Practical Work 3
DAI - HEIG-VD

Loïc Herman Massimo Stefani

II. Unicast Distributed System Monitoring Protocol

A. Overview

UDSMP serves a critical role in the realm of distributed systems, offering a robust solution for tracking resource utilization across various components within a distributed network. Its primary objective is to transmit essential data related to CPU, RAM, and disk consumption from individual machines to system administrators and operators to make informed decisions regarding system performance and resource allocation.

B. Protocol

The protocol follows a request-response pattern, where readers send requests to the monitoring server and receive responses in return using the port 6343. This pattern facilitates two-way communication, enabling clients to query the server for specific information.

UDSMP enforces strict case sensitivity for all its messages. This consistency ensures that data is accurately interpreted and processed by both the sending and receiving ends, eliminating potential issues caused by case mismatches.

All messages exchanged within the UDSMP protocol must be encoded in UTF-8. This standardized encoding scheme ensures compatibility across diverse systems and prevents data corruption during transmission.

Clients must send the first request to which the server might answer, it is recommended to have a timeout of 1 second in all implementations in case the message gets lost or the server is unavailable.

C. Messages

1) *From reader to aggregator:*

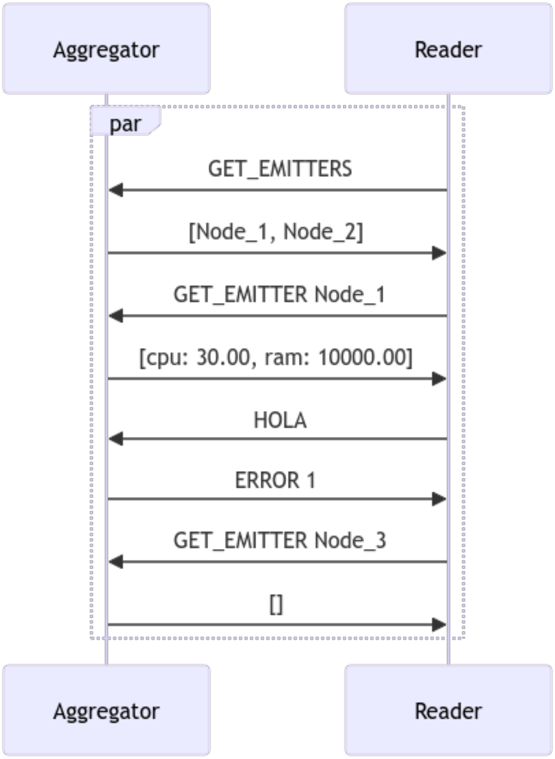
1. `GET_EMITTERS` : Request to get the list of emitters
2. `GET_EMITTER <emitter>` : Request to get the values of a specific emitter
   1. `<emitter>` : Hostname of the emitter to get the values from

2) *From aggregator to reader:*

1. `[<emitter>, ...]` : List of emitters. Empty if there is no emitter.
   1. `<emitter>` : Hostname of the emitter
2. `[<metric>: <value>, ...]` : List of values for the requested emitter. The list limit is 10 values and can be empty if the emitter does not exist.
   1. `<metric>` : Type of the data sent (CPU, RAM, DSK)
   2. `<value>` : Value of the sent data
      1. cpu : CPU consumption in percentage
      2. ram : RAM consumption in MB
      3. dsk : Disk consumption in MB
3. `ERROR <ErrorNum>` : Send an error to the reader
   1. `<ErrorNum>` : Error number
      1. `1` : The message could not be parsed

D. Examples

The next figure contains an example of a couple readers that may be sending messages in parallel to the server. The server will then respond to each reader with the requested information.


Aggregator
Reader
par
GET_EMITTERS
[Node_1, Node_2]
GET_EMITTER Node_1
[cpu: 30.00, ram: 10000.00]
HOLA
ERROR 1
GET_EMITTER Node_3
[]
Aggregator
Reader