Workgroup: CATS
Internet-Draft:
draft-du-cats-computing-modeling-
description-00
Published: 5 March 2023
Intended Status: Informational
Expires: 6 September 2023
Authors: Z. Du
China Mobile
Y. Fu
China Mobile
C. Li
Huawei Technologies
G. Huang
ZTE

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 in 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 6 September 2023.

Table of Contents

1. Introduction

Computing-Aware Traffic Steering (CATS) is proposed to support steering the traffic among different edge 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): Aiming at computing and network resource optimization by steering traffic to appropriate computing resources considering not only routing metric but also computing resource metric.

Service: A monolithic functionality that is provided by an endpoint according to the specification for said service. A composite service can be built by orchestrating monolithic services.

Service instance: Running environment (e.g., a node) that makes the functionality of a service available. One service can have several instances running at different network locations.

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.

Service transaction: Has one or more service request that has several flows which require the affinity because of the transaction related state.

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

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.

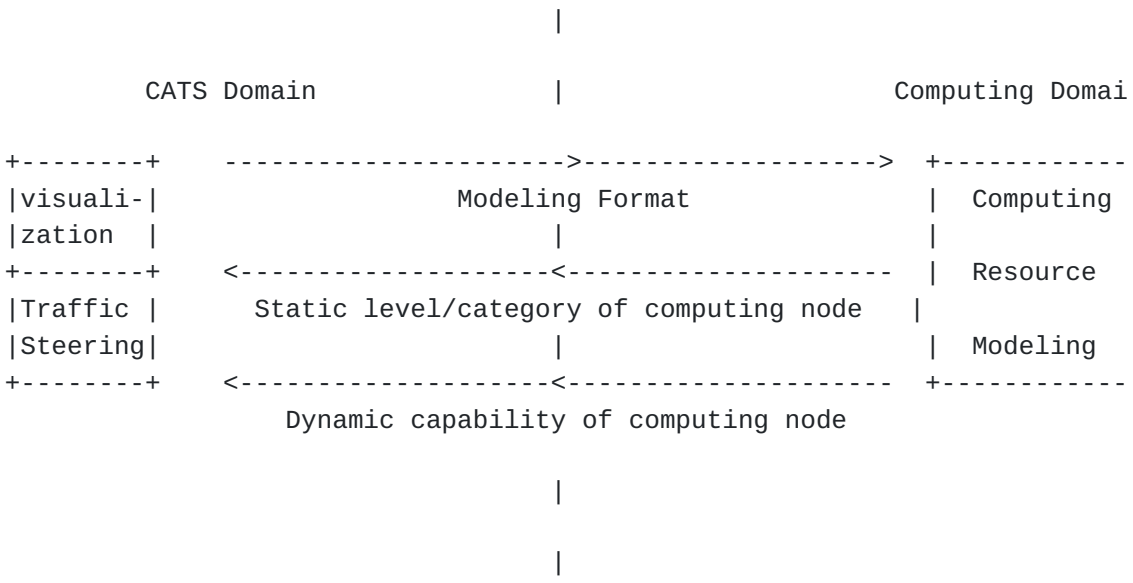

Figure 1: Modeling Based on CATS-defined Format

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.

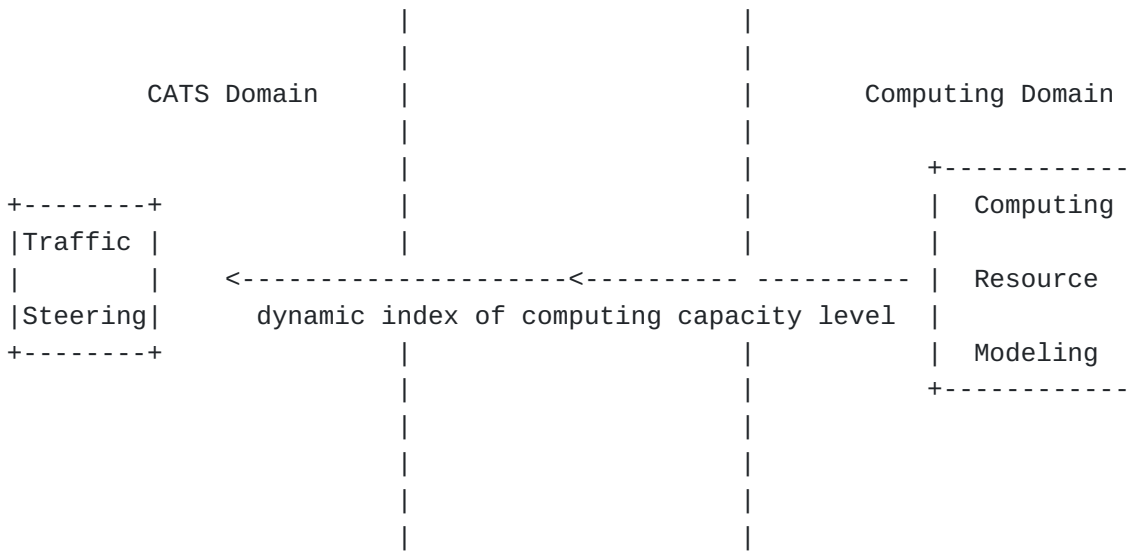

Figure 2: Modeling Based on Application-defined Method

5. Computing Resource Modeling

To support a computing service, we need to evaluate the comprehensive service performance in a service point, 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.

We can build up a computing resource modeling system which contains three levels of indicators, and the architecture can refer to Appendix B.

The first level is about the heterogeneous hardware computing capability. The indexes of this level can be the performance parameters provided by the manufacturer, such as CPU model, main frequency, number of cores, GPU model, single-precision floating-point performance, etc. Meanwhile, the indexes can also be the test values of commonly used benchmark programs.

The second-level indexes are abstracted from the first-level indexes, which are mainly used for the comprehensive evaluation of node's computing capability. They can provide the ability of a certain aspect of the node, such as in the aspect of computing, communication, cache, and storage, or a general comprehensive service ability of the node.

Level 3 indexes are related to the services deployed on the nodes. They mainly provide service-related evaluation parameters, such as the actual processing throughput that nodes can provide for a specific computing service. It can also be a test value, but it is generated by running the real service.

5.1. Consideration of Using in CATS

It is assumed that the same service can be provided in multiple places in the CATS. In the different service points, 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. In fact, the decision would influenced more by the dynamic indexes. An example of the decision process is described below.

Firstly, the decision point needs to make sure the candidate service points can still access a new session. If a service point claims that it is busy, no packet of new clients should be steered to it.

Secondly, the decision point can select a service point with a higher comprehensive performance evaluation value for the service. If Level 3 indexes for the service are available in the decision point, the decision point can use it directly because it is most straightforward. If not, for example the service is not tested in the service point yet, Level 2 indicators can be used instead with configurable weight values for the four aspects as mentioned in Appendix B.

In this example, the index in the first step is dynamic, and is related to the service status. The index in the second step is relatively static, and is related to computing efficiency for the service.

For a specific service, more indicators can still be provided. It is to say that the computing information could be customized for different services. Additionally, besides the computing information, the energy consumption index can also be included if it is considered necessary when making a decision.

Therefore, in this example, CATS needs two indexes at least, one for the service status, and another for service ability. Optionally, other information can also be provided if it is subscribed by the decision point. The detailed decision process in the decision point is out of scope of this document.

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 is usually composed of several sub service that complete different functions, and the service is usually composed of several sub transactions, which would be the smallest schedulable unit.

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.

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 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., Eardley, P., Trossen, D., Boucadair, M., Contreras, L. M., Li, C., Li, Y., and P. Liu, "Computing-Aware Traffic Steering (CATS) Problem Statement and Use Cases", Work in Progress, Internet-Draft, draft-yao-cats-ps-usecases-00, 3 March 2023, <https://datatracker.ietf.org/doc/html/draft-yao-cats-ps-usecases-00>.

[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>.

[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.

Appendix B. Architecture of Computing Modeling

This Appendix describes the potential architecture of computing resource modeling, regardless of any ways of the further usage of traffic steering of CATS, neither of the usage ways described in Section 4.

According to the computing indicators and related work described in Section 2, computing capacity includes the types of computing resources and tasks, and also need to consider multi-dimensional capabilities such as communication, memory, and storage. Because every factor will affect each others. For instance, with the rapid growth of modern computer CPU performance, the communication bottleneck between CPU and cache has become increasingly prominent. Moreover, the storage capacity greatly affects the processing speed of a computer. So the architecture of computing capacity modeling could be seen in Figure 3.

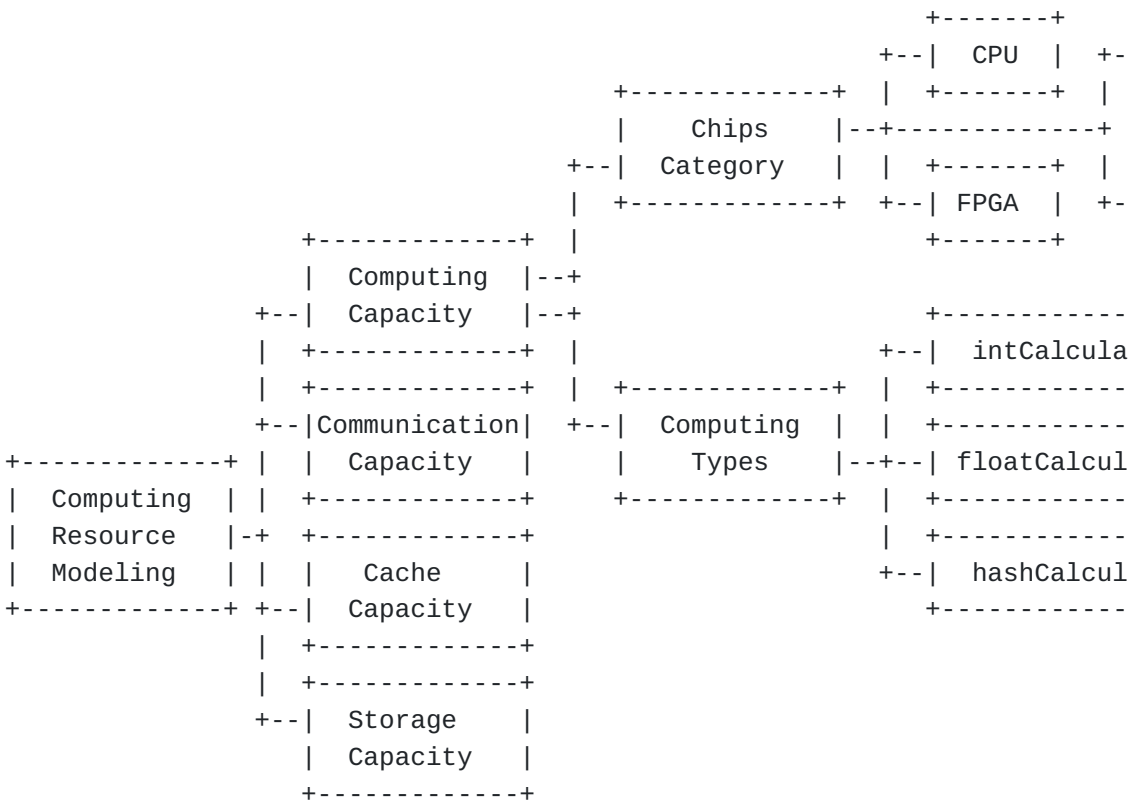

Figure 3: Referecen Architecture of Computing Modeling Format

B.1. Computing Capacity

The computing capacity includes the chips category and computing types. Common chip types include CPU, GPU, FPGA and ASIC. CPU and GPU belong to von Neumann structure, with instruction decoding and execution and shared memory. According to the different characteristics and requirements of computing programs, the computing performance can be divided into integer computing performance, floating-point computing performance and hash computing performance.

B.1.1. Types of Chips

CPU (Central Processing Unit) is a general-purpose processor needs to be able to handle comprehensive and complex tasks, as well as the synchronization and coordination between tasks. Therefore, a lot of space is required on the chip to perform branch prediction and optimization and save various states to reduce the delay during task switching. This also makes it more suitable for logic control, serial operation and universal type data operation.

GPU (Graphics Processing Unit) has a large-scale parallel computing framework composed of thousands of smaller and more efficient Alu

cores. Most transistors are mainly used to build control circuits and caches, and the control circuits are relatively simple.

FPGA (Field Programmable Gate Array) is essentially an architecture without instructions and shared memory, which is more efficient than GPU and CPU. The main advantage of FPGA in data processing tasks is its stability and extremely low latency, which is suitable for streaming computing intensive tasks and communication intensive tasks.

ASIC (Application Specific Integrated Circuit) is a special integrated circuit, and its performance is actually better than FPGA. However, for customized customers, its cost is much higher than FPGA.

On this basis, according to different computing task requirements, chip manufacturers have also developed various "xpus", including APU (Accelerated Processing Unit), DPU (Deep-learning Processing Unit), TPU (Tensor Processing Unit), NPU (Neural-network Processing Unit) and BPU (Brain Processing Unit), which are made based on the CPU, GPU, FPGA and ASIC.

B.1.2. Type of Computing

At present, the computing type in computer mainly includes integer calculation, floating-point calculation, and hash calculation.

The integer calculation rate is expressed as the calculation rate of the integer data operation benchmark program running on the CPU. Integer computing capability has its specific application scenarios, such as discrete-time processing, data compression, search, sorting algorithm, encryption algorithm, decryption algorithm, etc.

Floating point calculation rate is expressed as the calculation rate of the floating-point data operation benchmark program running on the CPU. There are many kinds of benchmark programs, each of which can reflect the floating-point computing performance of nodes from different aspects.

The hash calculation rate refers to the output speed of the hash function when the computer performs intensive mathematical and encryption related operations. For example, in the process of obtaining bitcoin through "mining", how many hash collisions can a mining machine do per second, and the unit is hash/s.

B.1.3. Relation of Computing Types and Chips

The differences computing capacity of the above different chip types is summarized as figure 4 shows. CPU is good at intCalculation, GPU

and FPGA are good at floatCalculation, and ASIC is good at intCalculation.

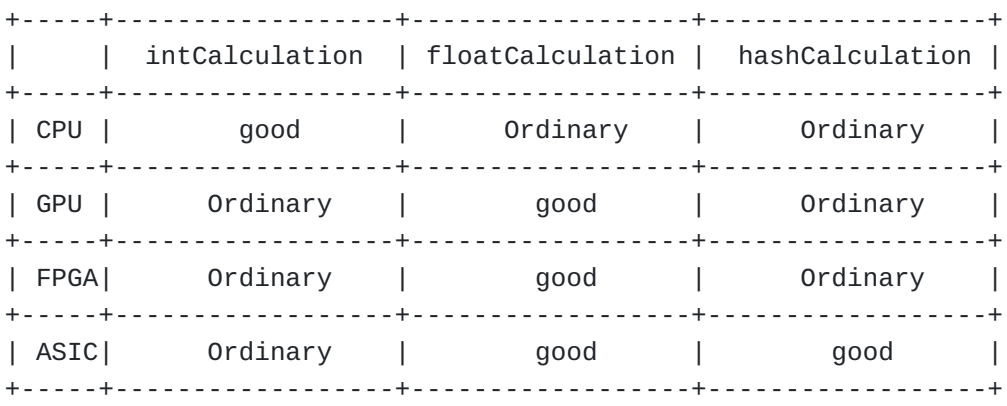

Figure 4: Relation of Computing Types and Chips

B.2. Communication, Cache and Storage Capacity

Besides the computing capacity, the communication, cache, and storage capacity should also be considered because each of them can potentially influence the comprehensive capacity of computing resource nodes.

The communication capacity is the external communication rate of computing nodes. From the point of view of a single node, the communication capability indicator of a node mainly includes the network bandwidth. Moreover, it is often to have cluster of service instances for one task (like Hadoop architecture). Therefore the network capacity among those instances are also important factor in assessing the capability of the cluster of the service nodes for one task.

The cache(memory) capacity describers the amount of of the cache unit on a node. The memory (CACHE) indicator mainly includes the cache(memory) capacity and cache(memory) bandwidth.

The storage capacity is the external storage (for example, hard disk) of the computing node. The storage indicators of a node mainly includes the storage capacity, storage bandwidth, operations per second (IOPs) and response time of the node.

B.3. Comprehensive Computing Capability Evaluation

Based on the architecture of computing resource modeling, this Section proposes the comprehensive performance evaluation methods based on the vectors to represent each capability of computing, communication, cache, and storage.

Figure 5~8 shows the vector of computing node(i) including each aspects.

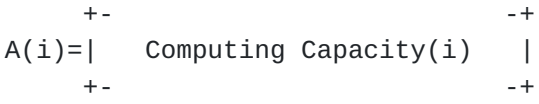

Figure 5: Computing Performance Vector

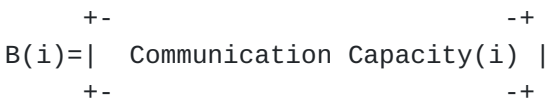

Figure 6: Comunication Performance Vector

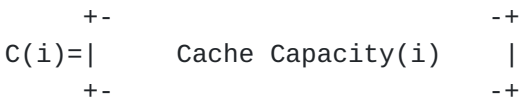

Figure 7: Cache Performance Vector

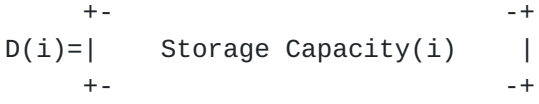

Figure 8: Storage Performance Vector

The vector of computing capacity, communication capacity, cache capacity and storage capacity could be further weighted to a comprehensive vector.

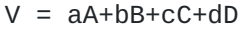

Figure 9: Comprehensive Vector

Where, a, b, c and d are the weight coefficients corresponding to the evaluation indicators of computing capacity, communication capacity, cache capacity and storage capacity respectively, and a+b+c+d=1.

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