Workgroup: ALTO WG
Internet-Draft:
draft-yang-alto-deliver-functions-over-
networks-03
Published: 21 March 2022
Intended Status: Standards Track
Expires: 22 September 2022
Authors: S. Yang, Shenzhen University
L. Cui, Shenzhen University
M. Xu, Tsinghua University
Y. Yang, Yale University
W. Xiao, Research Institute of Tsinghua University in Shenzhen

Delivering Functions over Networks: Traffic and Performance Optimization for Edge Computing using ALTO

Abstract

As the rapid development of internet, massive data are produced. Service providers typically need to deploy services near the edge networks to better satisfy user_s demand. In order to obtain better quality of the networks, computing functions and user traffic need to be scheduled properly. However, it is challenging to efficiently schedule resources among the distributed edge servers because of the lack of network information, such as network topology, traffic distribution, link delay/bandwidth and utilization/capability of computing servers. In this standard, we employed the ALTO protocol to help deliver functions and schedule traffic at the edge computing platform. This protocol supplied information of multiple resources for the distributed edge computing platform, thus enhancing the efficiency of function delivery in edge computing platform.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

For recent years internet has been developing rapidly since it is promising to be applied in industrial upgrading. In many scenarios of industrial internet, massive data are produced and require to be processed with high efficiency in real time. Typically, various functions or services are delivered in these scenarios according to the users_ demand. These functions or services could be (1) surveillance videos that need analysis by AI, (2) Hi-Definition videos that require to be encoded/decoded and (3) contents stored in the edge network. It is noted that functions and services are deployed widely in industrial internet. For instance, Function as a service (FaaS) is applied and delivered more and more frequently in the cloud computation service for industrial internet.

Many functions and services demand high quality of the supporting networks. For instance, the delay and jitter are expected to be as tiny as possible in order to obtain good user_s experience. Different from Kubernets and Mesos that are able to efficiently schedule computing resources in a single computing cluster, deployment of functions in wide area networks usually encounters much more complexity.

Many resources, including network traffic, bandwidth, topology, link delay and the computing capacity/utilization of each computing cluster, should be taken in to considerations when functions being deployed over distributed networks. This network status or information requires to be collected with unified interfaces and protocols because the resources typically need to be scheduled crossing different domains for satisfying user_s demands. In addition, resources scheduling algorithms SHOULD and network performances, such as load balancing, also needs to be optimized to improve user_s experience. In this standard, we propose an efficient method to utilize the computing and network resources by delivering functions over the edge computing networks.

We use the ALTO (Application-Layer Traffic Optimization) [RFC7285] to optimize network traffic and performance by delivering functions over the edge computing network. ALTO can provide global network information for the distributed applications, while the information can not be retrieved or computed by the applications themselves [RFC5693]. Specifically, the information of network for the distributed edge cluster, such as network traffic, link delay and other cost metrics, is collected and computed by ALTO. Subsequently, by using the pre-defined scheduling algorithms, functions will be delivered by system to the most suitable edge clusters based on the information offered by ALTO.

For brevity, in this document, we will use the terminologies introduced in [RFC7285] and [I-D.ietf-alto-unified-props-new].

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Background

3.1. Edge computing

There are many applications and scenario of network that are very sensitive to the quality of network. For instance, remote controlling for machine tool typically needs enhanced broadband and low-latency communication in real time. However, the scale of the data generated by the equipment or sensors are usually very large (up to be hundreds of GB per day). In addition, the data produced from various equipment will require support of heterogeneous computing. In this case, uploading these data to the centralized cloud computation platform is difficult, expensive but not so useful. Similarly, delivering functions or services from the cloud computation platform to the local equipment is also hard and the real-time performance will not be guaranteed. Alternatively, new framework of computation and network SHOULD be adopted to address these problems that hinder the edge computation to acquire broader promotion.

Edge computing was designed to enhance the quality of network, including the packet loss, security, bandwidth and latency. In order to decrease the distance between the servers and users, people typically will deploy servers at the edge nearing the users in the edge computing infrastructure. In this framework, User_s tasks can be submitted to the edge servers that will handle with the tasks close to the user and return the computation output back to the user promptly. As a consequence, the latency, bandwidth and network traffic performance of edge computing will be better than that of the centralized cloud computing. With these advantages, edge computing has been applied in various applications in network, such as AI supporting HD videos and VR/AR.

In this standard, functions and services will be delivered over the edge computing so that we can schedule the computing functions dynamically in a distributed edge computing network to enhance the performance of network. Nevertheless, it should be noted that during the deployment of the functions to edge servers, there are multiple resources, such as bandwidth, computing and link resources, that SHOULD be allocated to satisfy the requirements in terms of latency and throughput.

3.2. Features of ALTO protocol

Application-Layer Traffic Optimization (ALTO) [RFC7285] is designed to provide network information for distributed applications. Specifically, the resource scheduling process for distributed applications will be guided by the network states and information that is provided by the ALTO server. Otherwise, the information will fail to be retrieved by the applications. In this case, much essential network information in the resource selection process, such as network traffic, cost map, and cost metrics, will be offered by the ALTO protocol. As a consequence, network traffic can be managed by the distributed applications. In addition, they could make better choice in selecting the path that contains lower delay to access the network and handle with the computation tasks.

Because of being distributed over different areas of network, the edge computing clusters would comprise various network states, such as topology, network traffic, link delay and the computing capacity/ utilization. Generally, in order to obtain better performance, the scheduling decisions require to be adaptive to the network states during the delivering of functions. Thus, the network information and traffic can be managed by ALTO according to its mentioned advantage. As a consequence, functions and services will be delivered to a proper edge computing cluster.

3.3. Resources and services/functions

Generally, we have both limited resources and massive network services/functions on the network. Some common limited resources are as below.

*Computing resource: it usually represents the computing powers of CPU or GPU. CPUs have different architectures, including ARM and x86. The power of a CPU is affected by the working status, such as current load, total space and available space.

*Link/Path: They are a physical or logical communication channel between network devices, such as routers, servers and clients. Properties of a link or path include hop count, bandwidth and communication latency.

*Storage: that refers to space to store the data. The property of storage includes the amount of space to save the data.

*Radio resource: It means the radio information in wireless communication systems, such as cellular networks and wireless local area networks. In 5G network, radio resources can be combined with slicing technology to provide customized network property for user_s differentiated network demand.

Except the limited resources above, as the network technology rapidly develops, there are many network services and functions that could supply abundant computation service to users.

*Software as a service: It supplies software services as a platform. Software services, such as wechat and Alipay, are deployed on the SaaS vendors_ servers, for users to access to and use.

*AI as a service: it supplies artificial intelligence services as a platform. Various artificial intelligence-based services, such as face recognition, speech recognition and big data analysis, are provided by vendors.

*Encoding/Decoding as a service: it supplies encoding and decoding services for high-definition videos and VR/AR videos. It has been widely applied in areas of smart city and online entertainment.

*Function as a service: it supplies function services as a platform. Functions can be encapsulated docker images or pieces of code. Usually, in order to let user access to FaaS easily and conveniently, the vendor will expose the function APIs. One of the main features of FaaS is allowing network resources to be dynamically allocated to computing clusters. Therefore, users do not need to handle with the complicated environment configuration and resource management process, as they can utilize the function-based computation services, such as face recognition, speech recognition and big data analysis.

*Content: it supplies storage services as a platform. Utilizing the content service, users can store their data such that users could spare their limited local storage. Meanwhile, users can retrieve the data from different terminals.

4. Scenario of delivering function

Suppose a scenario in IoT, in which unmanned aerial vehicle (UAV) are connected via the network to apply for the face recognition computing services. When a UAV submits a task, the face recognition function will be delivered to an edge server to process the task. Then the recognition results will be returned to the UAV. During this process, network information, such as link delay and other cost metrics, would be requested and retrieved by the ALTO protocols from ALTO servers and clients in the network system. According to the information supplied by ALTO, the function and task will be delivered to the most suitable edge server that would provide best performance to the UAVs. The schematic diagram of this framework is shown in Figure 1 below.

```
+---------------+                  +-------------------+
|               |                  |                   |
|               |                  |                   |
|  ALTO Server  |<---------------->|    ALTO Client    |
|               |                  |                   |
|               |                  |                   |
+---------------+                  +------^-----+------+
                                          |     |
                                          |     |
                                          |     |
                                       +--+-----v--+
                                       |  Cluster  |
                               +-------+  Client   +------+
                               |       +-----------+      |
                               |                          |
                               |                          |
                               |                          |
                        +------v-------+          +-------v------+
                        |Edge Computing|          |Edge Computing|
                        |              |  ......  |              |
                        |  Cluster 1   |          |  Cluster N   |
                        +--------------+          +--------------+
```

Figure 1. Scenario of delivering function over edge network in IoT

5. Delivering functions by ALTO over edge computing

Since lots of edge clusters and servers are distributing in the network, the system MUST handle the huge amount of edge devices and their corresponding network traffic. A cluster client is employed to manage the connectivity and traffic information of the distributed edge clusters. The ALTO client will communicate with the cluster client and provide the necessary network information. The usage of ALTO is to optimize the network traffic and guide the function delivering process in edge computing. It will provide the overall network states with information for the distributed edge clusters, and decide the appropriate edge cluster to deploy the functions.

More specifically, the ALTO server will collect and compute the network cost metrics; including the link delay, availability, network traffic, bandwidth, and etc. The information will then be sent to the ALTO client. The ALTO client will select the target appropriate edge clusters to deploy the target function. Finally, the system will connect and deploy the function to the target servers, so that users can submit their computation task to the selected edge clusters.

```
+---------------+                  +-------------------+
|               |   (1) Network    |                   |
|               |   Information    |                   |
|  ALTO Server  |<---------------->|    ALTO Client    |
|               |                  |                   |
|               |                  |                   |
+---------------+                  +------^-----+------+
                                          |     |
                          (2)Get clusters |     | (3)Select Cluster List
                                          |     |
                                       +--+-----v--+
                                       |  Cluster  |
                               +-------+  Client   +------+
                               |       +-----------+      |
                               |                          |
                               |  (4) Connect to Cluster  |
                               |    and deliver function  |
                        +------v-------+         +-------v------+
                        |Edge Computing|         |Edge Computing|
                        |              |  ......  |              |
                        |  Cluster 1   |         |  Cluster N   |
                        +--------------+         +--------------+
```

Figure 2. Delivering process in edge computing platform with ALTO

Figure 2 illustrates the infrastructure and function delivering process of the edge computing platform.

1. The ALTO client requests the information, such as network map and cost map of distributed edge clusters from the ALTO server, by using ALTO protocol.

2. The Cluster Client requests an edge cluster list of the network.

3. The ALTO Client returns the edge cluster list and corresponding resource information about the clusters computed by ALTO servers according to the network state.

4. The Cluster Client connects and delivers function to the corresponding edge computing cluster according to the information, and the cluster will process and return the computation results to users.

Note that the data transfer process is using the ALTO protocol described in [RFC7285] to guarantee the efficiency and security of the delivering process. In this case, the edge computing clusters are allowed to retrieve the network information, so that the function can be delivered to the proper ones to achieve a better performance in terms of latency, throughput, etc.

6. Implementation and Deployment

We have implemented a prototype, where the edge cluster resources are managed by K8S and Docker. When users request for edge computing services, the appropriate edge cluster will be selected by ALTO according to network map and cost information.

We have deployed the prototype in real network in the China Mobile network. The preliminary results of our deployment show that 1) the performance of edge computing will be greatly improved by using the supplied network information and 2) collection and scheduling policies of this information need to be standardized to obtain coordination among different domains.

7. Management of Functions

As function standardization in our system could be useful in managing functions efficiently, We will introduce this technique as below. Our system can standardize the functions and expose standard APIs for users to easily access to and apply for function-based computation services. Above the function-based computation services, certain function codes and docker images can be updated and replaced based on the user_s demand or standard upgrade. This would be useful to the function management of the platform. Specifically, function standardization is composed of several steps below.

Specifically, function standardization consists of:

*Function repository: Generate the repository that stores all the functions for users to apply for.

*Function registry/discovery: A service MUST be registered before being applied for. After the registry, the information of service will be broadcast to a registry server. In this circumstance, when delivering the functions, the system will recognize which node is registered with the function information by accessing to the registry server. As a consequence, appropriate node can be determined by our system in order to deliver functions with high efficiency.

*Function status update: When there are updates, functions in all the network nodes MUST be updated accordingly.

As function standardization is powerful in delivering function, users can conveniently process their tasks by sending requests to the interfaces of the system in which the standard APIs are exposed by the process of function standardization. As described above, this mechanism could help users to bypass the complexity of resource deployment and configuration. In addition, the function standardization is also useful in managing the system, because system operators can update or replace the target functions more efficiently. When users applying for functions, they can easily locate the target edge servers since each function on the platform is saved and registered in certain c edge servers before.

8. Multi-domain System

A function delivery platform can be a multi-domain system. For example, there may be multiple service providers offering the function-based computation service. In this case, we should consider how to collect and manage the network information from different domains, in order to achieve better function delivery performance in networks. Consequently, we SHOULD develop additional designs for our platform.

On the one hand, we introduce the layered design for function delivery. More specifically, we deploy multiple distributed registry servers in the lower layer, each of which processes the function registry in its domain. Then we deploy a centralized registry server in the upper layer to collect and manage the distributed registry servers in the lower layer. A server in the lower layer will report and send network information of its domain to the centralized server in the upper layer periodically. And the centralized server will coordinate the domains by sending instructions to the distributed servers in the lower layer, which will make adjustment according to the instructions of the centralized registry server. In this case, the centralized registry server is able to manage the distributed function and network information easily and efficiently, which is beneficial to multi-domain system management.

On the other hand, we introduce the policy management for multiple domains. Note that different domains MAY have various delivery policies, thus we need to provide a policy management tool for multiple domains. When delivering functions in a multi-domain system, the tool will provide the overall management policy to synchronize and coordinate the distributed local policies in each individual domain. In this case, the distributed multiple domains in different policies are able to communicate and coordinate with each other, with the help of the policy management tool. Therefore, by utilizing the policy management tool, we can manage the multiple domains for efficient function delivery.

9. Scheduling Framework

Recently, with the development of high-capacity computing devices, the computing power of networks has improved much. However, due to the lack of efficient scheduling strategies, the current computing platforms cannot achieve better computing throughput, i.e., the ability to schedule the distributed computing power over a long period. To improve the scheduling efficiency of the computing power, researchers proposed some high-throughput computing scheduling frameworks, for example, HTCondor, PBS, CPUsage, etc., which are able to schedule the limited distributed computing power to achieve better throughput of the network in a long period. Inspired by the high-throughput computing scheduling frameworks, we develop the scheduling framework for function delivery, in order to achieve better performance of networks.

The objective of our scheduling framework for function delivery is to minimize the computational latency. The basic idea is, our platform will compute the function scheduling schemes, according to the information collected by the ALTO server, including the network congestion, resource utilization, etc. The users will access the most appropriate edge server, which will provide the function-based computation service and return the results to the users.

More specifically, when a user applies for the function delivery service, it will send requests to the interface provided by the ALTO server, along with its location and task information. The ALTO server will also collect the resource utilization and network information of the decentralized edge servers. Then, according to the collected information, the ALTO server will compute the function scheduling scheme, to determine the function delivery destination of a specific edge server. The platform will select the edge server with lowest computation latency for user. However, if the selected edge server is overloaded, the platform will proceed to search other edge server that satisfies the load balance demand, along with achieving considerable latency performance. Finally, the user will establish the communication channel with the target edge server, which will provide the function-based service and return the results to the users.

By developing the scheduling framework and strategy for function delivery, our platform can maintain the stable network condition and guarantee the load balance over a long period, which is beneficial to the reliability of system. And users can enjoy a low-latency and high-throughput function delivery service at the same time.

10. Security Considerations

T.B.D.

11. IANA Considerations

This document includes no requests to IANA.

12. References

12.1. Normative References

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <https://www.rfc-editor.org/info/rfc5693>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <https://www.rfc-editor.org/info/rfc7285>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

12.2. Informative References

[I-D.ietf-alto-unified-props-new] Roome, W., Randriamasy, S., Yang, Y., Zhang, J., and K. Gao, "Unified Properties for the ALTO Protocol", Work in Progress, Internet-Draft, draft-ietf-alto-unified-props-new-09, 4 September 2019, <http://www.ietf.org/internet-drafts/draft-ietf-alto-unified-props-new-09.txt>.

Authors' Addresses

Shu Yang
Shenzhen University
South Campus, Shenzhen University
Shenzhen
518060
P.R. China

Phone: +86-755-2653-4078
Email: yang.shu@szu.edu.cn

Laizhong Cui
Shenzhen University
South Campus, Shenzhen University
Shenzhen
518060
P.R. China

Phone: +86-755-8695-6280
Email: cuilz@szu.edu.cn

Mingwei Xu

Tsinghua University
Department of Computer Science, Tsinghua University
Beijing
100084
P.R. China

Phone: +86-10-6278-5822
Email: xumw@tsinghua.edu.cn

Richard Yang
Yale University
51 Prospect St
New Haven, CT, 06511
United States of America

Email: yry@cs.yale.edu

Wei Xiao
Research Institute of Tsinghua University in Shenzhen
Nanshan Hi-new Technology and Industry Park
Shenzhen
518060
P.R. China

Email: xiaow@tsinghua-sz.org