KADAI
envite consulting GmbH

envite

Anlage 7 - Blauer Engel für Software

KADAI – OPEN Source Software für Task Management
(KADAI Documentation)

Diese Anlage enthält:

a) Minimale Systemvoraussetzungen (siehe Abschnitt 3.1.1.1)

b) Dokumentation des Messsystems (siehe Abschnitt 3.1.1.2)

c) Messergebnisse im Leerlaufzustand (siehe Abschnitt 3.1.1.3)

d) Messergebnisse bei der Nutzung (siehe Abschnitt 3.1.1.4)

e) Kalenderjahr und Daten zur Abwärtskompatibilität (siehe Abschnitt 3.1.2.1)

f) Kontinuität des Softwareproduktes (Siehe Abschnitt 3.1.3.3)

Minimale Systemvoraussetzungen

(Alle Werte in SI-Einheiten)

- CPU: Es wird mindestens ein SSE2 kompatibler Prozessor vorausgesetzt.
- Architektur: x86-64 oder ARM64
- Prozessor: 1GHz+ Dual-core Prozessor
- Arbeitsspeicher: 2048 MB
- Festplattenspeicher: 5120 MB
- Betriebssystem und Softwarepakete:
    - Linux Betriebssystem (z.B. Ubuntu 18.04 oder neuer); Windows 10+; Windows Server 2019+
    - Java 17 oder 21 (beides Long-Term Supported Versionen)
- Erforderliche externe Dienste: Keine
- Zusätzliche Hardware: Keine

Dokumentation des Messaufbaus

(Alle Werte in SI-Einheiten)

Die Messung wurde durchgeführt auf einem dedizierten Bare-Metal System (Fujitsu PRIMERGY TX1330 M2) an welches das Leistungs-Messgerät (MCP39F511N-https://www.microchip.com/en-us/development-tool/ADM00706) via USB angeschlossen wurde.

Eine Synchronisation der Zeitstempel zwischen mehreren Systemen ist nicht erforderlich, da lediglich auf einem System gemessen wird.

Das Leistungs-Messgerät wurde auf eine kontinuierliche Übertragung der Leistungs-Messdaten eingestellt in einem 99ms Intervall.

Entsprechend der Anforderung der Vergabekriterien wurden die Zeitstempel der Leistungsdaten direkt beim Auslesen aus dem Messgerät erzeugt und in einer Log-Datei abgelegt.
Das Messgerät führt keine Mittelung der Werte durch sondern liefert instantan-Werte.
Durch das gewählte Abtastintervall ist dies ist jedoch identisch mit den in den Vergabekriterien vorgeschlagenen Geräten (z.B. Janitza UMG-604 - https://www.janitza.de/files/download/manuals/current/UMG604-PRO/janitza-bhb-umg604pro-en.pdf). Das Janitza UMG-604 z.B nimmt laut Datenblatt intern eine 200ms Mittelung der Effektivwerte vor.

Das System wurde weiterhin auf eine feste CPU-Frequenz eingestellt und das Intel Turbo-Boost Feature sowie Intel SpeedStep deaktiviert.

Vor jeder Messung wurde weiterhin folgendes ausgeführt:

- sync
    - Offene Schreiboperationen auf der Festplatte werden ausgeführt, damit diese nicht während der Messung passieren
- sudo /usr/sbin/sysctl -w vm.drop_caches=3
    - Weist das Betriebssystem an alle bestehenden Caches freizugeben, damit das Szenario nicht "falsch-niedrige" Werte ausliefert durch bestehende Speicher- oder Datei-Caches

Mess-Software / Version

GREEN CODING;

Als Mess-Software wurde das Green Metrics Tool genutzt in der Version v2.2
mit Commit-Version 52bf063e722ceb487355373f56912e22922ab9f4

Automatisierungs-Software

Als Automatisierungs-Software wurde das Green Metrics Tool genutzt in der Version v2.2
mit Commit-Version 52bf063e722ceb487355373f56912e22922ab9f4.
Sowie grafana/k6:1.1.0 als Client / API-Lasttreiber.

Erhobene Kennwerte

Für die Kennwerte innerhalb des Systems wurden aus dem OS die Werte über Sammel-Skripte ausgelesen.

Hierbei respektive:
- Permanentspeicherbelegung über stavfs syscall als C Programm
- Arbeitsspeicherbelegung über /proc/meminfo
- Netzwerkverkehr auf IP Ebene über tcpdump
- Netzwerkauslastung über /proc/net/dev
- CPU-Auslastung über /proc/stat

Messmethode: Einsatz von Virtualisierung

Wie in den Vergabekriterien unter 3.1.1.2 als Option genannt, wird hier als Messmethode der Einsatz von Virtualisierung genutzt.

Hierbei werden die Komponenten Client/Server auf einem System gestartet und durch Virtualisierung voneinander getrennt.

Dabei haben Client / Server jeweils folgende Unterkomponenten:
- Client
    - Dargestellt durch einen Lasttreiber (k6)
- Server
    - app
    - postgres

Bei Einsatz einer Virtualisierung werden zunächst die Werte für die Grundauslastung normal erhoben. Zu diesem Zeitpunkt findet noch keine Virtualisierung statt.

Die Werte für die Leerlaufauslastung werden ebenfalls kumuliert erhoben und auch als solche ausgewiesen. Zuzüglich müssen hier jedoch die unter "Erhobene Kennwerte" genannten Metriken für Client / Server getrennt erhoben werden.

Gleiches gilt analog für das Nutzungsszenario.

Im Sonderfall der Messung dieser Client/Server Anwendung, welches mit einem API-Lasttreiber instrumentiert wird, ist eine Aufsplittung der Messdaten im Leerlauf nicht notwendig, da diese identisch mit den Werten des Gesamtsystems sind (Siehe Anmerkung zu Qualitäts-Anforderung 1.4)

Anmerkung zu Qualitäts-Anforderung 1.4

In den Vergabekriterien wird für den Leerlauf gefordert, dass die Anwendung aktiv ist und gestartet ist.

In Fall dieser Client/Server Anwendung findet die Steuerung nicht durch eine eigenständige App, sondern durch einen API-Lasttreiber statt. Diese hat keinen Zustand in dem er geöffnet ist. Er kann nur ausgeführt werden und beginnt dann sofort mit seiner Lastgenerierung

Durchführung der Messung zu Anforderung 3.1.1.4

In den Vergabekriterien wird gefordert, dass:
• Arbeitsspeicherarbeit (MByte*s)
• Permanentspeicherarbeit (Lesen und Schreiben) (MByte/s*s)
• Übertragene Datenmenge außerhalb des lokalen Netzwerks (Mbit/s*s)
• Energiebedarf (Wh) (netto beim Szenario-Test, brutto beim Langzeit-Test)
• Liste aller aufgerufenen Internetadressen (IP-Adresse oder Domain Name), örtliche Zuordnung (Ländercode nach ISO 3166-1), Angabe zum Eigentümer (eigener oder externer Dienst), Häufigkeit des Aufrufs (Frequenz oder Anzahl pro Zeiteinheit)

erhoben werden. Entsprechend müssen diese Werte beim Einsatz von Virtualisierung ebenfalls getrennt ermittelt werden.

Eine Besonderheit entsteht hierbei wie, der Overhead aus dem Automatisierungs-Tool und dem Mess-Tool zu berücksichtigen ist.

In einer Messung ohne Einsatz von Virtualisierung findet der Overhead durch die Mess-Tools Einzug in die Grundauslastung und der Overhead durch die Automatisierung Einzug in die Messung während der Nutzung.
Somit ist das Mess-Tool, welches ebenfalls in der Grundauslastung schon läuft, aus der später auszuweisenden Arbeit bereits herausgerechnet.
Die Automatisierungs-Software und der damit entstehende Overhead werden jedoch den Kosten der Software selbst in der Messung während der Nutzung zugeschlagen.
Entsprechend ist dies auch bei Einsatz von Virtualisierung zu handhaben. Hierzu muss der Overhead der Automatisierungs-Software rechnerisch aus der Differenz zwischen den getrennt für Client / Server erhobenen Kennwert berechnet werden.

Dies geschieht durch:
$$CPUBruttoauslastung_{Gesamtsystem} - CPUNettoauslastung_{Client} - CPUNettoauslastung_{Server} - CPUBaselineauslastung_{Gesamtsystem} = Overhead_{Automatisierung}$$

Der Overhead der Automatisierungs-Software muss dann dort zugeschlagen werden, wo in einem nicht-virtualisierten System die Automatisierungs-Software laufen würde. In diesem Fall wäre dies der Client.

Für die Energie sind diese Werte nicht getrennt zu erheben, daher wird hier ein Virtualisierungs-Allokations-Faktor (VAF) erstellt, welcher sich aus der jeweiligen CPU-Auslastung von Server / Client berechnet und mit der Netto-Auslastung des Systems in Bezug gesetzt wird.

Konkret:

$$VAF_{Server} = \frac{CPUNettoauslastung_{Server}}{CPUNettoauslastung_{Gesamtsystem}}$$

$$VAF_{Client} = \frac{CPUNettoauslastung_{Client} + CPUNettoauslastung_{AutomatisierungsOverhead}}{CPUNettoauslastung_{Gesamtsystem}}$$

Und es muss gelten:

$$VAF_{Server} + VAF_{Client} = 1$$

Die Verwendung der CPU-Auslastung ist hierbei ein plausibel nutzbarer Wert, da diese repräsentativ für die Arbeit aller Komponenten gilt.
Schlussendlich ist die CPU dafür verantwortlich, sowohl die elektrische Arbeit der CPU selbst, des Speichers als auch des Permanentspeichers und des Netzwerkadapters zu verursachen.

Die Methodik wird in folgenden akademischen und technischen Quellen beschrieben:

- Scaphandre Dokumentation - https://hubblo-org.github.io/scaphandre-documentation/explanations/how-scaph-computes-per-process-power-consumption.html
- Kepler: A Framework to Calculate the Energy Consumption of Containerized Applications - https://ieeexplore.ieee.org/abstract/document/10254956
- Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—Green Software Measurement Model (GSMM) - https://www.sciencedirect.com/science/article/pii/S0167739X24000384

Wichtig: Zu beachten hier ist, dass diese Methodik nur dann erlaubt ist, wenn das Hardware-System mit einer festen CPU Frequenz und ohne DVFS (Dynamic Voltage and Frequency Scaling) eingestellt ist.
Der Hintergrund ist, dass sonst die relativen Prozentangaben keine gleiche Grundgesamtheit haben. Im Falle von DVFS kann die CPU während einer Messung den Takt ändern und somit in einer Zeiteinheit unterschiedlich viele CPU-Zyklen durchlaufen.

Plausibilitätsbetrachtung des Zuschlags zum Client

Um die virtualisierte Testmethode anzuwenden ist eine Plausibilitätsbetrachtung notwendig, welche darlegt, welcher Komponente der Overhead zuzuschlagen ist oder der Overhead muss so klein sein, dass er zu vernachlässigen ist.
Die Begründung, dass dieser dort zugeschlagen wird, wo auch das Automatisierungs-Tool läuft, ist hinreichend, jedoch nicht abschließend.
Es könnte ebenfalls der Fall sein, dass der Kernel im Betriebssystem Arbeiten für die *Server Komponente* erledigt.

Die Berechnung des Overheads der Virtualisierung auf CPU-Ebene ergibt den Wert -0,09 %. Dass dieser Wert negativ ist, ist erwartbar. Die Nutzung von KADAI sorgt für wenig Kernel-Operationen, so dass es entsprechend einen sehr geringen Overhead gibt. Der negative Wert erklärt sich durch die genutzte Rechenvorschrift, bei der vom Overhead vom OS+Automatisierung während der Nutzung (0,25 %) die zuvor gemessene Grundauslastung (0,34 %) abgezogen wird.

Der „negative“ bzw. geringe Overhead zeigt, dass dieser als vernachlässigbar angesehen und dort zugeschlagen werden kann, wo auch das Automatisierungstool läuft.

Die notwendige Annahme, für die Nutzung der Messmethode *Einsatz von Virtualisierung* jeglichen Overhead hier dem Client zuzuschlagen, ist somit gerechtfertigt.

Die gleiche Betrachtung muss auch für den Arbeitsspeicher und auch das Schreiben auf die Festplatte gemacht werden. Kleine Abweichungen (wenige % / wenige Hundert-MB) sind hier wieder völlig normal, da die Werte sequentiell erfasst werden und somit auch zwischen dem Auslesen Speicher neu allokiert werden kann und ebenfalls Daten auf der Festplatte in einen temporären Cache geschrieben und erst später final auf die Festplatte gespeichert werden.

Anmerkung zur Übertragenen Datenmenge für Netzzugang

Für den Netzwerkverkehr wird kein Virtualisierungsfaktor bestimmt, da der Server / Client Komponenten bereits über virtuelle Interfaces den Netzwerkverkehr schicken. Dies bedeutet, dass jeglicher Netzwerktraffic der Komponenten voll in diesen virtualisierten Interfaces erfasst wird.
Würde man diese in Bezug zu dem Netzwerkverkehr des Gesamtsystems setzen, würde hier ein "Double-Counting" erfolgen, da der Traffic im unterliegenden Betriebssystem ebenfalls separat gezählt wird.
Es ist in Folge ausreichend, nur den Netzwerktraffic der virtualisierten Komponenten zu betrachten und als Arbeit auszuweisen.

Anmerkung zu den Markern in den Zeitreihen (Start / Ende des Szenarios)

Das Green Metrics Tool führt immer einen vollen Durchlauf von Baseline, Install, Boot, Idle, Runtime und Remove durch.
Für die Messprotokolle des Blauen Engel relevant ist immer der Teil zwischen "Starting Phase [RUNTIME]" und "Ending Phase [RUNTIME]".
Ausnahme: Die Grundauslastung. Hier wird die Phase "Baseline" des Green Metrics Tool genutzt welche analog beginnt bei "Starting Phase [BASELINE]" und endet bei "Ending Phase [BASELINE]".

Anmerkung zum Hilfsblatt in der Tabellenkalkulations-Datei

Die Permanentspeicherauslastung von Container benötigt zur Ermittlung drei Komponenten:

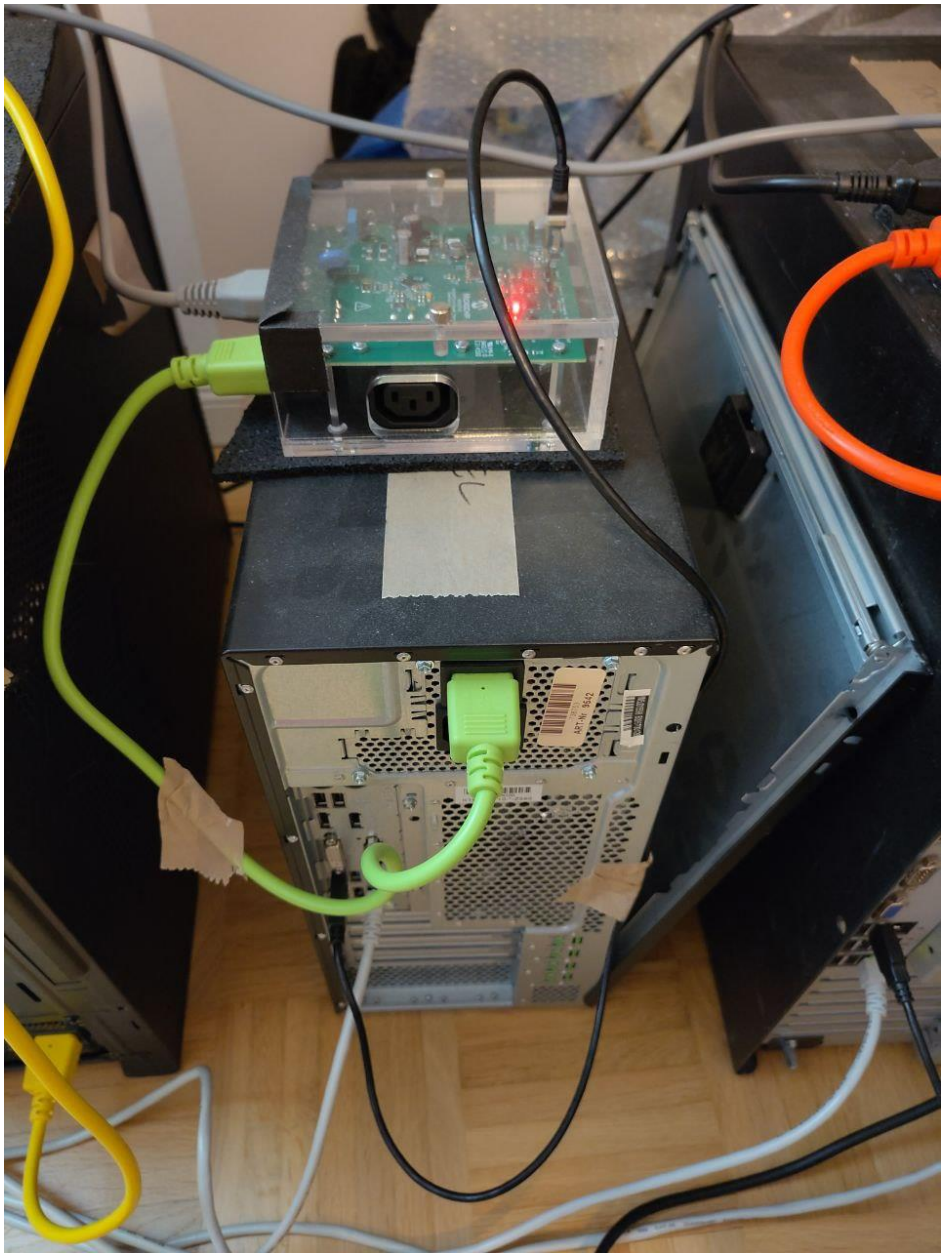
- A: Die Größer des Container Image
- B: Alle Daten die während der Ausführung des Standardnutzungs-Szenario im Container geschrieben werden

A ist im Hilfsblatt eingetragen und werden in dem Berechnungsbogen über deine Formel zu B hinzuaddiert.

Die Ermittlung der Werte von A sind in der Anlage 3 im Ordner "Image Size Logs" hinterlegt. Dies erfolgt über die Container-Orchestrierungs-Software Docker.

Die Ermittlung der Werte von B erfolgt über die Messprotokolle.

Foto des Messaufbau



ID	Bezeichnung	Hinweis	Ergebnis	Einheit
Messung der Grundauslastung				
3.1.1.3 a)	Mittlere Prozessorauslastung		0.34	%
3.1.1.3 b)	Mittlere Arbeitsspeicherbelegung		716.77	MByte
3.1.1.3 c)	Mittlere Permanentspeicher-belegung		12718.5	MByte
3.1.1.3 d)	Mittlere beanspruchte Bandbreite für Datenübertragung	Die mittlere Bandbreite des Messsystems wird ohne das installierte Softwareprodukt gemessen	0	Mbit/s
3.1.1.3 e)	Mittlere elektrische Leistungsaufnahme (brutto)		18.45	W
Messung der Last des Softwareprodukts im Leerlaufzustand				
3.1.1.3 a1)	Mittlere Prozessorauslastung		0.23	%
3.1.1.3 b1)	Mittlere Arbeitsspeicherbelegung		581	MByte
3.1.1.3 c1)	Mittlere Permanentspeicher-belegung		1,809	MByte
3.1.1.3 d1)	Mittlere beanspruchte Bandbreite für Datenübertragung	Die zusätzliche mittlere Bandbreite, die durch den Leerlaufbetrieb der Software entsteht wird gemessen	0	Mbit/s
3.1.1.3 e1)	Mittlere elektrische Leistungsaufnahme (netto)		0.06	W

(Alle Werte in SI-Einheiten)

Messergebnisse während der Nutzung

(Alle Werte in SI-Einheiten)

- **Szenario**
  - Task erstellen
  - Workbaskets suchen
  - Workbasket lesen
  - Tasks aus Workbasket lesen
  - Task lesen
  - Task editieren
  - Task verschieben
  - Task zuweisen
  - Kommentare lesen
  - Kommentar schreiben
  - Kommentar löschen
  - Task abschließen
  - Suche Tasks mit Ordnungsbegriff
  - Erstelle Task Status Report

ID	Bezeichnung	Hinweis	Ergebnis	Einheit
3.1.1.4 a)	Prozessorarbeit		866	%*s
3.1.1.4 b)	Arbeitsspeicherarbeit		85,058	MByte*s
3.1.1.4 c)	Permanentspeicherarbeit (Lesen und Schreiben)		94,672	MByte/s*s
3.1.1.4 d)	Übertragene Datenmenge für Netzzugang		0	Mbit/s*s
3.1.1.4 e)	Energiebedarf	netto beim Szenario-Test; brutto beim Langzeit-Test	0.04	Wh

Messung des Softwareprodukts während der Nutzung - Für virtualisierte Systeme - Server

ID	Bezeichnung	Hinweis	Ergebnis	Einheit
3.1.1.4 a)	Prozessorarbeit		841	%*s
3.1.1.4 b)	Arbeitsspeicherarbeit		73,211	MByte*s
3.1.1.4 c)	Permanentspeicherarbeit (Lesen und Schreiben)		51,787	MByte/s*s
3.1.1.4 d)	Übertragene Datenmenge für Netzzugang		342	Mbit/s*s
3.1.1.4 e)	Energiebedarf	netto beim Szenario-Test; brutto beim Langzeit-Test	0.04	Wh

Messung des Softwareprodukts während der Nutzung - Für virtualisierte Systeme - Client

ID	Bezeichnung	Hinweis	Ergebnis	Einheit
3.1.1.4 a)	Prozessorarbeit		26	%*s
3.1.1.4 b)	Arbeitsspeicherarbeit		11,847	MByte*s
3.1.1.4 c)	Permanentspeicherarbeit (Lesen und Schreiben)		42,885	MByte/s*s
3.1.1.4 d)	Übertragene Datenmenge für Netzzugang		138	Mbit/s*s
3.1.1.4 e)	Energiebedarf	netto beim Szenario-Test; brutto beim Langzeit-Test	0.00	Wh

Dokumentation des Szenariotests

Ziel und Kontext

Ziel des Szenario-Tests ist es, KADAI unter realitätsnaher Nutzung zu vermessen. Das Szenario bildet typische Nutzeraktionen ab, startet mit einer Warm-up-Phase und geht anschließend in eine Messphase mit parallelen Nutzern über.

Randbedingungen

Workbaskets (Postkörbe):
20 Stück (*Sammelbehälter für Tasks/Aufgaben; beispielsweise Team- oder Themen-basiert)*

Tasks (Aufgaben):
100.000 bereits in der Datenbank vorhanden

Klassifikationen:
rund 200 (*Kategorisierung von Aufgaben)*

Datenbank:
PostgreSQL

Lastgeber:
k6

Abgedeckte Aktivitäten

KADAI bietet eine Vielzahl von Funktionalitäten. Die im Abschnitt Messergebnisse ausgewählten Schritte repräsentieren den realistischen Umfang für das Nutzungsszenario.

Durchführung einer Warm-up-Phase

Ziel der Warm-up-Phase ist es, den JIT-Compiler der JVM, die die Java-Anwendung (KADAI) ausführt, sowie Caches zu stabilisieren, damit die Messung nicht durch Kaltstart-Effekte verzerrt wird.

Umsetzung: Dreiteilung des Tests in Warm-up → Pause → Messung des Standardnutzungsszenarios mittels "Flows". Flows sind ein Konzept beim Green Metrics Tool für die Unterteilung eines Szenarios in mehrere Phasen. Die Warm-up-Messwerte sind nicht Teil der Zertifizierung.

Anzahl der Nutzer

Die Aktivitäten werden von 100 gleichzeitigen Nutzern ausgeführt. Dafür gibt es zwei zentrale Gründe:

1. Repräsentativität: In der Praxis arbeiten mehrere Personen parallel im System (z. B. in denselben Postkörben oder an denselben Aufgaben). Parallelität ist daher realistischer als ein Einzelnutzer-Szenario.

2. Messstabilität: Bei nur einem Nutzer sind CPU-Auslastung und Energieverbräuche sehr niedrig; schon kleine Messungenauigkeiten wirken dann relativ stark und verfälschen die Ergebnisse. Mit 100 virtuellen Usern (VUs) wird eine gut messbare und stabile Auslastung (+16 % mittlere CPU-Auslastung) erreicht.

Parallelisierungs- und Ablaufstrategie

Es wurden in der Vorbereitung zwei Ansätze betrachtet:

1. Gesamtes Szenario parallelisieren, das heißt alle 100 VUs starten gleichzeitig in das komplette Szenario.
   - Einfach umsetzbar
   - Einzelne Schritte können sich überlappen; das Erkennen energieintensiver Aktionen ist somit schwierig.

2. Einzelne Schritte synchron starten, sodass eine Parallelisierung innerhalb der einzelnen Aktionen stattfindet.
   - Alle VUs starten denselben Schritt zeitgleich, anschließend gibt es eine Pause bis der nächste Schritt synchron gestartet wird.
   - Vorteile:
     - Die Schritte lassen sich klar unterscheiden und können in der Visualisierung sichtbar gemacht werden.
     - Die maximale CPU-Auslastung lässt sich erkennen und der relative Energiebedarf lässt sich visuell vergleichen.

Aufgrund der besseren Darstellung in den Ergebnisgraphen (u.a. auch für zukünftige Messungen) wird die 2. Variante genutzt, mit einer Pause von 4s zwischen den einzelnen Schritten, um eine Überlappung zuverlässig ausschließen zu können.

Umgang mit zufälligen Zugriffen und Caching

Bei einzelnen Aktionen wird zufällig aus vorhandenen IDs gewählt:

- Workbasket lesen / Task verschieben: Zufällige Auswahl aus 17 Workbasket-IDs (die anderen 3 Workbasket-IDs werden im Warm-up genutzt)
- Einzelne Task lesen: Zufällige Auswahl aus 100 Task-IDs

Konsequenz: Mehrere VUs können dieselbe ID treffen → DB-Cache-Treffer sind möglich.

Diese Implementierung wurde trotzdem gewählt, da:

- Es ist realistisch, dass mehrere Nutzer gleichzeitig in denselben Postkörben arbeiten.
- Es ist realistisch, dass mehrere Nutzer dieselbe Aufgabe öffnen (Status prüfen, Abstimmung etc.).
- Vorherige Messungen haben gezeigt: Selbst im Extremfall (alle 100 VUs greifen auf dieselbe Task in der Datenbank zu) ergibt sich kein signifikanter Unterschied in den Ergebniswerten.

Zusammenfassung des Messablaufs

1. Warm-up: 10 VUs führen 15 Iterationen des gleichen Szenarios durch, jedoch mit einer anderen Menge von IDs im Vergleich zum Nutzungsszenario in der Messphase.
2. Pause: Zwischen der Warm-up-Phase und der eigentlichen Messphase liegen 30s Pause.
3. Messphase: 100 parallele Nutzer, die synchronisierte Schritte durchführen. Die Schritte beginnen jeweils im 4 Sekunden Takt.

Kalenderjahr und Daten zur Abwärtskompatibilität

(Alle Werte in SI-Einheiten)

Das System ist mindestens mit einem 5 Jahre alten Betriebssystem kompatibel, respektive:

- Linux Betriebssystem (z.B. Ubuntu 18.04 oder neuer) mit Kernel 4.0 oder größer
- Windows 10 (Mid-2019) / Windows Server 2019+ oder neuer

System Jahr	2016	Jahr
Modell	FUJITSU PRIMERGY TX1330 M2	Text
Prozessor:	Intel(R) Xeon(R) CPU E3-1240L v5 @ 2.10GHz	Text
Cores	4	Zahl
Taktfrequenz	2.1	GHz
RAM:	8 GB	Text
Festplatte (SSD / HDD)	SATADOM-ML 3SE3, 64GB	Text
Grafikkarte:	Matrox Electronics Systems Ltd. MGA G200e	Text
Netzwerk:	Intel Corporation I210 Gigabit Network Connection	Text
Netzteil:	Fujitsu 450W hot-plug, 94% (Platinum efficiency), 100-240V, 50 / 60Hz	Text
Mainboard:	Fujitsu D3373 mit Intel® C236 Chipset	Text
Betriebssystem:	Ubuntu 24.04.3 LTS	Text
Konfiguration (Software):	NOP Linux (https://docs.green-coding.io/docs/cluster/nop-linux/); Um die Testmethode mit "Einsatz von Virtualisierung" zu nutzen wurde die CPU Taktfrequenz auf 2.1 GHz fixiert.	Text

Kontinuität des Softwareproduktes

Hinweise zu Software-Updates: Die generelle Update Policy für KADAI ist in der Dokumentation beschrieben: https://kadai-io.github.io/kadai-doc/docs/user-guide/declarations/updatePolicies

Zudem verpflichtet sich envite, für 5 Jahre kostenlose Sicherheitsupdates durchzuführen.