Quality Assurance: Tests, Strategy, Reports

fabric-test repository

- fabric-test
- Daily Test Suite Description
- Steps to create a release branch for fabric-test

Reports

- v1.0.0-alpha release testing description
- Release System Verification Test (SVT) Report - SVT test descriptions and results
- SVT Daily Test Reports for ongoing testing on master development branch - Behave, LTE, OTE, and more and PTE test results
- #fabric-quality channel in RocketChat - Ask questions and become active in our testing community
- CI Gap Analysis - High-level analysis of Continuous Integration pipeline
- HFRD Test Tool (Hyperledger Fabric Regression Driver)

Test Strategy Presentations

During the lifecycle of the hyperledger fabric project the test strategy and process changes to allow for improvements with each release. The following presentations have outlined the different strategies at different points during the project.

<table>
<thead>
<tr>
<th>File</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Powerpoint Presentation 2019 Test Suite Summary.pptx</td>
<td>Apr 05, 2019 by David Enyeart</td>
</tr>
<tr>
<td>Microsoft Powerpoint Presentation Test Strategy-v1.2.pptx</td>
<td>Apr 05, 2019 by Latitia Haskins</td>
</tr>
<tr>
<td>Microsoft Powerpoint Presentation Interoperability Tests.pptx</td>
<td>Apr 05, 2019 by Latitia Haskins</td>
</tr>
</tbody>
</table>

Disk Space and RAM Requirements

Handling Large Transactions

The default largest blocksize is 100MB, due to a limitation in GRPC. Remember that each block on the chain is duplicated in every peer and orderer, so using such large transactions would consume disk space quickly! It is preferable to store smaller transactions on the chain, possibly including a hash of an object stored off-chain.

To handle a maximum size proposal payload approaching 50MB (which would result in a 100 MB transaction, including the payload read-write set in the response, plus typically 4 to 10 KB for identify/signature certs on the block) in our k8s cluster network tests, the only increase required for the default configuration settings is for the AbsoluteMaxBytes. If you are using fast disks, then set the WAL SnapshotIntervalSize (required for raft orderers only) to be big enough for multiple blocks, to avoid individual writes for every block. Of course, one must be sure to use adequate pod memory GB RAM of the k8s hosts. Below are the relevant config settings for a test in k8s cluster, as used in this sample network specification file being used with a launcher tool in automation tests:

- Increase orderer.batchsize.absolutemaxbytes up to 100 MB (Refer to GRPC for exact payload size limitation.)
- Increase orderer.batchsize.preferredmaxbytes (optional)
- Increase orderer.etcdraft_options.SnapshotIntervalSize up to 100 MB (optional)
- Increase the pod memory to minimum 4Gi for the peers, cc containers, and kafkas
- Increase the pod memory to minimum 2Gi for the orderers, couchdb

Note: to process multiple huge msgs, it is recommended to allocate more GB for the pods, to avoid causing the peers or orderers to panic due to resource limitations.

Note: in raft networks, the orderers update the Write Ahead Log (WAL) periodically. If disk-write speeds are too slow (eg. slower than 10 IOPS/GB), then you may experience seconds of delays when processing huge messages - and possibly even dropped transactions if a leadership change is triggered due to the slow WAL disk write (which may block the raft orderer channel leader from sending heartbeats to other orderers).

How much disk space is required for a blockchain?

- First, be aware that, in general terms, a full copy of a blockchain channel ledger is stored on each peer that joins the channel, and on each orderer, and (when using Kafka) on each kafka broker in the RF set. Also, system managers could set up storage systems to store multiple copies of the peer ledger.
The ones and zeroes stored in each location are not identical; the transactions in the peer are modified to contain the VSCC result (TxValidationCode code: valid, or reason for being invalid), which differentiates it from that of the orderer. And the KBs store transactions and markers, not blocks, so one estimate puts the size requirement in the KB as 10% less than the orderer and peer ledger size.

Beyond that, the answer is different for each blockchain. To devise an accurate number for disk space requirements, system managers must determine the average size of transactions. Multiple things impact the size of each transaction, including:

- the endorsement policies used in the channels - which affect the number of endorser signatures attached to each transaction;
- the average size of each signature attached by each requested endorser peer and by the orderer that delivers the transaction block (an estimate for one signature with identity and root cert is 1KB size);
- the average size of data stored for each transaction of the specific chaincode (ranging from a few bytes to MegaBytes or larger, depending on the documents or whatever is processed by the business app chaincode).
- And to a lesser factor, a larger batchsize for the orderer service would mean fewer blocks, which reduces overhead (note a simple block header could be 2KB - 4KB).

Of course, to compute space requirements, multiplication factors include the average size of each transaction, and the number of transactions you expect to process and store on the blockchain forever.

Here is one example:

- For a very simple chaincode that stores only a few bytes of data with each transaction, requiring a single peer endorsement, using root certs generated without intermediary certificate authority signatures: one could observe that the size of each block (containing 1 transaction each) is approximately 6KB.
- In a blockchain network with 10 blockchain channels, using a single chaincode, each processing only 10 transactions per day for 10 years, we compute that a ledger would require $365,000 \times 6KB = 2.2 \text{ GB}$.
- Remember that every orderer, every Kafka Broker in RF set, and each peer that joins the channel would each require 2.2 GB disk space for the projected timeframe and usage. Therefore, for a network like this standard test network containing 2 organizations with 2 peers each, and a basic kafka ordering service as defined here, that would mean 10 nodes, each requiring 2.2 GB.