Benchmark Performance with Camunda Platform’s Zeebe Engine

February 2023
Agenda

1. Scalable cloud-native architecture
   What makes Zeebe fast?

2. Deep dive into performance benchmarking
   Tools for load testing Zeebe

3. Best practices
   How to perform benchmarking and optimize performance
Introductions

Olga Inozemtceva
Senior Product Marketing Manager
Camunda

Falko Menge
Senior Principal Solution Architect
Camunda
How Projects Really Work (version 1.0)

How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
How the business consultant described it

How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed
Camunda Platform: The Universal Process Orchestrator

**Design**
Developers & business users collaborate to design & deploy processes with Camunda.

- **Modeler**
  Model & deploy business process diagrams with BPMN & DMN.
  Available via web and desktop app.

- **Connectors**
  Out-of-the-box integrations to easily communicate with popular enterprise applications & protocols.

- **Integration Framework**
  Build & provision your own connectors to any system including home-grown & legacy applications.

- **Forms**
  Drag & drop creation of forms that power workflows that require human interaction.

**Automate**
Enterprise-grade automation platform. Built for today’s business complexity, loved by developers.

- **Workflow Engine**
  Next-generation, cloud-native BPMN workflow engine that unlocks unparalleled speed, scale & resilience.

- **Decision Engine**
  Automate decisions in end-to-end business processes via DMN.

- **Tasklist**
  Assign and execute tasks that require human interaction via easy-to-use Forms or via your own apps with the Tasklist API.

**Operate**
Real-time visibility to monitor, analyze and resolve problems with any process instance.

**Improve**
Teams have the insights to fix the highest ROI problems for their business processes.

- **Optimize**
  Get the insights you need to understand and continuously improve your business processes.

**Connectors**
Integration Framework
Forms
Out-of-the-box integrations to easily communicate with popular enterprise applications & protocols.

**Modeler**
Model & deploy business process diagrams with BPMN & DMN.
Available via web and desktop app.

**POWERED BY ZEEBE**
Automate
Enterprise-grade automation platform. Built for today’s business complexity, loved by developers.

**Tasklist**
Assign and execute tasks that require human interaction via easy-to-use Forms or via your own apps with the Tasklist API.

**Operate**
Real-time visibility to monitor, analyze and resolve problems with any process instance.
Scalable Cloud-Native Architecture

- **Tasklist**
- **Operate**
- **Streaming Exporter**
- **Event Bus / Messaging System**
- **Connector (Http, ...)**
- **Workflow Clients**
- **Microservices / Apps**
- **Ingress/Controller**
- **Gateway**
- **Zeebe Cluster**
- **Broker**
- **Automation Apps**
- **Operate**
- **Tasklist**
- **Optimize**
- **Scalable Cloud-Native Architecture**
Command Query Responsibility Segregation (CQRS)

Write-optimized

Zeebe Cluster

Broker

Broker

Broker

Streaming Exporter

gRPC Gateway

Read-optimized

Automation Apps

Operate

Tasklist

Optimize

elasticsearch

Workflow Clients

Microservices / Apps

Event Bus / Messaging System

Connector (Http, ...)

Event Bus / Messaging System

Connector (Http, ...)

Workflow Clients
Partitions (Shards) and Replication using Raft

Example:

- 5 Brokers
- 5 Partitions
- Replication factor 3

- \( L = \text{Leader} \)
- \( F = \text{Follower} \)
**Process Execution interpreted as Stream Processing**

1. **Send & Append Command**
2. **Replicate & Commit Command**
3. **Validate & Process Command**
4. **Apply to State & Write Event**
5. **Send Response**
Dual-region active-passive

Region 1

Zone 1
- Gateway 0
- Broker 0

Zone 2
- Broker 1

Region 2

Zone 3
- Broker 2

replication factor 3 => quorum 2 => commits stay local
replication factor 4 => quorum 3 => commits must go cross-region
Key Process Performance Metrics

- **Throughput**
  - Number of process instances completed per second (PI/s)

- **Process size**
  - Number of tasks in the BPMN process model (tasks/PI)
  - Tasks dominate performance; gateways & events almost negligible

- **Process latency (cycle time/process instance duration)**
  - Time to execute process instance from start to end (ms)

- **Inter-region network latency**
  - Traveling time of network packets between geographically distant regions (ms)
Key Process Performance Metrics

- **Throughput**
  - Number of process instances completed per second (PI/s)

- **Process size**
  - Number of tasks in the BPMN process model (tasks/PI)

- **Process latency (cycle time/process instance duration)**
  - Time to execute process instance from start to end (ms)

- **Inter-region network latency**
  - Traveling time of network packets between geographically distant regions (ms)
How many process instances per second (PI/s) are you running in production?

Start presenting to display the poll results on this slide.
## Workload Characteristics of Customers

<table>
<thead>
<tr>
<th>Throughput (PI/s)</th>
<th>Process size (#tasks)</th>
<th>Latency (ms)</th>
<th>Multi-Region Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000</td>
<td>8 tasks</td>
<td>500 ms</td>
<td>active-passive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-west 60ms</td>
</tr>
<tr>
<td>500</td>
<td>3 tasks + 2 messages + 2 call activities</td>
<td>1,000 ms</td>
<td>active-active</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10ms avg / 35ms max</td>
</tr>
<tr>
<td>2,400</td>
<td>10 tasks</td>
<td>1,200 ms</td>
<td>active-passive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>52ms one way</td>
</tr>
<tr>
<td>1,700</td>
<td>10 tasks</td>
<td>120,000 ms</td>
<td>active-active-passive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2x east coast + 1x central</td>
</tr>
<tr>
<td>800</td>
<td>8 tasks</td>
<td>200 ms</td>
<td>active-passive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>62ms</td>
</tr>
<tr>
<td>3,000</td>
<td>3 tasks</td>
<td>300 ms</td>
<td>single-region</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>replication factor = 1</td>
</tr>
</tbody>
</table>
Zeebe Grafana Dashboard

docs.camunda.io/docs/next/self-managed/zeebe-deployment/operations/metrics/
Load Generator: Camunda 8 Benchmark

github.com/camunda-community-hub/camunda-8-benchmark

- Java-based load generator for Zeebe
- Simulates the gRPC workload of clients
- Starts thousands of process instances at fixed/increasing rate
  - Overcomes Java scheduler limitations
- Completes tens of thousands of jobs
  - Configurable delay & payload
  - Implemented as asynchronous/reactive as possible, i.e. no blocking of threads
Benchmark Setup – Don’t try this at home ;-)
# Zeebe Benchmark Template

<table>
<thead>
<tr>
<th>Test Case Name</th>
<th>Load Generator/Start</th>
<th>Process Model</th>
<th>Process Replicas</th>
<th>Total Duration (s)</th>
<th>Start Throughput (Pps)</th>
<th>Start Pi Increase Factor</th>
<th>Message TTL (ms)</th>
<th>engine</th>
<th>Engine Version</th>
<th>Machine Type</th>
<th>Machine Size (nodes)</th>
<th>CPU (GHz)</th>
<th>RAM (Gigabytes)</th>
<th>DB Thread Pool Size/Pool Size</th>
<th>Jvm</th>
<th>RocksDB</th>
<th>Gateway</th>
<th>Job Woes</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>1</td>
<td>1</td>
<td>18000</td>
<td>0.60</td>
<td>1</td>
<td>80</td>
<td>none</td>
<td>camunda/zeebe</td>
<td>latest</td>
<td>n1 - 5</td>
<td>4 elastic</td>
<td>2 2 2 1 1 126</td>
<td>disk 128/ssd-ex4</td>
<td>NO 2</td>
<td>2</td>
<td>25 64 128</td>
<td>3</td>
<td>0.08</td>
<td>0.32</td>
</tr>
<tr>
<td>Min tested</td>
<td>1</td>
<td>1</td>
<td>18000</td>
<td>0.60</td>
<td>1</td>
<td>80</td>
<td>none</td>
<td>camunda/zeebe</td>
<td>latest</td>
<td>n2 - 14</td>
<td>32 metrics</td>
<td>14 25 124 36 45 512</td>
<td>disk 500</td>
<td>NO 2</td>
<td>2</td>
<td>25 64 128</td>
<td>2</td>
<td>0.08</td>
<td>0.32</td>
</tr>
<tr>
<td>Max tested</td>
<td>100</td>
<td>18000</td>
<td>36000</td>
<td>0.60</td>
<td>1</td>
<td>80</td>
<td>none</td>
<td>camunda/zeebe</td>
<td>latest</td>
<td>n2 - 14</td>
<td>32 metrics</td>
<td>14 25 124 36 45 512</td>
<td>disk 500</td>
<td>NO 2</td>
<td>2</td>
<td>25 64 128</td>
<td>3</td>
<td>0.42</td>
<td>0.60</td>
</tr>
</tbody>
</table>

**Notes:**
- singleProc = 1
- 2010 = 3600
- 30 = 3600
- 60 = 3600

**Performance Metrics:**
- Jvm Max/MaxPercentage
- rocksDB Max/MaxPercentage
- Gateway Max/MaxPercentage
- Job Woes
- Performance

**Key Values:**
- 0.08
- 0.32
- 0.5
- 0.1
- 0.02
- 21
- 112000

**Additional Details:**
- Performance improvements observed with increased CPU and RAM configurations.
Zeebe Benchmark Result Template

- Precisely documents
  - Configuration parameters
  - Test results, e.g.
    - Throughput
    - Duration
- Graphical diff between configurations
- Heatmap of best results
- Planning new benchmark runs
- Driving k8s automation to iterate over planned configurations

Zeebe Tuner (parameterized Kubernetes tests)
Iterative Benchmark Setup with Zeebe Tuner

- Load Generator
- Gateway
- Zeebe Cluster
- Broker
- Prometheus Metrics
- Grafana
- Kubernetes Cluster
- Google Sheets
- Chaos Mesh
- Camunda 8 Benchmark
- Camunda Platform Helm Chart
- Camunda 8 Helm Profiles
- Zeebe Tuner
- Helm

github.com/camunda-community-hub/camunda-8-helm-profiles
Zeebe Tuner (parameterized Kubernetes tests)

- Zeebe Tuner project (Spring Boot)
  - Programmatically reads Benchmark Template Spreadsheet
  - Creates directory + scripts to run each test
  - Tests can be shared and re-run
  - One Bash script to run multiple tests in sequence
  - Saves url to easily view results
  - Able to run tests unattended
  - Results can be viewed as Grafana Chart and analyzed

[github.com/camunda-consulting/zeebe-tuner](https://github.com/camunda-consulting/zeebe-tuner)
Throughput (PI/s)

Target: 500 PI/s
Message Throughput & Backpressure

Target: 1500 msg/s
Process Instance Duration (Latency)

Target 99% < 1s
Test Strategies

- Exploratory tests: starting from a baseline change one parameter at a time to find new directions

- Navigating the terrain: iterate through various values within a parameter’s value range to find local optimum, then iterate over other parameters to find global optimum
Optimize Performance First, Hardware Cost Second

- First test with “unlimited” hardware, e.g. reserve more CPUs and memory than the brokers could possibly use
  - That reduces the number benchmark parameters to iterate over
  - Find optimal number of partitions per broker and other parameters
- Then measure CPU and memory consumption and reduce hardware limits to optimize costs
- Also long-running tests to check stability should be done later
Performance engineering is a process

- A change in the code may invalidate prior optimization results, e.g.
  - Number of workers
  - Number of job types

- Parameters are interrelated, i.e. changing one requires changing others, e.g.
  - Number of partitions & brokers
  - vCPUs & thread pool sizes

- That's why optimization is an ongoing process
Current Tuning Best Practices

- Always enforce leader balancing
- Scale partitions & brokers
- Latest generation CPUs (Arm)
- Fastest possible disks & file systems, e.g. XFS
- Disable RocksDB WAL
- Disable explicit Raft flush (takes disk out of critical path)
- Experimental
  - Job exporter
  - Prefer local brokers by selecting correlation key
Resources

Bernd Ruecker’s Blog Articles
- How to Benchmark Your Camunda 8 Cluster
- How to Achieve Geo-redundancy with Zeebe

GitHub
- camunda-consulting/zeebe-tuner
- camunda-community-hub/camunda-8-benchmark

Camunda Platform 8 Docs
- Metrics
- Deployment options

Contact Us
- Contact Form
-mailto:info@camunda.com
- Try Camunda Platform 8 for free
The [hybrid] conference for Camunda Developers

May 10-11

The Camunda Community Summit will be a unique, interactive two-day event exclusively for developers, enterprise architects, and process automation experts.

Featuring deep-dive technical topics, live coding demonstrations, and the latest advances in process automation, our annual summit will be held in an immersive, hybrid and collaborative format.

THANK YOU

twitter.com/camunda

linkedin.com/company/camunda

camunda.com