Workgroup: CATS
Internet-Draft:
draft-du-cats-computing-modeling-
description-02
Published: 23 October 2023
Intended Status: Informational
Expires: 25 April 2024
Authors: Z. Du
China Mobile
Y. Fu
China Mobile
C. Li
Huawei Technologies
G. Huang
ZTE
Z. Fu
New H3C Technologies

Computing Information Description in Computing-Aware Traffic Steering

Abstract

This document describes the considerations and the potential architecture of the computing information that needs to be notified into the network in Computing-Aware Traffic Steering (CATS).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 25 April 2024.

Table of Contents

1. Introduction

Computing-Aware Traffic Steering (CATS) is proposed to support steering the traffic among different service sites according to both the real-time network and computing resource status as mentioned in [I-D.yao-cats-ps-usecases] and [I-D.yao-cats-gap-reqs]. It requires the network to be aware of computing resource information and select a service instance based on the joint metric of computing and networking.

In order to generate steering strategies, the modeling of computing capacity is required. Different from the network, computing capacity is more complex to be measured. For instance, it is hard to predict how long will be used to process a specific computing task based on the different computing resource. It is hard to calculate and will be influenced by the whole internal environments of computing nodes. But there are some indicators has been used to describe the computing capacity of hardware and computing service, as mentioned in Appendix A.

Based on the related works and the demand of CATS traffic steering, this document analyzes the types of computing resources and tasks, providing the factors to be considered when modeling and evaluating the computing resource capacity. The detailed modeling job of the computing resource is not the object of this document.

2. Definition of Terms

This document makes use of the following terms:

Computing-Aware Traffic Steering (CATS): A traffic engineering approach [I-D.ietf-teas-rfc3272bis] that takes into account the dynamic nature of computing resources and network state to optimize service-specific traffic forwarding towards a given service contact instance. Various relevant metrics may be used to enforce such computing-aware traffic steering policies.

Service: An offering that is made available by a provider by orchestrating a set of resources (networking, compute, storage, etc.).

Service instance: An instance of running resources according to a given service logic.

Service identifier: Used to uniquely identify a service, at the same time identifying the whole set of service instances that each represents the same service behavior, no matter where those service instances are running.

Computing Capacity: The ability of nodes with computing resource achieve specific result output through data processing, including but not limited to computing, communication, memory and storage capacity.

3. Problem Statement in Computing Resource Modeling

3.1. Heterogeneous Chips and Different Computing Types

Different heterogeneous computing resources have different characteristics. For example, CPUs usually deal with pervasive computing and are most widely used. GPUs usually handle parallel computing, such as rendering of display tasks, and are widely used in artificial intelligence and neural network algorithm computing. FPGA and ASIC are usually used to handle customized computing. At the same time, different computing tasks need to call different calculation types, such as integer calculation, floating-point calculation, hash calculation, etc.

3.2. Multi-dimensional Modeling

The network and computing have multi-dimensional and hierarchical resources, such as cache, storage, communication, etc., and these dimensions will affect each other and further affect the overall level of computing capacity. Other factors besides the computing itself need to be considered in modeling. At the same time, the form of computing resources is also hierarchical, such as computing type, chip type, hardware type, and converging with the network. For different computing forms, such as gateway, all-in-one machine, edge cloud and central cloud, the computing capacity, and types provided are also different. It is necessary to comprehensively consider multi-dimensional and multi-modal resources, and provide multi-level modeling according to application demands.

3.3. Support to be used for Further Representation

Modeling itself provides a general method to evaluate the capacities of computing resource. For CATS, modeling-based computing resource representation is the basis for subsequent traffic steering. In addition, for different applications, it may be optimized based on general modeling methods to establish a set of models that conform to their own characteristics, so as to generate corresponding representation methods. Moreover, in order to use computing resource status more efficiently and protect privacy, modeling for the further representation of resource information needs to support the necessary simplification and obfuscation.

4. Usage of Computing Resource Modeling of CATS

We need to use the computing resource modeling in two procedures. The first is the service deployment, and the second is the traffic steering, in which the later is more related to the CATS work. However, the service deployment is the precondition of CATS, which enables the assumption that the service can be accessed in multiple places.

In the procedure of service deployment, a control or management device either in the CATS domain or in the Computing domain can collect the computing information and make the service deployment decisions. As the procedure is not that real time, it can collect more information about the service points. Many existing jobs can be reused here such as the ones used in the data centers.

In the procedure of traffic steering, we can use limited metrics to trigger the change of the policy for the service on path, so that a quick response can be ensured for the change of the computing status.

For the modeling mechanism based on CATS-defined format, the decision point can collect more information to support both the service deployment and the traffic steering. On the contrary, the mechanism based on application-defined method will be more suitable for the CATS, in which only necessary metrics need to be notified into the network or called the CATS domain. The detailed metric design can be found in Section 5.

4.1. Modeling Based on CATS-defined Format

Figure 1 shows the case of modeling based on CATS-defined Format. CATS provides the modeling format to the computing domain to evaluate the computing resource capacity of computing domain and then get the result based on the unified interface, which will define the properties should be notified to CATS. Then CATS could select the specific service instance based on the computing resource and network resource status.

In this way, the CATS domain and computing domain has the relative loose boundary based on the situation that the CATS service and computing resource belongs to the same provider, CATS could be aware of computing resource more or less, depending on the privacy preserving demand of the computing domain at the same time. The exposed computing capacity includes the static information of computing node category/level and the dynamic capabilities information of computing node.

Based on the static information, some visualization functions can be implemented on the management plane to know the global view of computing resources, which could also help the deployment of applications considering the overall distributed status of computing and network resource. Based on the dynamic information, CATS could steer category-based applications traffic based on the unified modeling format and interface.

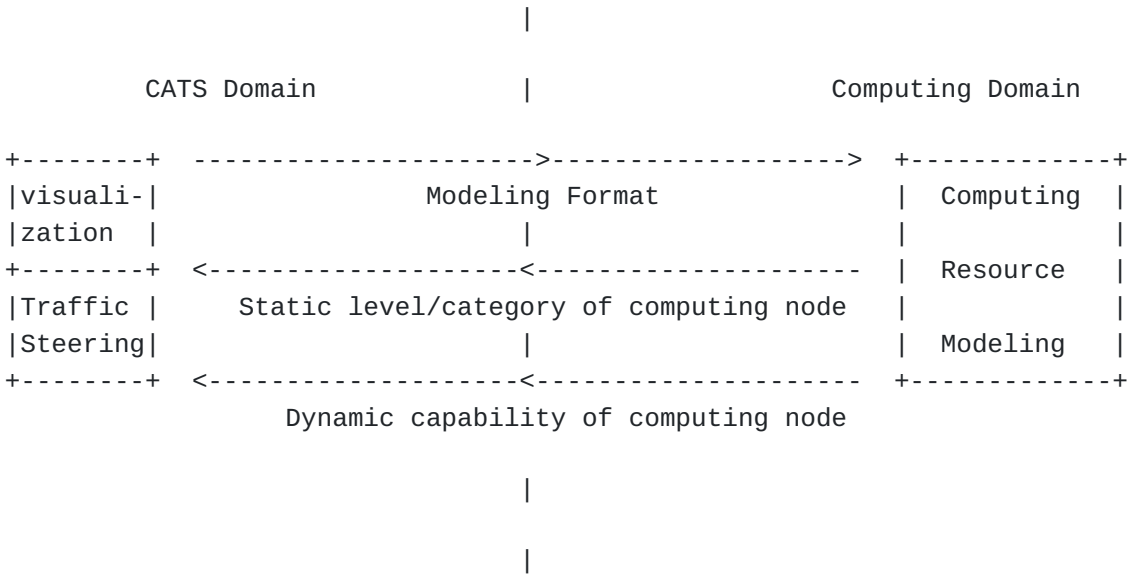

Figure 1: Modeling Based on CATS-defined Format

4.2. Modeling Based on Application-defined Method

Figure 2 shows the case of modeling based on application-defined method. Computing resource of the specific application evaluates its computing capacity by itself, and then notifies the result which might be the index of real time computing level to CATS. Then CATS selects the specific service instance based on the computing index.

In this way, the CATS domain and computing domain has the strict boundary based on the situation that the CATS service and computing resource belongs to the different providers. CATS is just aware of the index of computing resource which is defined by application, don't know the real status of computing domain, and the traffic steering right is potentially controlled under application itself. If CATS is authorized by application, it could steer traffic based on network status at the same time.

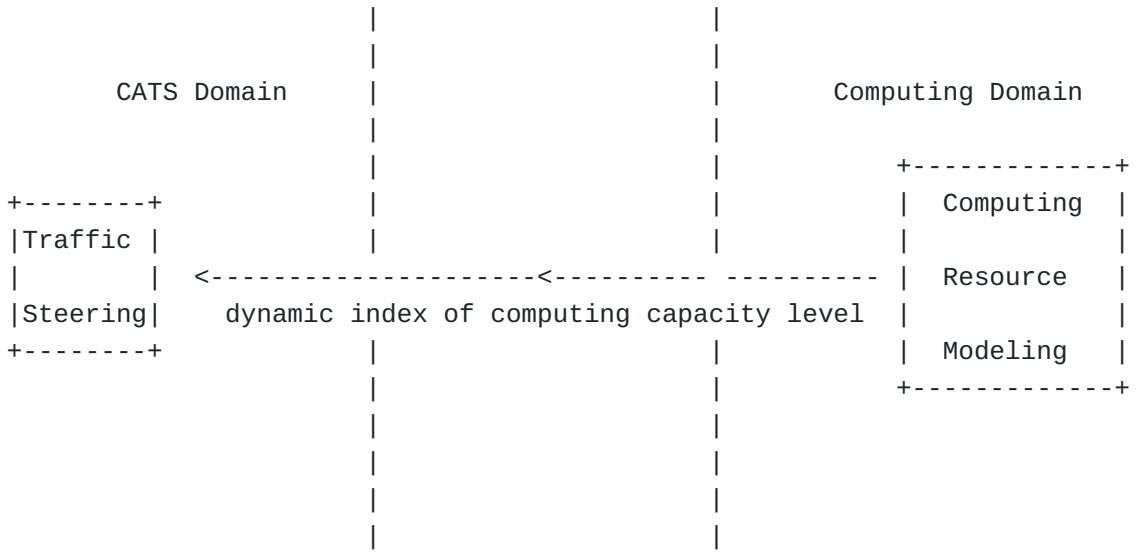

Figure 2: Modeling Based on Application-defined Method

5. Computing Resource Modeling

To support a computing service, we need to evaluate the comprehensive service performance in a service instance, which is influenced by the coordination of chip, storage, network, platform software, etc. It is to say that the service support capabilities are influenced by multidimensional factors. Therefore, in the modeling of the computing metric, we can provide not only the specification computing values provided by the manufacturer, such as FLOPS, but also some integrated index values that can comprehensively reflect the service support capabilities.

5.1. Requirements of Using in CATS

It is assumed that the same service can be provided in multiple places in the CATS. In the different service instances, it is common that they have different kinds of computing resources, and different utilization rate of the computing resources.

In the CATS, the decision point, which should be a node in the network, should be aware of the network status and the computing status, and accordingly choose a proper service point for the client.

A general process to steer the CATS traffic is described as below. The CATS packets have an destination address as the service ID that is announced by the different service points.

Firstly, the service points need to collect some specific computing information that need to be sent into the network following a uniform format so that the decision point can understand the computing information. In this step, only necessary computing

information needs to be considered, so as to avoid exposing too much information of the service points.

Secondly, the service instances send the computing information into the network by some means, and update it periodic or on demand.

Thirdly, the decision point receives the computing information, and makes a decision for the specific service related to the service ID. Hence, the route for the service ID on the Ingress is established or updated.

Fourthly, the traffic for the service ID reaching the Ingress node would be identified and steered according to the policy in the step3.

In fact, what to send, how to send, and the optimization objective of the policy are all related to the design of the computing resource modeling in CATS, meanwhile they would influence each other. Some requirements are listed below.

1. The optimization objective of the policy in the decision point may be various. For example, it may be the lowest latency of the sum of the network delay and the computing delay, or it may be an overall better load balance result, in which we would prefer the service points that could support more clients.

2. The update frequency of the computing metrics may be various. Some of the metrics may be more dynamic, and some are relatively static.

3. The notification ways of the computing metrics may be various. According to its update frequency, we may choose different ways to update the metric.

4. Metric merging process should be supported when multiple service instances are behind the same Egress.

The target in CATS mainly concerns about the service point selection and traffic steering in Layer3, in which we do not need all computing information of the service points. Hence, we can start with simple cases in the work of the computing resource modeling in CATS. Some design principles can be considered.

1. The computing metrics in CATS should be few and simple, so as to avoid exposing too much information of the service points.

2. The computing metrics in CATS should be evolveable for the future extensions.

3. The computing metrics in CATS should be vendor-independent, and OS-independent.

5.2. Consideration of Using in CATS

Various metrics can be considered in CATS, and perhaps different services would need different metrics. However, we can start with simple cases.

In CATS, a straightforward intent is to minimal the total delay in the network domain and the computing domain. Thus, we can have a start point for the metric designation in CATS considering only the delay information. In this case, the decision point can collect the network delay and the computing delay, and make a decision about the optimal service point accordingly. The advantage of this method is that it is simple and easy to start; meanwhile, the network metric and the computing metric have the same unit of measure. The network delay can be the latency between the Ingress node and Egress node in the network. The computing delay can be generated by the server, which has the meaning of “the estimate of the duration of my processing of request”. It is usually an average value for the service request. The optimization objective of traffic steering in this scenario is the minimal total delay for the client.

Another metric that can be considered is the server capability. For example, one server can support 100 simultaneous sessions and another can support 10,000 simultaneous sessions. The value can be generated by the server when deploying the service instance. The metric can work alone. In this scenario, the decision point can do a Load Balance job according to the server capability. For example, the decision process can be load balancing after pruning the service points with poor network latency metrics. Also, the metric can work with the computing delay metric. For example, in this scenario, we can prune the service points with poor total latency metrics before the load balancing.

In future, we can also consider other metrics, which may be more dynamic. Besides, for some other optimization objectives, we can consider other metrics, even metrics about energy consumption. However, in this cases, the decision point needs to consider more dimensions of metrics. A suggestion is that we should firstly make sure the service point is available, which means the service point can still accept more sessions, and then select a optimal target service point according to the optimization objective.

6. Network Resource Modeling

The modeling of the network resource is optional, which depends on how to select the service instance and network path. For some

applications which care both network and computing resource, the CATS service provider also need to consider the modeling of network and computing together.

The network structure can be represented as graphs, where the nodes represent the network devices and the edges represent the network path. It should evaluate the single node, the network links and the E2E performance.

6.1. Consideration of Using in CATS

When to consider both the computing and network status at the same time, the comprehensive modeling of computing and network might be used. For example, to measure all the resource in a unified dimension, such as latency, reliability, etc.

If there is no strict demand of consider them at same time, for instance, consider computing status first and then network status. CATS could select the service instance at first, then to mark identifier for network path selection of network itself. In this situation, the network modeling is not that needed. Existing mechanisms on the control plane or the management plane in the network can be used to obtain the network metrics.

7. Application Demands Modeling

The application always has its own demands for network and computing resource, for instance we can see the HD video always requires the high bandwidth and the PC game always requires the better GPU and memory. The application is identified by using the Service Identifier in the network, which can indicate its demands in a certain degree.

7.1. Consideration of Using in CATS

The modeling of the application demand is optional, which depends on whether the application could tell the demands to the network, or what it could tell. Once the CATS knows the application's demand, there should be a mapping between application demand and the modeling of the computing and/or network resource.

8. Security Considerations

TBD.

9. IANA Considerations

TBD.

10. Acknowledgements

The author would like to thank Adrian Farrel, Joel Halpern, Tony Li, Thomas Fossati, Dirk Trossen, Linda Dunbar for their valuable suggestions to this document.

11. Contributors

The following people have substantially contributed to this document:

Jing Wang
China Mobile
wangjingjc@chinamobile.com

Peng Liu
China Mobile
liupengyjy@chinamobile.com

Wenjing Li
Beijing University of Posts and Telecommunications
wjli@bupt.edu.cn

Lanlan Rui
Beijing University of Posts and Telecommunications
llrui@bupt.edu.cn

12. Informative References

[I-D.yao-cats-ps-usecases]
Yao, K., Trossen, D., Boucadair, M., Contreras, L. M., Shi, H., Li, Y., and S. Zhang, "Computing-Aware Traffic Steering (CATS) Problem Statement, Use Cases, and Requirements", Work in Progress, Internet-Draft, draft-yao-cats-ps-usecases-03, 30 June 2023, <https://datatracker.ietf.org/doc/html/draft-yao-cats-ps-usecases-03>.

[I-D.yao-cats-gap-reqs] Yao, K., Jiang, T., Eardley, P., Trossen, D., Li, C., and D. Huang, "Computing-Aware Traffic Steering (CATS) Gap Analysis and Requirements", Work in Progress, Internet-Draft, draft-yao-cats-gap-reqs-00, 3 March 2023, <https://datatracker.ietf.org/doc/html/draft-yao-cats-gap-reqs-00>.

[I-D.ietf-teas-rfc3272bis]
Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-27, 12 August 2023, <https://

datatracker.ietf.org/doc/html/draft-ietf-teas-rfc3272bis-27>.

[One-api] One-api, "http://www.oneapi.net.cn/", 2020.

[Amazon] Amaozn, "https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-target-tracking.html#available-metrics", 2022.

[Aliyun] Aliyun, "https://help.aliyun.com/?spm=a2c4g.11186623.6.538.34063af89EIb5v", 2022.

[Tencent-cloud] Tencent-cloud, "https://buy.cloud.tencent.com/pricing", 2022.

[cloud-network-edge] cloud-network-edge, "A new edge computing scheme based on cloud, network and edge fusion", 2020.

[heterogeneous-multicore-architectures] access, I., "Towards energy-efficient heterogeneous multicore architectures for edge computing", 2019.

[ARM-based] Guide, S., "A heterogeneous CPU-GPU cluster scheduling model based on ARM", 2017.

Appendix A. Related Works on Computing Capacity Modeling

Some related work has been proposed to measurement and evaluate the computing capacity, which could be the basis of computing capacity modeling.

[cloud-network-edge] proposed to allocate and adjust corresponding resources to users according to the demands of computing, storage and network resources.

[heterogeneous-multicore-architectures] proposed to design heterogeneous multi-core architectures according to different customization, such as CPU microprocessors with ultra-low power consumption and high code density, low power microprocessor with FPU, and a high-performance application processor with FPU and MMU support based on a completely unordered multi problem architecture.

[ARM-based] proposed the cluster scheduling model that is combined with GPU virtualization and designed a hierarchical cluster resource management framework, which can make the heterogeneous CPU-GPU cluster be effectively used.

The hardware cloud service providers have also disclosed their parameter indicators for computing services:

[One-api] provides a collection of programming languages and cross architecture libraries across different architectures, to be compatible with heterogeneous computing resources, including CPU, GPU, FPGA, and others. [Amazon] uses the computing resource parameters when evaluating the performance, including the average CPU utilization, average number of bytes received and sent out, and average application load balancer. Alibaba cloud [Aliyun] gives the indicators including vcpu, memory, local storage, network basic and burst bandwidth capacity, network receiving and contracting capacity, etc., when providing cloud servers service. [Tencent-cloud] uses vcpu, memory (GB), network receiving and sending (PPS), number of queues, intranet bandwidth capacity (Gbps), dominant frequency, etc.

Authors' Addresses

Zongpeng Du
China Mobile
No.32 XuanWuMen West Street
Beijing
100053
China

Email: duzongpeng@foxmail.com

Yuexia Fu
China Mobile
No.32 XuanWuMen West Street
Beijing
100053
China

Email: fuyuexia@chinamobile.com

Cheng Li
Huawei Technologies

Email: c.l@huawei.com

Guangping Huang
ZTE

Email: huang.guangping@zte.com.cn

Zhihua Fu
New H3C Technologies

Email: fuzhihua@h3c.com