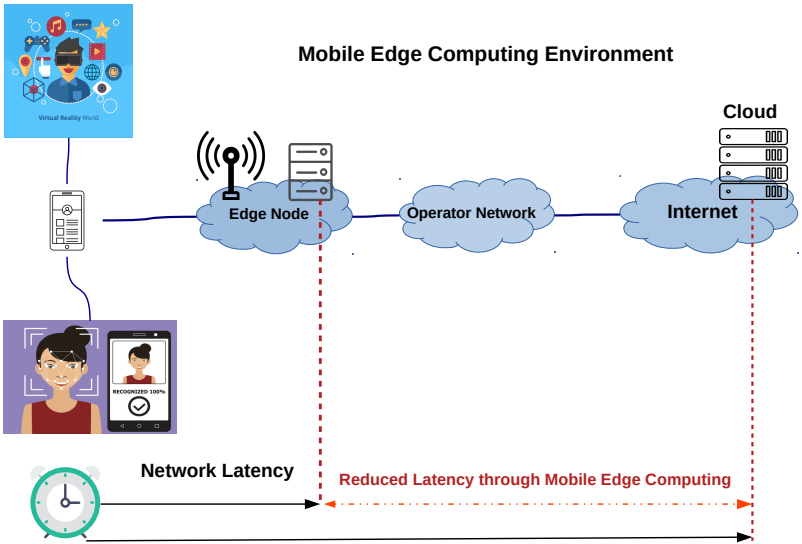
$\Rightarrow$ Ultra-low latency.

$\Rightarrow$ Small capacity.

Network Latency

Reduced Latency through Mobile Edge Computing

The new emerging computing paradigm: extension



$\Rightarrow$ Ultra-low latency.

$\Rightarrow$ **Small capacity**.

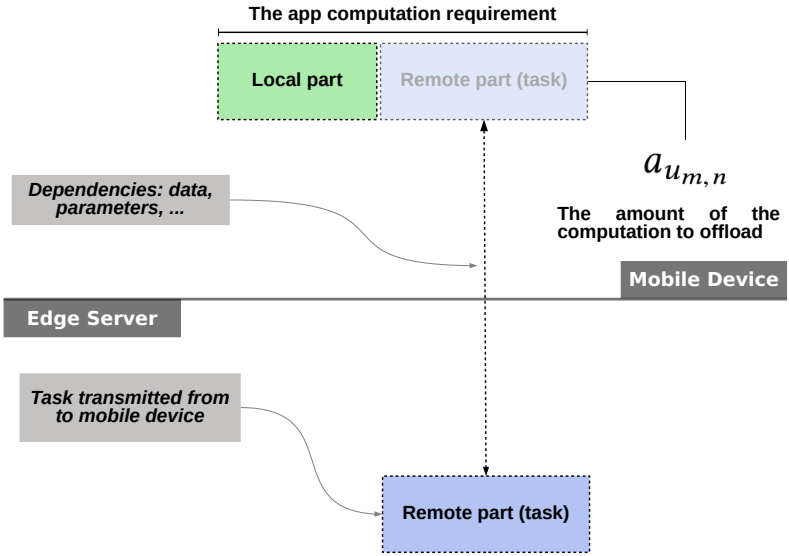
The new emerging computing paradigm: MEC Challenges

1. Placement of the Edge Server (cloudlet) in the network
2. Selection of the Edge Server for whom a user offloads its computation
3. Model of the mobile application: `define the offloadable parts, offloading condition, virtualization technology`
4. Computing resource allocation at the edge server
5. Bandwidth allocation

The new emerging computing paradigm: MEC Challenges

1. Placement of the Edge Server (cloudlet) in the network
2. **Selection of the Edge Server for whom a user offloads its computation**
3. **Model of the mobile application:** `define the offloadable parts, offloading condition, virtualization technology`
4. **Computing resource allocation at the edge server**
5. **Bandwidth allocation**

Computation offloading: model of the application

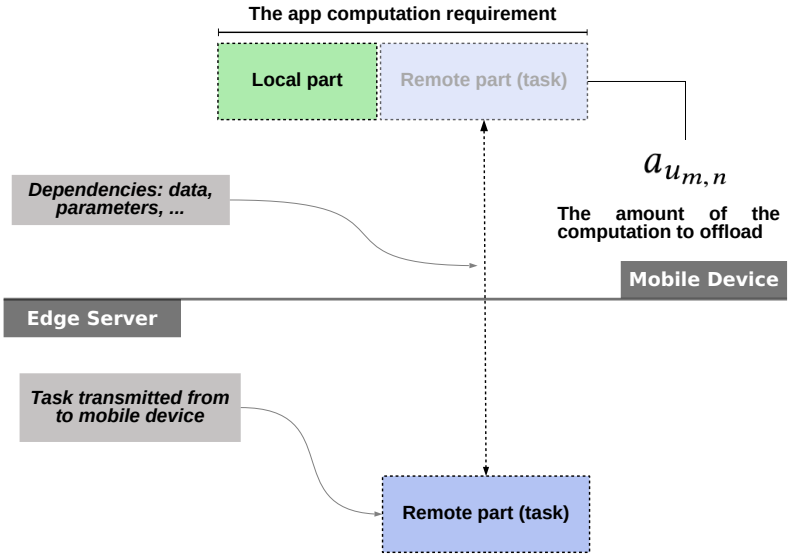


Determine the remote part:

⇒ At the design time: **static offloading decision app**

⇒ At the runtime: **dynamic offloading decision app**

Computation offloading: model of the application

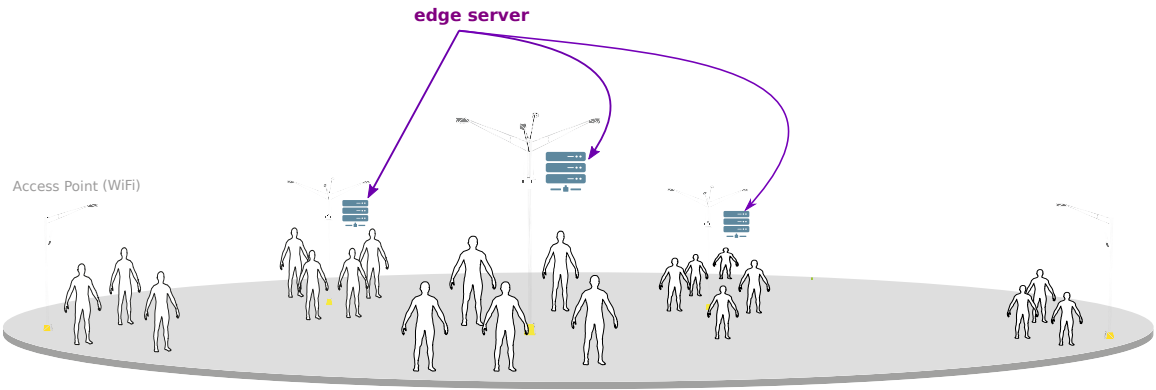


Determine the remote part:

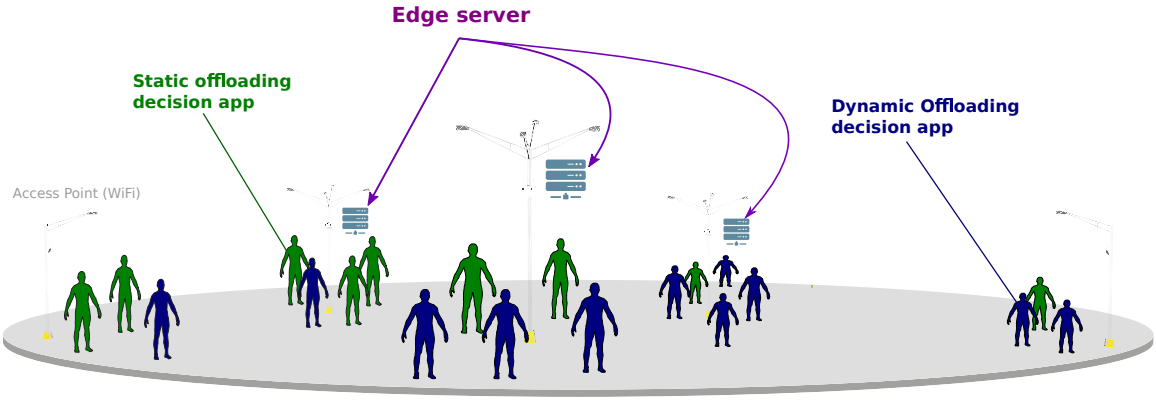
⇒ At the design time: **static offloading decision app**

⇒ At the runtime: **dynamic offloading decision app**

Large MEC: Computation offloading

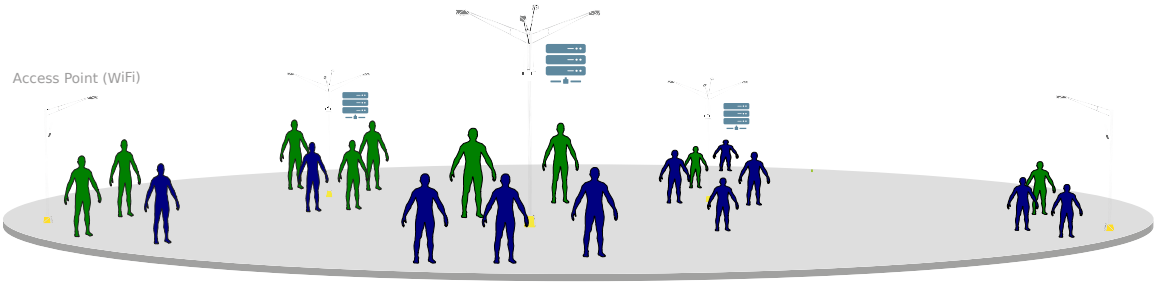


Large MEC: Computation offloading



Large MEC: Computation offloading

Which user should offload? How much computation?
And to which edge server?



Our Offloading Policy

⇒ **Goal:** Determine which user should offload, select an edge server and the amount of the computation to offload.

- Allocate the bandwidth to each user.
- minimize the offloading cost: $cost = \beta * \text{Energy} + (1 - \beta) * \text{Time}$
- assumptions:
  - ⇒ For static offloading decision: $a_{u_{m,n}} = 1$, `the whole computation must be offloaded to MEC.`
  - ⇒ For dynamic offloading decision: $a_{u_{m,n}} \in [0, 1]$, `we must find its optimal value.`

Problem Formulation: multi-user multi-edge server offloading

Minimize $\sum_{m}^{M} \sum_{n}^{N_m} \mathcal{Z}_{u_{m,n}}$

$C1 : \sum_{k=1}^{K} x_{u_{m,n},k} \leq 1, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m$	$\Rightarrow$ Each task can be offload to at most one Edge server
$C2 : y_{u_{m,n}} - \sum_{k=1}^{K} x_{u_{m,n},k} \leq 0, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m$	$\Rightarrow$ Static offloading Decision app must be offloaded
$C3 : T_{u_{m,n}} \leq t_{u_{m,n}}, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m$	$\Rightarrow$ QoS constraint
$C4 : x_{u_{m,n},k} \leq g_{u_{m,n},k}, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m, k \in \mathcal{K}$	$\Rightarrow$ Edge server support Constraint
$C5 : \sum_{m}^{M} (\sum_{n}^{N_m} x_{u_{m,n},k} * c_k) \leq F_k, \forall k \in \mathcal{K}$	$\Rightarrow$ Edge server capacity
$C6 : x_{u_{m,n},k} \in \{0,1\}, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m, k \in \mathcal{K}$	
$C7 : a_{u_{m,n}} \in [0,1], a_{u_{m,n}} \geq y_{u_{m,n}}, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m$	

This problem is NP-hard.

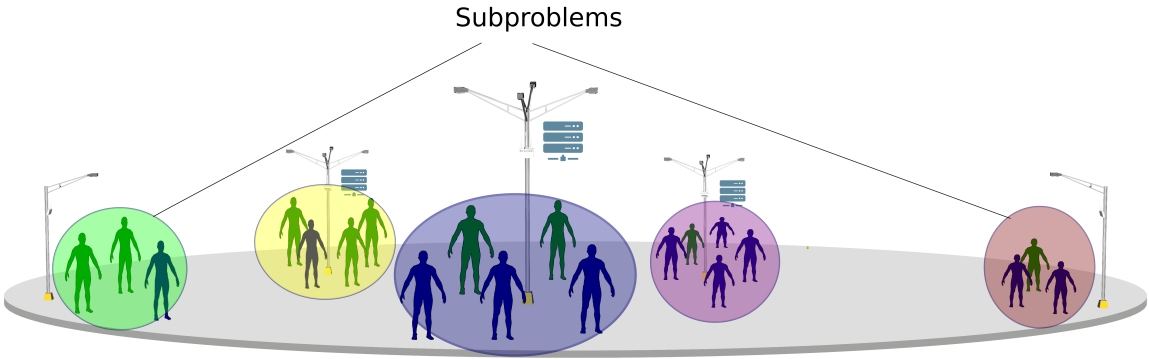
Problem Formulation: multi-user multi-edge server offloading

Minimize $\sum_{m}^{M} \sum_{n}^{N_m} \mathcal{Z}_{u_{m,n}}$

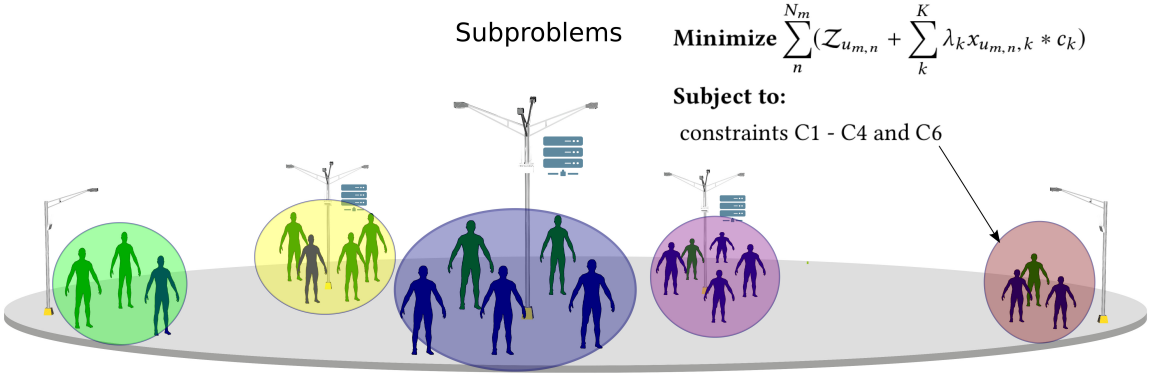
$C1 : \sum_{k=1}^{K} x_{u_{m,n},k} \leq 1, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m$	$\Rightarrow$ Each task can be offload to at most one Edge server
$C2 : y_{u_{m,n}} - \sum_{k=1}^{K} x_{u_{m,n},k} \leq 0, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m$	$\Rightarrow$ Static offloading Decision app must be offloaded
$C3 : T_{u_{m,n}} \leq t_{u_{m,n}}, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m$	$\Rightarrow$ QoS constraint
$C4 : x_{u_{m,n},k} \leq g_{u_{m,n},k}, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m, k \in \mathcal{K}$	$\Rightarrow$ Edge server support Constraint
$C5 : \sum_{m}^{M} (\sum_{n}^{N_m} x_{u_{m,n},k} * c_k) \leq F_k, \forall k \in \mathcal{K}$	$\Rightarrow$ Edge server capacity
$C6 : x_{u_{m,n},k} \in \{0,1\}, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m, k \in \mathcal{K}$	
$C7 : a_{u_{m,n}} \in [0,1], a_{u_{m,n}} \geq y_{u_{m,n}}, \forall m \in \mathcal{M}, u_{m,n} \in \mathcal{N}_m$	

This problem is NP-hard.

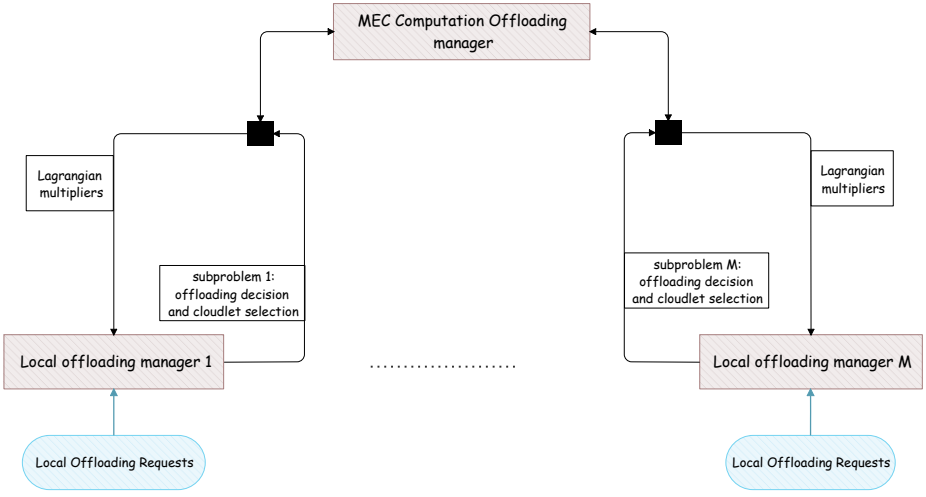
Our proposal: DM2-ECOP algorithm



Our proposal: DM2-ECOP algorithm



Our proposal: DM2-ECOP algorithm



DM2-ECOP: Local offloading manager

1- Estimate the bandwidth allocation to each user using Bianchi model:

$$w_{u_{m,n}} = \frac{B_m(\pi_m)}{\pi_m}$$

- B_m: `is the estimated bandwidth at the AP` m
- π_m: `is the number of users that offload`

2- For each `Static offloading decision` task, select the cloudlet that minimizes $\mathcal{Z}^e_{u_{m,n},k} + \lambda_k c_k$.

DM2-ECOP: Local offloading manager

3- For each `Dynamic offloading decision` task, compute the offloading priority:

$$\xi_{u_{m,n}} = \mathcal{Z}^{l}_{u_{m,n}} - \min_{k \in \mathcal{K}} (\mathcal{Z}^{e}_{u_{m,n},k}); \quad \text{under } a_{u_{m,n}} = 1$$

4- Sort `dynamic offloading decision apps` in decreasing order of $\xi_{u_{m,n}}$

5- Select the cloudlet k that minimizes $\mathcal{Z}^{e}_{u_{m,n},k} + \lambda_k c_k$

6- Compute the optimal value of $a_{u_{m,n}}$

7- when the offloaded task is equal to π_m, all the remaining apps will be performed locally

DM2-ECOP: find the optimal amount of computation to offload

$\Rightarrow$ For each user, the optimal $a_{u_{m,n}}$ is the solution of:

$$min(\mathcal{Z}^{e}_{u_{m,n},k} + \mathcal{Z}^{l}_{u_{m,n}})$$

Subject to: $a_{u_{m,n}} \in [0, 1]$.

$\Rightarrow$ **the optimal value of $a_{u_{m,n}}$ is 1 if and only if :** $\psi_{u_{m,n}} < \mu_{u_{m,n}}$

$\Rightarrow$ Where:

- $\psi_{u_{m,n}} = \dfrac{up_{u_{m,n}}}{\gamma_{u_{m,n}}}$
- $\mu_{u_{m,n}} = \dfrac{w_{u_{m,n}} \cdot [\kappa \cdot f^{3}_{u_{m,n}} \cdot c_k \cdot \beta_{u_{m,n}} + (1 - \beta_{u_{m,n}}) \cdot (c_k - f_{u_{m,n}}) - \beta_{u_{m,n}} \cdot P^{idle}_{u_{m,n}} \cdot f_{u_{m,n}}]}{c_k \cdot f_{u_{m,n}} \cdot (P^{tx/rx}_{u_{m,n}} \cdot \beta_{u_{m,n}} + 1 - \beta_{u_{m,n}})}$

DM2-ECOP: find the optimal amount of computation to offload

$\Rightarrow$ For each user, the optimal $a_{u_{m,n}}$ is the solution of:

$$min(\mathcal{Z}^{e}_{u_{m,n},k} + \mathcal{Z}^{l}_{u_{m,n}})$$
$$\textbf{Subject to: } a_{u_{m,n}} \in [0,1].$$

$\Rightarrow$ **the optimal value of $a_{u_{m,n}}$ is 1 if and only if :** $\psi_{u_{m,n}} < \mu_{u_{m,n}}$

$\Rightarrow$ Where:

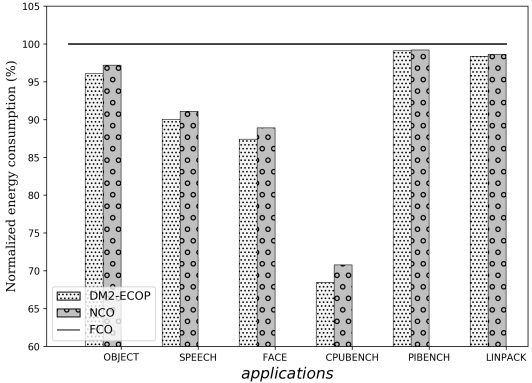
- $\psi_{u_{m,n}} = \dfrac{up_{u_{m,n}}}{\gamma_{u_{m,n}}}$
- $\mu_{u_{m,n}} = \dfrac{w_{u_{m,n}} \cdot [\kappa \cdot f^{3}_{u_{m,n}} \cdot c_k \cdot \beta_{u_{m,n}} + (1-\beta_{u_{m,n}}) \cdot (c_k - f_{u_{m,n}}) - \beta_{u_{m,n}} \cdot P^{idle}_{u_{m,n}} \cdot f_{u_{m,n}}]}{c_k \cdot f_{u_{m,n}} \cdot (P^{tx/rx}_{u_{m,n}} \cdot \beta_{u_{m,n}} + 1 - \beta_{u_{m,n}})}$

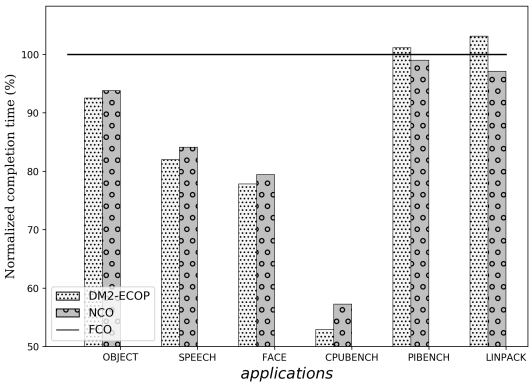
Testbed

Application	$\gamma_{u_{m,n}}$ (Giga CPU cycles)	$up_{u_{m,n}}$ (Kilobyte)	$dw_{u_{m,n}}$ (Byte)	$t_{u_{m,n}}$ (Second)
static offloading decision tasks				
FACE	12.3	62	60	5
SPEECH	15	243	50	5.1
OBJECT	44.6	73	50	13
dynamic offloading decision tasks				
Linpack	50	10240	120	62.5
CPUBENCH	3.36	80	80	4.21
PI BENCH	130	10240	200	163

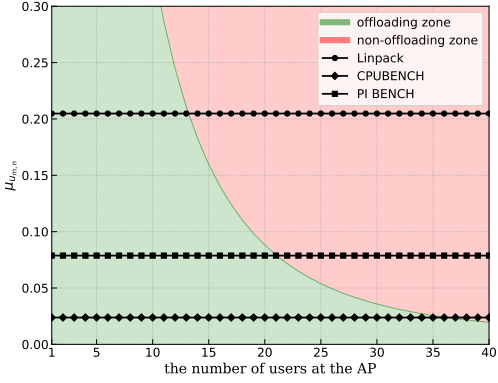
- 20 access point and 4 edge servers.
- WiFi Bandwidth: 150 Mbps.
- Access delay : 5 ms
- Internet delay: 200 ms
- compared to offloading algorithms:
  - NCO: Nearest Cloudlet Offloading.
  - FCO: Full Offloading to Cloud

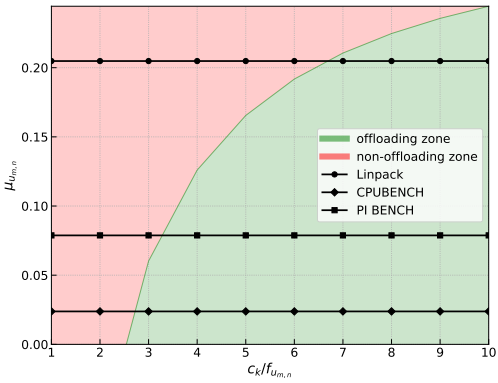
Numerical Results: Energy consumption and Completion time



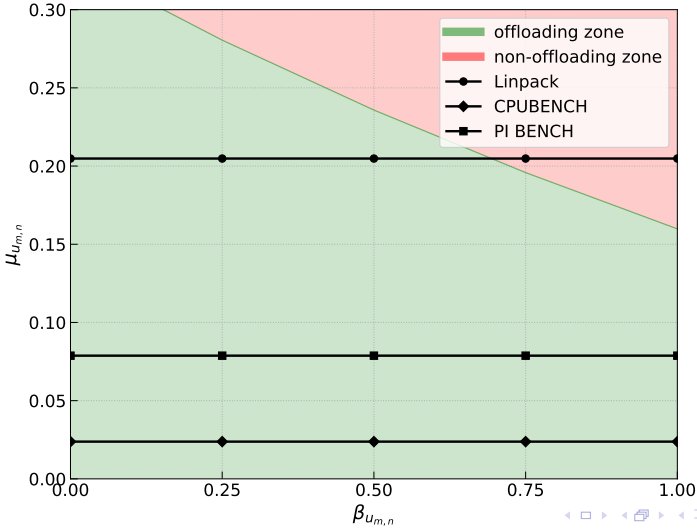


Numerical Results: Optimal $a_{u_{m,n}}$





Numerical Results: Optimal $a_{u_{m,n}}$



Conclusion

- Mobile Edge computing is a very powerful approach to extend the mobile's resource.
- Mobile apps offloading model has an important role on MEC.
- It is very important to consider a dynamic edge server selection.
- Offloading cost model play a key role to determine the efficiency of the offloading policy.

On going work

We are working on:

- Dynamic apps partitioning into offloaded part and local part.
- Designing a framework for offloading in MEC.
- Introducing the operator part in the optimization problem (Operator cost and pricing model)

Thank you