Grouping Computational Data in Resource Caches of Edge-Fog Cloud

Nitinder Mohan† Pengyuan Zhou† Keerthana Govindaraj‡ Jussi Kangasharju†
†University of Helsinki ‡Robert Bosch GmbH

Abstract

Edge-Fog clouds provide an attractive platform for bringing data processing closer to its source in a networked environment. In this paper we extend our work on Edge-Fog clouds and to an industrial automation scenario, where we show how grouping of computational data in resource caches lowers network traffic and shortens application-experienced latency. Our preliminary results are promising and in our future work we plan to evaluate more realistic scenarios and apply the solutions in real industrial automation cases.

1. Introduction

Large-scale data generation and analysis from high volume Internet of Things applications put stringent requirements on the network architecture. In [2], we proposed an Edge-Fog cloud model which aims to solve this problem by processing data as close to data generators as possible. The cloud model not only decreases the upstream load but also helps in decreasing the end-user experience latency.

Edge-Fog cloud follows a three- tier hierarchy which consists of outermost lower-powered edge devices, fog devices with more computational power, and a central data store for permanent archival of data. Edge and Fog resources have local memory for maintaining real-time data requirements to fulfill computational demands. At computation time, data from Data Store must be loaded into caches of resources involved in processing that data. In spite providing reliability, this approach incorporates unnecessary computational delay, marring the benefit of computing at the network edge.

Several works have proposed to deploy caches at the edge of the network to limit requests for data from remote repositories [3]. Most of these approaches are modeled as CDNs, the properties of which significantly differ from a "computation-first" network. Typical data in Edge-Fog cloud has shorter temporal relevance and receives more frequent updates when compared to CDNs which renders available solutions inapplicable.

In this work, we propose an efficient edge caching mechanism which leverages the cloud resource caches to predict and store data required for upcoming computations. The caching mechanism clusters Edge-Fog resources in collaborative cache pools by profiling upstream computation data demands. To the best of our knowledge, this work is the first to propose a resource caching strategy in edge computing domain. We motivate the benefits of our approach via real-world industrial application scenarios. We also provide a preliminary evaluation of our mechanism on a real-world network topology in an Edge-Fog cloud simulator.

2. Application Scenario

Collaborative robotics, such as Bosch APAS [1] require efficient coordination between robotic operations and decision-making capabilities of humans. However, this requires continuous sensing and monitoring of the environment to allow robots to work at an efficient speed while ensuring human safety. APAS uses many self-mounted proximity sensors in combination with 2D and 3D cameras to perform programmed tasks on an embedded server. However, to improve performance, decoupling onboard data collection and processing is required. Howver, remote processing must ensure rigid requirements for overall latency, reaction delays, task processing etc.

Edge-Fog cloud is a prime candidate for offloading robot's processing tasks. The task allocation algorithm proposed in [2] ensures that the APAS's task deployment on the cloud will achieve the least possible processing and network cost as required by the end-application. Moreover, Edge-Fog cloud can also provide a common collaboration network for robots developed by different companies across factories. However, the required data must be pre-cached at edge computing node to mitigate latency for requesting data from the remote Data Store.

CrossCloud'17, April 23, 2017, Belgrade, Serbia.

http://dx.doi.org/10.1145/3069383.3069391

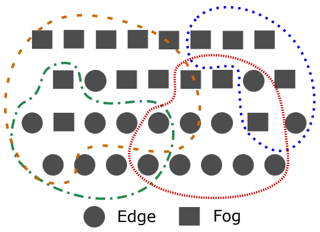

Figure 1: Resource cache grouping on workload classification

3. Solution

In [5], we proposed an edge caching mechanism which groups content requests based on pattern similarity at edge caches thereby significantly improving the overall hit rate. We utilize this approach in computational environment as computation in Edge-Fog cloud are easily groupable. Several properties such as origin (data generated by temperature sensors), locality (data from sensors in a localized environment) or computation type (video camera feed analysis) are common in each computation. Even in collaborative robotics, the data required for an operation can be categorized based on the workbench/task type, along with several other properties to analyze and group data at edge resource caches.

Figure 1 shows an example scenario of Edge and Fog resource caches which have been grouped based on workload data cached locally. As shown in the figure, a cache can store data conforming to several properties and thus become a member of several application groups. Membership to a group is independent of the computing power or whether the resource belongs at the Edge or Fog layer. The task deployment algorithm [2] will use group membership of a resource which will complement resource groups with application type while calculating ideal deployment.

Grouped caching mechanism offers several advantages. First, as there can be a significant number of resources at the Edge and Fog layer, such a caching strategy at the edge would lower the complexity of task deployment by limiting the size of the available resource pool. Second, group membership of resource caches can be advertised for sharing data from neighbor resources rather than requesting from the central data store. Third, based on current group membership, a resource can cache data before application deployment thus eliminating any delay at computation time. Finally, the caching strategy aims to remove any unnecessary network delay for transporting data from the remote central repository.

4. Preliminary Results

To evaluate the effectiveness of resource cache grouping in Edge-Fog cloud, we simulate our mechanism on Icarus [4] and Edge-Fog cloud simulator [2]. Figure 2 compares the network cost for grouped and non-grouped caching strategies on 160 node Edge-Fog cloud with increasing workload types. As seen from the figure, grouped caching strategy significantly lowers the network cost in edge task computing by 7-10%. Moreover, the caching strategy can maintain relevant data in simultaneous workload computation situations.

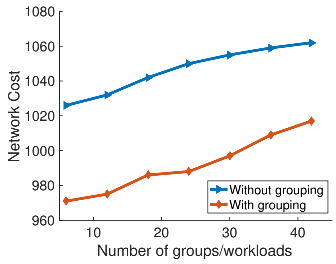

Figure 2: Network cost with grouped workload caching

5. Conclusion and Future Work

In this paper, we proposed a caching strategy for computing resource caches in an Edge-Fog cloud. The strategy profiles the data at the edge and groups resources based on their cache profiles. Further, we provide preliminary results depicting significant decrease in network cost for computing tasks on Edge-Fog cloud after employing this caching strategy.

As future work, we plan on doing an in-depth analysis of grouping classifier properties and its impact on overall number of groups. Further, effect of data sharing within a group on network cost needs to analyzed.

Acknowledgments

This research was partially funded by the joint EU FP7 Marie Curie Actions Cleansky Project 607584

References

[1] Bosch apas description. `http://www.bosch-apas.com/en/apas/produkte/assistant/apas_assistant_3.html`.

[2] Mohan, N., and Kangasharju, J. Edge-fog cloud: A distributed cloud for internet of things computations. In *Proc. CIoT* (November 2016), IEEE.

[3] Ramaswamy, L., Liu, L., and Iyengar, A. Scalable delivery of dynamic content using a cooperative edge cache grid. *IEEE Trans. on KDE 19*.

[4] Saino, L., Psaras, I., and Pavlou, G. Icarus: A caching simulator for information centric networking (icn). In *ICST SimuTools* (2014).

[5] Zhou, P., and Kangasharju, J. Profiling and grouping users to edge resources according to user interest similarity. In *ACM Workshop CAN* (December 2016).