Compute Modeling and Metrics

IETF 119

CATS (Computing-Aware Traffic Steering) WG

https://datatracker.ietf.org/doc/draft-du-cats-computing-modeling-description/

Background and Motivation

- Motivation of computing and network convergence
  - services with a lot of computing resource involved are increasing
  - meanwhile, users expect a low-latency access to these services
- Therefore, the operators need to
  - deploy “Cloud + MECs” service structure, in which multiple points can provide the same service
  - meanwhile, provide a proper scheduling method for user requests, so as to LB both the network resource and computing resource
- However,
  - current anycast mechanism will only consider the network distance without computing information
  - current LB will mainly be done within a DC, and a user request can be redirected to other places
- We are short of an on-path LB mechanism based on the network, in which traffic steering will be near to users, with knowledge of computing information
  - thus, the decision point in the network should be aware of certain degree of computing information

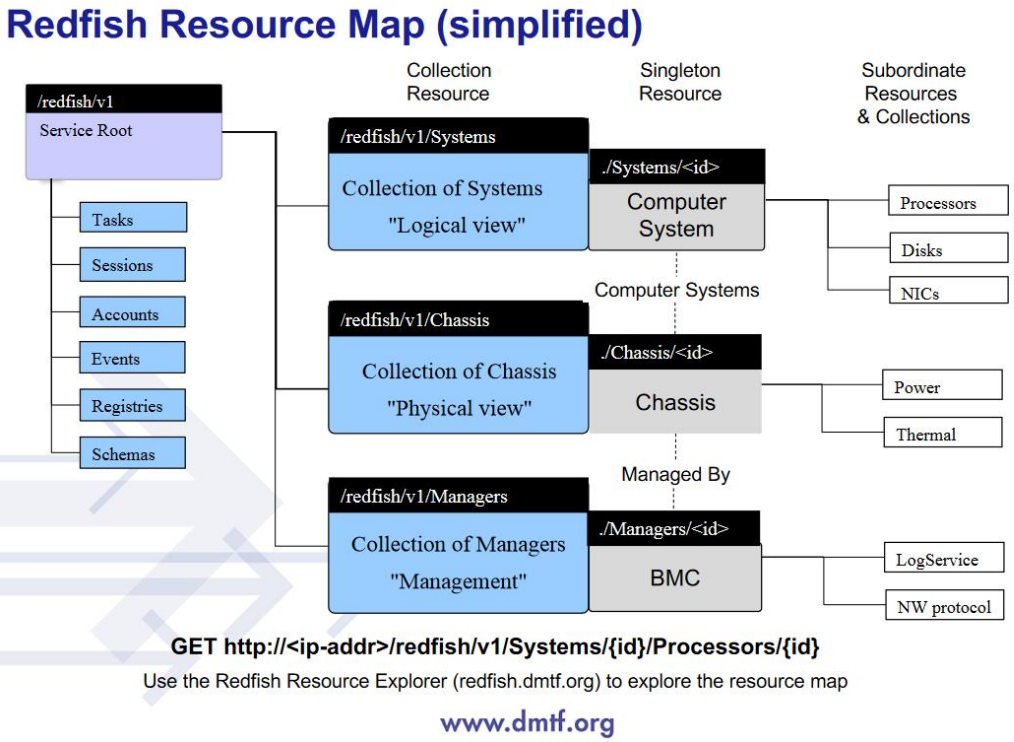
Usage of Computing Metrics

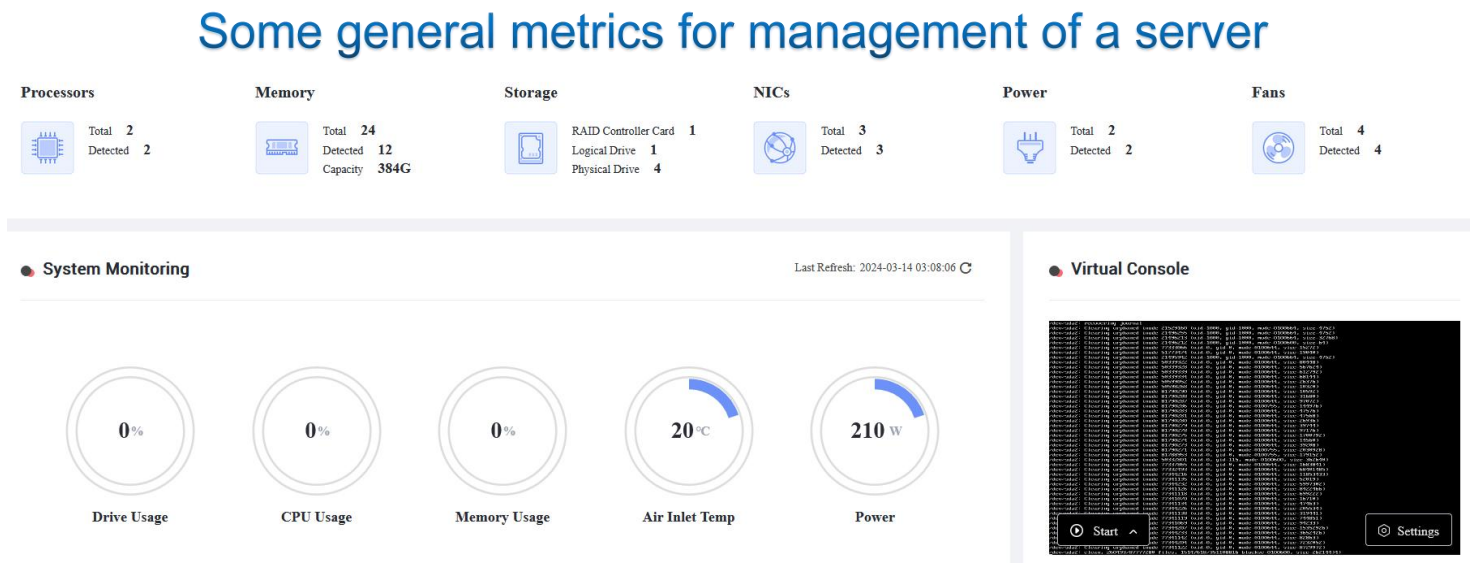
- In the maillist of CATS, it is talked about that “computing model and metrics” can refer to the work in DC management planes, for example
  - The Redfish
  - The Anuket Project
- However, it is mainly for computing resource management, which can be used for
  - service deployment, i.e., find a place with enough computing resource to establish a service
  - service adjustment, i.e., modify the resource configured previously
- Potentially, a uniform metrics for billing would also make sense, if we want to make the service of heterogeneous computing resource tradable
- However in CATS, we mainly care about the computing metrics that need to be announced into the network
  - which will influence the route for the Service ID (normally an anycast IP address)
  - The CATS metrics can even be a subset of the metrics used for management planes

Some metrics examples: Redfish

Redfish https://www.dmtf.org/standards/redfish

- DMTF’ s Redfish® is a standard designed to deliver simple and secure management for converged, hybrid IT and the Software Defined Data Center (SDDC). Both human readable and machine capable, Redfish leverages common Internet and web services standards to expose information directly to the modern tool chain.





Some metrics examples: Anuket

https://anuket.io

- Anuket delivers a common model, standardized reference infrastructure specifications, and conformance and performance frameworks for virtualized and cloud native network functions, enabling faster, more robust onboarding into production, reducing costs and accelerating communications digital transformations.

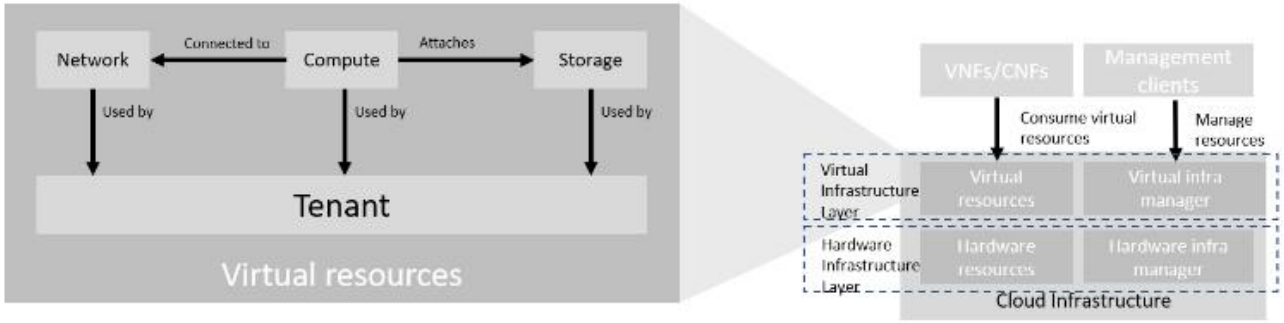


Figure 3.2 Virtual Infrastructure Resources provide virtual compute, storage and networks in a tenant context

Attribute	Description
`name`	name of the virtual host
`vcpus`	number of virtual CPUs
`ram`	size of random access memory in GB
`disk`	size of root disc in GB
`nics`	sorted list of network interfaces connecting the host to the virtual networks
`acceleration`	key/value pairs for selection of the appropriate acceleration technology
`metadata`	key/value pairs for selection of the appropriate redundancy domain

Table 3-2: Attributes of compute resources

Analysis of the requirements

- As mentioned before, we would mainly care about the CATS traffic steering
- For example, if the decision point of CATS is the Ingress, the job would be to select a proper Egress, which connects to a proper service site
- To select an Egress, the Ingress would not need all the computing information on the management plane
- Some suggestions have been received here
  - S1: The computing metrics in CATS should be few and simple, to avoid exposing too much information of the service points
  - S2: The computing metrics in CATS should be evolveable for the future extensions
  - S3: The update frequency of computing metrics needs to be considered carefully, to avoid too much pressure on the network node
- Generally speaking, to start with simple cases...

To start with simple cases

- So far from the mailist, we have three potential metrics that are considered as the start point, i.e., worthy to be announced to the Ingress and influence the selection of the Egress/service site
- The first one is the “predicted computing delay”
  - the meaning: “the estimate of the duration of my processing of request”
- The second one is the “server capability”
  - For example, one server can support 100 simultaneous sessions and another can support 10,000 simultaneous sessions
- The third one is the “status indication”
  - For example, an indication of "please stop sending new sessions to instance A”
- Other metrics can also be defined in future if it is considered valuable

Corresponding of metrics and optimization objectives

- Indeed, which metric to announce is relevant to the policy in the decision point, and the policy in the decision point is relevant to the optimization objective of CATS
- We have two main optimization objectives in CATS
  - The first one is to minimal the total delay in the network domain and the computing domain
    - the “predicted computing delay” metric is valuable for this purpose
  - The second one is to LB the network and computing resources
    - the “server capability” metric is valuable for the purpose, because it can work as a weight value for LB
    - The operator will care about the second optimization objective more, while the “server capability” is relatively static
- The third metric “status indication” would be helpful when the server is heavy load
  - However, it is a temp and dynamic value, so can only work additionally
- Hence, we can choose a default optimization objective (currently with two candidates) for services in CATS, have one or more default metrics for it, and make things happen

Thanks and welcome for comments