Cloud-Native Scalability Benchmarking with Theodolite

Sören Henning, Benedikt Wetzel, Wilhelm Hasselbring
Scalability Benchmarking of Cloud-Native Applications

- Scalability is a main driver for adopting cloud-native applications and microservices
  (Knoche & Hasselbring 2019; Soldani et al. 2018; Kratzke & Quint 2017)
Scalability Benchmarking of Cloud-Native Applications

• Scalability is a main driver for adopting cloud-native applications and microservices
  (Knoche & Hasselbring 2019; Soldani et al. 2018; Kratzke & Quint 2017)

• Engineers and researchers use benchmarking to evaluate and compare quality of frameworks, configuration, etc.
  (Kounev et al. 2020; Hasselbring 2021)
Scalability Benchmarking of Cloud-Native Applications

- Scalability is a main driver for adopting cloud-native applications and microservices (Knoche & Hasselbring 2019; Soldani et al. 2018; Kratzke & Quint 2017)

- Engineers and researchers use benchmarking to evaluate and compare quality of frameworks, configuration, etc. (Kounev et al. 2020; Hasselbring 2021)

- However, no commonly used method for benchmarking scalability of cloud-native applications (Henning & Hasselbring 2022)
Goal:
Make Scalability Benchmarking more Usable and Reproducible!
The Theodolite Scalability Benchmarking Framework

- Creating benchmarks based on existing Kubernetes resource files
- Defining SLO based on existing Prometheus metrics via PromQL queries
- Kubernetes Operator for declarative definition of benchmarks and executions
In Search of Microservice
Show Case...
TeaStore Architecture
TeaStore Architecture
Desired Outcome
Designing a Scalability Benchmark for the TeaStore

```yaml
apiVersion: theodolite.rocks/v1beta1
kind: benchmark
metadata:
  name: teastore
spec:
  waitForResourcesEnabled: true
  sut:
    resources:
      - configMap:
          name: teastore-deployment
          files:
            - teastore-auth-deployment.yaml
            - teastore-auth-service.yaml
            - teastore-db-deployment.yaml
            - teastore-db-service.yaml
            - teastore-image-deployment.yaml
            - teastore-image-service.yaml
            - teastore-persistence-deployment.yaml
            - teastore-persistence-service.yaml
            - teastore-recommender-deployment.yaml
            - teastore-recommender-service.yaml
            - teastore-registry-deployment.yaml
            - teastore-registry-service.yaml
            - teastore-webui-deployment.yaml
            - teastore-webui-nodeport.yaml
  loadGenerator:
    resources:
      - configMap:
          name: teastore-imeter-deployment
```

12
System under Test (SUT)
TeaStore - benchmark.yaml

```yaml
apiVersion: theodolite.rocks/v1beta1
class: benchmark
metadata:
  name: teastore
spec:
  waitForResourcesEnabled: true
  sut:
    resources:
      - configMap:
          name: teastore-deployment
          files:
            - teastore-auth-deployment.yaml
            - teastore-auth-service.yaml
            - teastore-db-deployment.yaml
            - teastore-db-service.yaml
            - teastore-image-deployment.yaml
            - teastore-image-service.yaml
            - teastore-persistence-deployment.yaml
            - teastore-persistence-service.yaml
            - teastore-recommender-deployment.yaml
            - teastore-recommender-service.yaml
            - teastore-registry-deployment.yaml
            - teastore-registry-service.yaml
            - teastore-webui-deployment.yaml
            - teastore-webui-service.yaml
          loadGenerator:
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
            resources:
              - configMap:
                  name: teastore-jmeter-deployment
                  files:
                    - jmeter.yaml
```

TeaStore - jmeter.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: jmeter
  annotations:
    openservicemesh.io/sidecar-injection: enabled
spec:
  replicas: 1
  template:
    metadata:
      labels:
        teastore: true
    spec:
      containers:
        - name: jmeter
          image: justb4/jmeter:5.4
          command: ["/bin/bash", "-e", "-c", "-"]
          args:
            - >
              java -jar bin/ApacheJMeter.jar \
              -t profile/teastore_browse_nogui.jmx \
              -n -Dhostname teastore-webui -Dport 8080 \
              -JnumUser $NUM_USERS -JrampUp 1 \
              -l mylogfile.log -n
              env:
              - name: NUM_USERS
                value: "20"
              volumeMounts:
```

Load Generator

- JMeter
- Envoy (OSM)
Resource Dimension

(1) Horizontal Scalability
Resource Dimension

(2) Vertical Scalability
Load Dimension

```
loadTypes:
  - typeName: NumUsers
    patchers:
      - type: EnvVarPatch
        resource: jmeter-profile
        properties:
          container: jmeter
          variableName: NUM_USERS
        slo:
          - sloType: generic
            name: uilatency
            prometheusUrl: "http://prometheus-operated:9090"
            offset: 0
            properties:
              externalSloUrl: "http://localhost:8082"
              promQLQuery: "histogram_quantile(0.95, sum(irate(osm_request_duration_ms_bucket(destination_name='teastore_webui') [ms]))) by (le, destination_name)"
            warmup: 600 # in seconds
            queryAggregation: max
            repetitionAggregation: median
            operator: lte
            threshold: 200
apiVersion: apps/v1
kind: Deployment
metadata:
  name: jmeter
  annotations:
    openservicemesh.io/sidecar-injection: enabled
spec:
  replicas: 1
containers:
  - name: jmeter
    image: justb4/jmeter:5.4
    command: ["/bin/bash", "-e", "--"]
    args:
      - >
        java -jar bin/ApacheJMeter.jar \
        -t profile/teastore_browse_nogui.jmx \
        -Dhostname teastore-webui -Jport 8080 \
        -JnumUser $NUM_USERS -JrampUp 1 \
        -l mylog/logfile.log -n
        - env:
            - name: NUM_USERS
              value: "20"
        volumeMounts:
          - name: jmeter-profile
            mountPath: /opt/apache-jmeter-5.4.3/profile
            ports:
              - containerPort: 59999
              - containerPort: 60000
              - containerPort: 1899
    volume:
```
SLO

p95 latency <= 200 ms?
Deployment

```
apiVersion: theodolite.rocks/v1beta1
class: benchmark
metadata:
  name: teststore
spec:
  waitForResourcesEnabled: true
  suite:
    resources:
      - configMap:
          name: teststore-deployment
          files:
            - teststore-auth-deployment.yaml
            - teststore-auth-service.yaml
            - teststore-db-deployment.yaml
            - teststore-db-service.yaml
            - teststore-image-deployment.yaml
            - teststore-image-service.yaml
            - teststore-persistence-deployment.yaml
            - teststore-persistence-service.yaml
            - teststore-recommender-deployment.yaml
            - teststore-recommender-service.yaml
            - teststore-registry-deployment.yaml
            - teststore-registry-service.yaml
            - teststore-webui-deployment.yaml
            - teststore-webui-nodeport.yaml

  loadGenerator:
    resources:
      - configMap:
          name: teststore-jmeter-deployment
          files:
            - jmeter.yaml

  resourceTypes:
    - typeName: Instances
      patchers:
        - type: ReplicaPatcher
          resource: teststore-auth-deployment.yaml
        - type: ReplicaPatcher
          resource: teststore-image-deployment.yaml
        - type: ReplicaPatcher
          resource: teststore-persistence-deployment.yaml
```

```
$ kubectl apply -f benchmark.yaml
benchmark.theodolite.rocks/teststore created
```
apiVersion: theodolite.rocks/v1beta1
kind: benchmark
metadata:
  name: teastore
spec:
  waitForResourcesEnabled: true
  sut:
    resources:
      - configMap:
          name: teastore-deployment
          files:
            - teastore-auth-deployment.yaml
            - teastore-auth-service.yaml
            - teastore-db-deployment.yaml
            - teastore-db-service.yaml
            - teastore-image-deployment.yaml
            - teastore-image-service.yaml
            - teastore-persistence-deployment.yaml
            - teastore-persistence-service.yaml
            - teastore-recommender-deployment.yaml
            - teastore-recommender-service.yaml
            - teastore-registry-deployment.yaml
            - teastore-registry-service.yaml
            - teastore-webui-deployment.yaml
            - teastore-webui-nodeport.yaml
  loadGenerator:
    resources:
      - configMap:
          name: teastore-jmeter-deployment
          files:
            - jmeter.yaml
  resourceTypes:
    - resourceName: Instances
      patchers:
        - type: ReplicaPatcher
          resource: teastore-auth-deployment.yaml
        - type: ReplicaPatcher
          resource: teastore-image-deployment.yaml
        - type: ReplicaPatcher
          resource: teastore-persistence-deployment.yaml

$ kubectl apply -f benchmark.yaml
benchmark.theodolite.rocks/teastore created

$ kubectl get benchmarks
NAME     AGE     STATUS
teastore  25s     Ready
Running the Benchmark

```
apiVersion: theodolite.rocks/v1beta1
kind: execution
metadata:
  name: teastore-horizontal
spec:
  benchmark: teastore
  load:
    loadType: NumUsers
    loadValues: [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
  resources:
    resourceType: Instances
    resourceName: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
  slos:
    - name: uiLatency
      properties:
        warmup: 600 # in seconds
        threshold: 200
  execution:
    strategy:
      name: RestrictionSearch
  restrictions:
    - LowerBound
    searchStrategy: LinearSearch
    duration: 1200 # in seconds
    repetitions: 1
  configOverrides: []
```

```
$ kubectl apply -f execution-horizontal.yaml
execution.theodolite.rocks/teastore-horizontal created
```
Running the Benchmark

```
apiVersion: theodolite.rocks/v1beta1
kind: execution
metadata:
  name: teastore-horizontal
spec:
  benchmark: teastore
  load:
    loadType: NumUsers
    loadValues: [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
  resources:
    resourceType: Instances
    resourceValues: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
  slos:
    - name: uILatency
      properties:
        warmup: 600 # in seconds
        threshold: 200
  execution:
    strategy:
      name: RestrictionSearch
      restrictions:
        - LowerBound
      searchStrategy: LinearSearch
      duration: 1200 # in seconds
      repetitions: 1
  configOverrides: []
```
Running the Benchmark

TeaStore - execution-horizontal.yaml

```yaml
apiVersion: theodolite.rocks/v1beta1
class: execution
metadata:
  name: teastore-horizontal
spec:
  benchmark: teastore
  load:
    loadType: NumUsers
    loadValues: [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
  resources:
    resourceType: Instances
    resourceValues: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
  slos:
    - name: uiLatency
      properties:
      - warmup: 600 # in seconds
      - threshold: 200
  execution:
    strategy:
      name: RestrictionSearch
      restrictions:
        - LowerBound
      searchStrategy: LinearSearch
    duration: 1200 # in seconds
    repetitions: 1
    configOverrides: []
```

TeaStore - bash

```bash
$ kubectl apply -f execution-horizontal.yaml
execution.teodolite.rocks/teastore-horizontal created

$ kubectl get executions
NAME   STATUS    DURATION    AGE
teastore-horizontal Pending 3s

$ kubectl get executions
NAME        STATUS    DURATION    AGE
teastore-horizontal Running 9s 13s
```
Running the Benchmark

TeaStore - execution-horizontal.yaml

```yaml
apiVersion: theodolite.rocks/v1beta1
kind: execution
metadata:
  name: teastore-horizontal
spec:
  benchmark: teastore
  load:
    loadType: NumUsers
    loadValues: [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
  resources:
    resourceType: Instances
    resourceValues: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
  slos:
    - name: uiLatency
      properties:
        warmup: 600 # in seconds
        threshold: 200
      execution:
        strategy:
          name: RestrictionSearch
          restrictions:
            - LowerBound
          searchStrategy: LinearSearch
          duration: 1200 # in seconds
          repetitions: 1
      configOverrides: []
```

TeaStore - bash

```bash
$ kubectl apply -f execution-horizontal.yaml
execution.teodolite.rocks/teastore-horizontal created

$ kubectl get executions
NAME   STATUS     DURATION   AGE
teastore-horizontal  Pending  3s

$ kubectl get executions
NAME   STATUS     DURATION   AGE
teastore-horizontal  Running 9s  13s
```
So, what’s going on inside?
So, what’s going on inside?

Load: 5 Users
Resources: 1 Pod
So, what’s going on inside?
So, what’s going on inside?

Load: 5 Users
Resources: 1 Pod
So, what’s going on inside?
So, what’s going on inside?

Load: 5 Users
Resources: 1 Pod
So, what’s going on inside?

- **Benchmark Results**
- **Theodolite**
  - queries
- **Prometheus**
- **Grafana**

**Load:** 5 Users

**Resources:** 1 Pod

p95 latency = 298 ms
So, what’s going on inside?

Load: 5 Users
Resources: 2 Pods
So, what’s going on inside?
So, what’s going on inside?
So, what’s going on inside?

Load: 5 Users
Resources: 2 Pods

p95 latency = 195 ms
So, what’s going on inside?
...and the result is:
Benchmarking Vertical Scalability

TeaStore - execution-vertical.yaml

apiVersion: theodolite.rocks/v1beta1
class: execution
metadata:
  name: teastore-vertical
spec:
  benchmark: teastore
  load:
    loadType: NumUsers
    loadValues: [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
  resources:
    resourceType: PodResources
    resourceValues: [500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500, 8000, 8500, 9000, 9500, 10000]
  slos:
    name: uilatency
    properties:
      warmup: 600 # in seconds
      threshold: 200
    execution:
      strategy:
        name: RestrictionSearch
        restrictions:
          - LowerBound
        searchStrategy: LinearSearch
        duration: 1200 # in seconds
        repetitions: 1
        configOverrides: [{}]

mCPU vs number of concurrent users
Conclusions & Lessons Learned

• Favor open workload over closed workload models
• Finding good SLOs is hard
• Cloud-native-ness of the TeaStore could be improved
• Interested? Theodolite’s TeaStore benchmark is now ready to use!