



dojo®

Pushing Your Streaming Platform to the Limit



Elad Leev

**#DistributedSystems #DataStream
#Scalability #DataMesh #Kafka #Flink**

 @eladleev

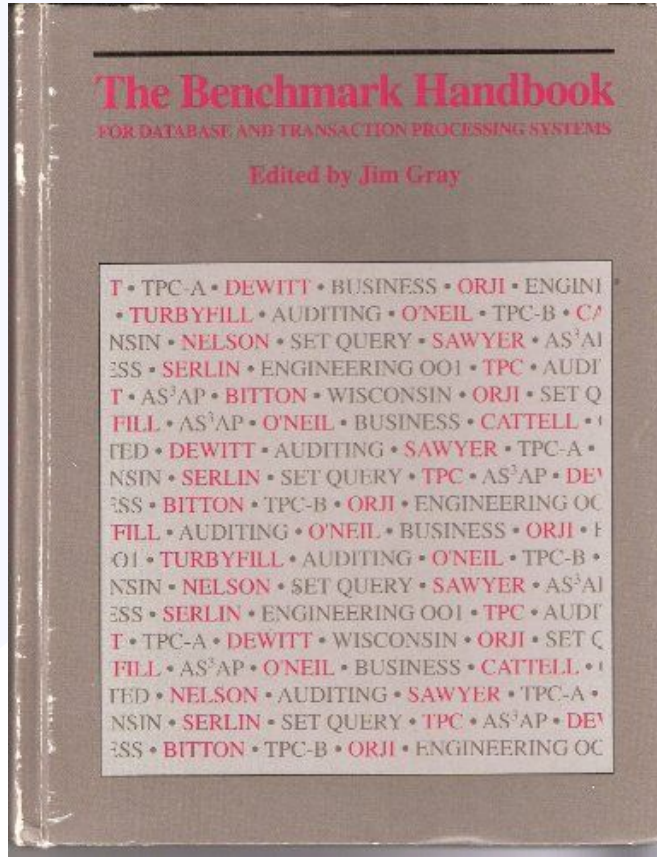
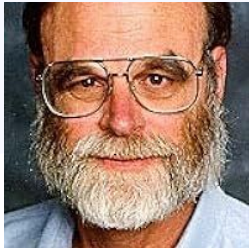
dojo.

www.dojo.tech

One of the largest Fintechs in Europe

Enabling 130,000 businesses to take card
payments from 4 million consumers per day





“

No single metric can measure the performance of computer systems on all applications. System performance **varies enormously** from **one application domain to another**. Each system is typically designed for a few problem domains and may be incapable of performing other tasks.

Jim Gray, 1991

”

“

... The system that does the job with the lowest cost-of-ownership.

Jim Gray, 1991

”

Your vendor's Benchmarks = Marketing job

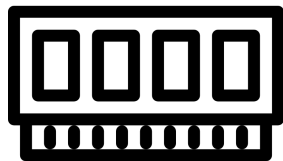


Know your limits

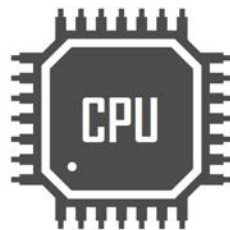




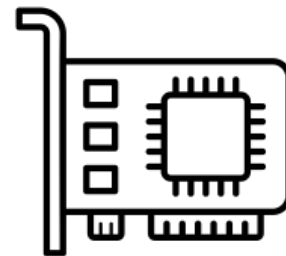
Disk



Memory



CPU



Network

The Key Criteria For a Benchmark

No special tricks.

The benchmark should test the exact same deployment and configuration as we planned to use in Production.



Mock our environment

Aim for the peak.

Do an advanced research. Know your traffic characteristics. Know your SLAs/SLOs. Expect the unexpected.



Mock our environment



Test peak performance

“

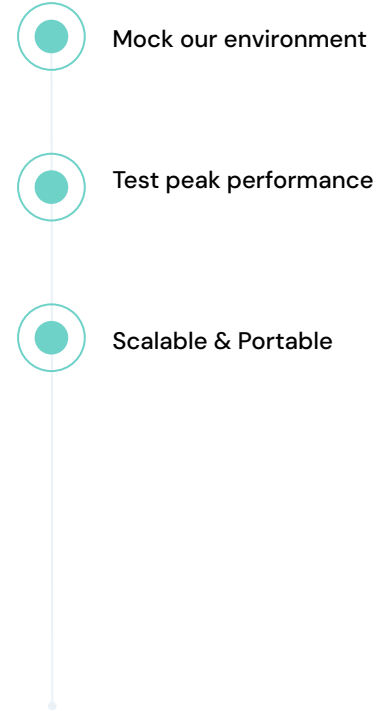
Failures are a given and everything will **eventually fail over time**: from routers to hard disks, from operating systems to memory units corrupting TCP packets, from transient errors to permanent failures. This is a given, whether you are using the highest quality hardware or **lowest cost components**.

Werner Vogels, AWS CTO, 2016

”

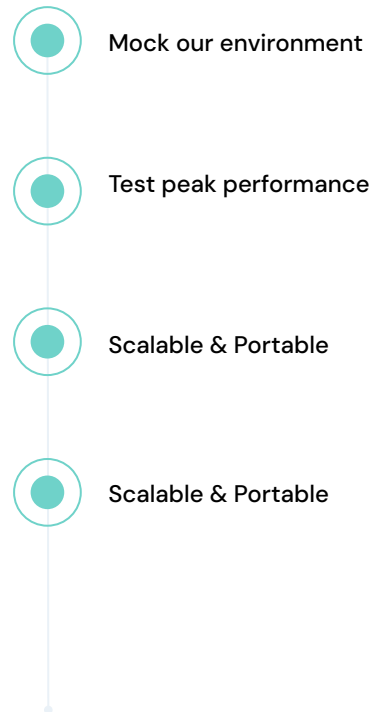
Make it scalable.

Our system is a living organisms - it scale up and down. Embrace the change, know the impact.



Simplicity is key.

The benchmark must be understandable, otherwise it will lack credibility.



What to look for?

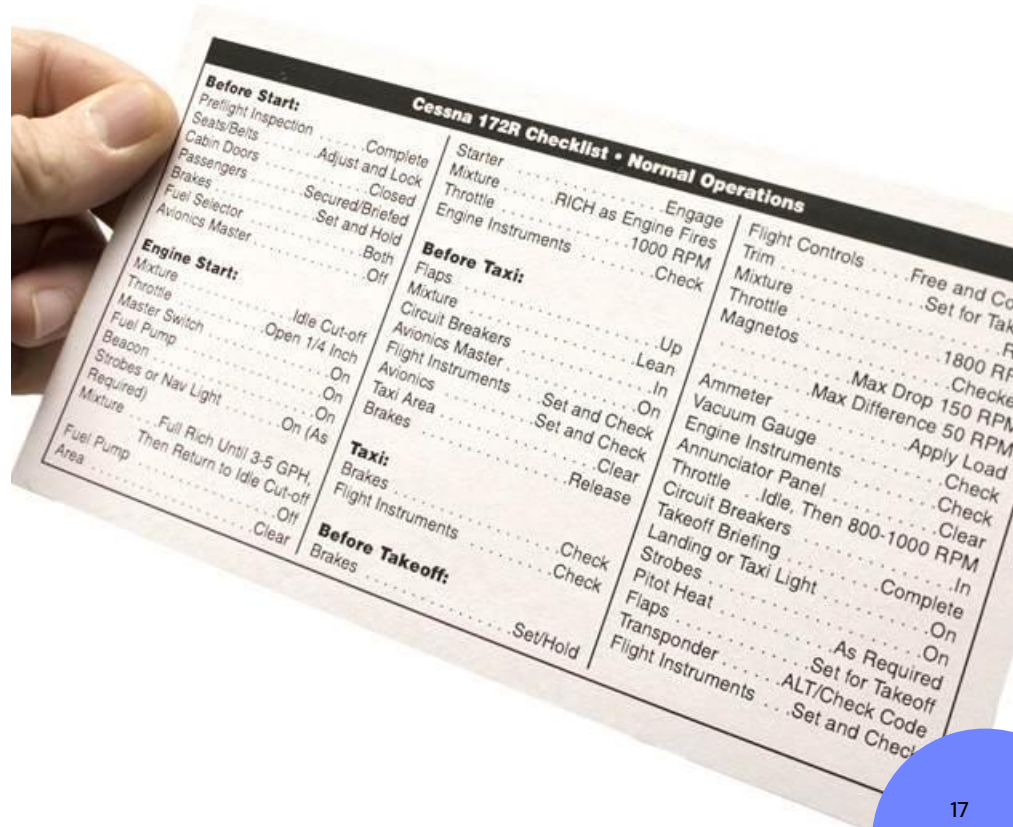
The USE Method

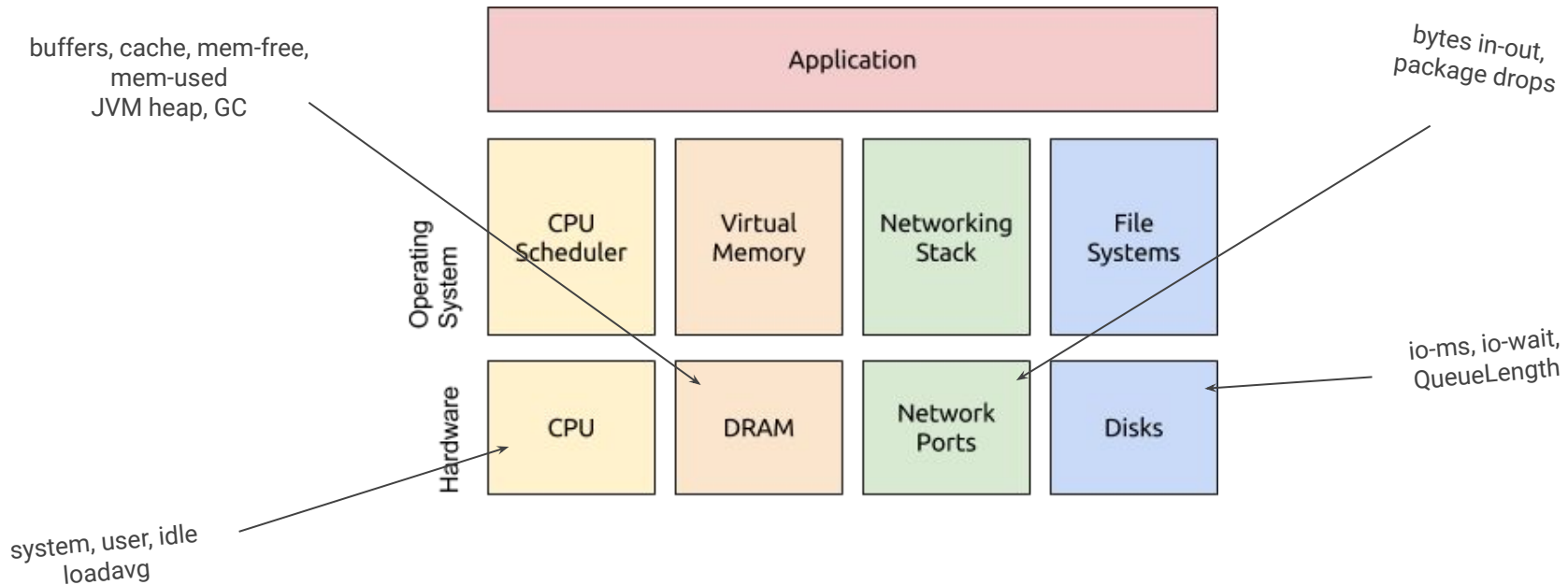
For every resource in the system, we should check:

Utilization

Saturation

Errors

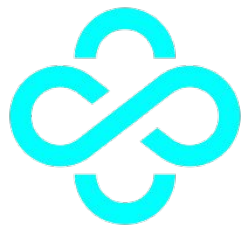




How to Benchmark?

How to Benchmark?





OpenMessaging Initiative

<https://openmessaging.cloud>



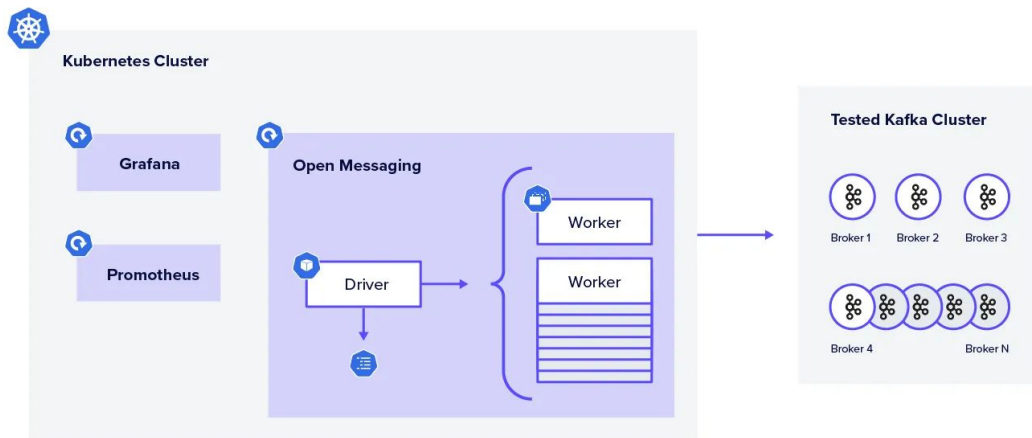
Assign tasks, create topics, orchestrate



The benchmark executor.

How to Benchmark?

Kubernetes-based, easy to scale, runs as a simple producer/consumer, same as our service



Kafka

```
1 ---
2 name: Kafka
3 driverClass: io.openmessaging.benchmark.driver.kafka.KafkaBenchmarkDriver
4
5 # Kafka client-specific configuration
6 replicationFactor: 3
7
8 topicConfig: |
9   min.insync.replicas=2
10
11 commonConfig: |
12   bootstrap.servers=localhost:9092
13   default.api.timeout.ms=1200000
14   request.timeout.ms=1200000
15
16 producerConfig: |
17   acks=all
18   linger.ms=1
19   batch.size=1048576
20
21 consumerConfig: |
22   auto.offset.reset=earliest
23   enable.auto.commit=false
24   max.partition.fetch.bytes=10485760
25
```

Pulsar

```
1 ---
2 name: Pulsar-effectively-once
3 driverClass: io.openmessaging.benchmark.driver.pulsar.PulsarBenchmarkDriver
4
5 # Pulsar client-specific configuration
6 client:
7   serviceUrl: pulsar://localhost:6650
8   httpUrl: http://localhost:8080
9   ioThreads: 4
10  connectionsPerBroker: 8
11  clusterName: local
12  namespacePrefix: benchmark/ns
13  topicType: persistent
14  persistence:
15    ensembleSize: 3
16    writeQuorum: 3
17    ackQuorum: 2
18    deduplicationEnabled: true
19  tlsAllowInsecureConnection: false
20  tlsEnableHostnameVerification: false
21  tlsTrustCertsFilePath:
22  authentication:
23    plugin:
24    data:
25
26 # Producer configuration
27 producer:
28   batchingEnabled: true
29   batchingMaxPublishDelayMs: 1
30   blockIfQueueFull: true
31
32 consumer:
33   receiverQueueSize: 10000
34
```


Kafka

```
1 ---
2 name: Kafka
3 driverClass: io.openmessaging.benchmark.driver.kafka.KafkaBenchmarkDriver
4
5 # Kafka client-specific configuration
6 replicationFactor: 3
7
8 topicConfig: |
9   min.insync.replicas=2
10
11 commonConfig: |
12   bootstrap.servers=localhost:9092
13   default.api.timeout.ms=1200000
14   request.timeout.ms=1200000
15
16 producerConfig: |
17   acks=all
18   linger.ms=1
19   batch.size=1048576
20
21 consumerConfig: |
22   auto.offset.reset=earliest
23   enable.auto.commit=false
24   max.partition.fetch.bytes=10485760
25
```

Pulsar

```
1 ---
2 name: Pulsar-effectively-once
3 driverClass: io.openmessaging.benchmark.driver.pulsar.PulsarBenchmarkDriver
4
5 # Pulsar client-specific configuration
6 client:
7   serviceUrl: pulsar://localhost:6650
8   httpUrl: http://localhost:8080
9   ioThreads: 4
10  connectionsPerBroker: 8
11  clusterName: local
12  namespacePrefix: benchmark/ns
13  topicType: persistent
14  persistence:
15    ensembleSize: 3
16    writeQuorum: 3
17    ackQuorum: 2
18    deduplicationEnabled: true
19  tlsAllowInsecureConnection: false
20  tlsEnableHostnameVerification: false
21  tlsTrustCertsFilePath:
22  authentication:
23    plugin:
24    data:
25
26 # Producer configuration
27 producer:
28   batchingEnabled: true
29   batchingMaxPublishDelayMs: 1
30   blockIfQueueFull: true
31
32 consumer:
33   receiverQueueSize: 10000
34
```

Kafka

```
1 ---
2 name: Kafka
3 driverClass: io.openmessaging.benchmark.driver.kafka.KafkaBenchmarkDriver
4
5 # Kafka client-specific configuration
6 replicationFactor: 3
7
8 topicConfig: |
9   min.insync.replicas=2
10
11 commonConfig: |
12   bootstrap.servers=localhost:9092
13   default.api.timeout.ms=1200000
14   request.timeout.ms=1200000
15
16 producerConfig: |
17   acks=all
18   linger.ms=1
19   batch.size=1048576
20
21 consumerConfig: |
22   auto.offset.reset=earliest
23   enable.auto.commit=false
24   max.partition.fetch.bytes=10485760
25
```

Pulsar

```
1 ---
2 name: Pulsar-effectively-once
3 driverClass: io.openmessaging.benchmark.driver.pulsar.PulsarBenchmarkDriver
4
5 # Pulsar client-specific configuration
6 client:
7   serviceUrl: pulsar://localhost:6650
8   httpUrl: http://localhost:8080
9   ioThreads: 4
10  connectionsPerBroker: 8
11  clusterName: local
12  namespacePrefix: benchmark/ns
13  topicType: persistent
14  persistence:
15    ensembleSize: 3
16    writeQuorum: 3
17    ackQuorum: 2
18    deduplicationEnabled: true
19  tlsAllowInsecureConnection: false
20  tlsEnableHostnameVerification: false
21  tlsTrustCertsFilePath:
22  authentication:
23    plugin:
24    data:
25
26 # Producer configuration
27 producer:
28   batchingEnabled: true
29   batchingMaxPublishDelayMs: 1
30   blockIfQueueFull: true
31
32 consumer:
33   receiverQueueSize: 10000
34
```

Kafka

```
1 ---
2 name: Kafka
3 driverClass: io.openmessaging.benchmark.driver.kafka.KafkaBenchmarkDriver
4
5 # Kafka client-specific configuration
6 replicationFactor: 3
7
8 topicConfig: |
9   min.insync.replicas=2
10
11 commonConfig: |
12   bootstrap.servers=localhost:9092
13   default.api.timeout.ms=1200000
14   request.timeout.ms=1200000
15
16 producerConfig: |
17   acks=all
18   linger.ms=1
19   batch.size=1048576
20
21 consumerConfig: |
22   auto.offset.reset=earliest
23   enable.auto.commit=false
24   max.partition.fetch.bytes=10485760
25
```

Pulsar

```
1 ---
2 name: Pulsar-effectively-once
3 driverClass: io.openmessaging.benchmark.driver.pulsar.PulsarBenchmarkDriver
4
5 # Pulsar client-specific configuration
6 client:
7   serviceUrl: pulsar://localhost:6650
8   httpUrl: http://localhost:8080
9   ioThreads: 4
10  connectionsPerBroker: 8
11  clusterName: local
12  namespacePrefix: benchmark/ns
13  topicType: persistent
14  persistence:
15    ensembleSize: 3
16    writeQuorum: 3
17    ackQuorum: 2
18    deduplicationEnabled: true
19  tlsAllowInsecureConnection: false
20  tlsEnableHostnameVerification: false
21  tlsTrustCertsFilePath:
22  authentication:
23    plugin:
24    data:
25
26 # Producer configuration
27 producer:
28   batchingEnabled: true
29   batchingMaxPublishDelayMs: 1
30   blockIfQueueFull: true
31
32 consumer:
33   receiverQueueSize: 10000
34
```

Kafka

```
1 ---
2 name: Kafka
3 driverClass: io.openmessaging.benchmark.driver.kafka.KafkaBenchmarkDriver
4
5 # Kafka client-specific configuration
6 replicationFactor: 3
7
8 topicConfig: |
9   min.insync.replicas=2
10
11 commonConfig: |
12   bootstrap.servers=localhost:9092
13   default.api.timeout.ms=1200000
14   request.timeout.ms=1200000
15
16 producerConfig: |
17   acks=all
18   linger.ms=1
19   batch.size=1048576
20
21 consumerConfig: |
22   auto.offset.reset=earliest
23   enable.auto.commit=false
24   max.partition.fetch.bytes=10485760
25
```

Pulsar

```
1 ---
2 name: Pulsar-effectively-once
3 driverClass: io.openmessaging.benchmark.driver.pulsar.PulsarBenchmarkDriver
4
5 # Pulsar client-specific configuration
6 client:
7   serviceUrl: pulsar://localhost:6650
8   httpUrl: http://localhost:8080
9   ioThreads: 4
10  connectionsPerBroker: 8
11  clusterName: local
12  namespacePrefix: benchmark/ns
13  topicType: persistent
14  persistence:
15    ensembleSize: 3
16    writeQuorum: 3
17    ackQuorum: 2
18    deduplicationEnabled: true
19  tlsAllowInsecureConnection: false
20  tlsEnableHostnameVerification: false
21  tlsTrustCertsFilePath:
22  authentication:
23    plugin:
24    data:
25
26 # Producer configuration
27 producer:
28   batchingEnabled: true
29   batchingMaxPublishDelayMs: 1
30   blockIfQueueFull: true
31
32 consumer:
33   receiverQueueSize: 10000
34
```

Workload Files

```
1 ---
2 name: 1000k rate 4 producers and 4 consumers on 1 topic / 10000 partition
3
4 topics: 1
5 partitionsPerTopic: 10000
6
7 messageSize: 1024
8 useRandomizedPayloads: true
9 randomBytesRatio: 0.5
10 randomizedPayloadPoolSize: 1000
11
12 subscriptionsPerTopic: 1
13 consumerPerSubscription: 4
14 producersPerTopic: 4
15
16 # Discover max-sustainable rate
17 producerRate: 1000000
18
19 consumerBacklogSizeGB: 0
20 testDurationMinutes: 5
21
```

Workload Files

```
1 ---
2 name: 1000k rate 4 producers and 4 consumers on 1 topic / 10000 partition
3
4 topics: 1
5 partitionsPerTopic: 10000
6
7 messageSize: 1024
8 useRandomizedPayloads: true
9 randomBytesRatio: 0.5
10 randomizedPayloadPoolSize: 1000
11
12 subscriptionsPerTopic: 1
13 consumerPerSubscription: 4
14 producersPerTopic: 4
15
16 # Discover max-sustainable rate
17 producerRate: 1000000
18
19 consumerBacklogSizeGB: 0
20 testDurationMinutes: 5
21
```



Kubernetes Cluster



Grafana



Prometheus



Open Messaging



Driver



Tested Kafka Cluster



Broker 1



Broker 2



Broker 3



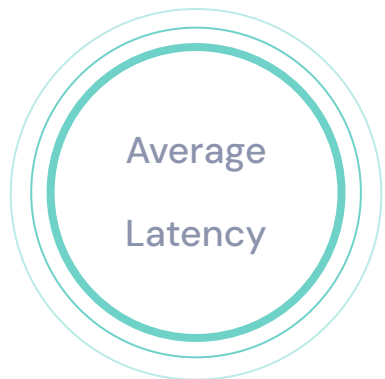
Broker 4



Broker N

Benchmark Insights


Benchmark Insights



Benchmark Insights

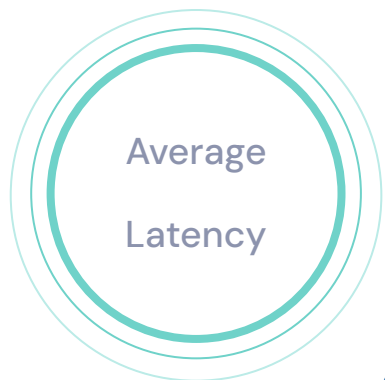


Average
Latency



Test Our
Monitoring &
Dashboards

Benchmark Insights



Benchmark Insights



Benchmark Insights




Average
Latency

A circular graphic with a light teal double-line border. The text "Average Latency" is centered inside.




Potential
Bottlenecks

A circular graphic with a dark grey double-line border. The text "Potential Bottlenecks" is centered inside.



Configuration
Suggestion

A circular graphic with a green double-line border. The text "Configuration Suggestion" is centered inside.



Test Our
Monitoring &
Dashboards

A circular graphic with a dark blue double-line border. The text "Test Our Monitoring & Dashboards" is centered inside.



Scale-up
Ballpark

A circular graphic with a blue double-line border. The text "Scale-up Ballpark" is centered inside.

*Thank you
for your time!*



<https://leevs.dev>



@eladleev



[linkedin.com/in/elad-leev](https://www.linkedin.com/in/elad-leev)