Q. How does your script work?

A. I have used JMeter (an open source tool form within the apache family) to construct the API service. Below is the API Request:

GET https://s1.ripple.com:51234/

GET data:

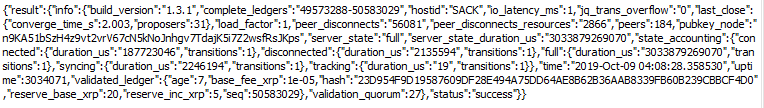
{

"method": "server\_info"

}

The “method:”server\_info” is passed as the payload with the request.

In the response it returns the details of the ledger with the timestamp. Post calling this API service in a loop showcased that the "seq" value changing along with the “time”.



In the script, I have extracted the “seq” value and the “time” in 2 different parameters such as “time” and “Sequence” and have written a custom code in “JSR223 sampler” (an element of JMeter) to write these value in real time into a “data.dat” file.

Q. How did you decide on your polling interval?

A. When calling this API service in a loop, it is observed that the “seq” value gets incremented by 1 after every 3 secs. Hence the polling interval factor value is 3 secs. This can be achieved by keeping the sleep timer (called as constant timer)

Q. What do the results tell you?

A. As per my understanding the results tells the completed ledgers id’s and the transaction details on each validated ledger with the age of the ‘validated ledger’, sequence number and the hash key value ensuring that every server in the network to have the complete copy of the exact same ledger state.

It also tells that the “seq” value is incremented in every 3 secs of time interval.

On plotting the results between the time and the sequence number, it has been seen that the sequence number increases linearly with the time.

Q. What might explain the variation in time between new ledgers?

A. The variation in time can be because of few factors (below listed are 2 according to me) –

1. There can be occasionally times where the participants in the network are out of sync and cause weird overlaps when it comes to consensus, hence increasing the time of closure of the ledger by few secs.

2. At times when network congestion happens, the ‘leader’ node isn’t able to respond with the closure time and hence at this point the network chooses a new leader to establish the close time and this additional time can add a few secs to the overall time.

Bonus question #1: Enhance your script to calculate the min, max, and average time that it took for a new ledger to be validated during the span of time captured.

Answer: On running the script for the interval between the validation of the new ledger, below are the captured time responses (in ms):

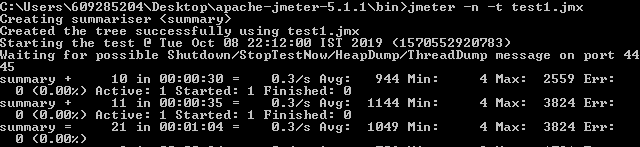


The above values have been captured when the ‘complete\_ledgers’ value changes.

Bonus question #2: There are some other (better) ways that you could use the rippled API to find how long each ledger took to close/validate. Using the API documentation, find and describe one of these methods (you don’t need to actually implement it).

Answer: Using the ‘Get Ledger’ API method: data.ripple.com/v2/ledgers/{sequencenumber}, one can retrieve the “date” field which depicts the creation of the ledger and also the ledger index and using the ‘GET /v2/ledgers/{identifier}’ one can retrieve the end time of the ledger. The difference between the two can result the time taken to close/validate the ledger.

Execution snapshot



Gnuplot

