PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT

IEEE GLOBECOM 2019

Authors: **Shubha Brata Nath**[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]

[1]Indian Institute of Technology Kharagpur, India,
[2]Indian Institute of Technology Guwahati, India,
[3]Techno International New Town, India

Introduction

- **Cloud Computing and Internet of Things (IoT) applications:** Short bursty flows generated by the IoT applications increase the response time in cloud.
- **Fog Computing** [1]**:** Fog computing delivers user desired quality of service for IoT applications.
- **Micro-service architecture** [2]**:** In this architecture, each application is developed in the form of a bunch of loosely coupled lightweight services.
- **Containers:** It is a lightweight virtualization technology that provides service isolation, lightweight migration.
- **Pick-Test-Choose (PTC) framework**: PTC uses bayesian optimization [3] based iterative reinforcement learning algorithm.

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Challenges

- How can we cater to the primary workloads of the fog devices?
- How to provide micro-service isolation?
- How can we handle migration of micro-services?
- How to make the deployment and migration framework lightweight?
- How to monitor the highly dynamic in-network processing architecture?

PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT
Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Objective

- Given the communication graph and available resources, the micro-service placement problem finds an allocation schedule for each micro-service with required instructions and resources such that the maximum response time taken by the applications is minimized.

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Related Works

- Saurez et al., ACM DEBS 2016 [4]: Developed a programming infrastructure which launches the application modules and performs the migration of these modules as containers.
- Souza et al., FGCS 2018 [5]: Developed a micro-service offloading framework by exploiting traditional memory allocation strategies (best-fit,first-fit).
- Goncalves et al., IEEE ISCC 2018 [6]: Provided a VM placement and migration framework to maximize the number of applications placed in fog while reducing the overall application latency.
- Wang et al., IEEE CloudNet 2017 [7]: Placed the applications as virtual machines in the edge or fog devices.

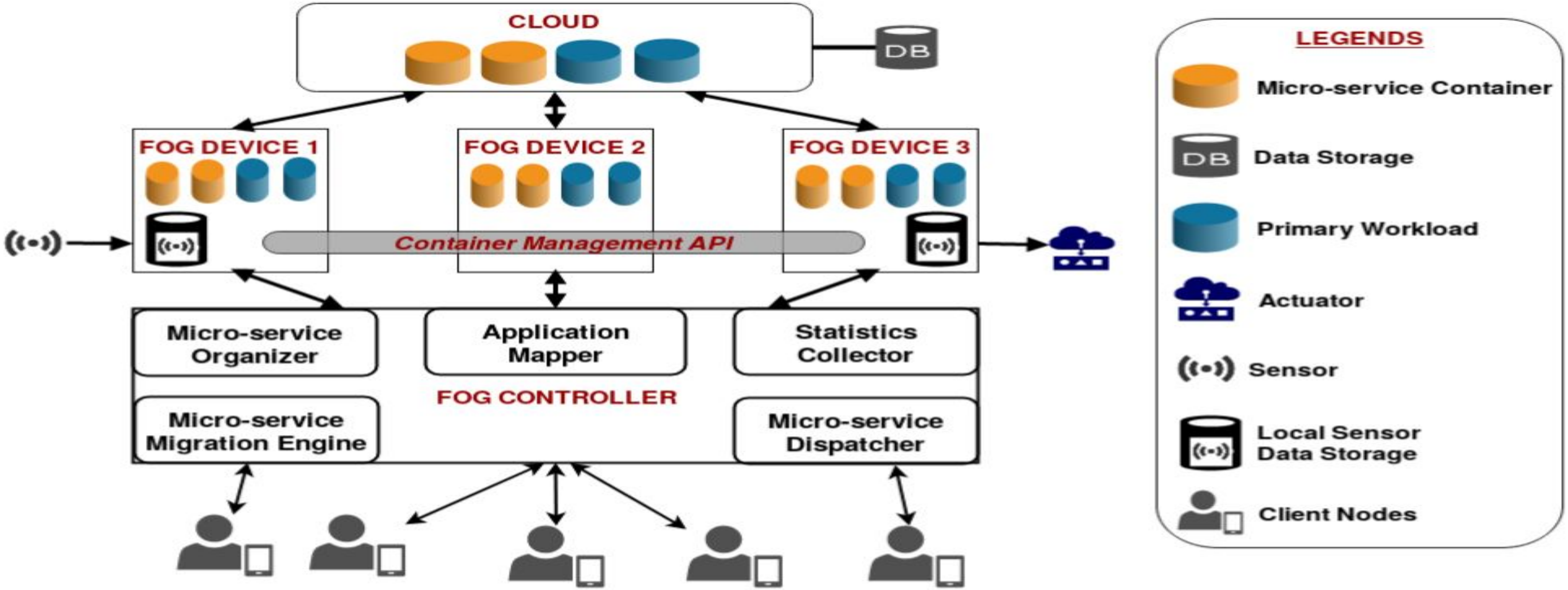
Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Related Works (contd.)

- Ahmed et al., IEEE EDGE 2018 [8]: Proposed a container driven framework to speed-up application deployment procedure.
- Yigitoglu et al., IEEE AIMS 2017 [9]: Proposed task scheduling mechanisms using containers in fog computing architecture.
- Taneja et al., IFIP/IEEE IM 2017 [10]: Proposed a module mapping algorithm for efficient utilization of resources in Fog-Cloud infrastructure.
- Elgamal et al., IEEE CLOUD 2018 [11]: Proposed a scalable dynamic programming algorithm called DROPLET, to partition operations in IoT applications across shared edge and cloud resources, while minimizing completion time of the end-to-end operations.

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

System Architecture with PTC framework



PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

System Architecture: Components

- Client Node: It generates a fog service request.
- Fog Controller Device: It manages the computation offloading in the fog devices.
- Fog Device: Fog devices are responsible for the actual computation and storage related to the applications provided by the fog-cloud.
- Cloud: It manages the overloaded scenarios when it is not possible to cater to all the micro-service demands inside the fog.

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Micro-service Placement as an Optimization

- Objective Function with Constraints:

$$\underset{A}{\text{minimize}} \qquad T_{Resp}^{max}(A)$$

subject to:

$$\mathcal{R}(A) \geq \vec{Z}$$

- $T_{Resp}^{max}(A)$ is maximum response time taken by the applications.
- $\mathcal{R}(A)$ is the resource availability vector.

$\vec{Z}$ = $(z_{i,q,t} = 0 : i = (0, \ldots, n), q = (0, \ldots, f), t = (0, \ldots, l))$.

- n is the number of fog nodes.
- f is total number of resource types.
- l is the number of time slots.

PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT
Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Solution: Micro-service Placement using BO

- Bayesian Optimization:
  - As the problem is NP-hard and difficult to implement due to high monitoring overhead, we propose a reinforcement learning framework which requires very little monitoring and can perform in the presence of noise. For this purpose, we design Bayesian Optimization (BO) based mechanism, PTC.
- We assume that the utility function $T_{Resp}^{max}(A)$ follows a normal distribution.
- Acquisition Function: It notifies BO to choose the configurations for subsequent experiments. It leads the framework towards optimum configuration.

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Solution: Micro-service Placement using BO (contd.)

- Acquisition Function:

$$\mu(a_u) = \mathbb{E}(T^{max}_{Resp}(a_u))$$
$$K(a_u, a_v) = \mathbb{E}\left(\left(T^{max}_{Resp}(a_u) - \mu(a_u)\right)\left(T^{max}_{Resp}(a_v) - \mu(a_v)\right)\right)$$

- Mean function.
- Covariance kernel function.

Where, a_u and a_v are the allocation matrices.

$$U_{min} = \min_{a \in \mathcal{D}_d}(T^{max}_{Resp}(a))$$
$$\pi = \frac{U_{min} - \mu(a_d)}{\sigma(a_{d-1}, a_d)}$$
$$\sigma(a_{d-1}, a_d) = \sqrt{K(a_{d-1}, a_d)}.$$

- U_{min} is the minimum value of the average response time observed till iteration d.
- $\sigma(a_{d-1}, a_d)$ is the standard deviation between the allocation matrices a_{d-1} and a_d.

PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT
Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Solution: Micro-service Placement using BO (contd.)

- Acquisition Function:

$$EI(A|\mathcal{D}_d) = \begin{cases} 0 & \text{if:}\sigma(a_{d-1}, a_d) = 0 \\ ((U_{min} - \mu(a_d))\,\Phi(\pi)) + (\phi(\pi)\sigma(a_{d-1}, a_d)) \\ & \text{Otherwise} \end{cases}$$

- A is an allocation matrix. D_d is the set of prior observations after d iterations. Φ,φ as standard normal cumulative distribution function and standard normal density function respectively.

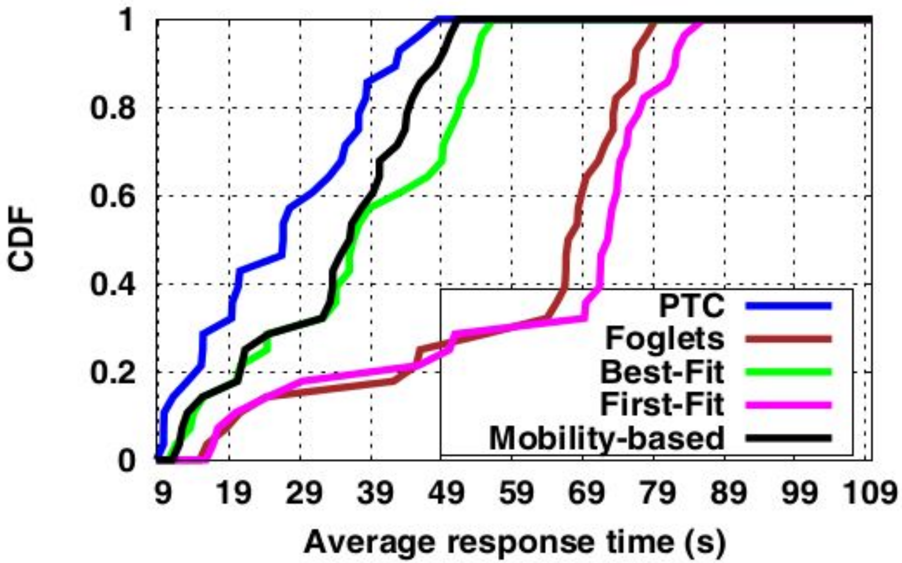
PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT
Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

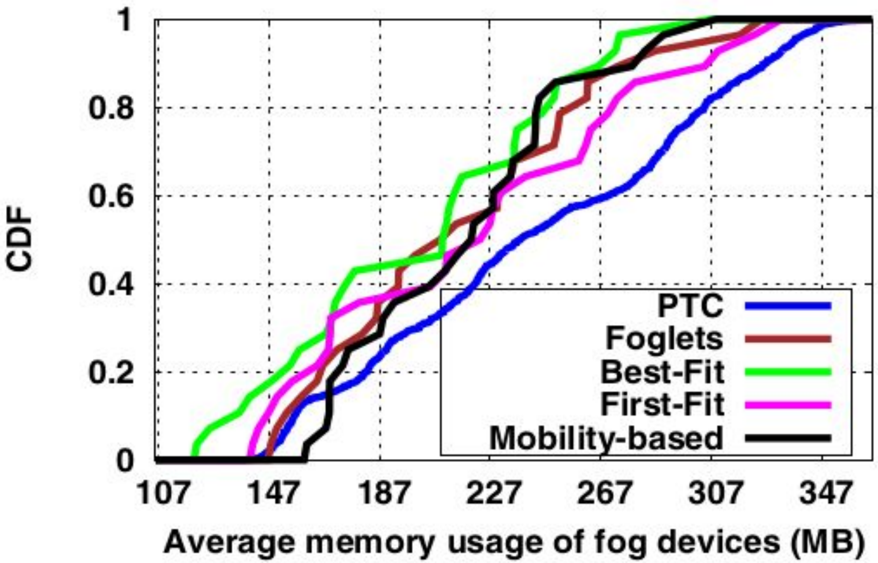
Evaluation: Implementation Details

- Testbed Components:
  - Raspberry Pi 3 model b single-board computers as the fog devices
  - Docker for micro-service isolation
  - Institute private cloud
- Baselines:
  - Foglets [4]
  - Best-Fit [5], First-Fit [5]
  - Mobility-based [6]
- Simulator Used:
  - iFogSim [12]

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Evaluation in Testbed: Results

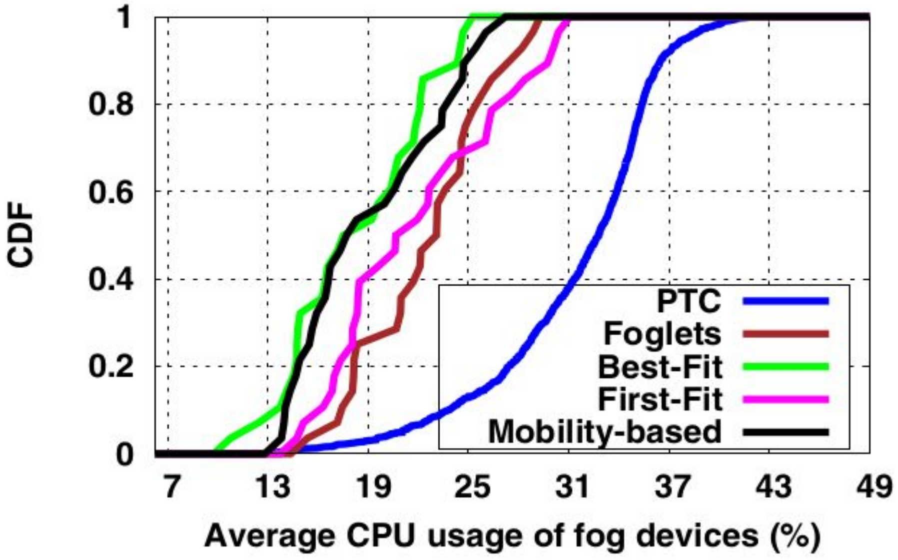


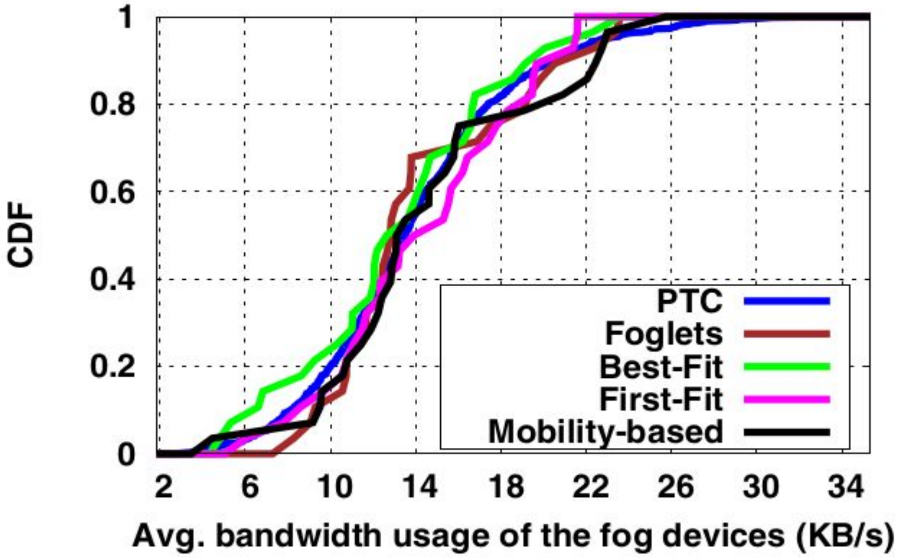


- PTC reduces the average response time by BO based reinforcement learning.
- PTC has more average memory usage of the fog devices than the baselines.

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Evaluation in Testbed: Results (contd.)

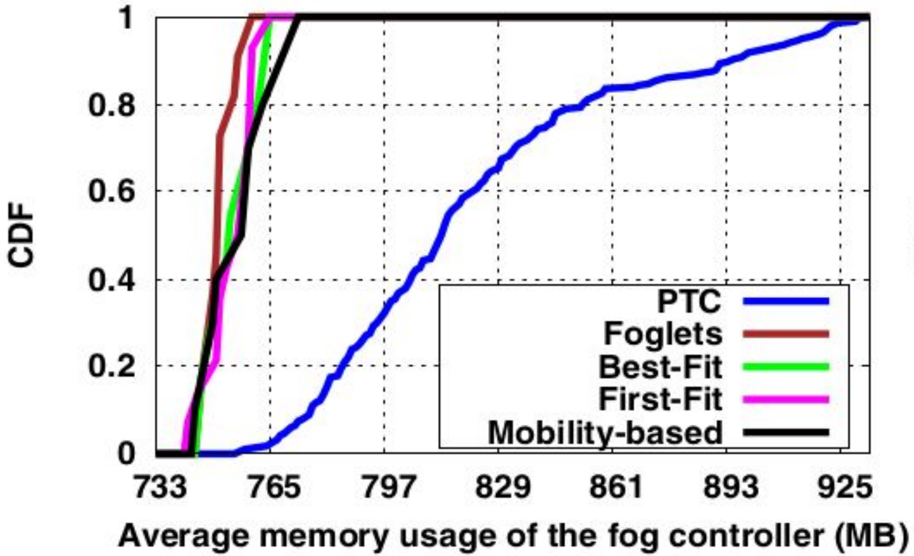


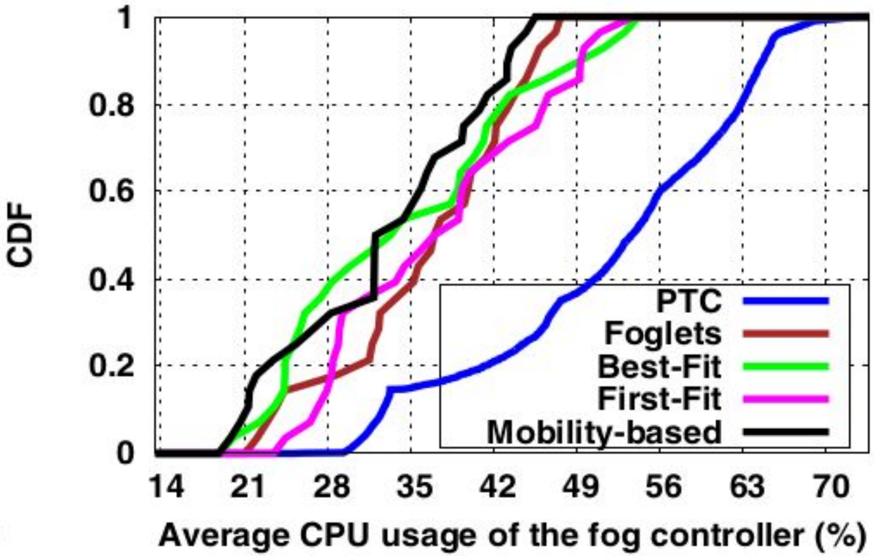


- The average CPU usage of the fog devices is higher in PTC.
- The bandwidth usage of the fog is in the range of 3 KB/s -36 KB/s

PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT
Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Evaluation in Testbed: Results (contd.)

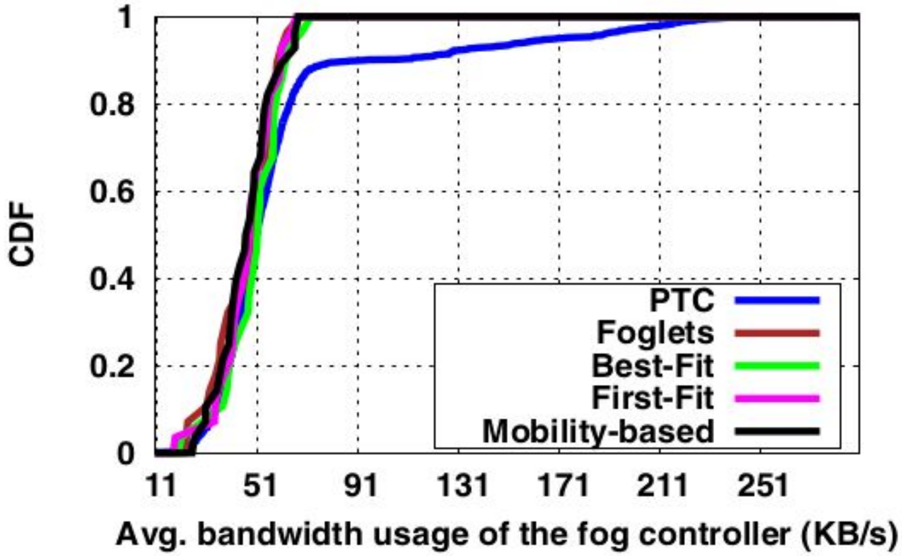


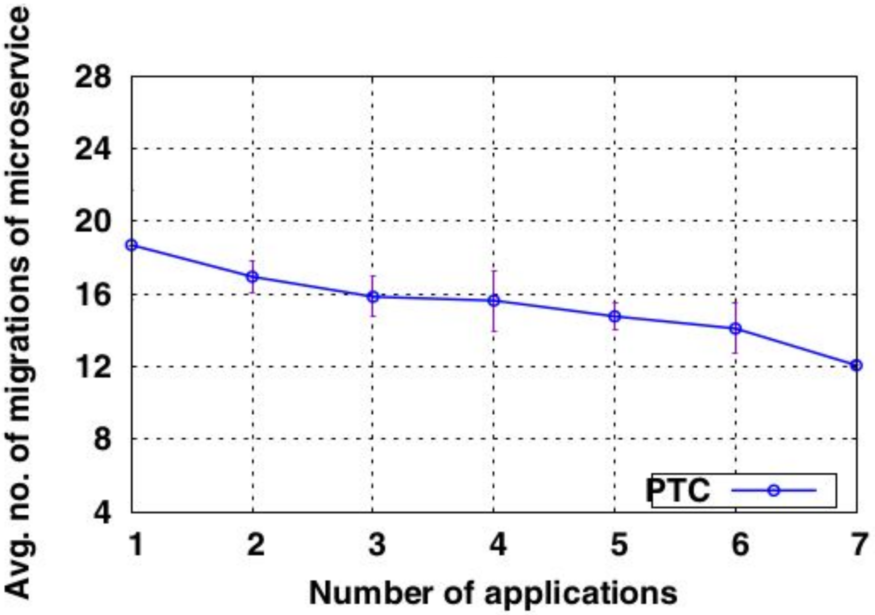


- PTC has more average memory usage for the controller device due to the overhead of multiple iterations.
- The average CPU usage of the controller device is higher in PTC.

PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT
Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Evaluation in Testbed: Results (contd.)

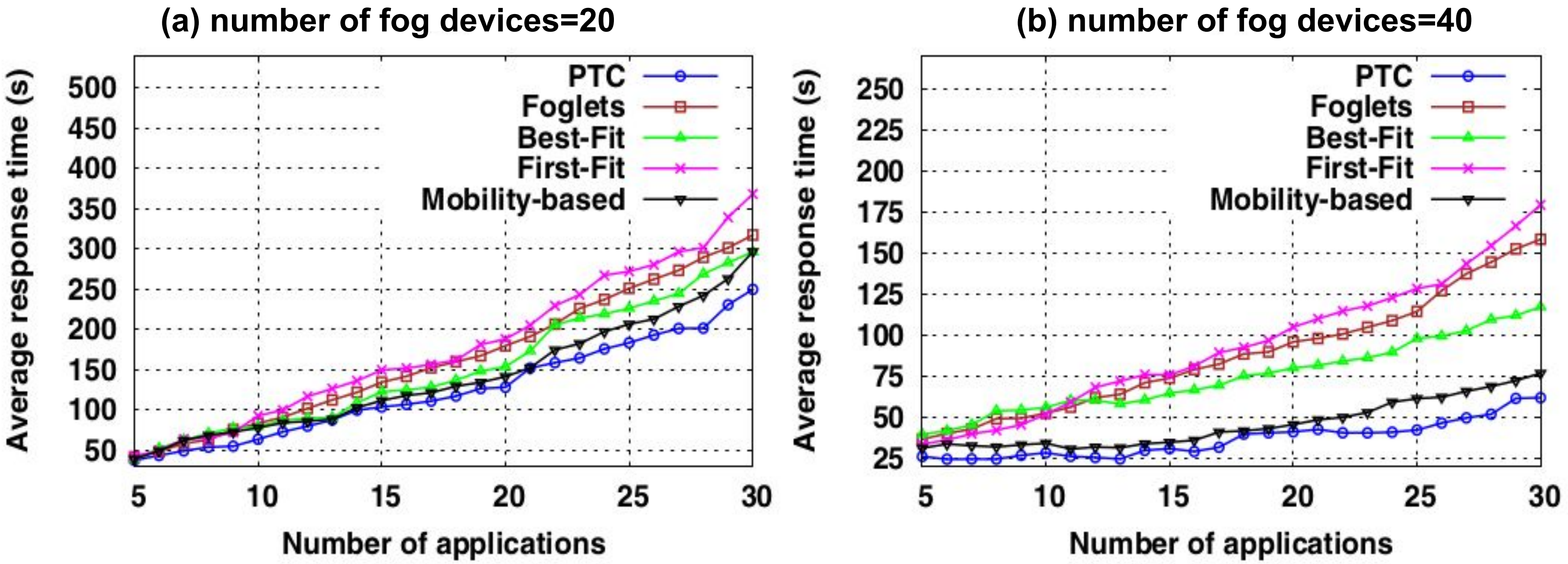




- The bandwidth usage of the fog controller is in the range of 11 KB/s -291 KB/s
- Though the total number of migrations increases, the average number of migrations decreases with the increase in the number of applications.

PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT
Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Evaluation in iFogSim Simulator: Results



- The average response time is reduced significantly in PTC, as the number of available fog devices increases.

PTC: Pick-Test-Choose to Place Containerized Micro-services in IoT

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]

[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

Conclusion

- Primary workload of the fog devices needs to be considered.
- The problem of containerized micro-service placement in IoT has been studied.
- We have formulated the micro-service placement problem as an optimization.
- BO-based iterative reinforcement learning mechanism is proposed.
- PTC is tested in an in-house testbed setup as well as in iFogSim simulator.
- It is observed that PTC can minimize the response time of the system.
- In the future, we plan to evaluate performance of PTC under different services.

Shubha Brata Nath[1], Subhrendu Chattopadhyay[2], Raja Karmakar[3], Sourav Kanti Addya[1], Sandip Chakraborty[1] and Soumya K Ghosh[1]
[1]Indian Institute of Technology Kharagpur, India, [2]Indian Institute of Technology Guwahati, India, [3]Techno International New Town, India

References

1. F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog computing: A platform for internet of things and analytics,” in Big data and internet of things: A roadmap for smart environments. Springer, 2014, pp. 169–186.
2. S. Hassan and R. Bahsoon, “Microservices and their design trade-offs: A self-adaptive roadmap,” in Proc. of the 13th IEEE SCC. IEEE, 2016, pp. 813–818.
3. J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” in Advances in neural information processing systems, 2012, pp. 2951–2959.
4. E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, and B. Ottenwälder, “Incremental deployment and migration of geo-distributed situation awareness applications in the fog,” in Proc. of the 10th ACM DEBS. ACM, 2016, pp. 258–269.

References (contd.)

5. V. Souza, X. Masip-Bruin, E. Marín-Tordera, S. Sànchez-López, J. Garcia, G.-J. Ren, A. Jukan, and A. J. Ferrer, “Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures,” Future Generation Computer Systems, vol. 87, pp. 1–15, 2018.
6. D. Gonçalves, K. Velasquez, M. Curado, L. Bittencourt, and E. Madeira, “Proactive virtual machine migration in fog environments,” in Proc. of the 23rd IEEE ISCC. IEEE, 2018, pp. 00 742–00 745.
7. W. Wang, Y. Zhao, M. Tornatore, A. Gupta, J. Zhang, and B. Mukherjee, “Virtual machine placement and workload assignment for mobile edge computing,” in Proc. of the 6th IEEE CloudNet. IEEE, 2017, pp. 1–6.

References (contd.)

8. A. Ahmed and G. Pierre, “Docker Container Deployment in Fog Computing Infrastructures,” in Proc. of the IEEE EDGE. IEEE, July 2018, pp. 1–8.
9. E. Yigitoglu, M. Mohamed, L. Liu, and H. Ludwig, “Foggy: A Framework for Continuous Automated IoT Application Deployment in Fog Computing,” in Proc. of the 6th IEEE AIMS. IEEE, 2017, pp. 38–45.
10. M. Taneja and A. Davy, “Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm,” in Proc. of the 15th IFIP/IEEE IM. IFIP/IEEE, 2017, pp. 1222–1228.
11. T. Elgamal, A. Sandur, P. Nguyen, K. Nahrstedt, and G. Agha, “DROPLET: Distributed Operator Placement for IoT Applications Spanning Edge and Cloud Resources,” in Proc. of the 11th IEEE CLOUD. IEEE, July 2018, pp. 1–8.

References (contd.)

12. H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, “iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments,” Weily Software: Practice and Experience, vol. 47, no. 9, pp. 1275–1296, 2017.

Thank You