# CAMUNDA CON 2024

## The State of Performance

Falko Menge

## @falko\_menge



15 years at Camunda

- #team-pre-sales-emea-apac
- Proving to prospects that Camunda is the solution
- Representing Camunda in Open Standards, e.g.
   BPMN 2.0, DMN 1.6, ...

Falko Menge Senior Principal Solution Architect Open Standards Ambassador github.com/falko

twitter.com/falko\_menge

in <u>linkedin.com/in/falko-menge</u>



### Stateful, long-running process orchestration <sup>C</sup>



## **High-Performance Use Cases**

- Instant payments (Echtzeitüberweisungen)
- Stock trade matching & settlement
- End-of-day asset balance
- Merchant payment batch clearing
- Insurance compliance checks
- Pre-paid mobile order rallies after ad campaigns
- End-of-month/year bulk orders of network equipment





## **Typical Questions from Customers**

Can you handle X million transactions per day?
 => Default answer: Yes, it's horizontally scalable. Let's talk!

- Can you prove it?
  - => Existing benchmark data?
  - => Performance tests in a Proof of Concept (PoC) workshop
- How much hardware do I need?
   => Sizing based on performance tests

### **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)

### Workload Characteristics of Customers

Throughput (PI/s)	Process size (#tasks)	99% Latency (ms)	Multi-Region Setup
10,000	8 tasks	500 ms	active-passive east-west 60ms
500	3 tasks + 2 messages + 2 call activities	1,000 ms	active-active 10ms avg / 35ms max
2,400	10 tasks	1,200 ms	active-passive 52ms one way
1,700	10 tasks	120,000 ms	active-active-passive 2x east coast + 1x central
800	8 tasks	<b>200</b> ms	active-passive 62ms
3,000	3 tasks	300 ms	single-region replication factor = 1



C

### **Command Query Responsibility Segregation (CQRS)**



C

### **Process Execution interpreted as Stream Processing**



### **Partitions (Shards) and Replication using Raft**



### **Dual-region active-passive**





replication factor 3 => quorum 2 => commits stay local

### **Dual-region active-active**





replication factor 4 => quorum 3 => commits must go cross-region

### Benchmarking a Process Engine

C

### Benchmark Setup – Don't try this at home\*



\*but on a proper server environment, i.e. neither your laptop nor a SaaS trial cluster

helm.camunda.io

#### 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 => Follows our best practices for <u>writing good workers</u>





#### Zeebe Benchmark Template

		Load Generator/Starter engine						ie																JVM	Rocks	DB	ateway	1	Job	Wor F	Performance								
Test Case Name	Timestamp (CET)	Process Model	Starter Replicas Load Generator Threads	Ramp up time (s)	Run Duration (s)	Start Throughput (PI/s)	Kate Adjustment Strategy	Start PI Increase Factor Message TTL (min)	·····	mage	Engine Version	Machine Type	Cluster Size (nodes) vCPUs (Hyperthreads/node)	RAM (GiB/node) Exporters	CPU Thread Pool Size/Node	Partitions	Replication Factor Log Segment Size (MB)	pre Allocate Segment Files	Disable Explicit Raft Flush	Disk Type	Disk Size	File System (Storage Class)	Backpressure Inter-Region Latency (ms)	Network Compression	max Appends Fer Follower max Append Batch Size (KiB)	JVM MaxRAMPercentage	Write Buffer Size (MiB)	Buffer to Maintain (MiB)	Keplicas (nodes) vCPUs (Hyperthreads)	RAM (GiB) Number of Threads	Inter-Region Latency (ms) Client for Job Worker	Job duration (ms)	Throughput (finished kPI/s)	Throughput with 99% < 1s Throughput (kTasks/s)	Throughput (kFNI/s)	Process Duration 99% (s)	Process Duration 50% (s)	Process Duration avg (s)	Standard Deviation (s) Flow Node Duration (ms) Test Duration (min)
		Default	1			200	0.	.4 60	) camur	nda/zeebe	lates	t n1-	1 5	4 elastic	2 2	1	1 1 128	3 🗸		ssd	128 s	ssd-ext4	$\checkmark$	NO	2 32	25	64	128	3		java	a <u>50</u>							
		Min tested	1	10	1800	18	0.	.0 60	j			0	3 2	4	2 2	3	3 3 16	5 🖌			14					25	64	128	2			0 (	9.08		0.3	2		0.1	0 0 21
_		Max tested	100	180	3600	500	0.	.1 60	camur	nda/zeebe		0 n2-	24 14	32 metrics	14 25	36	6 4 512	$2 \checkmark$		disk	500				_	25	64	128	3		job	ex 50 (	).42		0.6	)		112000.	0 0 30
4		singleProce	1 20	10 :	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metric	s 4 4	24	4 4 128	3 🔽		disk	100 s	ssd-ext4	$\checkmark$	NO	8 32	25	64	128	2 3	3 3	0 job	e: 0 (	<mark>).08</mark> 0	.08	0.3	2 0.5		0.3	30
5	"2022-10-27	singleProces	1 20	10 :	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metrics	4 4	32	2 4 128	3 🗸		disk	200 s	ssd-ext4	$\checkmark$	NO	8 32	25	64	128	2 3	3 3	0 job	ex 0 (	<b>J.08</b> 0	.08	0.3	2 0.5		0.4	30
<u>6</u>	"2022-10-27	singleProces	1 20	10 :	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metrics	4 4	24	4 4 128	3 🗹		disk	200 s	ssd-ext4	$\checkmark$	NO	8 32	25	64	128	2 3	3 3	0 job	ex 0 (	<b>J.25</b> 0	.00		>10	1	112000.	30
7		singleProces	1 20	10 3	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metrics	4 4	24	4 4 128	3 🖌		disk	100 s	ssd-ext4	35	NO	8 32	25	64	128	2 3	3 3	0 job	ex O							
8		singleProces	1 20	10	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metrics	4 4	18	8 4 128	3 🗸		disk	100 s	ssd-ext4	≤ 35	NO	8 32	25	64	128	2 3	3 3	0 job	ex 0 (	).08 O	.08		0.5		0.3	30
9	11/2 9:28	singleProces	1 20	10 :	3600	160 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metrics	4 4	24	4 4 128	3 🗹		disk	100 s	ssd-ext4	≤ 35	NO	8 32	25	64	128	2 3	3 3	0 job	ex 0 (	).16 0	.16		0.8		0.5	30
0	11/2 10:37	singleProces	1 20	10 3	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 6	16 metrics	6 6	24	4 4 128	3 🗹		disk	100 s	ssd-ext4	≤ 35	NO	8 32	25	64	128	2 3	3 3	0 job	ex 0 (	0.30			2.5		0.7	30
1	11/2 11:32	singleProces	1 20	10 3	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 8	16 metrics	8 8	24	4 4 128	3 🗹		disk	100 s	ssd-ext4	≤ 35	NO	8 32	25	64	128	2 3	3 3	0 job	ex 0 (	0.30			2.0			30
2	11/2 12:47	singleProces	1 20	10 :	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 8	16 metrics	8 8	32	2 4 128	3 🗹	Ц	disk	100 s	ssd-ext4	≤ 35	NO	8 32	25	64	128	2 3	3 3	0 job	ex 0 (	0.30			2.5		0.7	30
3	11/2 13:28	singleProces	1 20	10 3	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 12	16 metrics	12 12	24	4 4 128	3	Ц	disk	100 s	ssd-ext4	≤ 35	NO	2 32	25	64	128	2 3	3 3	0 job	ex 0 (	0.30			1.1		0.6	30
.4	11/2 14:38	singleProces	1 20	10 :	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 14	16 metrics	14 14	24	4 4 128	3	Ц	disk	100 s	ssd-ext4	≤ 35	NO	2 32	25	64	128	2 3	3 3	0 job	ex 0 (	1.30 0	.30		0.8		0.5	30
.5	11/3 16:02	singleProces	1 20	10 :	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metrics	4 4	24	4 4 128	3 🗹	Ц	disk	100 s	ssd-ext4	✓ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.08			5.0		2.0	30
6	11/3 15:25	singleProces	1 20	10 :	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metrics	s 4 4	24	4 4 128	3 🗹	Ц	disk	100 s	ssd-ext4	✓ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	0.08			5.0		3.4	30
.7	11/4 11:04	singleProces	1 20	10 :	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metrics	s 4 4	24	4 4 128	3		disk	100 s	ssd-ext4	≤ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.08			5.0		3.4	30
.8	11/4 9:20	singleProces	1 20	10	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 4	16 metrics	5 4 4	24	4 4 128			disk	100 s	ssd-ext4	✓ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.08			5.0		3.3	30
9	11/4 13:09	singleProces	1 20	10 3	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 7	16 metrics	5 7 7	24	4 4 128	3		disk	100 s	ssd-ext4	✓ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.08			5.0		3.5	30
20	11/4 13:37	singleProces	1 20	10 :	3600	80 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 7	16 metrics	5 7 7	24	4 4 128	3		disk	100 s	ssd-ext4	✓ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.08			5.0		3.1	30
21	11/4 14:41	singleProces	1 20	10 3	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 7	16 metrics	5 7 7	24	4 4 128	3		disk	100 s	ssd-ext4	≤ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.13			>10			30
22	11/4 13:56	singleProces	1 20	10 :	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	8 7	16 metrics	5 7 7	24	4 4 128	3		disk	100 s	ssd-ext4	✓ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.14			9.8		6.5	30
23	11/4 15:15	singleProces	1 20	10 3	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	12 7	16 metrics	5 7 7	24	4 4 128			disk	100 s	ssd-ext4	≤ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.14			>10		91000	30
24	11/4 15:43	singleProces	1 20	10 :	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	12 7	16 metrics	5 7 7	24	4 4 128	3 🗠		disk	100 s	ssd-ext4	35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.15			9.7		5.9	30
	11/4 16:13								camur			n1-	24 7									ssd-ext4	<ul> <li>✓ 35</li> </ul>			25		128			35 job	ex 0 0	1.30			4.9			
25	11/7 14:16	singleProces	1 20	10 3	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	24 7	16 metrics	5 7 7	24	4 4 128	3 🗹		disk	100 s	ssd-ext4	≤ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.25			9.5	1	??	30
26	11/7 10:18	singleProces	1 20	10 :	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	24 7	16 metrics	5 7 7	24	4 4 128			men	r 100 s	ssd-ext4	≤ 35	NO	2 32	25	64	128	2 3	3 3	35 job	ex 0 (	1.26			7.5		??	30
27	11/7 10:38	singleProces	1 20	10 :	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	24 7	16 metrics	5 7 7	24	4 4 128			disk	< 100 s	ssd-ext4	<ul> <li>✓ 0</li> </ul>	NO	2 32	25	64	128	2 3	3 3	0 job	ex 0 (	1.30			0.1		0.1	30
28	11/7 12:27	singleProces	1 20	10	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	24 7	16 metrics	5 7 7	24	4 4 128		$\sim$	disk	100 s	ssd-ext4	× 35	GZ	2 32	25	64	128	2 3	3 3	35 Job	ex 0 0	1.30			5.0		3.0	30
9	11/7 13:10	singleProces	1 20	10	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	24 7	16 metrics	5 7 7	24	4 4 128			disk	100 s	ssd-ext4	35	SN	2 32	25	64	128	2 3	3 3	35 Job	ex 0 0	1.30			5.0		3.0	30
<u>sO</u>	11// 1/:23	singleProces	1 20	10	3600	300 no	ne -	60	camur	nda/zeebe	8.1.2	n1-	24 7	16 metrics		24	4 4 128		H	disk	100 5	ssd-ext4	35	GZ	4 64	25	64	128	2 3	3 3	35 Job		1.30			5.0		2.7	
51	11/7 17:00	singleProces	1 20	10	3000	300 no	ne -	60	camur	nua/zeebe	8.1.2	n1-	24 7	16 metrics		24	4 4 128		H	disk	100 5	ssd-ext4	35	GZ	2 64	25	64	128	2 3	3 3	35 JOD		1.30			5.0		3.1	
2	11/0 0:40	singleProces	1 20	10	3600	300 no	ne -	60	camur	nua/zeebe	8.1.2	n1-	24 7	16 metrics		24	4 4 128		H	uisk	100 5	ssu-ext4	35	GZ	8 64	25	64	128	2 3	3 3	35 JOD		1.30			5.0		2.7	
3	11/8 8:49	singleProces	1 20	10	3000	300 no	ne -	60	camur	nua/zeebe	0.1.2	n1-	24 8	16 metrics	8 8	24	4 4 128			dial	100 5	ssu-ext4	35	GZ	2 128	25	64	120	2 3	3 3	35 JOD		0.30			5.0		2.7	
24	TTIO 3:00	singleProce:	1 20	10 .	3000	300 110	ne -	00	camur	nud/zeebe	0.1.2	IIT-	64 8	TO methos	0 8	20	4 4 128			UISK	100 5	sou-exi4	35	962	c 175	25	04	120	2 3	3 3	20100		1.30			5.0		2.1	

#### **Iterative Benchmark Setup with Zeebe Tuner**



Ľ

### **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

Google Sheets





C

#### **Zeebe Tuner for iterative Performance Tests**

#### **Input: Zeebe Configurations**

		Load Generaton@tarter								engine									_						JVN	Rocks	DB	Batew	ay .		Job Wor
feat Case Name	Tressary (CET)	Process Madel	Starter Replicate	Lond Generator Threads	(x) white our during	start Throughput (Pha)	Pade Adjustment Strelegy	Start Pillnereases Factor	Message TTL (min)	JÖRN	Engine Version	Machine Type	Cluster Som (nodes) ACPUs (heusethreadstrade)	RAM IN Bunchet	Esperiers	CPU Thread Pool Strethofe IO Thread Pool Strethofe	Partitions	Reglication Factor Log Segment Size (MB)	pre Allocate Begreent Files	Disable Explicit Ruft Flush	Disk Type Disk Size File Spatem (Sterage Chear)	Electronomia Inter-Region Latency (ms)	Newsork Compression	max Append Beich Stee (KB	24M NoveMARPENDERINGE	Webs Buffer Size (MB)	Buffer to Mairton (MB)	Replican (notice) vCirus (hypertheods)	(urs) www.	Number of Threads Inter-Region Latency (mid	Cherre for Job Worker Job duration (res)
		Defeuli Min tested		1 1	10 18	200		0.4	60	camenda/20010	hatest C	rd.	3 3	2 4	4 classic 4	2 2	1	3 128			14 128 stal-ce4	2	NO	2 32	25	64	125	2			1000 <u>52</u> 0
-		Max tested	1	00_11	92 <u>36</u>	00 50		-84	- 66	cam, ndateete		n2-2	4 14	113	S WHICH	14 25	- 26	4 612		믕	disk 970	-			25	64	1281	a			10044.90
4		singleProce	1 3	20 1	10 35	00 00	non		80	camorela/sease	81.2	n1-	8 4	1 14	6 matrica	4 4	24	4 121		H	thic 100 stal-act-7	<u> </u>	ND	8 32	25	84	125	2 3	3	3 0	joba: 0
-	2022-10-5	is rectroce	1	20 1	10 35	00 9.	1.000		60	camunda/zecce	8.1.2	nı	•		5 metrics	4.4	32	4 125	100	H	dist 200 ssd-cm4		NO	8 32	25	64	125	2 3		3 0	00C+ 0
-	1.000000	and Fines	1	20 1	0 35	~ ~	1000		100	camunda/zeece	0.1.0	-	2.3		a mento	1 1	24	4 1 22	100	H	disk 210 sod exa		NO.	0 02	20	5.5	1.20	0.0	1	2 2	inter of
15		a mial more	1	20 1	10 35	201 100			190	certurala/secter	8.1.2	nle		1.	S meltes	4 4	10	4 125		ň	they 100 satisfied	10 18	ND	8 32	25	54	125	2 3	1	3 0	sther C
9	11/2 9:25	angleProce	1	20 1	10 35	00 180	1000		60	cainunda/zeete	8.1.2	14	8 4	1	5 metrics	4 4	24	4 125		Ö	disk 100 sed-ooA	2 25	NO	8 32	25	64	125	2 3	8	3 0	ober 0
10	1112 10:37	singleProce	1 :	20 1	LC 36	00 300	1000		60	camunda/zeete	8.1.2	nt	8 6	1 11	6 metrics	6.6	24	4 128			disk 100 sed exc4	35	NO	8 82	25	64	128	2 3	. a	8 0	jobce o
11	11/2 11:32	singleFraces	1 :	20	0 35	00 300	i cone	()	66	camunda/zeete	9.1.2	nl-	8 8	1.1	C metrice	8.8	24	4 175			disk 100 sad-end	25	NO I	8 32	25	64	128	2 3	3	3 0	jobes 0
12	11/2 1.2 /47	angleFriese	1 :	20 1	10 35	201 200	11414		190	cerrurabaSeebe	8.1.2	nl-	0 1	1 13	s meltus	a n	32	4 120			they 100 sosterol-1	2 25	ND	1 32	25	54	1.75	2 3	3	3 0	paters C
-12	11/2 13:25	angleProce.	1 :	20 1	10 38	00 300	1000		BU.	camanda/acche	8.1.2	14	8 11	11	6 matrics	12 12	24	4 126		님	disk 100 sad-cetA	Si 25	NO	2 32	25	64	125	2 3	3	3 6	phone 0
24	11/2 14:38	singleProce	1 :	20 1	10 36	00 300	1000		60	camunda/zeete	8.1.2	11	e 14	1	6 metrics	14 14	24	4 125		H	disk 100 ssd crist	35	NO	2 82	25	64	128	2 3	8	3 0	jobce 0
16	30.0 10.05	singlehitice	1.1	20 1	0 36	00 80	100		00	camundazeete	8.1.2	11.		1	6 mence	4 4	24	4 1.28		H	disc 100 sea-ensi		10	2 32	03	64	126	2 3		3 39	0021 0
10	12/4 22/22	* HIPPENER			0 35		11414		191	CATTLINE PHILE	9.1.2	na-			D ITHINK	4 4	201	4 1.01	10	H	they that some of	10 IN	NIN .	1 91		61	1.10	2 4	4	1 10	pane o
-	114.0.22	- restationer		30. 1	0.96	00 D.	1000		-	cate of attories	81.0	114			a realized	4 4	20	4 1.10			dels 100 sed orea		10	2 92		64	138	1.1	-	2 22	inter o
19	114 13:09	snahemce	1	20 1	10 35	00 90	1001		60	cam indateete	812	nt		1	6 metrics	7 7	24	4 125		n	disk 100 sed-one	36	NO	2 37	25	64	128	2 3		3 35	johne 0
20	114 13:37	angieFrage.	1	20 1	10 35	00 00			10	camundaberte.	812	nl		1 11	netres.	7.7	24	4 1.78			risk 100 subject	2 25	ND .	2 32	25	54	178	2 2		2 35	mbra G
21	124 1441	angleFrome	1 :	282 1	10 35	200 200			50	certurals/seater	8.1.2	n1-	8 1	1 11	ereinen 1	7.7	24	4 125	1	D	their 100 seal-east	2 25	ND :	2 32	25	54	125	2 3	. 3	3 33	paten C
22	114 13:55	angialmore:	1 :	202 1	10 35	00 300	1000		ы	camunda/seese	8.1.2	61-	ð )	1.1	e matrics	7.7	24	4 128	: 🖾		disk 100 ssd-ext4	2 35	NO	2 32	25	64	125	2 3	3	3 33	jobar o
23	11:4 15:15	singleProcer	1	20 1	10.96	00 300	1000		66	camunda/zeebe	8.1.2	rd 1	2 7	11	6 menics	7 7	24	4 128	2		disk 100 ssd ensk	35	NO	2 82	25	64	128	2 3	8	8 85	Jobce 0
24	1144 1.6:43	singleFincer	1	20 1	(C 35	00 300	1001		66	camunda/zeete	812	nl-1	2 7	7.13	. metree	7 7	24	4 128			disk 100 exd-exel	26	NO	2 32	25	64	128	2 3	a	3 87	Jobes 0
	11.94 10:13								001			14.4							10	0	sheri (203) heatenint	20120			28		131				infant 12
22	11/7 14:15	angehous	1 :	201 1	10 38	00 200	C CON	1.	B)]	Caminda/Jeste	8.1.2	14-2	4	11	s metres	1.7	24	4 122		1	that 100 stal-and4	28	ND :	2 32	25	84	125	2 3	3	3 35	phie 0
26	11/7 10:18	singleProce	1	20 1	10 38	00 300	1000		60	camunda/zeebe	8.1.2	rd 2	4 1	1	6 mothes	7.7	24	4 125			men 100 ssd-citi4	35	10	2 32	25	64	128	2 3	8	3 37	JODCA C
	107 10:35	singlehitice	1	20	0.46	00 300	1003		60	camundazeete	8.1.2	n1-2		1	mence	11	29	4 128	80	873	diss 100 650-600	0	80	2 82	03	64	125	2.2	4	3 0	0024 0
211	107 1027	angernen	-	201	0 38	AL 120			101	Centing Server	81.2	n1-2	4 1		C melers	2 7	20	4 120	100	10	they 100 sectored	10.00		2 12		8.5	126	4 4	1	1 10	inter C
50	1107 17:25	sourchores	1	20 1	10 96	00 500	1000		60	cam or a treate	812	11.2	4	1	6 proteins	2.2	24	4 126		ñ	dia 1/0 soluma	5 95	02	3 84	25	64	128	2 3	. 9	3 95	intern it
81	11/7 17:00	sindirFrocer	1	20 1	0.96	00 300	1001		60	camunda/zente	812	n1 2	4	1	6 metics	7 7	24	4 125			disk 1/0 sed-ensi	2 25	62	2 64	25	64	128	2 3		3 35	jobre 0
32	11/7 17:52	e regie Fromes	1	20 1	0 35	00 200			10	camundabaste	8.1.2	n1-2	4 1	1 11	retirs.	7 7	28	4 175	1	Ē	they 100 subweld	2 25	67	8 64	25	54	178	2 2		2 33	mbes C
23	116 8:40	angiaFronat.	1 :	20 1	10 35	00 300			190	ormunate/sector	8.1.2	14-2	4 1	1 11	t maines		24	4 121		D	their 100 soul-sec-1	2 IN	GZ :	2 125	25	84	125	2 3		3 38	patres C
*	11/5 9.00	sighthous	1 :	20 1	10 35	00 300	-		60	camunda/zeebe	812	12-2	4 2	\$ 15	e matica	8 8	24	4 125		2	disk 100 ssd-cos4	2 35	62 :	2 125	25	84	125	2. 3	3	3 33	polace O
25	11/8 9:28	singleProce	1 :	20 1	10 36	80 300	i nond		60	camunda/zeebe	812	n1 2	4 8	3 13	6 motilics	8.8	24	4 126		2	disk 100 ssd on4	<b>1</b> 0	GZ :	2 128	25	64	128	2 3	8	\$ 10	jobc+ 0
20	10.010.00	* ngieEnces	1.	20 1	ic an	00 200	1000	-	66	camundabeete	8.1.2	12-2	4 6	8 11	i metter	8 8	24	4 178	1		disk 100 statered	35	67	2 123	25	nd	178	2 3		3 35	johes 0
27	11-11-12-12	angeFrees	1 :	20 1	10 38	201 200	1001		191	ormundabeette	0.1.2	12-		1 11	S melites	8 0	24	4 121		Н	their 100 seal-rest	10	GZ :	2 121	25	54	126	2 2	3	3 10	piters 0
- 28	11/5 11:54	angefrixer	4	203 1	10 35	00 150	TOTAL	-	683	CATTURNAL	8.1.2	12-	0 8	5 11	o matrica	0 8	24	4 128	1	H	they 100 states	10	62	2 64	25	54	125	2 3	3	3 10	DOUT C
- 22	1100 10:00	singleProce	1 1	202 3	LC 35	00 300	1000		60	ca nunda/zeese	812	112	0 0	5 11	e metres	3 8	24	4 128	100	H	disk 100 ssd citiA	39	02	2 64	22	54	125	2 3		5 37	Jobcs C
- 40	1149 10-27	s rateFrome	1	20 1	0 35	00 200	1000		100	cam inda heate	a12	12.			i maine	4 8	28	4 120	100	H	disk 100 sed-ensi	10	57	8 32	70	64	178	2 3	1	3 10	mbes C
42	11.91 1.0-23	anualizer	1	20 1	10 38	00 19	1000		197	centurale/seeter	8.1.2	10-		1 1	i metary	0.8	24	4 129	10	H	their 100 states of	2 10	az	8 32	25	54	125	2 3	1	3 10	ster C
43	1149 16:45	singisfrace	1	20 1	0 35	00 80	1000		60	camunda/seese	812	12		1 1	6 metres	8 8	24	4 125		ŏ	disk 100 sed-cet4	10	02	8 32	25	64	128	2 3	9	3 10	obcs 0
44	118 17:03	SingleProcer	1	20 1	0.86	00 300	1000		60	camunda/zeete	812	re	8 6	11	6 menies	8 8	24	4 128		$\sim$	disk 100 ssd entil	10	GZ I	8 82	25	64	128	2 3	. a	8 10	Jobce O
45	11/9 9:24	angieFrices	1	20	10 35	00 300			60	camundateete	51.8	n2.		1 13	retire	8 8	24	4 128			disc 100 and-end	10	SN	8 32	25	64	128	2 3	3	3 10	inbes 0
16	11/0 9:53	angieFreese	1 :	20 1	10 35	00 300	1100		190	cemurate/seeter	0.1.2	12-	5 1	1 11	s metrics	a 11	24	4 1.21	1	0	their 100 soul-rol-1	10	54	4 32	25	84	125	2 3	3	3 10	patres 0
47	11/1 10:10	angicfrome	1 :	20 1	10 35	00 500	TOPS		60	camunda/secte	8.1.2	12-	8 8	3 13	earlan d	8 8	24	4 125		0	this 100 sal-cetA	10	574 1	2 32	25	84	125	2 3	9	3 10	edace C
48	11/9 10:30	singlefroce:	1 :	20 1	10 36	00 300	1000	1	60	camunda/zeese	812	12	8 6	1	6 metrics	8 8	24	4 126			disk 100 ssd exte	10	BNI I	8 32	25	64	125	2 3	8	8 0	jobce 0
10	11/9 10/54	singleFrocer	1	20	10 30	00 190	100		66	camunda/zeete	8.1.2	n2-	8 8	1	metace	8 8	24	4 128		님	disk 100 ssd-ext4	10	54	8 32	25	64	12B	2 3	3	3 0	Jobes 0
- 20	11/2 11:10	angoFrone	4 1	AL 1	0 35	CH1 500	1.1834		n/l	CAMILINASHICH	4.1.2	12-			D FEMILES		24	4 171		·H	mee and soul-real	10	tere i	0 32	25	24	178	2 2	1	2 0	Distant C
	169 11 23	a right house	1 3	0. 1	10 35	00 500	THE	817 -	DU	CHALLER WELF	8.1.2	1621	0 1	. 11	o tradito	e 8	24	* 123	. 68	1.1.1	DEAC 200 SOLD-DOA	NE 10	21.6	a 38	25	D+3	145	2 3	- 4	5 10	Dance of

#### **Output: Performance Metrics**



#### **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

=> Optimization is an ongoing process



C

### **Throughput (PI/s)**



**Configuration Tuning** 

**Engine Hacking** 

#### **Message Throughput & Backpressure**





#### **Configuration Tuning**

#### **Engine Hacking**

### **Process Instance Duration (Latency)**



#### **Configuration Tuning**

#### **Engine Hacking**

# General / Zeebe Performance & Genera



## General / Zeebe Performance & CO 2022-0-05 13:22:31 to 2023-09-05 13:49:30 CPU Use General / Zeebe Performance & Construction of the second se



() 2023-09-10

partition All ~

## **Job Streaming: PI Duration -56%**



## Job Streaming: CPU Usage -50%



## **Predictable Scalability**



Խ

GCP N2D (3rd Generation AMD)

Throughput in TI/s

## GCP N2D vs C3D (3rd & 4th gen AMD)







### **Price Performance for Zeebe Brokers**



#### C

### **Current Tuning Best Practices**

C

- Job Streaming
- Priority Election
- Always <u>enforce Leader Balancing</u>
- Scale partitions & brokers
- Latest generation CPUs
- Fastest possible disks & file systems, e.g. XFS
- Enable <u>RocksDB SST file Partitioning</u> for large state
- <u>Raft flush delay time</u> (takes disk out of critical path)
- Multi-region: prefer local brokers by selecting correlation key



## **Key Takeaway**



### Yes, it's horizontally scalable. Let's talk!

## Resources

#### **Bernd Ruecker's Blog Articles**

- How to Benchmark Your Camunda 8 Cluster
- How to Achieve Geo-redundancy with Zeebe

#### GitHub

- <u>camunda-consulting/zeebe-tuner</u>
- <u>camunda-community-hub/camunda-8-benchmark</u>

#### **Camunda Platform 8 Docs**

- <u>Metrics</u>
- Deployment options

#### **Contact Us**

- Contact Form
- mailto:info@camunda.com

Try Camunda Platform 8 for free

# Thank You

🔀 fm@camunda.com

in linkedin.com/in/falko-menge

github.com/falko

