# An Energy Consumption Oriented Offloading Algorithm for Fog Computing

Xiaohui Zhao(✉), Liqiang Zhao, and Kai Liang

State Key Laboratory of Integrated Service Networks,
Xidian University, Xian 710071, Shaanxi, China
`xxhzhao@hotmail.com`

**Abstract.** Fog computing is a promising method for computation offloading by bringing the computation at arms reach, which is characterized by low latency and significant for the delay-sensitive applications. Offloading is effectively to extend the lifetime of battery of mobile device by executing some applications remotely. In this paper, we provide an energy consumption oriented offloading algorithm to save mobile devices energy while satisfying given application response time requirement. We formulate the offloading algorithm as minimizing energy consumption with the constraints of time tolerance and the maximum transmission power. It dynamically selects cloud computing or fog computing to offload computing instead of only relying on cloud computing. The numerical results show that our offloading algorithm can reduce the energy consumption obviously.

**Keywords:** Offloading · Cloud computing · Fog computing · Energy consumption

## 1 Introduction

Nowadays, mobile devices have become an indispensable part of People's Daily life. The demands for mobile devices to run applications with high computation are increasing. The local computation resources are insufficient to run sophisticated task, compared to desktop counterparts. Due to the physical size constraint, mobile devices have limited computation and battery life. Thus, the limitation of energy has been the bottleneck of mobile devices.

Computing offloading [1,2] is a possible strategy to overcome the above bottleneck. It enables resource constrained mobile devices to offload their most energy-consuming tasks to nearby more resourceful servers. The mobile device can offload computing to the cloud. Then the cloud server executes the tasks and provides the mobile device with the results. However, recently research [3] shows that offloading computing to the cloud is not always a good choice because some delay sensitive applications should be completed within their delay tolerance, and the cloud computing has a large transmission delay, especially for the network edge devices. With the availability of nearby resources via fog computing,

J.-H. Lee and S. Pack (Eds.): QShine 2016, LNICST 199, pp. 293–301, 2017.
DOI: 10.1007/978-3-319-60717-7_29

mobile device offer computation offloading with low latency, which is beneficial for the delay sensitive task. Fog computing is a recent computing paradigm that is extending cloud computing towards the edge of network. The computation capacity of fog computing is weaker than that of cloud computing. We formulate the offloading problem as minimizing energy consumption with the constraints of time tolerance and the maximum transmission power.

In the paper, we provide an energy consumption oriented offloading algorithm to overcome the problem above with the consideration of latency tolerance and the transmission power of mobile device. The contributions of our work can be summarized as follows:

We provide the energy consumption oriented offloading algorithm. Firstly, we compute the energy consumption of fog computing and cloud computing, respectively. Then mobile device makes a choice after comparing the magnitude between them. Numerical result shows that the energy consumption oriented offloading algorithm has less energy than that of adopting cloud computing alone.

The rest of the paper is organized as follows. Section 2 presents system model. Section 3 presents the problem formulation and solutions. Numerical results are provided in Sect. 4. Section 5 concludes the paper.

## 2 System Model

We first introduce the system model. This paper shows a three-layer structure consisting of cloud layer, fog layer and UE (user equipment) layer. The smartphone provides mobile computing functionalities to the end user via different applications. The fog node in the fog layer is the connections with cloud layer and the end user, which can provide fog computing capability. The cloud server can provide cloud computing capacity. For simplicity, we assume that there has one mobile device that has a computational intensive task to be completed. In this case, a mobile device is able to find a radio access point within a short distance in the fog server. Similar to many previous studies in mobile cloud computing [4] and mobile networking [5], to enable tractable analysis and get useful insights, we consider a quasi-static scenario where the mobile device remains unchanged during a computation offloading period (e.g., within several seconds), while may change across different periods.

### 2.1 Delay Analysis

#### (1) Fog Computing

In the case of computation offloading, the latency incorporates the time to transmit the input bits to the fog server necessary to enable the execution, the time necessary for the fog server to execute the instructions, and the time to send the result back to the UE. More specifically, the overall latency experienced by mobile device can be written as

$$\Delta_f = \Delta_{f,1}^t + \Delta_{f,2}^t \tag{1}$$

Where $\varDelta_{f,1}^{t}$ is the time for the mobile device to transfer the input bits to its fog server, $\varDelta_{f,2}^{t}$ is the time for the fog server to execute instructions.

$$\varDelta_{f,1}^{t} = \frac{D}{R\left(p_{tr}\right)} \tag{2}$$

$$\varDelta_{f,2}^{t} = \frac{M}{S_f} \tag{3}$$

Where $D$ is the size of the input bits, and $M$ is the number of instructions, which is necessary for the fog server to execute, and $S_f$ is the computation capacity of the fog server. $R\left(p_{tr}\right)$ is the wireless transmission rate.[1]

$$R\left(p_{tr}\right) = \log\left(1 + \frac{p_{tr}h^2}{N_0}\right) \tag{4}$$

Where $h$ is the channel fading coefficient, and $N_0$ denotes the noise power.

### (2) Cloud Computing

The latency incorporates the time to transmit the input bits to the fog server necessary to enable the execution transfer, the time necessary for the fog server to transfer the input bits to the cloud server, and the time necessary for the cloud server to execute the instructions, and the time necessary to send result back to the UE. The overall latency experienced by the mobile device is

$$\varDelta_c = \varDelta_{c,1}^{t} + \varDelta_{c,2}^{t} + \varDelta_{c,3}^{t} \tag{5}$$

Where $\varDelta_{c,1}^{t}$ is the time necessary for the mobile device to transfer the input bits to its fog server; $\varDelta_{c,2}^{t}$ is the time necessary for the fog server to transfer the input bits to the cloud server; $\varDelta_{c,3}^{t}$ is the time for the cloud server to execute the instructions.

$$\varDelta_{c,1}^{t} = \frac{D}{R\left(p_{tr}\right)} \tag{6}$$

$$\varDelta_{c,2}^{t} = \frac{D}{B} \tag{7}$$

$$\varDelta_{c,3}^{t} = \frac{M}{S_c} \tag{8}$$

Where $B$ denotes the bandwidth between fog and cloud server, and $S_c$ is the computation capacity of cloud server.

For many applications (e.g., face recognition), the size of the computation outcome in general is much smaller than that of computation input data including the mobile system settings, program codes and input parameters. We ignore the time it needs to return the outcome to the mobile device.

---

[1] All the $\log(\cdot)$ functions are of base 2 by default.

### 2.2 Energy Consumption Analysis

#### (1) **Fog Computing**

Energy consumption of offloaded services is

$$E_f\left(p_{tr}\right)=E_{f,1}+E_{f,2} \tag{9}$$

Where $E_{f,1}$ is the idle energy consumption of mobile device when the fog server executes the instructions, and $E_{f,2}$ is the energy consumption for the mobile device necessary to transfer the input bits to the fog.

$$E_{f,1}=\frac{Mp_i}{S_f} \tag{10}$$

$$E_{f,2}=\frac{Dp_{tr}}{R\left(p_{tr}\right)} \tag{11}$$

Where $p_i$ and $p_{tr}$ are the idle power and the transmission power of the mobile device, respectively. $M$ denotes the size of instructions which are executed on the fog server.

#### (2) **Cloud Computing**

Energy consumption of offloaded service is

$$E_c\left(p_{tr}\right)=E_{c,1}+E_{c,2}+E_{c,3} \tag{12}$$

Where $E_{c,1}$ is the energy consumption for the mobile device transmitting the input bits to the fog server, $E_{c,2}$ is the idle energy consumption of the mobile device when the fog server transmit the input bits to the cloud server, and $E_{c,3}$ is the idle energy consumption during the cloud server executes the instructions.

$$E_{c,1}=\frac{Dp_{tr}}{R\left(p_{tr}\right)} \tag{13}$$

$$E_{c,2}=\frac{Dp_i}{B} \tag{14}$$

$$E_{c,3}=\frac{Mp_i}{S_c} \tag{15}$$

## 3 Problem Formulation and Solution

### 3.1 Problem Formulation

In this paper, we aim at minimizing the energy consumption at the mobile device. The delay tolerance constraints based on the $p_{tr}$ can be given by

$$\varDelta_{(\cdot)}\leq T \tag{16}$$

Where $T$ is the delay tolerance and $\Delta_{(\cdot)}$ denotes the overall time of fog computing when $(\cdot)$ represents $f$. $\Delta_{(\cdot)}$ denotes the overall time of cloud computing when $(\cdot)$ represents $c$.

The transmission power constraints based on $P_{\max}$ can be given by

$$0 \le p_{tr} \le p_{\max} \tag{17}$$

Combining (16) with (17), we have

$$p_{\min} \le p_{tr} \le p_{\max} \tag{18}$$

We formulate the optimization problem as minimizing the energy consumption for the mobile device computation offloading while satisfying the transmission power of mobile device constraints as follows:

$$\begin{aligned} &\arg \min_{p_{tr}} \; E\left(p_{tr}\right) \\ &\text{s.t.} \;\; (18) \end{aligned} \tag{19}$$

## 3.2 Solution

To achieve the minimum energy consumption, we introduce the solution in fog computing as an example.

$$\begin{aligned} &\min_{p_{tr}} \;\; \frac{Mp_i}{S_f} + \frac{Dp_{tr}}{\log\left(1+\frac{p_{tr}h^2}{N_0}\right)} \\ &\text{s.t.} \; p_{\min} \le p_{tr} \le p_{\max} \end{aligned} \tag{20}$$

Because $\frac{Mp_i}{S_f}$ is a constant, the above optimization problem can be equivalent to

$$\begin{aligned} &\min_{p_{tr}} \; \frac{Dp_{tr}}{\log\left(1+\frac{p_{tr}h^2}{N_0}\right)} \\ &\text{s.t.} \; p_{\min} \le p_{tr} \le p_{\max} \end{aligned} \tag{21}$$

Due to $\frac{Dp_{tr}}{\log\left(1+\frac{p_{tr}h^2}{N_0}\right)}$ is non-convex, the problem (21) is non-convex for $p_{tr}$.

## 3.3 Algorithm

In this subsection, we provide an energy consumption oriented offloading algorithm, as shown in Algorithm 1. We know that the optimization problem in (21) is non-convex due to the fractional form of the objective function. We note that there is no standard approach for solving non-convex optimization problems. In order to solve the optimization problem above, we introduce a transformation to handle the objective function via nonlinear fractional programming [6].

**Algorithm 1.** Optimal energy consumption algorithm design

```
Input: $D, h^2, N_0$
Output: Optimal energy consumption and transmission power of mobile device $p_{tr}$
Initialize the maximum number of iteration $I_{max}$ and the maximum tolerance $\varDelta$;
Set minimize energy efficiency $q = 0$ and iteration index $n = 0$;
Repeat Main Loop
    Solve the loop problem in $\frac{Dp_{tr}}{\log\left(1+\frac{p_{tr}h^2}{N_0}\right)} = q$ for a given $p_{tr}$ and obtain the energy efficiency $q^*$;
    $p_{tr} = p_{min}, q = 0$;
    if $Dp_{tr} - q\log\left(1+\frac{p_{tr}h^2}{N_0}\right) < \varDelta$ then
        Convergence=true;
        Return $p_{tr}^* = p_{tr}$ and $q^* = \frac{Dp_{tr}}{\log\left(1+\frac{p_{tr}h^2}{N_0}\right)}$.
    Else
        Set $q = \frac{Dp_{tr}}{\log\left(1+\frac{p_{tr}h^2}{N_0}\right)}$ and $n = n + 1$;
        Convergence=false;
Until convergence=true or $n = I_{max}$
End
```

Without loss of generality, we define the minimum energy consumption $q^*$ of the considered system as

$$q^* = \frac{Dp_{tr}^*}{\log\left(1+\frac{p_{tr}^*h^2}{N_0}\right)} = \min_{p_{tr}} \frac{Dp_{tr}}{\log\left(1+\frac{p_{tr}h^2}{N_0}\right)} \tag{22}$$

We are now ready to introduce the following Theorem 1 in [6].

Theorem 1: The maximum weighted energy efficiency $q^*$ is achieved if and only if

$$\min_{p_{tr}} Dp_{tr} - q^*\log\left(1+\frac{p_{tr}h^2}{N_0}\right) = Dp_{tr}^* - q^*\log\left(1+\frac{p_{tr}^*h^2}{N_0}\right) = 0 \tag{23}$$

for $Dp_{tr} \geq 0$ and $\log\left(1+\frac{p_{tr}h^2}{N_0}\right) \geq 0$.

The proposed algorithm is summarized in Algorithm 1. Its convergence to the optimal energy consumption is guaranteed if we are able to solve the inner problem (23) in each iteration.

## 4 Numerical Analysis

In this section, we evaluate the performance for the proposed energy consumption oriented offloading algorithm. We have used Matlab in the simulation. Throughout the simulations, some assumptions are made unless stated otherwise. For simplicity, the system is made up of a fog server, a cloud server and a mobile device. Assume that the mobile device has a delay sensitive application to be executed remotely. The basic parameters throughout the simulations unless otherwise specified, are as follows. The power of the wireless channel noise is $N_0 = 10^{-7}$ watt. We assume that the channel is ideal and the fading coefficient is $h^2 = 1$. The idle power and maximum transmission power of the mobile device are $p_i = 0.3\,\mathrm{W}$ and $p_{\max} = 1.5\,\mathrm{W}$ [7], respectively. The delay tolerance of mobile device is considered as $T = 1\,\mathrm{s}$. The bandwidth is assumed to be $B = 500$ MHz. The size of the exchange data is $D = 20$. The capacity of the fog server is $F = 200$. $C$ denotes the capacity of the cloud server. The ratio between the capacity of the cloud server and the fog server is $S$.

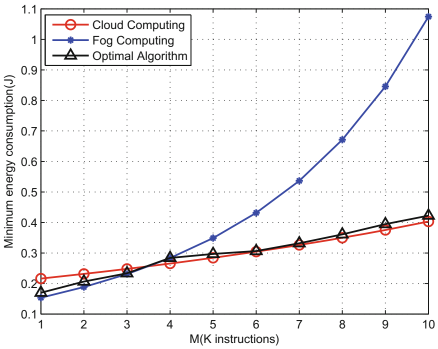

**Fig. 1.** Minimum energy cost versus instruction

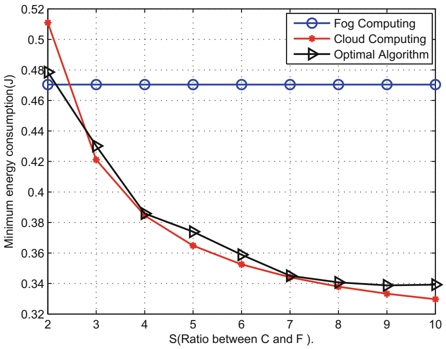

**Fig. 2.** Minimum energy consumption versus capacity coefficient with fog and cloud computing

Figure 1 shows the impact of the number of instructions on the energy consumption of the cloud computing and the fog computing. From the figure, we can see that the minimum energy consumption of the fog computing raises faster than the other two methods with the increase of the size of instructions. Basically, the proposed algorithm can achieves the minimum energy consumption with the consideration of the execution time. When the size of instructions is smaller than 3.5, the proposed algorithm choose fog computing and vice versa. The overall energy consumption is less than that of adopting fog computing or cloud computing alone.

Figure 2 depicts the minimum energy cost versus the coefficient between the capability of the cloud and fog computing. With the increase of $S$, the capacity of the fog computing remains unchanged. Nevertheless, the capacity of the cloud computing is increasing linearly. This leads to a reduction in the execution time of the cloud computing which can significantly reduce the energy consumption of mobile device.

Figure 3 shows the minimum energy consumption by the proposed algorithm. The figure shows that the algorithm converges in ten times iterations which demonstrates that the proposed algorithm can obtain the minimum energy consumption in a finite number of iterations.

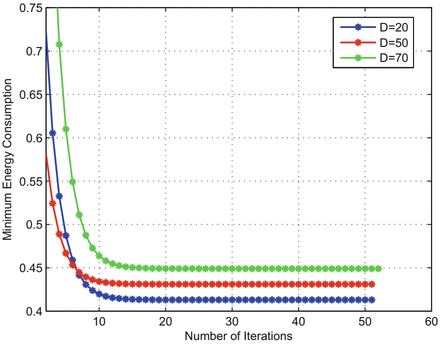

**Fig. 3.** Minimum energy cost versus iterations

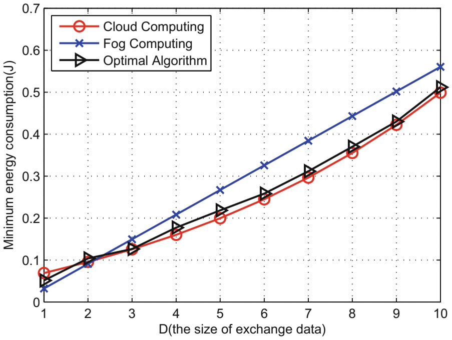

**Fig. 4.** Minimum energy cost versus the size of exchange data for fog and cloud computing

Figure 4 shows the minimum energy cost versus the size of the exchange data between the mobile device and the remote server. Both the energy of the fog computing and cloud computing increase with the increase of the size of $D$. The energy consumption of the proposed algorithm is better than that adopts cloud computing alone obviously.

## 5 Conclusion

In this paper, we introduce fog computing in a simple three layer architecture to minimize the energy consumption of mobile device. We propose an energy consumption oriented offloading algorithm to minimize the energy consumption. By adjust the transmitting power, the paper minimize the energy consumption of the system. The numerical results show that the proposed algorithm can achieve better performance. For simplicity, we introduces one user system model, but in the future, we will study the multi-user model.

**Acknowledgement.** This work was supported in part by National Natural Science Foundation of China (61372070), Natural Science Basic Research Plan in Shaanxi Province of China (2015JM6324), Ningbo Natural Science Foundation (2015A610117), National Science and Technology Major Project of the Ministry of Science and Technology of China (2015zx03002006-003), and the 111 Project (B08038).

## References

1. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for VM-based cloudlets in mobile computing. In: IEEE Pervasive Computing, vol. 8, no. 4, October 2009
2. Kwon, Y.W., Tilevich, E.: Power-efficient and fault-tolerant distributed mobile execution. In: Proceedings of ICDCS (2012)

3. Hassan, M.A., Bhattarai, K., Chen, S.: vUPS: virtually unifying personal storage for fast and pervasive data accesses. In: Uhler, D., Mehta, K., Wong, J.L. (eds.) MobiCASE 2012. LNICSSITE, vol. 110, pp. 186–204. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36632-1_11
4. Wen, Y., Zhang, W., Luo, H.: Energy-optimal mobile application execution: taming resource-poor mobile devices with cloud clones. In: Proceedings of IEEE INFOCOM, pp. 2716–2720 (2012)
5. Wu, S., Tseng, Y., Lin, C., Sheu, J.: A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks. Comput. J. **45**(1), 101–110 (2002)
6. Dinkelbach, W.: On nonlinear fractional programming. Manag. Sci. **13**, 492–498 (1967). http://www.jstor.org/stable/2627691
7. Deng, S., Huang, L., Taheri, J., Zomaya, A.Y.: Computation offloading for service workflow in mobile cloud computing. IEEE Trans. Parallel Distrib. Syst. **26**(12), 3317–3329 (2015)