Fabric Version: 1.4.1
Caliper Version :
The deployed smart contract :

The smart contract used in the experiments is the Marble Chain code provided in the sample in Caliper Benchmark samples.
Link:
https://github.com/tharindupr/caliper-benchmarks/blob/master/src/fabric/samples/marbles/go/marbles.go

Chain Code Logic :

Transaction details: This transaction will initialize the counts of marbles belongs to 4 people. Each person will have marbles of the colours red, blue, green, black, white, pink and rainbow.

Key Words :

Expected Send Rate (TPS): Send rate defined in the benchmark configuration that needed be achieved in each round during the benchmark.

Send Rate(TPS): Number of transactions per second is generated by all the clients. These generated transactions are equally divided among all the clients. The number of clients can be defined in the configuration.

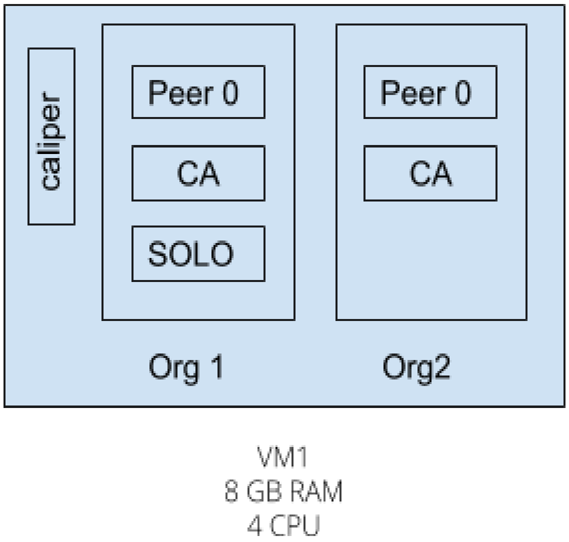
Succ: The number of successful transactions that are committed to the ledger.

Fail: The number of transactions that are not executed properly and committed to the ledger.

Following are the benchmarking results of the conducted experiments up to now.

Benchmark 01.

Maximum number of transactions per round: 1000
System Under Test :



StateDB : GoLevelDB
Network: 2 Organization with 1 peer.
Setup: Single Host
Ordering: Solo
Number of Clients: 5

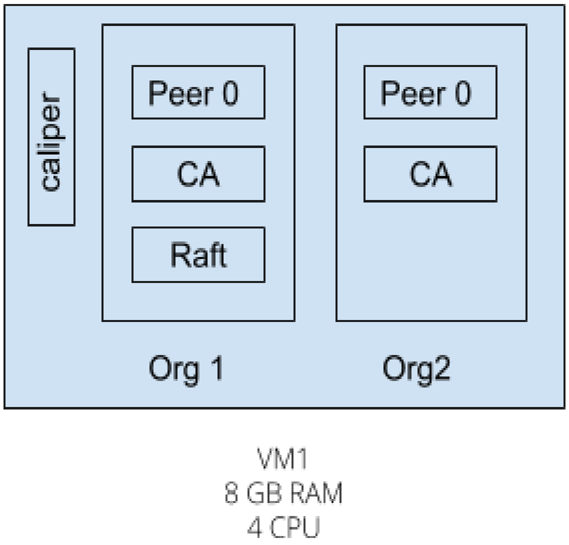
Name	Expected Send Rate (TPS)	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Round1	50	1000	0	45.4	13.50	0.69	6.32	41.7
Round2	75	982	18	61.7	12.38	0.40	5.18	26.7
Round3	100	851	149	60.1	13.08	0.40	7.26	24.5
Round4	125	691	309	64.0	14.53	0.93	9.14	19.7
Round5	150	884	116	64.1	14.55	0.39	8.22	24.5

Round6	175	727	273	70.3	14.07	0.41	8.76	21.3
Round7	200	880	120	69.1	13.50	0.41	8.25	24.6
Round8	300	763	237	69.8	13.30	0.56	8.23	21.2
Round9	400	774	226	72.1	13.89	0.85	8.26	22.0
Round10	500	795	205	66.9	16.77	0.55	8.40	22.2

This setup is on a single virtual machine as shown in the above diagram. Only **Round 1** has achieved a 100% transaction success (No transaction failures). The maximum throughput was 41.7 TPS for a 45.4 TPS send rate. The maximum send rate that is achieved in this setup is 72.1 TPS due to the resource constraints in the VM1. Another reason could be due to a large number of docker containers trying to utilize the same hardware.

Benchmark 02.

Maximum number of transactions per round: 1000
System Under Test :



StateDB : GoLevelDB
Network: 2 Organization with 1 peer.
Setup: Single Host
Ordering: Raft

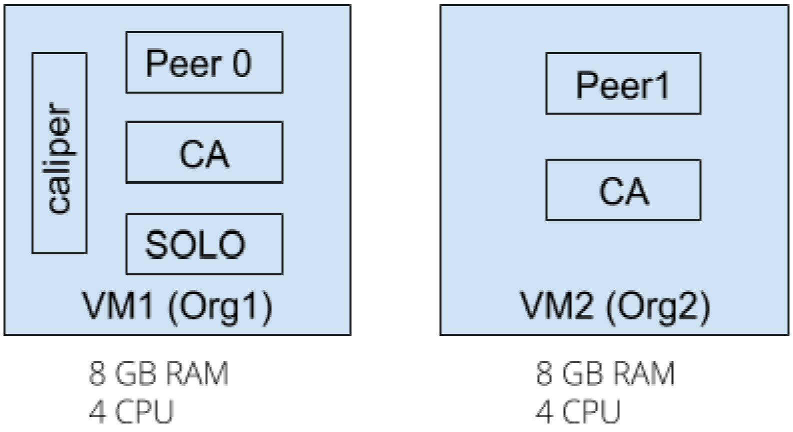
Number of Clients: 5

Name	Expected Send Rate (TPS)	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Round1	50	954	33.4	14.49	2.06	7.54	0.8	0.8
Round2	75	442	558	48.9	20.94	0.89	14.19	8.9
Round3	100	390	610	46.7	20.37	1.77	13.60	7.8
Round4	125	246	754	43.6	12.68	1.54	9.42	5.0
Round5	150	245	755	47.1	13.92	1.91	9.23	5.0
Round6	175	507	493	40.5	14.67	1.64	11.52	9.4
Round7	200	195	805	44.9	12.09	1.80	8.54	4.0
Round8	300	239	761	46.0	11.54	1.73	9.15	4.8
Round9	400	88	912	44.4	11.07	2.34	8.50	1.7
Round10	500	300	700	46.0	13.84	1.98	10.23	5.8

This setup is almost the same as the SUT in benchmark 1 except the ordering service. The ordering service used in the setup is Raft. Raft add more complexity as the ordering service than SOLO which will consume more resources (RAM and CPU) resulted in reducing the network throughput. Since all reside in a single VM calliper is unable to generate the transaction load as mentioned in the benchmark configuration. The maximum send rate this setup has achieved is 48.9 TPS at Round2. Due to the extra overhead created by Raft on the resources, none of the rounds had achieved a 100% success rate.

Benchmark 03.

Maximum number of transactions per round: 1000
System Under Test :



StateDB : GoLevelDB
Network: 2 Organization with 1 peer.
Setup: Multiple Host
Ordering: Solo
Number of Clients: 5

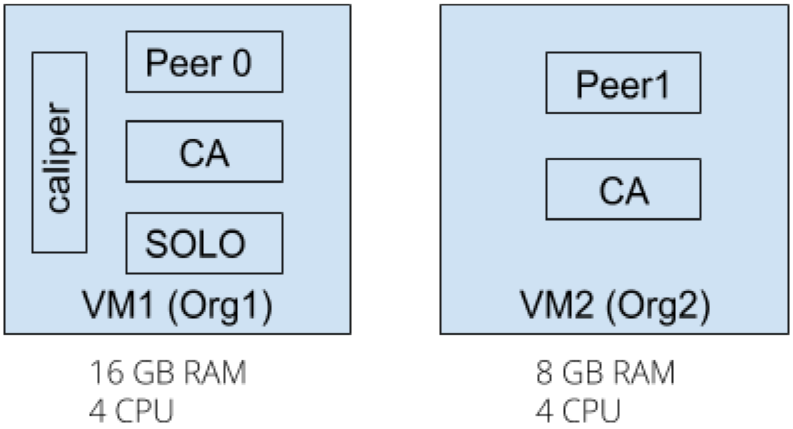
Name	Expected Send Rate (TPS)	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Round1	50	1000	0	50.1	4.10	0.13	1.32	49.3
Round2	75	1000	0	70.4	4.25	0.20	1.83	61.8
Round3	100	1000	0	93.2	12.01	0.22	5.31	54.9
Round4	125	988	12	89.1	16.08	0.47	6.65	29.6
Round5	150	879	121	91.3	14.34	0.57	7.72	27.1

Round6	175	870	130	94.2	14.37	0.39	7.73	26.7
Round7	200	916	84	86.0	14.61	0.71	7.05	27.2
Round8	300	904	96	86.2	14.43	0.60	7.14	26.8
Round9	400	888	112	95.8	15.66	0.62	9.09	27.6
Round10	500	931	69	85.6	14.00	0.55	7.17	28.3

This is the first benchmark performed on a distributed setup as shown in the above diagram. Caliper was able to produce a maximum transaction load of 95.8 TPS (Round9) with this setup. However, there’s a bottleneck in Caliper load generation due to the resource limitations in VM1. And still, there are transaction failures when the send rate increases.

Benchmark 04.

Maximum number of transactions per round: 1000
System Under Test :



StateDB : GoLevelDB
Network: 2 Organization with 1 peer.

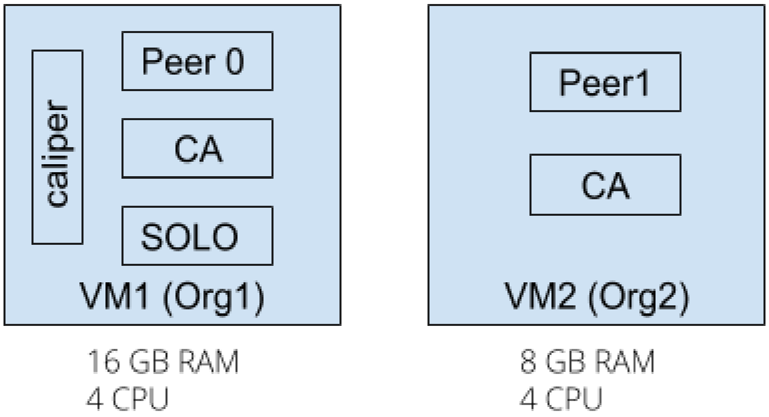
Setup: Multiple Host
Ordering: Solo
Number of Clients: 5

Name	Expected Send Rate (TPS)	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Round1	50	1000	0	50.2	0.24	0.05	0.13	50.0
Round2	75	1000	0	75.3	0.31	0.05	0.12	74.9
Round3	100	1000	0	100.1	0.23	0.06	0.12	99.4
Round4	125	1000	0	125.1	0.44	0.06	0.14	124.0
Round5	150	1000	0	149.7	0.54	0.05	0.26	148.5
Round6	175	1000	0	164.4	1.42	0.17	0.85	149.7
Round7	200	1000	0	178.7	2.00	0.12	1.18	153.1
Round8	300	1000	0	173.6	1.81	0.10	1.06	154.1
Round9	400	1000	0	171.0	2.01	0.17	1.20	154.1
Round10	500	1000	0	177.1	2.29	0.26	1.25	154.3

This benchmark was run on two VMs on Google compute engine. The system configuration of VM2 was similar to the one in benchmark 3. The RAM of VM1 was increased to 16GB since it has Caliper clients as well. In this setup, no failure transactions occurred in the Blockchain network, unlike other systems under test. The problem with this setup is the failure to achieve the send rates (Caliper load generations)defined in the benchmark configurations. For example, in Round8 expected send rate is 300 but the actual send rate is 173.6 and in Round10 the expected send rate is 500 but the actual send rate is 177.1. This could be due to two reasons either Caliper docker containers unable to consume hardware resources properly or just 5 clients can’t produce a load like 300 TPS. No in the benchmark 05 number clients was increased to 15.

Benchmark 05.

Maximum number of transactions per round: 1000
System Under Test :



StateDB : GoLevelDB
Network: 2 Organization with 1 peer.
Setup: Multiple Host
Ordering: Solo
Number of Clients: **15**

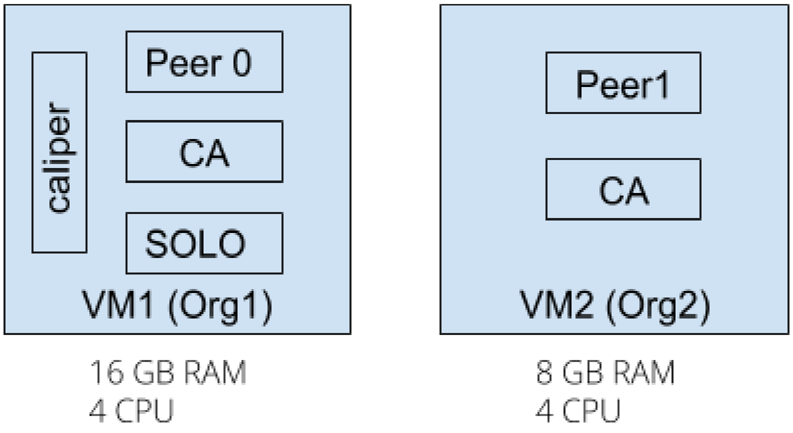
Name	Expected Send Rate (TPS)	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Round1	50	1000	0	43.5	0.63	0.06	0.18	42.9
Round2	75	1000	0	66.3	0.35	0.07	0.17	65.8
Round3	100	1000	0	88.0	1.33	0.07	0.21	86.9
Round4	125	1000	0	110.7	1.71	0.06	0.35	108.6

Round5	150	1000	0	131.4	2.12	0.16	1.04	126.6
Round6	175	1000	0	141.5	1.81	0.13	1.00	130.1
Round7	200	1000	0	131.1	2.45	0.11	1.05	126.6
Round8	300	1000	0	145.9	2.61	0.11	1.59	127.1
Round9	400	1000	0	125.9	3.47	0.33	1.87	112.8
Round10	500	1000	0	159.1	3.14	0.32	2.00	128.9

The number of clients is increased to 15 (In previous it was 5) to make sure the low send rate is not caused by a bottleneck in the number of clients. It was observed that the maximum send rate that it could achieve was 159.1 which is even less than the benchmark 04. This is due to the more resource consumption of the additional clients who reside in the VM1.

Benchmark 06.

Maximum number of transactions per round: Variable (See the tab)
System Under Test :



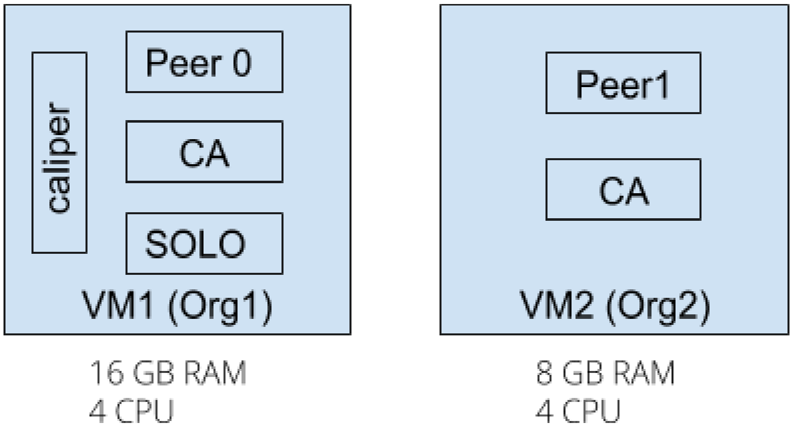
StateDB : GoLevelDB
Network: 2 Organization with 1 peer.
Setup: Multiple Host
Ordering: Solo
Number of Clients: **15**

Name	Expected Send Rate (TPS)	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Round1	**200**	**2000**	0	114.6	2.08	0.23	1.22	112.5
Round2	**300**	**3000**	0	131.0	3.85	0.30	1.78	125.6
Round3	**400**	**4000**	0	134.6	4.41	0.31	2.16	131.4
Round4	**500**	**5000**	0	146.0	6.51	0.13	3.94	135.4

The same setup was tested under different transaction counts in each round (2000, 3000, 4000, 5000). There were no failures in transaction accepted by the Blockchain network. However, the max load generated was 146 TPS due to same reasons as benchmark 05.

Benchmark 07.

Maximum number of transactions per round: 1000
System Under Test :



StateDB : GoLevelDB
Network: 2 Organization with 1 peer.
Setup: Multiple Host
Ordering: Solo

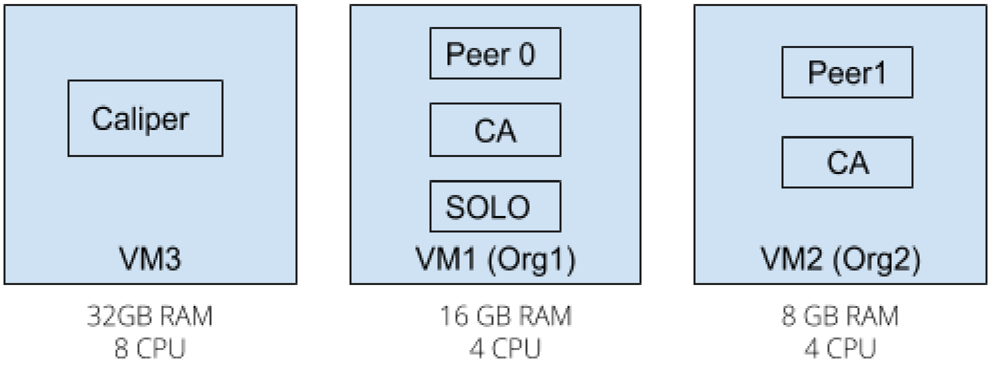
Number of Clients: **1**

Name	Expected Send Rate (TPS)	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Round1	50	1000	0	50.1	0.28	0.05	0.15	49.9
Round2	75	1000	0	75.1	0.32	0.05	0.14	74.7
Round3	100	1000	0	91.4	0.31	0.10	0.17	90.5
Round4	125	1000	0	94.0	0.33	0.07	0.17	93.4
Round5	150	1000	0	98.8	0.29	0.07	0.16	98.2
Round6	175	1000	0	96.5	0.31	0.07	0.17	95.9
Round7	200	1000	0	99.8	0.28	0.07	0.16	99.2
Round8	300	1000	0	97.5	0.30	0.07	0.17	96.8
Round9	400	1000	0	99.0	0.31	0.06	0.16	98.4
Round10	500	1000	0	99.7	0.30	0.07	0.16	99.0

This benchmark was performed to find the maximum load that a single Caliper client can produce. The number of clients in this benchmark is 1. The maximum send rate that a single client can achieve in Caliper was 99.8 TPS(approximately 100 TPS).

Benchmark 08.

Maximum number of transactions per round: 1000
System Under Test :



StateDB : GoLevelDB
Network: 2 Organization with 1 peer.
Setup: Multiple Host
Ordering: Solo
Number of Clients: **5**

Name	Expected Send Rate (TPS)	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Round1	50	1000	0	50.2	0.20	0.04	0.11	50.0
Round2	75	1000	0	75.3	0.14	0.04	0.09	75.0
Round3	100	1000	0	100.4	0.20	0.04	0.08	99.9
Round4	125	1000	0	125.5	0.11	0.04	0.08	124.8
Round5	150	1000	0	150.4	0.17	0.05	0.08	149.3
Round6	175	1000	0	175.6	0.15	0.04	0.08	174.1

Round7	200	1000	0	200.6	0.15	0.04	0.08	198.8
Round8	300	1000	0	300.4	0.64	0.06	0.40	265.8
Round9	400	1000	0	400.2	1.45	0.09	1.09	265.2
Round10	500	1000	0	500.5	1.95	0.09	1.54	258.6

Based on the observations of benchmark 05, 06, 07 these experiments were set up to avoid resource limitation bottleneck on the Caliper load generation. In this setup, Caliper instance is moved to a third VM with 32GM RAM and 8 CPUs. This setup was able to achieve the expected load generations (maximum of 500TPS per second). However, now the bottleneck is with transaction throughput due to the resource limitation in VM where the Blockchain nodes are. The maximum throughput achieved in this series of experiments was 265.8 TPS in Round8 of this benchmark.

Conclusions :

1. Caliper instance needed to be in a separate VM to get accurate benchmark values of a Fabric Blockchain network.
2. The throughput of a Fabric network can be increased by assigning a dedicated VM to each peer in a Blockchain.
3. Adding a CFT ordering service result in reducing the throughput.