Computing-Aware Traffic Steering   Y. Kehan
Internet-Draft   China Mobile
Intended status: Informational   H. Shi, Ed.
Expires: 24 April 2025   C. Li, Ed.
                                                    Huawei Technologies
                                                        21 October 2024

CATS ~~metric~~ Metrics Definition
draft-ysl-cats-metric-definition-01

Abstract

This document defines ~~the~~ a set of computing metrics used ~~in~~ for Computing-Aware Traffic Steering (CATS).

Commenté [MB1]: I don't think the intent is be exhaustive/restrict future metrics to be defined in the future.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Many ~~modern~~ computing services are deployed in a distributed way. In ~~this~~ such deployment mode, multiple service instances are deployed in multiple service sites to provide equivalent ~~function~~ service to end users. In order to provide better service to end users, a framework called ~~CATS~~ ~~(~~Computing-Aware Traffic Steering (CATS) [I-D.ietf-cats-framework] is ~~Proposed~~defined.

CATS ~~(Computing-Aware Traffic Steering) [I-D.ietf-cats-framework]~~ is a traffic engineering approach 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 [I-D.ietf-cats-framework]. Various ~~relevant~~ metrics may be used to enforce such computing-aware traffic steering policies.

To ~~effectively~~ steer traffic to ~~the appropriate~~a service contact instance, ~~network devices~~CATS components (C-PS, CATS-Forwarders, etc.) need a model of the service instance's computing status. A common definition of relevant computing metrics is essential for effective coordination between network devices and computing systems. Without standardized computing metrics, ~~devices on the network~~CATS functional elements may interpret and respond to traffic conditions and computing load differently, leading to inefficiencies and potential conflicts. A standardized metric allows both network devices and computing systems to evaluate load consistently, enabling precise traffic steering

Commenté [MB2]: May not age well :-)

Commenté [MB3]: Consistent with the sentence right after

Commenté [MB4]: What is intended by «model» here?

Commenté [MB5]: In addition to network-related metrics. Right?

Commenté [MB6]: Not every computing metric would be useful. Only those that can influence steering.

Commenté [MB7]: Only CATS entities.

a mis en forme : Surlignage

Commenté [MB8]: I would simply delete this. We don't need such statement to motivate this work.

Commenté [MB9]: I don't parse this.

decisions that optimize resource utilization and improve overall system performance.

Various considerations for metrics definition are ~~proposed~~ discussed in [I-D.du-cats-computing-modeling-description], which are useful in defining computing metrics.

Commenté [MB10]: If these are useful, why not import these cons and include them here? That would rationalize the set of documents out there ;-)

Based on the considerations defined in [I-D.du-cats-computing-modeling-description], this document defines relevant computing metrics for CATS by categorizing the metrics into three levels based on their complexity and granularity ~~richness~~details.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the following terms defined in [I-D.ietf-cats-framework]~~. We~~: ~~list them below for clarification~~.

* Computing-Aware Traffic Steering (CATS)~~: An architecture that takes into account the dynamic nature of computing resources and network state to steer service traffic to a service instance. This dynamicity is expressed by means of relevant metrics.~~

* 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~~.

Commenté [MB11]: What about service contact instance?

* Ingress CATS-Frowarder

3. Definition of Metrics

~~Many metrics are being discussed and/or defined in routing and computing area.~~ Definition and usage of specific metrics are ~~highly~~ related to the intended use case, especially in IT use cases. However, when considering ~~distributing~~ disseminating compute metrics to network devices, appropriate ~~categorizing~~ categorization and abstraction is required in order to ~~not~~avoid ~~introduce~~ introducing extra complexity into the network.

Commenté [MB12]: Why calling out this one specifically?

Commenté [MB13]: Of what?

Commenté [MB14]: Other considerations may impact complexity even with «appropriate abstraction»: e/g., this may may depend on the underlying machinery to share the metric, provisioning epoch, etc.

~~Based on the abstraction level of metrics, t~~This document defines three abstract metric levels ~~of metric~~ to meet different requirements of different use cases:

Commenté [MB15]: Which ones?

* Level 0 (L0): Raw ~~m~~Metrics. ~~In this level, t~~The metrics are not

abstracted, so different metrics use their own unit and format as used within a compute orchestration domain.

* Level 1 (L1): Normalized ~~Metrics~~ metrics in ~~Categories~~categories. ~~In this level, t~~The metrics are categorized into multiple dimensions, such as network, computing, and storage. Each category metric is normalized into a value.

Commenté [MB16]: Only one?

* Level 2 (L2): Fully Normalized ~~Metric~~metrics. ~~In this level,~~M~~m~~etrics are normalized into a single value, the category information or raw metrics information cannot be interpreted from the value directly.

3.1. Level 0: Raw Metrics

~~The metrics without any abstraction are Level 0 metrics.~~ ~~Therefore~~, Level 0 metrics encompass detailed, raw metrics, including but not limit to:

Commenté [MB17]: Already said above.

* CPU: Base Frequency, Number of Cores, Boosted Frequency, Memory Bandwidth, Memory Size, Memory Utilization Ratio, Core Utilization Ratio, Power Consumption.

* GPU: Frequency, Number of Render Unit, Memory Bandwidth, Memory Size, Memory Utilization Ratio, Core Utilization Ratio, Power Consumption.

* NPU: Computing Power, ~~Utlization~~Utilization Ratio, Power Consumption.

* Network: Bandwidth, TXBytes, RXBytes, HostBusUtilization.

* Storage: Available Space, Read Speed, Write Speed.

* Delay: Time ~~takes~~ taken to process a request.

Commenté [MB18]: Capacity? Throughput?

Commenté [MB19]: At which layer?

Commenté [MB20]: To be defined.

Commenté [MB21]: This service specific. Right?

Commenté [MB22]: What about requests/s that can be handled by an instance?

~~In L0, d~~Detailed information of a metric in L0 can be encoded into the protocol, and different services ~~has~~ have ~~its~~ their own metrics with different information elements. This kind of metrics are used widely in IT systems.

Commenté [MB23]: Which protocol?

Commenté [MB24]: Can we have examples?

Commenté [MB25]: Do you mean L0? Else?

Regarding network related raw metrics, IPPM WG has defined many types of metrics in [performance-metrics]. [RFC9439] also defines ~~a lot of~~many metrics ~~of~~ for packet performance and Throughput/Bandwidth. Regarding computing metrics, [I-D.rcr-opsawg-operational-compute-metrics] ~~defines~~ lists a set of cloud resource metrics.

3.2. Level 1: Normalized Metrics in Categories

~~In Level 1, t~~The metrics in L1 ~~will be~~are categorized into different categories, and ~~appropriate~~ abstraction will be applied to each category.

~~The~~ Level 0 raw metrics can be ~~categorized~~ classiifed into multiple categories, such as computing, networking, storage, and delay. In

Commenté [MB26]: Why is this not part of networking/computing?

each category, the metrics are normalized into a value that present the state of ~~the~~ a resource~~, making it as a Level 1 metric~~. Potential categories are shown below:

* Computing: A normalized value generating from the computing related L0 metrics, such as CPU/GPU/NPU L0 metrics

* Networking: A normalized value generating from the network related L0 metrics.

* Storage: A normalized value generating from the storage L0 metrics.

* Delay: A normalized value generating from computing/networking/ storage metrics, reflecting the processing delay of a request.

Commenté [MB27]: Not sure to get what is intended here.

Editor note: detailed categories can be updated according to the CATS WG discussion.

The L0 metrics, such as the ones defined in [performance-metrics] ,[RFC9439] and [I-D.rcr-opsawg-operational-compute-metrics] can be categorized into above categories. Each category will use its own method(weighted summary, etc.) to generate the normalized value. In this way, the protocol only care about the metric categories and its normalized value, and avoid to process the detailed metrics.

3.3. Level 2: Fully Normalized Metric.

L2 metric is a one-dimensional value derived from a weighted sum of L1 metrics or from L0 metrics directly. ~~Different s~~Service may have ~~has~~ ~~its~~their own normalization method which might use different metrics with different weight. ~~For the An ingress~~ Ingress ~~CATS~~ CATS-~~router~~Forwarder~~, it~~ can compare ~~the~~a metric value to make the traffic steering decision (e.g., larger value has higher priority)~~-~~. ~~In some cases, s~~Some implementations may support ~~to configure~~ configuration of ~~the i~~Ingress CATS~~ router~~ -Forwarders ~~to know~~ with the metric normalizing method so that it can decode the affection from the L1 or L0 metrics.

Commenté [MB28]: Not sure to connect this with the previous text.

This method simplifies the complexity of transmission and management of multiple metrics by consolidating them into a single, unified measure.

a mis en forme : Surlignage

~~The below f~~Figure 1 shows the logic of metrics in Level 0, ~~level~~ Level 1, and L~~l~~evel 2.

```
                     +--------------+
Level 2       +------| Normalized M |-------+
              |      +--------------+       |
              |             |               |
              |             |  Normalizing  |
         +----------+    +---------+     +---------+
Level 1  | Category M1 |     | Category M2|      | Category M3|  ...
         +----------+    +---------+     +---------+
```

```
                | |             |                | |
                | |             |Normalizing     | |
        +-------+ +-------+   +-------+   +-------+ +-------+
Level 0 |Raw M1| |Raw M2|...|Raw M3|...|Raw M4| |Raw M5| ...
        +-------+ +-------+   +-------+   +-------+ +-------+
```

Figure 1: Logic of CATS Metrics in levels

4. Representation of Metrics

A hierarchical view of metrics has been shown in ~~the section above~~Figure 1. ~~In t~~This section~~,~~ includes ~~a~~the detailed representation of metrics ~~will be described~~.

[RFC9439] gives a good way to show the representation of some network metrics which is used for network capabilities exposure to applications. This document further ~~describe~~describes the representation of CATS metrics.

Basically, in each metric level and for each metric, there will be some common fields for representation, including metric type, unit, and precision. Metric type is a ~~name~~ label for network devices and protocols to recognize what the metric is. "unit" and "precision" are ~~necessary~~ usually associated with the ~~for~~ metric ~~descripition~~. How many bits a metric occupies in protocols is also required.

Beyond these basic representations, the source of the metrics ~~MUST~~must also be declared. As defined in [RFC9439], there are three cost-sources, nominal, sla, and estimation. This document further divide the estimation type into three sub-types, direct measurement, aggregation, and normalization, since different levels of metrics require different sources to acquire CATS metrics. Directly measured metrics have physical meanings and units without any processing. ~~Aggregation~~ Aggregated metrics can be either physically meaningful or not, and they maintain their meanings compared to the directly measured metrics. Normalized metrics can have physical meanings or not, but they do not have units, and they are just numbers that used for routing decision making.

To be more ~~fine~~ fine-grained, ~~This~~ this document ~~refer~~ refers to the definition of [RFC9439] on the metrics statistics.

4.1. Level 0 Metric Representation

Raw metrics have exact physical meanings and units. They are directly measured from the underlying computing resources providers. Lots of definition on this level of metrics have been defined in IT industry and other ~~standardisations~~standardizations [DMTF], and this document only show some examples for different categories of metrics for reference.

4.1.1. Compute Raw Metrics

Commenté [MB29]: These are already classified/categorized!

Commenté [MB30]: Can we simply use that, so?

Commenté [MB31]: ?

Commenté [MB32]: I guess, you mean the value range.

Commenté [MB33]: Why? where? And for what purpose?

a mis en forme : Surlignage

a mis en forme : Surlignage

* The metric type of compute resources are named as “compute_type: CPU” or “compute_type: GPU”. Their frequency unit is GHZ, the compute capabilities unit is FLOPS. Format should support integer and FP8. It will occupy 4 octets.

* Example[TBA].

4.1.2. Storage Raw Metrics

The metric type of storage resources like SSD are named as “storage_type: SSD”. The storage space unit is megaBytes(MBs). Format is integer. It will occupy 2 octets. The unit of read or write speed is denoted as MB per second.

* Example[TBA].

4.1.3. Network Raw Metrics

The metric type of network resources like bandwidth are named as “network_type: Bandwidth”. The unit is gigabits per second(Gb/s). Format is integer. It will occupy 2 octets. The unit of TXBytes and RXBytes is denoted as MB per second.

* Example[TBA].

4.1.4. Delay Raw Metrics

Commenté [MB34]: Is it about the exact delay for a specific transaction? An avg, a max, a percentile, etc.?

Delay is a kind of synthesized metric which is influenced by computing, storage access, and network transmission. It is named as “delay_raw”. Format should support integer and FP8. Its unit is microsecond. It will occupy 4 octets.

4.1.5. Considerations on the Sources of Metrics and the Statistics

The sources of L0 metrics can be nominal, directly measured, or aggregated. Nominal L0 metrics are provided initially by resource providers. Dynamic L0 metrics are measured and updated during service stage. L0 metrics also support aggregation, in case that there are multiple service instances.

The statistics of L0 metrics will follow the definition of ~~section~~Section 3.2 of [RFC9439].

4.2. Level 1 Metric Representation

Normalized metrics in categories have physical meanings but they do not have unit. They are numbers after some ways of abstraction, but they can represent their type, in case that in some use cases, some specific types of metrics require more attention.

4.2.1. Normalized Compute Metrics

Commenté [MB35]: The definition needs some elaboration

The metric type of normalized compute metrics is “compute_norm”, and its format is integer. It has no unit. It will occupy an octet.

Commenté [MB36]: How is this set?

* Example[TBA].

4.2.2. Normalized Storage Metrics

The metric type of normalized ~~compute~~ storage metrics is "storage_norm", and its format is integer. It has no unit. It will occupy an octet.

* Example[TBA].

4.2.3. Normalized Network Metrics

The metric type of normalized ~~compute~~ network metrics is "network_norm", and its format is integer. It has no unit. It will occupy an octet.

* Example[TBA].

4.2.4. Normalized Delay

The metric type of normalized ~~compute~~ delay metrics is "delay_norm", and its format is integer. It has no unit. It will occupy an octet.

* Example[TBA].

4.2.5. Considerations on the Sources of Metrics and the Statistics

The sources of L1 metrics ~~is~~ are normalized and support aggregation. Based on L0 metrics, service providers design their own algorithms to normalize metrics. For example, assigning different cost values to each raw metric and do summation. L1 metric do not need further statistical values.

4.3. Level 2 Metric Representation

~~The~~ A fully normalized metric is a single value which does not have any physical meaning or unit. Each provider may have its own methods to derive the value, but all providers ~~MUST~~ must follow the definition in this section to represent the fully normalized value.

Metric type is "Norm_fi". The format of the value is non-negative integer. It has no unit. It will occupy ~~a~~ an octet.

The fully normalized value also supports aggregation when there are multiple service instances providing these fully normalized values. When providing fully normalized values, service instances do not need to do further statistics.

5. Comparison of three layers of metric

From L0 to L1 to L2, the computing metric is consolidated. Different level of abstraction can meet the requirements from different services. Table 1 shows the comparison among metric levels.

```
+=======+=============+===============+===========+==========+
| Level | Encoding    | Extensibility | Stability | Accuracy |
|       | Complexity  |               |           |          |
```

Commenté [MB37]: How is this defined?

Commenté [MB38]: Idem as above

Commenté [MB39]: How is this defined?

Commenté [MB40]: That is?

a mis en forme : Surlignage

a mis en forme : Surlignage

Commenté [MB41]: ?

Level 0	Complicated	Bad	Bad	Good
Level 1	Medium	Medium	Medium	Medium
Level 2	Simple	Good	Good	Medium

Table 1: Comparison among Metrics Levels

Since Level 0 metrics are raw metrics, therefore, different services may have their own metrics, resulting in hundreds or thousands of metrics in total, this brings huge complexity in protocol encoding and standardization. Therefore, this kind of metrics are always used in customized IT systems case by case. In Level 1 metrics, metrics are categorized into several categories and each category is normalized into a value, therefore they can be encoded into the protocol and standardized. Regarding the Level 2 metrics, all the metrics are normalized into one single metric, it is easier to be encoded in protocols and standardized. Therefore, from the encoding complexity aspect, Level 2 and Level 1 metrics are suggested.

Similarly, when considering extensibility, new services can define their own new L0 metrics, which requires protocols to be extended as needed. Too many metrics type can create a lot of overhead to the protocol resulting in a bad extensibility of the protocol. Level 1 introduce only several metrics categories, which is acceptable for protocol extension. Level 2 metric only need one single metric, so it brings least burden to the protocol. Therefore, from the extensibility aspect, Level 2 and Level 1 metrics are suggested.

Regarding Stability, new Level 0 raw metrics may require new extension in protocol, which brings unstable format for protocol, therefore, this document does not recommend to standardize Level 0 metrics in protocol. Level 1 metrics request only few categories, and Level 2 Metric only introduce one metric to the protocol, so they are preferred from the stability aspect.

In conclusion, for computing-aware traffic steering, it is recommended to use the L2 metric due to its simplicity. If advanced scheduling is needed, L1 metric can be used. L2 metrics are the most comprehensive and dynamic, therefore transferring them to network devices is discouraged due to their high overhead.

Commenté [MB42]: Some concrete details about these Level metrics are needed.

Editor notes: this draft can be updated according to the discussion of metric definition in CATS WG.

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. References

8.1. Normative References

[I-D.ietf-cats-framework]
Li, C., Du, Z., Boucadair, M., Contreras, L. M., and J. Drake, "A Framework for Computing-Aware Traffic Steering (CATS)", Work in Progress, Internet-Draft, draft-ietf-cats-framework-04, 17 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-04>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

8.2. Informative References

[DMTF] "DMTF", n.d., <https://www.dmtf.org/>.

[I-D.du-cats-computing-modeling-description]
Du, Z., Yao, K., Li, C., Huang, D., and Z. Fu, "Computing Information Description in Computing-Aware Traffic Steering", Work in Progress, Internet-Draft, draft-du-cats-computing-modeling-description-03, 6 July 2024, <https://datatracker.ietf.org/doc/html/draft-du-cats-computing-modeling-description-03>.

[I-D.rcr-opsawg-operational-compute-metrics]
Randriamasy, S., Contreras, L. M., Ros-Giralt, J., and R. Schott, "Joint Exposure of Network and Compute Information for Infrastructure-Aware Service Deployment", Work in Progress, Internet-Draft, draft-rcr-opsawg-operational-compute-metrics-06, 7 July 2024, <https://datatracker.ietf.org/doc/html/draft-rcr-opsawg-operational-compute-metrics-06>.

[performance-metrics]
"performance-metrics", n.d., <https://www.iana.org/assignments/performance-metrics/performance-metrics.xhtml>.

[RFC9439] Wu, Q., Yang, Y., Lee, Y., Dhody, D., Randriamasy, S., and L. Contreras, "Application-Layer Traffic Optimization (ALTO) Performance Cost Metrics", RFC 9439, DOI 10.17487/RFC9439, August 2023, <https://www.rfc-editor.org/rfc/rfc9439>.

Authors' Addresses

Kehan Yao
China Mobile
China

Email: yaokehan@chinamobile.com

Hang Shi (editor)
Huawei Technologies
China
Email: shihang9@huawei.com

Cheng Li (editor)
Huawei Technologies
China
Email: c.l@huawei.com