Q2. Messung und Benchmarking

Gliederung

1. Lasten in Computersystemen und verteilten Systemen
2. Benchmarks
3. Monitore
4. Repräsentation von Daten
5. Lastprognosen

Literatur (primär)

•R. Jain. The Art of Computer Systems Performance Analysis. Wiley 1991. Teil II (Kap. 4, 7-11)

Weiteres

•P. McKerrow. Performance Measurement of Computer Systems. Addison-Wesley 1987.

•http://www.spec.org

Ziele:

- Festlegung adäquater Lasttypen und –maße
- Methoden der Lastanalyse und –messung kennen lernen
- Problem der Messung in verteilten Systemen einordnen können
- Einige Benchmarks kennen lernen
- Einordnen der Möglichkeiten unterschiedlicher Monitoring-Ansätze
- Verfahren zur Zusammenfassung und Darstellung von Daten kennen lernen
- Grenzen und Chancen von Prognosemethoden erkennen

Q 2.1 Lasten in Computersystemen und verteilten Systemen

Ziele von Messungen

- Bewertung der Leistungsfähigkeit, Zuverlässigkeit eines Systems oder eines Systemteils
- Bestimmung von Parametern als Teil der Modellbildung für die Leistungs-/Zuverlässigkeitsanalyse eines Systems oder eines Systemteils
- Erlangung einer Basis für die Vorhersage über das Verhalten zukünftiger Systeme (d.h. Projektion, Prognose)

immer bezüglich der Analyseziele

⇒ **Analyseziel definiert Maße, diese möglichst direkt (oder indirekt) messen, so dass Messung repräsentativ**

Messung ist auf vielen Ebenen möglich!
Bsp. Rechnernetz

Ende-zu-Ende

- Ladezeit für eine Web-Seite
- RTT für ein Ping-Paket
- Übertragung einer Datei per sftp

Netzintern

- Übertragungs-zeit eines Ethernet-Frames
- Bitfehlerrate einer Verbindung

Zustand

- Auslastung eines Routers
- Verfügbar-keit einer Verbindung

➢Unterschiedliche Größen
➢Unterschiedliche Zeitskalen

} Unterschiedliche Messmethoden

Zentrale Fragen: Wie misst man? Was misst man?

Wie:

- Passiv durch Beobachten
- Aktiv durch induzierte Signale/Pakete etc.
- Durch Software
- Durch Hardware
- Lokal
- Verteilt

Implikationen:

- Aufwand
- Auflösung
- Beeinflussung

Was?

Messung soll *typische Ergebnisse* für die zu untersuchende Situation liefern d.h. die Realität wiederspiegeln

- Messung im realen Betrieb (falls überhaupt möglich)
- Messung unter synthetischer Last (Benchmarks Q 2.2)

Vorteile Messung im realen Betrieb

- Reales Verhalten wird untersucht
- Keine Betriebsunterbrechung, oft nur geringfügige Beeinträchtigung

Vorteile Messung synthetischer Lasten

- Last ist kontrollierbar
- Last ist portierbar (z.B. zum Systemvergleich)

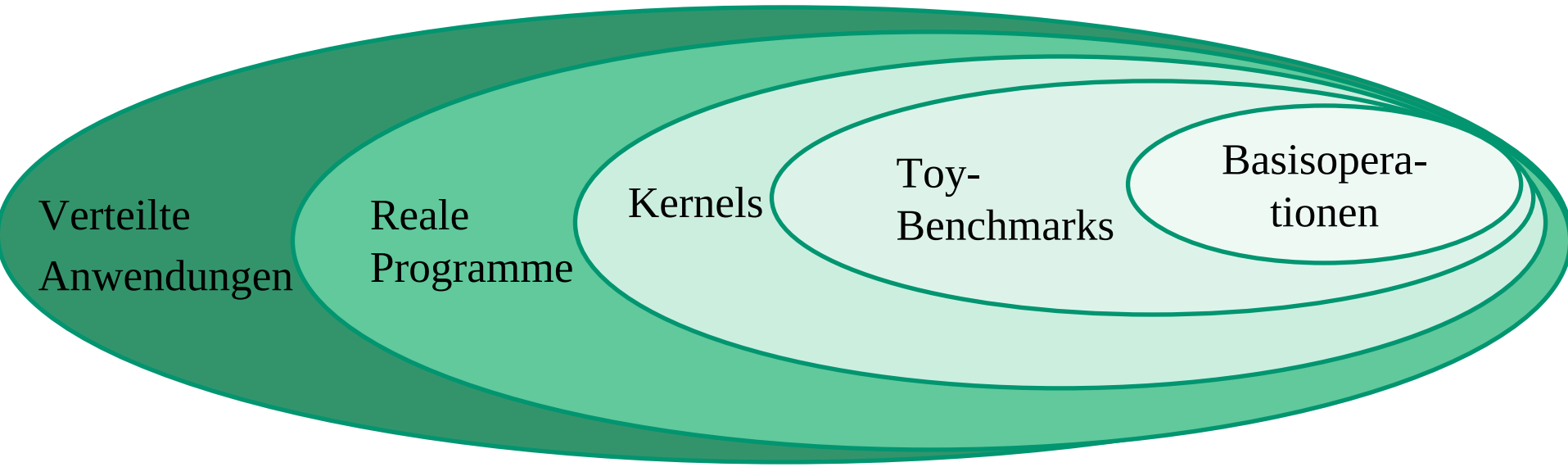
Messung am realen System

- Was soll gemessen werden?
  - I.d.R. Ankunftsrate/-intensität und Größe einer Lasteinheit (siehe SLA)
  - Alle Lasteinheiten oder Stichproben oder nur Lasteinheiten, die spezielle Bedingungen erfüllen
- Wann soll gemessen werden?
  - In Zeiträumen, in denen das System typisches Verhalten zeigt
- Wie werden die Messdaten repräsentiert ?
  - Messwerte i.d.R. statistisch, d.h. statistische Techniken verwenden
  - Modelltypen geben Parameter vor
    (z.B. nur Mittelwerte, nur Phasenverteilungen ,…)
- Wie werden Messdaten validiert?
  - Mehrfach Messen und auf statistische Ähnlichkeit testen
    (siehe auch Vorlesung MAO)

Q 2.2 Benchmarks

- Benchmarks sind synthetische Programme speziell zur Messung von Systemen (Computersystemen, -netzen etc.)

Benchmarkhierarchie

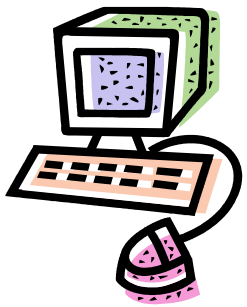


- Benchmarks für Basisoperationen
  Untersuchung der Geschwindigkeit einzelner Operationen wie Addition, Multiplikation, .. (z.B. Dhrystone, Whetstone, …)
- Toy-Benchmarks
  kleine klassische Programme mit geringer praktischer Relevanz
  (z.B. Türme von Hanoi, ….)
- Kernel-Benchmarks
  Teile realer Programme, die besonders viel Zeit verbrauchen
  (z.B. Livermore Loops, Linpack, …)
- Reale Programme
  Komplexe Programme aus vorgegebenen Anwendungsgebieten
  (z.B. TPC-C, SPEC-cpu, SPEC-viewperf,…)
- Verteilte Programme
  Verteilte Anwednungen, oft Web-Server oder verteilte DB
  (z.B. TPC-App, SPEC-web, SPECjAppServer, …)

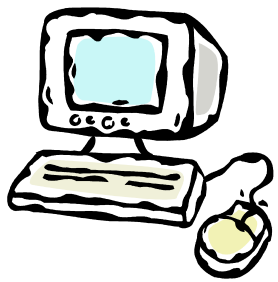
Wesentlicher Einsatzbereich von Benchmarks: Systemvergleich

Sei t(A,P,S) Zeit zur Abarbeitung von Benchmark A mit Parametern P im System S

S



S‘



t(A,P,S) < t(A,P,S‘)

S ist besser als S‘?

Offene Fragen:

- Sind die Unterschiede signifikant?
- Gilt evtl. t(A,P‘,S) > t(A,P‘,S‘) oder t(B,P,S) > t(B,P,S‘)
  Welcher Benchmark/ Parametersatz spiegelt am Besten die reale Last wieder?
- Misst de Benchmark das richtige Leitungsmaß?
  Was ist mit Kosten?

Einfache Benchmarks (Basisoperationen bis Kernels):

- Sind verständlich und nachvollziehbar
- Weisen keine/wenige Parameter auf
- Sind einfach zu installieren und zu messen
- Sind aber oft nicht repräsentativ

Komplexe Benchmarks (reale und verteilte Programme)

- Sind aufwändig zu parametrisieren und zu installieren
- Sind kaum nachvollziehbar
- Erlauben eine Vielzahl unterschiedlicher Messungen
- Können so konfiguriert werden, dass sie repräsentativ sind

Gefahr beim Vergleich mittels etablierter Benchmarks:
Lokale Optimierung durch Hersteller oder Betreiber!

Beispiel TOP 500 Liste

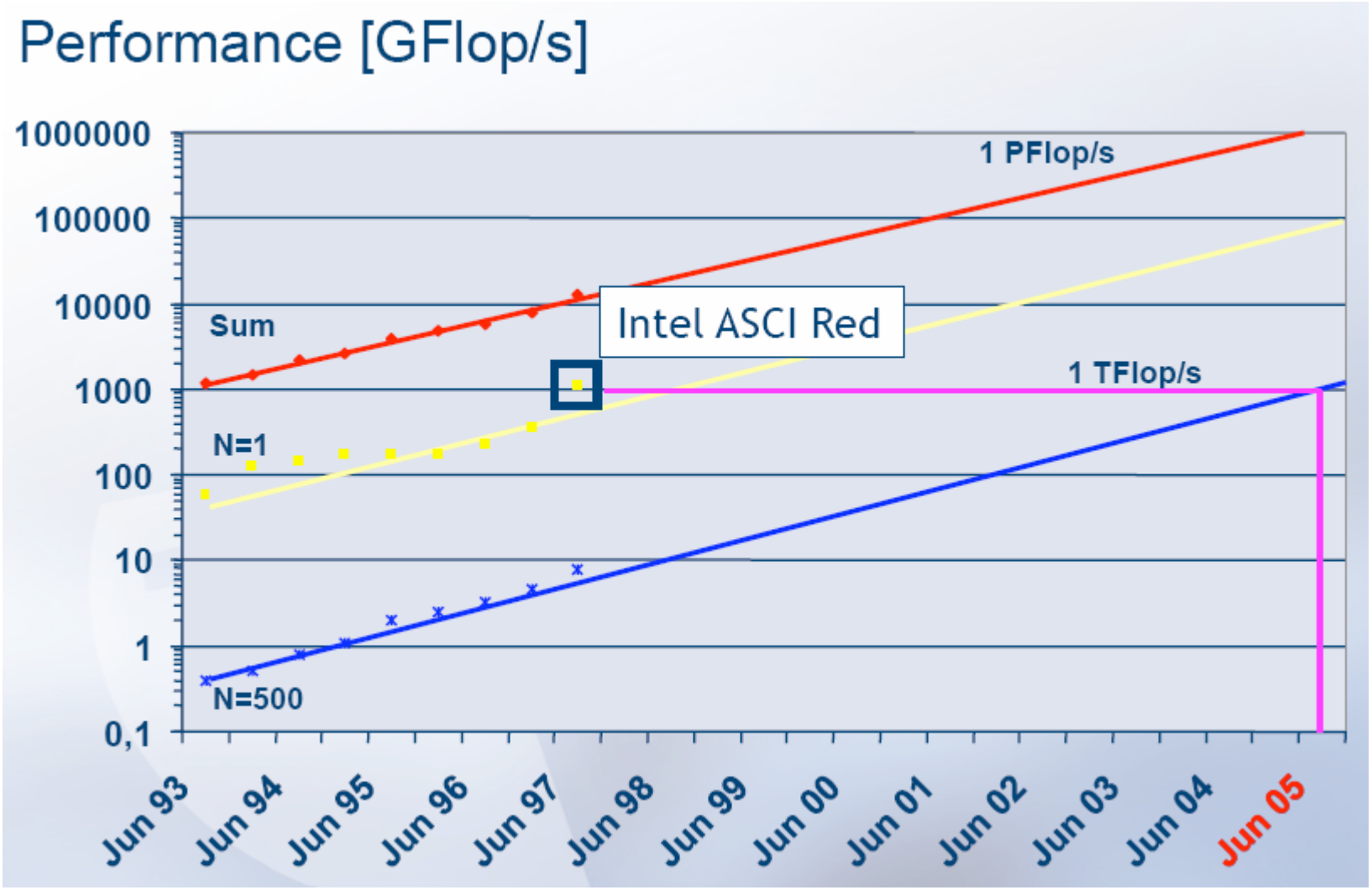
- Liste der 500 schnellsten Supercomputer, seit 1993 halbjährlich veröffentlicht
- Basis: Lösung des Gleichungssystems Ax = b mit einer dichten Matrix mittels der Gauß-Elimination
(LINPACK Programm, relativ altes FORTRAN-Programm)
- Leistungsmaß: R_{max} Linpack Durchsatz im Gegensatz zu R_{peak} theoretischer Maximaldurchsatz auf Grund der Prozessorleistungen
weiterhin relevant N_{max} Größe des Gleichungssystems für das R_{max} erreicht wurde und $N_{1/2}$ Größe des Gleichungssystems für die R_{max} /2 erreicht wird
- Klare Vorgaben, was im Algorithmus geändert werden darf (zur Parallelisierung)

Einfaches Vorgehen, oft kritisiert aber mit großer Wirkung!

Die ersten 10 der aktuellen Liste:

	Manufacturer	Computer	Rmax [TF/s]	Installation Site	Country	Power [MW]	#Cores
1	IBM	Roadrunner BladeCenter QS22/LS21	1026	DOE/NNSA/LANL	USA	2.35	122,400
2	IBM	BlueGene/L eServer Blue Gene Solution	478.2	DOE/NNSA/LLNL	USA	2.33	212,992
3	IBM	Intrepid Blue Gene/P Solution	450.3	DOE/ANL	USA	1.26	163,840
4	Sun	Ranger SunBlade x6420	326	TACC	USA	2.00	62,976
5	Cray	Jaguar Cray XT4 QuadCore	205	DOE/ORNL	USA	1.58	30,976
6	IBM	JUGENE Blue Gene/P Solution	180	Forschungszentrum Juelich (FZJ)	Germany	0.50	65,536
7	SGI	Encanto SGI Altix ICE 8200	133.2	New Mexico Computing Applications Center	USA	0.86	14,336
8	HP	EKA Cluster Platform 3000 BL460c	132.8	Computational Research Laboratories, TATA SONS	India	1.60	14,384
9	IBM	Blue Gene/P Solution	112.5	IDRIS	France	0.32	40,960
10	SGI	SGI Altix ICE 8200EX	106.1	Total Exploration Production	France	0.44	10,240

Entwicklung:



Methodik zeigt die Entwicklung und bildet eine Rangfolge (einfach, verständlich, publizierbar, ..)

Aber erlaubt keine Aussagen über Systemleistung und erst recht keine Systemanalyse

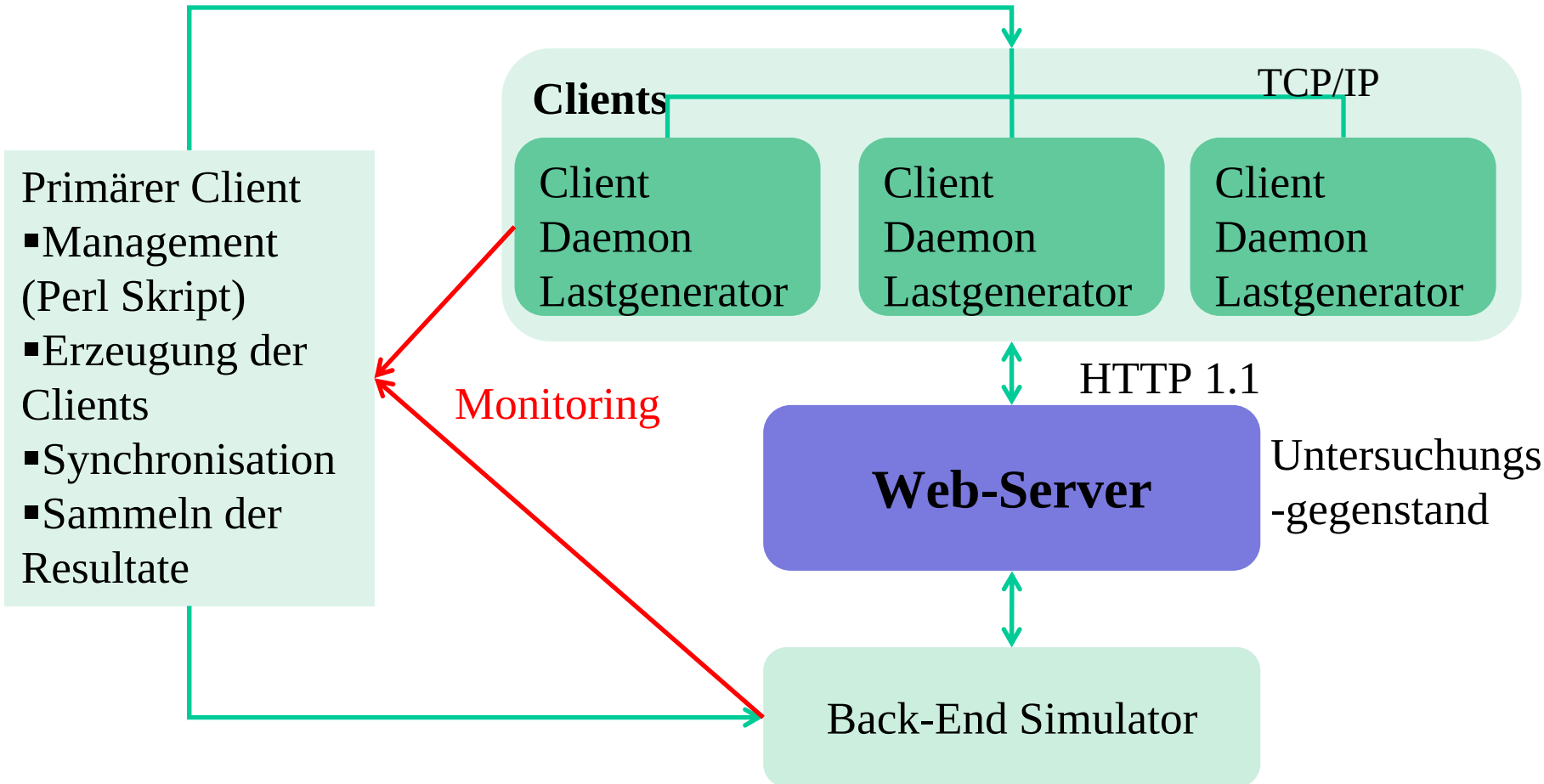
⇒Nicht für die Ziele der Vorlesung geeignet

Hier verwendet

- Einfache Benchmarks auf Komponentenebene
  - Messung der Übertragungsrate durch Paketströme
  - Messung der Zugriffszeit von Platten durch Datenströme (Mix aus Lese- und Schreibanforderungen)
- Komplexe Benchmarks auf Systemebene
  - Für spezielle Anwendungen entwickelt
  - Falls kein Benchmark vorhanden, u.U. eigene Anwendungen verwenden

Beispiel SPECweb (seit 99 aktuelle Version 2005)

Leistungsmessung von Unix und Windows basierten Web-Systemen



Drei Anwendungsszenarien:

- Bankanwendung

Sehr dynamisch, vollständig über HTTPS

- E-Commerce Anwendung

Sehr dynamisch, teilweise HTTP, teilweise HTTPS

- Standard-Anwendung

Relativ statisch, laden von Web-Seiten, ohne SSL

Eigenschaften:

- Caching wird nachgebildet
- PHP und JSP Implementierungen sind enthalten
- Bilder werden über 2 parallele HTTP-Verbindungen geladen

Benutzer kann den Benchmark nicht modifizieren (keine Parameter!)
Konfiguration durch Anzahl parallel laufender Clients

Wie viele Clients können parallel auf den Web-Server zugreifen, so dass eine vorgegebene Dienstqualität erreicht wird ? (= SPECwebNumber)

Dienstqualitäten:

- Bank und E-Commerce: 95% der Antworten in weniger als 2 Sekunden, 99% in weniger als 4 Sekunden
- Normal: Benutzer versucht Seite mit 100.000 Bytes/sek. zu laden, 95% der Seiten müssen mit 99.000 Bytes/Sek. geladen werden, 99% mit 95.000 Bytes/Sek.

Angabe der Leistungsmaße in Relation zum Referenzsystem

$$SPECwebNumber = \sqrt[3]{\frac{EComm\_Numb}{EComm\_Ref} \cdot \frac{Bank\_Numb}{Bank\_Ref} \cdot \frac{Normal\_Numb}{Normal\_Ref}}$$

Vergleichsresultate auf Basis der SPECwevNumber:

Third Quarter 2008 SPECweb2005 Results

These results have been submitted to SPEC; see the disclaimer before studying any results.

Published SPECweb2005 Results (3):

Last publication update: *Thu Aug 28 13:27:36 EDT 2008*

Tester Name	System Name	System Web Server Software	Script	CPU				Result
				Cores	Chips	Cores per chip	HW Multi-threading	
Fujitsu Siemens Computers	PRIMERGY TX150 S6, Intel Xeon processor X3360	Apache 2.2.9, PHP 5.1.6	PHP	4	1	4	No	2641
Fujitsu Siemens Computers	PRIMERGY TX150 S6, Intel Xeon processor X3360	Apache 2.2.3, Apache Tomcat 5.5.20	JSP	4	1	4	No	7945
Hewlett-Packard	HP ProLiant DL585 G5	Rock Web Server v1.4.7 (x86_64), Rock JSP/Servlet Container v1.3.2 (x86_64)	JSP	16	4	4	No	43854

Quelle: www.spec.org

Detaillierte Resultate:

Banking					
Simultaneous User Sessions	Test Iteration	Aggregate QOS Compliance			Validation Errors
		Good	Tolerable	Fail	
2700	1	99.0%	99.9%	0.1%	9
	2	98.4%	99.9%	0.1%	7
	3	98.4%	99.9%	0.1%	1
Ecommerce					
Simultaneous User Sessions	Test Iteration	Aggregate QOS Compliance			Validation Errors
		Good	Tolerable	Fail	
3600	1	99.0%	99.8%	0.2%	0
	2	97.6%	99.4%	0.6%	0
	3	97.3%	99.3%	0.7%	0

Quelle: www.spec.org

Konfigurationsbeschreibung (Ausschnitt):

Configuration

Availability Dates	
SUT Hardware	Mar-2008
Backend Simulator	Mar-2006
Web Server Software	Jun-2008 (HTTP Software) Sep-2007 (Script Engine)
Operating System	Nov-2007
Other Components	Dec-2006 (jdk)

System Under Test (SUT)	
# of SUTs	1
Vendor	Fujitsu Siemens Computers
Model	PRIMERGY TX150 S6, Intel Xeon processor X3360
Processor	Intel Xeon processor X3360
Processor Speed (MHz)	2833
# Processors	4 cores, 1 chip, 4 cores/chip
Primary Cache	32KB(I) + 32KB(D) on chip, per core
Secondary Cache	12 MB on chip, 6 MB shared/2 cores
Other Cache	N/A
Memory	8GB(4x2 GB DDR2 PC2-6400E, 2 rank, CAS 6-6-6)
Disk Subsystem	1x140GB SGT ST3160815AS, 1x80GB ST380013AS, 2x12x146GB ST3146855SS
Disk Controllers	Intel ICH9 4port Serial ATA Storage Controller, Emulex LPe11002,v8.1.10.9(PCIe)
Operating System	Red Hat Enterprise Linux 5.1 (2.6.18-53.el5 x86_64)
File System	ext2
Other Hardware	1xBX300 Blade: 4 Broadcom Switches(BCM5633), D-Link DGS-1224T switch, FibreCAT SX88 (2 enclosures)
Other Software	java version 1.6.0_04 (build 1.6.0_04-b12)

HTTP Software	
Vendor	Apache Source Foundation
Name/Version	Apache 2.2.9
Dynamic Scripts	N/A
Server Cache	N/A
Log Mode	Common Log Format

Script Engine	
Vendor	PHP Net
Name/Version	PHP 5.1.6
Dynamic Scripts	PHP
Server Cache	N/A
Log Mode	Common Log Format

Clients	
# of Clients	20
Model	PRIMERGY BX300 Blade
Processor	Pentium III
Processor Speed (MHz)	933
# Processors	2
Memory	1024 MB
Network Controller	2 Broadcom NetXtreme onboard
Operating System	MS Windows XP Professional(SP1)
JVM Version	Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_10-b03)
JIT Version	Java HotSpot(TM) Client VM (build 1.5.0_10-b03, mixed mode, sharing)
Other Hardware	see SUT -Other Hardware- (Switches)
Other Software	N/A

Zusammenfassung SPEC (und ähnliche Benchmarks):

- Installation, Ausführung und Messung sind aufwändig
- Ergebnisse im jeweiligen Anwendungskontext meistens repräsentativ
- Messergebnisse auch zur Modellparametrisierung nutzbar (dann aber oft detailliertere Messung und Modellierung des Benchmarks notwendig (siehe Q4))

Q 2.3 Monitore

Ein Monitor ist ein Tool zur Beobachtung von Aktivitäten in einem System

Idealerweise sollte ein Beobachter das Systemverhalten beobachten aber in keiner Weise beeinflussen

Man unterscheidet:

- Online Monitore: Messung und Auswertung zur Laufzeit
- Offline Monitore: Datensammlung zur Laufzeit, Auswertung später

Monitortypen:

- Software
- Hardware
- Hybrid

Softwaremonitore

- Software auf Benutzer- oder Systemebene
- Zugriff auf alle Informationen, die Nutzerprogrammen oder dem Betriebssystem zur Verfügung stehen
- Aktivierung:
  - Zeitgesteuertes Auslesen von Systeminformationen alle x Sekunden
  - Ereignisgesteuerte Aktivierung bei Ausführung bestimmter Operationen (z.B. Ein-/Ausgabe)
    (⇒ vom Betriebssystem zu aktivieren)
- Daten können direkt ausgewertet und aggregiert werden
- Aber als Programm mit anderen Programmen in Konkurrenz um Ressourcen

Hardwaremonitore

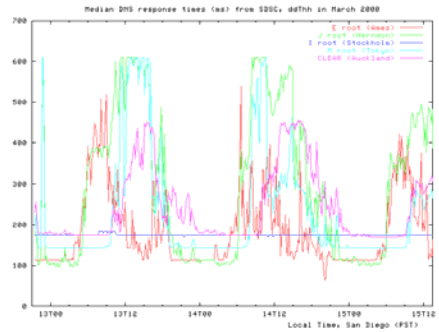
- Eigenständige Hardware zur Messung elektrischer Signale
- Messung festgelegter Signale der Hardware
(z.B. Pakete über ein Ethernet, Signale auf dem Systembus)
dazu müssen Messpunkte existieren
- Aktivierung:
  - i.d.R. Ereignisgesteuerte Aktivierung
- Kein Bezug zur Anwendung
- Als eigenständige Hardware kein zusätzlicher Ressourcenverbrauch auch bei hoher Eingangsrate und Auflösung

Hybride Monitore

- Kombination aus Hardware- und Softwaremonitoren

Netzwerkmonitore

- Grundsätzlich auch wieder Hardware oder Software
- Passive Messungen durch Auslesen von Paketinformationen
  - Netramet: Messung auf Routern (z.B. DNS-Messung)
  - Tcpdump, tcptrace: Ausgabe und Auswertung von TCP-Headern am Interface
- Aktive Messungen durch Injektion von Nachrichten
  - Ping: Messung von RTT
  - Timeit: Zugriffszeiten auf Web-Seiten

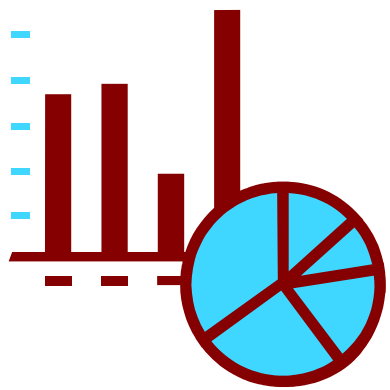


Log-Dateien

- Viele Programme schreiben Log-Dateien oder können so konfiguriert werden, dass sie Log-Dateien schreiben (z.B. Web-Server, DB-Systeme)
  - Einfache Methode, um an Daten zu gelangen
  - Aber eingeschränkter Messbereich (nur aus dem Benutzerprogramm heraus)
  - Zusätzlicher Aufwand durch Dateizugriffe
- Interpretation manuell oder durch spezielle Bearbeitungs- und Auswertungsprogramme

Q 2.4 Repräsentation von Daten

- Zentrale Frage:
  Wie sollen verfügbare Daten dargestellt werden?
- Anforderungen:
  - Repräsentativ
  - Informativ
  - Relevant
  - Kompakt
  - In Modellen/Analysen nutzbar



Darstellungsalternativen:

- Als Konstante
(z.B. durch den Mittelwert repräsentiert)
- Als theoretische Verteilung
  - Diskrete Verteilung
  - Kontinuierliche Verteilung

  in beiden Fällen Verteilungsbestimmung, Parameterschätzung, Anpassungstest
- Als empirische Verteilung
  - Diskrete empirische Verteilung
  - Kontinuierliche empirische Verteilung (per Interpolation)
- Als stochastischer Prozess

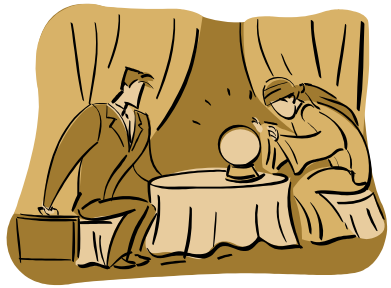
Siehe auch MAO

- Durch Maximal-/Minimalwerte
- Durch Intervalle
- Als vollständiger Trace (u.U. nach Löschen von Ausreißern)

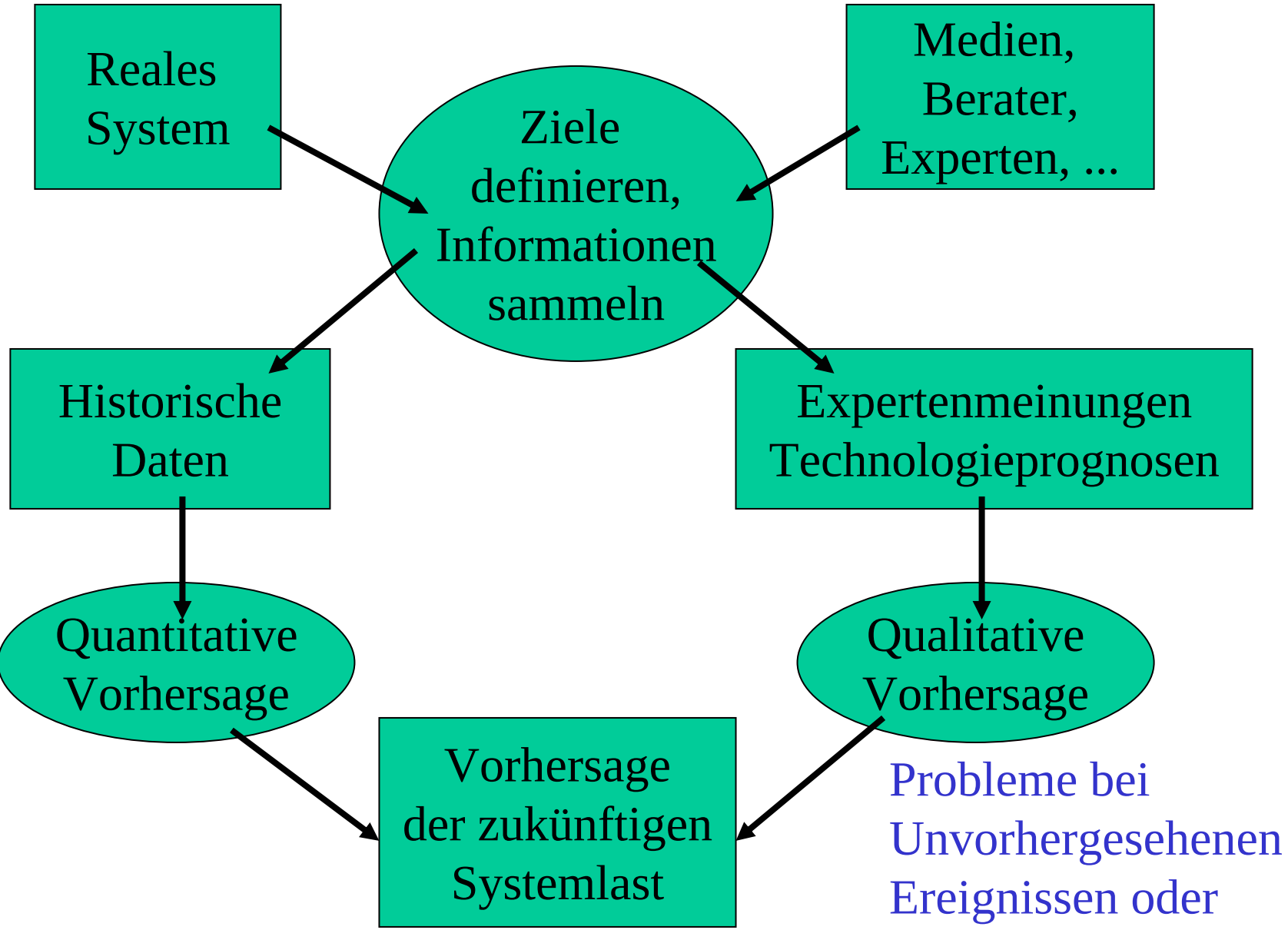
Entscheidung für eine Darstellungsform:

- Ziel der Analyse
(z.B. Schranken, Intervalle bei Realzeitsystemen, stochastische Beschreibung zur Leistungsanalyse)
- Qualität und Umfang der Daten
(z.B. geringer Datenumfang erlaubt keine Anpassung stochastischer Prozesse)

Q 2.4 Lastprognosen



- Modellierung und Analyse für zukünftige Systeme, Konfigurationen etc.
  Wie bestimmt man Parameter für interessante, fiktive Szenarien oder zukünftige Entwicklungen ?
- Man unterscheidet
  - Qualitative Methoden
    subjektive Formulierung von Erwartungen über die Zukunft auf Basis von
    Intuition, Expertenmeinungen, historischen Analogien, technischen Entwicklungen
  - Quantitative Methoden
    objektive mathematische Verfahren auf Basis historischer Daten und Bildung eines Modells

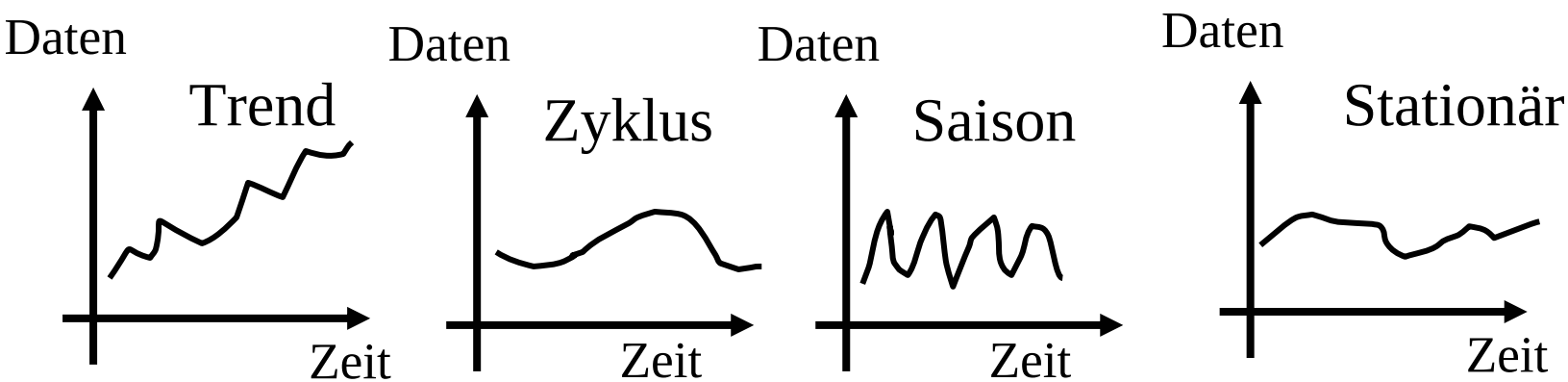


Probleme bei Unvorhergesehenen Ereignissen oder Technologiesprüngen

Patterns, Entwicklungsmuster

- Muster haben starken Einfluß auf die Auswahl und Güte von Prognoseverfahren.

  Es werden 4 Muster unterschieden:
  - Trend: Werte wachsen (oder sinken) über die Gesamtentwicklung hinweg
  - Zyklus: Schwankungen wiederholen sich
  - Saisonale Schwankungen: wie beim Zyklus nur ist die Wiederholungsperiode ist kürzer
  - Stationärität: kein Trend, Mittelwert über die Datenreihe hinweg ist konstant



Regressionsmethoden

- Grundidee: Regressionsmodelle schätzen den Wert einer (abhängigen) Variable als Funktion einer Menge von (unabhängigen) Variable.
- Die funktionale Form der Abhängigkeit wird vorgegeben

Entscheidungsmöglichkeiten

- Welche Variable haben Einfluss auf das Resultat, auf die abhängige Variable, die ich prognostizieren möchte?
- Wie viele unabhängige Variable sollen einfließen ? Einfachster Fall: 1 unabhängige Variable
- Von welchem Typ soll der funktionale Zusammenhang sein ? Einfachster Fall: linear

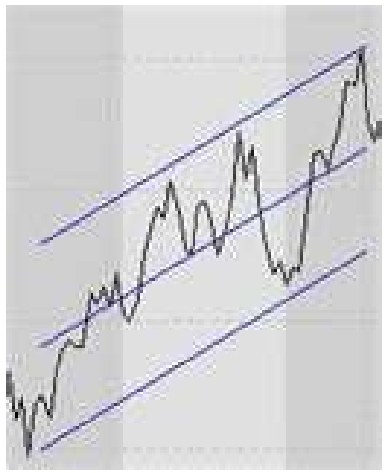
- Lineares Modell
  - unabhängige Variable x, abhängige Variable y
  - Regressionsgerade
  - Bestimmung von a und b mit der Methode der kleinsten (Fehler-)Quadrate

$$b = \frac{\sum_{i=1}^{n} x_i \cdot y_i - n \cdot \bar{x} \cdot \bar{y}}{\sum_{i=1}^{n} x_i^{\,2} - n \cdot (\bar{x}^2)}$$

$$a = \bar{y} - b \cdot \bar{x}$$

x_i, y_i sind reale Daten,

$\bar{x}, \bar{y}$ sind Mittelwerte, $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$



Weitere einfache Ansätze:

Gleitender Durchschnitt

- Die Entwicklung einer Variable wird lediglich über die Zeit aus sich selbst heraus erklärt.
- Schätzwert für den nächsten zukünftigen Wert ist der Mittelwert der letzten n Daten:

$$f_{t+1} = \left(y_t + y_{t-1} + \ldots + y_{t-n+1}\right)/n$$

wobei f_{t+1} prognostizierter Wert für Zeitpunkt t+1
y_t beobachteter Wert zum Zeitpunkt t
n Anzahl Beobachtungen, die Einfluss haben sollen

Exponentielle Glättung

- Grundidee: ähnlich wie bei gleitenden Durchschnitten nur werden Vergangenheitswerte unterschiedlich gewichtet
- Hypothese: neuere Werte sind von größerer Relevanz als ältere Werte

$$f_{t+1} = f_t + \alpha\left(y_t - f_t\right)$$

wobei f_{t+1} prognostizierter Wert für Zeitpunkt t+1
y_t beobachteter Wert zum Zeitpunkt t
α Glättungsfaktor