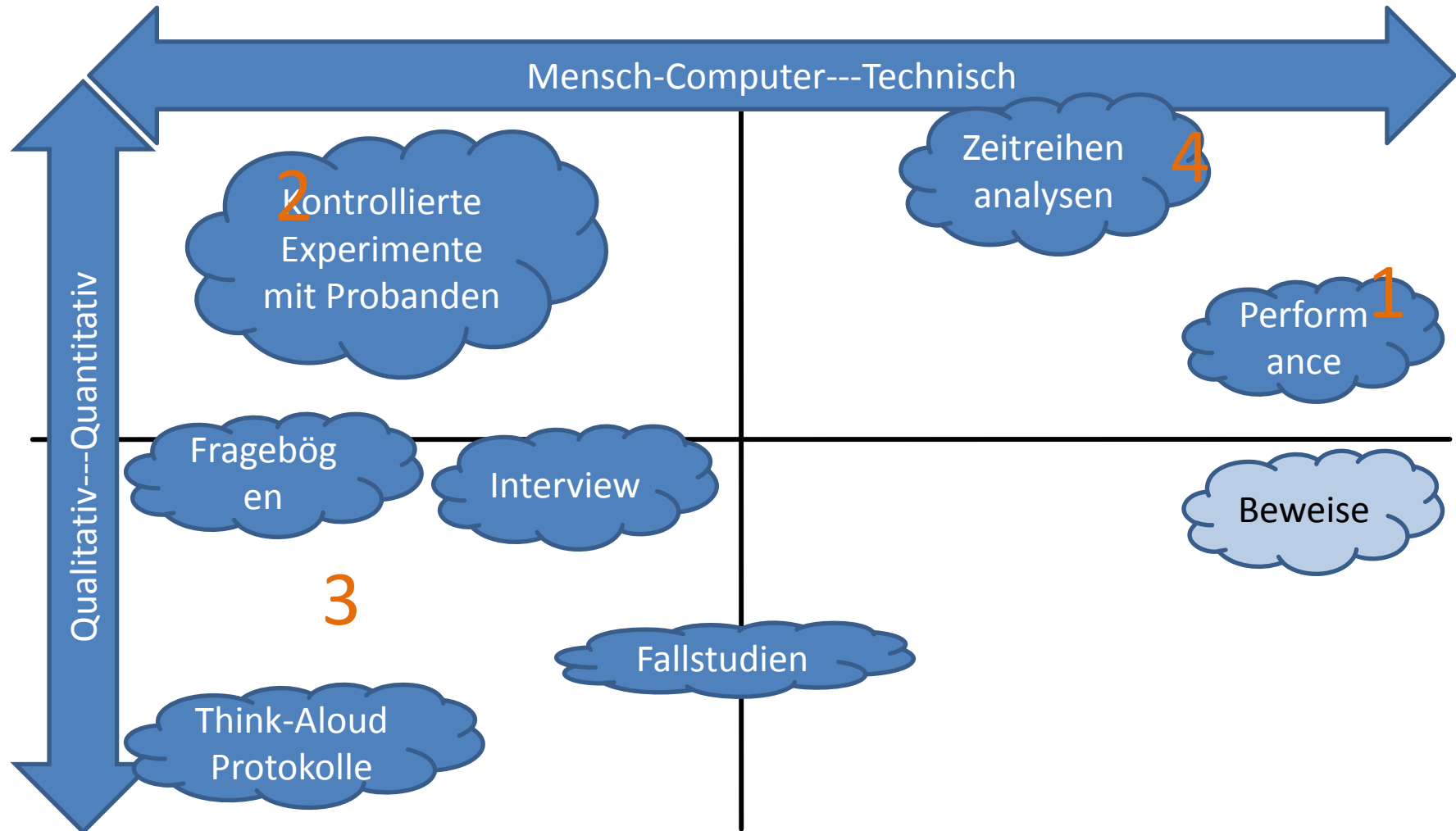


Performance Messungen

Einordnung



Aufgabe

- Bestimmen Sie die schnellste Sortierfunktion
 - Gruppe 1: Mergesort vs. Quicksort
 - Gruppe 2: Quicksort Rekursiv vs. Quicksort Iterativ
 - Gruppe 3: Quicksort Java vs. Quicksort C
 - Gruppe 4: Quicksort C++ vs. Quicksort Haskell
- Stellen Sie die Ergebnisse mit einem Poster vor
- Diskutieren Sie die Ergebnisse. Vertrauen Sie den Ergebnissen der anderen Teilnehmer?

Lernziele

- Aussagekräftige Performance-Analysen selbst durchführen können
- Performance-Analysen bewerten können



Warum Performanceanalyse?

- Alternativen vergleichen
- Einfluss eines Features
- System Tuning
- Relative Performance erkennen (über Zeit)
- Absolute Performance für ausgewählte Fälle
- Erwartungen setzen
- Analyse von Systemverhalten

Analysetechniken

- Messen
 - keine vereinfachenden Annahmen
 - i.d.R. am glaubwürdigsten
 - inflexibel, spezielles System
- Simulation
 - Abstraktion
 - Flexibel
- Analytisches Modellieren
 - Mathematische Beschreibung des Systems
 - Starke Abstraktion, i.d.R. kaum glaubwürdig
 - Insbesondere zur frühen Validierung

Benchmark

- Ausführen realer Programme/Hardwarekomponenten in realen Umgebungen (keine analytische Simulation)
- Messen von Performance, Speicherverbrauch, usw.
- Automatisierbar
- Kein menschlicher Einfluss

Benchmark - Beispiele

- 3DMark (Grafikkarte/System)
- TCP-H (Datawarehouse)
- TCP-C (OLTP)
- Sintel (Video-Encoder)

Was messen?

- Ausführungszeit
- CPU-Zyklen
- MIPS (Million instructions per second)
- MFLOPS (Million floating-point operations per second)
- SPEC (System Performance Evaluation Cooperative)
- QUIPS (Quality improvements per second)
- Transaktionen pro Sekunde

Aufgabe

- Welche Kriterien sollte eine gute Metrik erfüllen?
- Sind die vorgestellten Metriken gute Metriken nach Ihren Kriterien?

Kriterien

Kriterium	Execution Time	CPU Zyklen	MIPS	MFLOPS	SPEC	QUIPS	Transactions/second
Linearität	+	-	-	+	-	+	+
Reliabilität	+	-	-	-	-	-	+
Wiederholbarkeit	+	+	+	+	+	+	+
Einfache Messbarkeit	+	+	+	+	+	+	+
Konsistenz	+	+	-	-	+	+	+
Unabhängigkeit	+	+	+	-	-	+	+

Beispiel für Prüfungsfrage: Welche Metrik(en) würden Sie benutzen, um den schnellsten Sortieralgorithmus zu bestimmen?

Störvariablen

- Beeinflussen das Messergebnis systematisch oder unsystematisch
- Beispiele:
 - Hintergrundprozesse
 - Hardwareunterschiede
 - Temperaturunterschiede
 - Eingabedaten, zufällig?
 - Heap-Size
 - Hardware-Plattform
 - System-Interrupts
 - Parallelität in Single- und Multicore-Systemen
 - Garbage Collection

Aufgabe

- Wie kann man diese Störvariablen kontrollieren?

Typisches Vorgehen: Bester Wert

- Wiederholen
- Bester, zweitbester oder schlechtester Wert
- Bsp: Antwortzeiten für Programmieraufgabe
- R: Daten einlesen
 - `data <- read.csv("rt.csv", header=TRUE, sep = ";", dec = ".")`
 - header: gibt an, ob Variablen/Spaltennamen in der ersten Zeile stehen
 - sep: Separator für Datensätze in der selben Zeile
 - dec: Dezimaltrennzeichen
 - `min(data)/max(data)`

Typisches Vorgehen: Mittelwert

- Messung wiederholen
- Mittelwert bilden

$$\bar{x}_{arithm} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \dots + x_n}{n}$$

- R:
 - mean(data)

Median

- Wert, der in der Mitte liegt
- Robust gegen Ausreißer
- R:
 - `median(data)`
- Bei gerader Anzahl an Messwerten:
 - Arithmetisches Mittel der beiden mittleren Werte
 - Einen der beiden mittleren Werte angeben

Median oder Mittelwert?

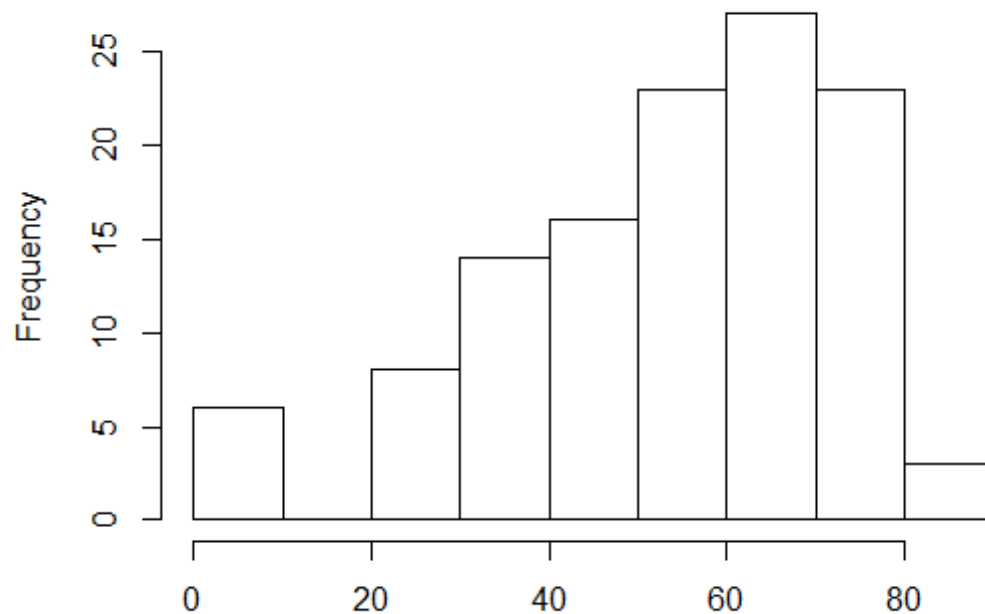
- Median statt arithmetisches Mittel, wenn
 - Ordinale Daten*
 - Wenig Messwerte
 - Asymmetrische Verteilung
 - Ausreißer
- *Skalenniveaus
 - Nominal (z.B. Geschlecht)
 - Ordinal (z.B. Platzierungen)
 - Metrisch (z.B. Temperatur, Antwortzeit)

Daten anschauen

- Überblick verschaffen
- Verteilung und Ausreißer einschätzen

Histogramme

- Häufigkeit von Messwerten in festgelegten Bereichen



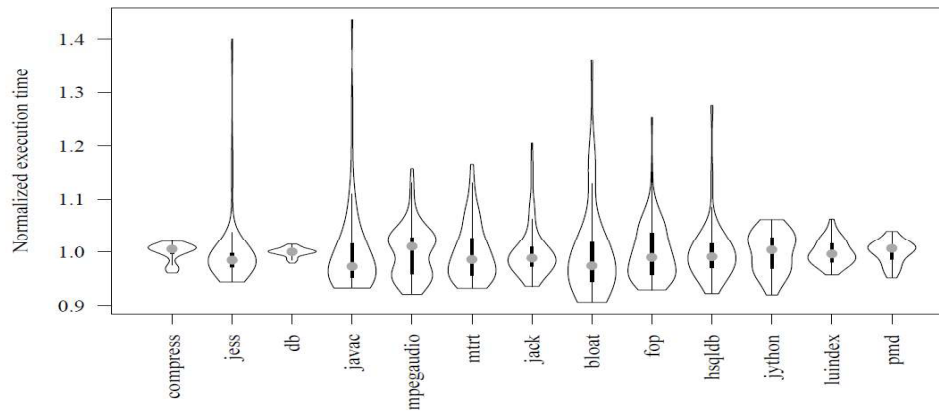
R: `hist(data)`

Boxplots

- Boxplot zeigt
 - Median als breite Linie
 - Quartile als Box (50% aller Werte in der Box)
 - Whiskers
 - Ausreisser als Punkte
- Graphische Darstellung von Verteilungen
- R: `boxplot(data)`

Violin-Plot

- Zeigt zusätzlich zu Boxplot die Verteilung der Daten



R: library(vioplot)
vioplot(data)