Buffering Target

A buffering target represents a logical target of data placement, i.e., parts of or a full blob can be placed there by the DPE. Buffering targets are logical constructs that are statically mapped by Hermes to underlying physical resources.

Contents

Terminology

A buffering target consists of two components:

Virtual Device

This represents a way to get to the actual storage. It could be a file handle and an offset, a memory address, a partition of a drive, etc.

NodeID

The identifier of the node that is responsible for the virtual device.

Tiers are the partitions of a partitioned set of targets order by a score, which is calculated based on a set of prioritized characteristics. Tier 1 represents the "best" targets according to the prioritized characteristics, and the tiers get "worse" as the tier number increases. For example, tier 1 might be a local RAM target when bandwidth is the ordering characteristic, but it might be a burst buffer target when remaining capacity is prioritized.

When the DPE runs, it is given an appropriate list of targets. If a placement fails, it can request an extended list of targets (neighborhood or global).

For now we map 1 `TargetID` to 1 (NodeID, VirtualDevice) pair, but the option is open for 1 to n and n to m.

The set of targets can be partitioned in the form of *topologies*. In some cases, the aggregate characteristics of such partitions can be defined based on the characteristics of the underlying targets.

Goals

- Provide a way for the DPE to operate on a reduced (or custom) set of resources.
- Remove certain resources from DPE consideration.
- Create orderings of resources based on characteristics (i.e., tiered groups).

Charateristics

Each buffering target has the following characteristics.

- Targets $d_i, i = 1, \ldots, D$
  - Target configuration/specs.
    - $Cap[d_i]$ - the total capacity of target d_i
    - $Wbw[d_i]$ - the HW max. write bandwidth of target d_i
    - $Rbw[d_i]$ - the HW max. read bandwidth of target d_i
    - $Alat[d_i]$ - the average HW access latency of target d_i (measured as time)
    - $Pwr[d_i]$ - the energy consumption of target d_i (measured in Watts)
    - $Concy[d_i]$ - the HW concurrency of target d_i (measured in lane count)
    - $End[d_i]$ - the endurance (wear and tear) of target d_i (measured as percentage of the expected storage cycles over the life time)
    - $Rrat[d_i]$ - the reliability rating of target d_i (measured in Trumps)
    - $Speed[d_i]$ - the average I/O speed of target d_i (measured as MB/s)
  - Variables
    - $Avail[d_i]$ - the availability of target d_i (Boolean)
    - $Rem[d_i]$ - the remaining capacity of target d_i
    - $Load[d_i]$ - the expected completion time of outstanding requests on target

Example

Assume a system with 3 nodes, each with three targets (RAM, NVMe, and burst buffer). Assume a neighborhood is any 2 of the three nodes. This means a local target list will consist of 3 targets, a neighborhood of 6, and the global target list of 9.

Kitchen Sink

From the OctopusFS paper (https://www.cut.ac.cy/digitalAssets/122/122275_1 From Wrike (https://www.wrike.com/open.htm?id=416733774): 00sigmod.pdf):

- Tiers $T_1, \ldots, T_k$
- Media m_i
  - $Tier[m_i]$ - the tier of medium m_i
  - $Cap[m_i]$ - the total capacity of medium m_i
  - $Rem[m_i]$ - the remaining capacity of medium m_i
  - $NrConn[m_i]$ - the number of active I/O connections to medium m_i
  - $WThru[m_i]$ - the sustained write throughput of medium m_i
  - $RThru[m_i]$ - the sustained read throughput of medium m_i
- Workers $W_1, \ldots, W_n$
  - Slightly different concept
    - Stores and manages file blocks on storage media
    - Serves read and write requests from clients
  - $W_i =< node, tier >$
  - Workers are a dedicated thread per tier available on the node
  - Worker characteristics:
    - Capacity
    - BW
    - Latency
    - Energy consumption
    - Concurrency (expressed as the number of lanes of the bus e.g., PCIex8 or SATA)
    - Queue pressure (outstanding requests)
      - Aggregate data size in queue
      - Number of pending requests

- Block creation, deletion, replication (instructed by name nodes HDFS...)

Retrieved from "https://hermes.page/index.php?title=Buffering_Target&oldid=819"

This page was last edited on 2 December 2020, at 15:25.