

1 Universität Hamburg

Fakultät für Mathematik, Informatik und Naturwissenschaften

Fachbereich Informatik

„Projekt: Parallelrechnerevaluation“

(64-194)

im Sommersemester 2013

Hausarbeit

Eingereicht von:

Jürgen Kosemund

Mat.-Nr.: 6356862

E-Mail: j.kosemund@gmx.net

Benjamin Stark

Mat.-Nr.: 6372914

E-Mail: benni@benjamin-stark.com

Daniel Vach

Mat.-Nr.: 6378106

E-Mail: davac@web.de

Tim Hauschild

Mat.-Nr.: 6390912

E-Mail: tim_hauschild@gmx.net

Betreuung:

Dr. Julian Kunkel

Hamburg, 30.10.2013

Inhaltsverzeichnis

1	Einleitung	6
2	Grundlagen der Evaluation	8
2.1	Qualität	8
2.2	Evaluation.....	9
2.3	Monitoring.....	11
3	Aufbau und Gestaltung von IT-Evaluationen	14
3.1	Definition Benchmark	14
3.2	Aufbau und Gestaltung einer Evaluation	15
3.2.1	Phase 1: Planen des Prozesses der Messung	16
3.2.2	Phase 2: Durchführung der Messung	18
3.2.3	Phase 3: Auswertung und Interpretation der Messergebnisse	19
4	Beispielhafte Evaluation des Testclusters	22
4.1	Evaluation des Testclusters mit dem Benchmark „Parabench“	22
4.1.1	Phase 1: Planung der Messung.....	22
4.1.2	Phase 2: Durchführung der Messung	24
4.1.3	Phase 3: Ergänzende Auswertung und Interpretation der Messergebnisse	32
4.2	Evaluation des Testclusters mit dem Benchmark „HPL“	52
4.2.1	Phase 1: Planen des Prozesses der Messung	52
4.2.2	Phase 2: Durchführung der Messung	52
4.2.3	Phase 3: Auswertung und Interpretation der Messergebnisse	54
5	Quellenverzeichnis	55
	Anhang 1: Exkurs Supercomputer.....	56
	Anhang 2: Exkurs Benchmark.....	83
	Anhang 3: Hardwareliste eines Knoten des Testclusters.....	92
	Anhang 4: Exkurs Erste Schritte mit parabench und HPL	109
	Anhang 5: Exkurs Statistische Grundlagen	112
	Anhang 6: Exkurs Evaluationsstandards	130

Abbildungsverzeichnis

Abbildung 1: Ebenen des Monitorings.....	11
Abbildung 2: Aufbau der Datentabelle	34
Abbildung 3: Gefüllter Datenbankbereich.....	35
Abbildung 4: Bestimmen des Datenbankbereichs.....	36
Abbildung 5: Auswahl der darzustellenden Variablen	37
Abbildung 6: Fehler im Datenbankbereich	38
Abbildung 7: Deskriptive Datentabelle für alle Variablen.....	39
Abbildung 8: Bereinigte deskriptive Tabelle	39
Abbildung 9: Ausreißer bei den Rohdaten.....	41
Abbildung 10: Gegenüberstellung ausgewählter Mittelwerte	42
Abbildung 11: Standardabweichungen und Mittelwert von (W)time und (R)time.....	43
Abbildung 12: Standardabweichung und Mittelwert für (W)loops und (R)loops.....	43
Abbildung 13: Standardabweichung und Mittelwert für (W)TP und (R)TP.....	44
Abbildung 14: Variablenbestimmung für Regression.....	45
Abbildung 15: Regression (R)time – (R)TP unbereinigt.....	46
Abbildung 16: Regression (R)time – (R)TP, bereinigt von Ausreißer	47
Abbildung 17: Regression (W)time – (W)TP, bereinigt von Ausreißer	48
Abbildung 18: Regression (R)time – (R)TP, nicht linear.....	49
Abbildung 19: Kontrollmessung Regression (R)time – (R)TP, unbereinigt	50
Abbildung 20: Kontrollmessung Regression (R)time – (R)TP, nicht linear	51
Abbildung 19: Von-Neumann-Architektur	57
Abbildung 20: Intel i7 (links), Apple A5 (mitte), AMD Athlon (rechts).....	58
Abbildung 21: Steuer-, Adress- und Datenbus.....	59
Abbildung 22: Arbeitsspeicher (Random Access Memory)	60
Abbildung 23: Formel für den Zeitgewinn.....	61
Abbildung 24: Strong und Weak Scaling	62
Abbildung 25: Strong und Weak Scaling	62
Abbildung 26: Serielle und Parallele Betriebsart.....	64
Abbildung 27: Maximale Speedup	67
Abbildung 28: Optimaler Speedup.....	67
Abbildung 29: Parallelrechner mit gemeinsamem Speicher (shared memory). Abstraktes Modell (oben) und realer Rechner (unten)	72
Abbildung 30: Parallelrechner mit verteiltem Speicher (distributed memory).....	72
Abbildung 31: Topologien für Netzwerke	74
Abbildung 32: Cluster Architektur wird immer wichtiger	75

<i>Abbildung 33: Grundlegender Aufbau eines kleinen Clusters.....</i>	<i>77</i>
<i>Abbildung 34: Histogramm (Beispiel)</i>	<i>125</i>
<i>Abbildung 35: Histogramm mit Polygonzug (Beispiel)</i>	<i>125</i>
<i>Abbildung 36: Arbeitslosenquote in den USA seit 1980 (als Zeitreihe dargestellt)</i>	<i>128</i>

Tabellenverzeichnis

<i>Tabelle 1: Unterschiede der Monitoring-Ebenen</i>	<i>12</i>
<i>Tabelle 2: Einheiten der Gleitkommarechenleistung mit Vorsätzen nach SI</i>	<i>81</i>
<i>Tabelle 3: Beispiele von nominalskalierten Merkmalen</i>	<i>114</i>
<i>Tabelle 4: Beispiele für ordinal skalierte Merkmale</i>	<i>114</i>
<i>Tabelle 5: Beispiele für intervallskalierte Merkmale.....</i>	<i>115</i>
<i>Tabelle 6: Beispiele für verhältnisskalierte Merkmale und dessen Nullpunkt.....</i>	<i>115</i>
<i>Tabelle 7: Streudiagramme.....</i>	<i>122</i>

2 Einleitung

Bei dem vorliegenden Dokument handelt es sich um eine Abhandlung zum Thema Evaluation von Parallelrechnersystemen. Es ist als Projektarbeit im Rahmen des Projektes „Parallelrechnerevaluation“ entstanden. Zielgruppe des Projektes „Parallelrechnerevaluation“ sind sowohl Studenten der Informatik in den Diplom- und Bachelorstudiengängen, als auch Lehramtsstudenten im Bachelorstudiengang mit Zweitfach Informatik.

Ziel des Projektes „Parallelrechnerevaluation“ ist es, den Studierenden einen ersten Einblick in die Welt der Parallelrechnersysteme zu verschaffen, da diese durch die Entwicklung und zunehmende Verbreitung von Mehrkernprozessoren und dem daraus resultierenden zunehmenden Bedarf an parallelen Programmen und Algorithmen an Bedeutung gewinnen. Das Projekt bietet dabei den Studierenden die Möglichkeiten, anhand eines Entwicklungsclusters unterschiedliche Aspekte eines Parallelrechners zu analysieren und zu evaluieren.

Zielstellung dieses Dokumentes ist es, das Themengebiet Evaluation anschaulich zu erschließen, wobei der Praxisbezug im Vordergrund stehen soll, um zukünftigen Teilnehmern des Projektes „Parallelrechnerevaluation“ einen ersten, fundierten Einstieg in die Materie zu ermöglichen. Die Evaluation von Parallelrechnersystemen ist dabei nicht ausschließlicher Fokus der Arbeit sondern vielmehr das praxisorientierte Beispiel, anhand derer die in dieser Arbeit vorangegangenen theoretischen Hintergründe praktisch erprobt und dargestellt werden.

Die hier vorgestellte Arbeit gliedert sich in drei Teilbereiche, die sich mit dem Thema Monitoring und Evaluation auseinandersetzen. Der Kern der eigentlichen Arbeit beschäftigt sich mit dem Themengebiet der Evaluation. Dabei wird in Kapitel 2 der Begriff Evaluation theoretisch und allgemeingültig hergeleitet. In Kapitel 3 wird der Bogen geschlagen zur Evaluation von Computersystemen. Dabei werden insbesondere die drei Phasen einer Evaluation von Computersystemen sowie die in den einzelnen Phasen erforderlichen Arbeitsschritte näher erläutert. In Kapitel 4 wird anhand eines praktischen Beispiels eine Evaluation eines Testclusters Schritt für Schritt geplant, durchgeführt und ausgewertet.

Um die Arbeit lesbar und interessant zu halten und um sicher zu stellen, dass sie als roter Informationsfaden für die heterogene Zielgruppe des Projektes nutzbar ist, wurde der Kern der Arbeit weitestgehend von theoretischen Hintergründen und wissenschaftlichem Ballast befreit. Dieser befindet sich als Anhang in der Arbeit und kann je nach Interessenslage und Wissensdefiziten der Leser genutzt werden.

Der Anhang besteht im wesentlichen aus in sich schlüssigen Exkursen zu einem speziellen Themengebiet, welches grundsätzlich mit der Evaluation von Supercomputern in Zusammenhang steht.

3 Grundlagen der Evaluation

Der Begriff der Evaluation hängt eng zusammen mit dem Begriff der Qualität. Beide Begriffe sind bisher nicht allgemeingültig definiert und spezifiziert, sondern sind von einem zeitlichem Wandel wissenschaftlicher Erkenntnisgewinnung geprägt sowie dem Kontext ihrer Verwendung unterworfen. Dabei stellt das Bestreben nach Bewertung und Verbesserung von Qualität, die Basis für die vorher notwendige Erfassung von „Daten zur Qualitätsmessung“ dar. Bevor der Zusammenhang der Begriffe Qualität und Evaluation spezifiziert wird, sollen beide Begriffe näher erläutert werden.

3.1 Qualität

Die Qualität eines Gegenstandes oder einer Dienstleistung kann nicht gemeingültig bestimmt werden. Dies liegt im Wesentlichen daran, dass sowohl objektive- als auch subjektive Merkmale die Qualität bestimmen. Objektive Merkmale stellen dabei zum Beispiel die technische Leistung eines Gegenstandes, seine Lebensdauer, seine einwandfreie Funktion und Fehlerfreiheit dar. Die subjektiven Merkmale stellen die Nützlichkeit und Verwendbarkeit des Gegenstandes in den Fokus der Qualitätsbeurteilung. Dabei werden Qualitätskriterien wie etwa Zweckmäßigkeit, Prestigegewinn, Design, Handhabbarkeit, Substituierbarkeit etc. zur Beurteilung herangezogen, wobei die individuelle Bewertung bestimmt wird durch die persönlichen Bedürfnisse des jeweiligen Betrachters.

Qualität bemisst sich aber nicht nur an objektiven und subjektiven Merkmalen, sondern bezieht kulturelle Gegebenheiten und den situativen Kontext in die Bewertung mit ein. So wird der Nutzen einer Axt von einem Schiffsbrüchigen auf einer einsamen Insel abhängig von dem Grad seiner Schärfe bemessen. Ein Holzfäller bemisst den Nutzen der Axt vielleicht in der Relation, der ihm zur Verfügung stehenden Alternativen (Kettensägen). Das Ergebnis wird, auch wenn der Bewertungsgegenstand absolut identisch ist, unterschiedlich ausfallen.

Die Qualität kann dementsprechend nicht allgemeingültig bestimmt werden. Sie bemisst sich vielmehr am generierten Nutzen des jeweiligen Individuums. Die Bewertung des Nutzens ist dabei zumeist subjektiv und hängt im Wesentlichen von den zur Bewertung herangezogenen Kriterien ab, welche je nach situativem Kontext unterschiedlich sein können.

3.2 Evaluation

Der Begriff Evaluation leitet sich aus dem Lateinischen ab (volor = Wert, e/ex = aus) und steht damit für „den Wert aus etwas ziehen“ bzw. der Beurteilung des Wertes eines Objektes.

Umgangssprachlich bedeutet dies, *„Irgendetwas wird von irgendjemandem nach irgendwelchen Kriterien in irgendeiner Weise bewertet.“* (Kromrey (2001): 106). Bei der Evaluation handelt es sich demzufolge um einen Prozess, *„...bei dem nach zuvor festgelegten Zielen und explizit auf den Sachverhalt bezogenen und begründeten Kriterien ein Evaluationsgegenstand bewertet wird“* (Balzer, (2005): 16). Entsprechend der Definition von Balzer setzt Evaluation zum einen ein klar definiertes (zu erreichendes) Ziel voraus, sowie begründete Kriterien, nach denen der Evaluationsgegenstand bewertet wird. Dabei müssen die Kriterien auf den jeweiligen Sachverhalt bezogen sein.

Evaluation, welche im wissenschaftlichen Kontext durchgeführt werden, unterscheidet sich durch folgende Kriterien von alltäglicher Bewertung:

- Informationsgewinnung durch Anwendung empirischer (sozialwissenschaftlicher) Forschungsmethoden zur Informationsgewinnung
- Auf einen klar definierten Gegenstand bezogen
- Von Experten durchgeführt
- systematische Bewertung (vergleichende Verfahren) anhand präzise festgelegter und offengelegter Kriterien (Evaluations-/Bewertungskriterien)
- Systematische Informationsbewertung um den Nutzen (Wert) eines Objekts (z.B. Programm) zu bestimmen.
- Ziel der Evaluation ist es, Entscheidungsprozesse durch Informationen zu unterstützen (Stockmann (2006): 65).

Funktionen von Evaluationen

Evaluationen dienen vor allem der Beschaffung von Informationen zur Entscheidungsfindung in Steuerungs- und Managementprozessen. Im Fokus der Evaluation kann dabei die:

- Erkenntnisfunktion (stellt Entscheidung auf rationale Grundlage)
- Kontrollfunktion (Abgleich der geplanten mit den erreichten Zielen)
- Transparenz- und Dialogfunktion (schafft Basis für gemeinschaftliche Lernprozesse)
- Legitimationsfunktion (Effektivität, Effizienz, Relevanz)

stehen (Stockmann (2006): 66-69).

Eine Evaluation kann durchaus mehrere Funktionen erfüllen. Dabei prägen die zu erfüllenden Funktionen das Design, Herangehensweise und Durchführung der Evaluation (ebd.).

Evaluationen werden je nach ihrem definierten Zweck zu einem bestimmten Zeitpunkt durchgeführt. Ex – ante Evaluierungen zielen dabei auf die Planung einer Intervention ab, d. h. sie wollen die Planung einer Maßnahme verbessern. On – going Evaluationen werden zu bestimmten, vorher definierten Zeitpunkten einer Intervention durchgeführt. Ziel dabei ist es, den Durchführungsprozess zu beobachten um Probleme zu identifizieren und zu beseitigen, die einer Zielerreichung entgegenstehen. Ex- post Evaluierungen finden nach der eigentlichen Intervention statt. Sie sollen Wirksamkeit und Nachhaltigkeit der Intervention bestimmen.

Evaluationen befassen sich dementsprechend mit tiefergehenden Ursache – Wirkungsfragestellungen. Typische Fragestellungen im Rahmen von Evaluationen könnte z. B. die Frage nach Ursachen für stattgefundene Veränderungen sein, oder ob der Erfolg oder Misserfolg einer oder mehrerer Interaktion zuzuordnen sind oder ob Veränderungen von Rahmenbedingungen eines Projektes beobachtbar sind und wenn ja, welche. Evaluation ist dabei nicht ausschließlich auf das Messen des Grades der Zielerreichung ausgerichtet sondern soll auch die Zielstellungen hinsichtlich ihrer realistischen Erreichbarkeit hinterfragen und diese gegebenenfalls anpassen (ebd.).

3.3 Monitoring

Evaluationen lassen sich zu einem kontinuierlichen Monitoring ausbauen. Dabei ist Monitoring definiert als die systematische Dokumentation aller Aspekte über die Leistungen eines Produktes die Auskunft darüber geben, ob es wie beabsichtigt (wie zugesichert) oder entsprechend für das Produkt geltenden Standards funktioniert. Es handelt sich um eine routinemäßige, fortlaufende Tätigkeit, bei der kontinuierlich Daten über ein Produkt bzw. den Nutzen eines Produktes gesammelt, verdichtet und ausgewertet werden, um so Informationen über dessen Nutzen (Input versus Output), den Grad der Zielerreichung und entsprechende Informationen über mögliche Wirkungen zu erlangen (in wie weit trägt das Produkt zur Zielerreichung bei).

Nachfolgende Grafik gibt Auskunft über Arten des Monitoring:

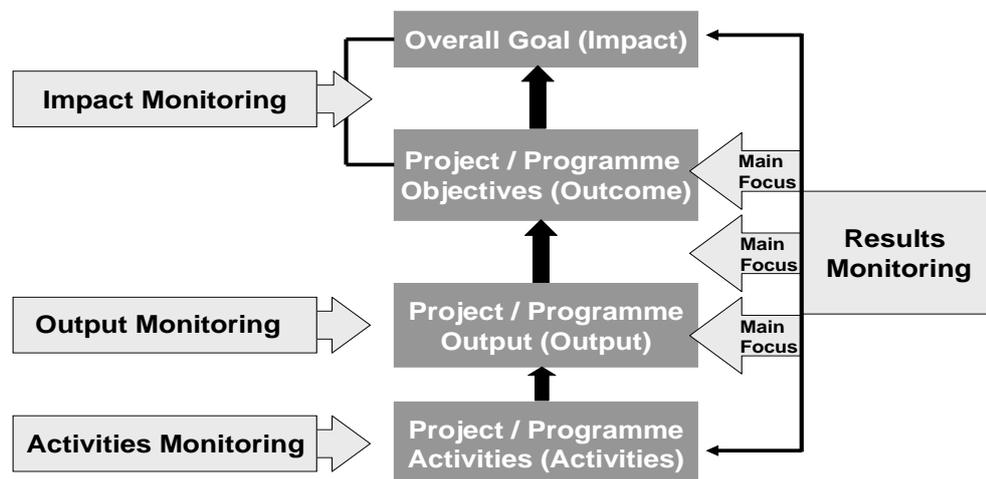


Abbildung 1: Ebenen des Monitorings

Quelle: (Reade (2008): 4)

Anhand der Grafik ist ersichtlich, dass es unterschiedliche Ebenen des Monitorings gibt. Nachfolgende Tabelle stellt die Unterschiede der einzelnen Ebenen des Monitorings an einem Beispiel dar

Tabelle 1: Unterschiede der Monitoring-Ebenen

Ebene	Gemessen wird	Beispiel
Activities	Aktivitäten, die zur Zielerreichung durchgeführt werden	- 4 Mitarbeiterschulungen zur Leitung von Projekten wurden durchgeführt
Output	Leistungen, die durch die Aktivitäten erreicht werden	- Insgesamt 40 Mitarbeiter wurden zu Projektleitern qualifiziert
Outcome	Direkte Wirkungen	- Zahl der erfolgreich durchgeführten Projekte steigt - Zufriedenheitsgrad der Kunden mit der Projektdurchführung nimmt zu - Motivation und Zufriedenheit der Mitarbeiter in den Projekten steigt
Impact	Indirekte Wirkungen	- Positive Auswirkungen auf Umsatz-/Gewinnwachstum - Verringerung der Mitarbeiterfluktuation

Quelle: eigene Darstellung

Nachfolgend werden die Ebenen des Monitorings kurz dargestellt

Activities Monitoring

Beim Monitoring der Aktivitäten handelt es sich um die einfachste Form. Wie der Begriff schon aussagt, geht es im Wesentlichen darum, die Aktivitäten eines Projektes zu messen. Grundgedanke dabei ist ein angenommener Wirkungszusammenhang zwischen den im Vorfeld geplanten und durchgeführten Aktivitäten sowie der Zielerreichung. Der Fokus des Monitorings ist dabei auf die Durchführung der Aktivitäten, nicht aber auf das Erreichen des Ziels gelegt.

Output Monitoring

Diese Form des Monitorings geht einen Schritt weiter als das Activities Monitoring. Auch hier werden die Aktivitäten hinsichtlich ihrer Durchführung gemessen (Activities Monitoring ist dementsprechend integraler Bestandteil des Output Monitorings) beschränkt sich aber nicht allein darauf. Es wird darüber hinaus das Output gemessen, d. h. ob bzw. in wie weit versprochene Leistungen der Aktivitäten auch tatsächlich erfüllt wurden.

Impact Monitoring

Wie aus der Grafik ersichtlich gibt es zwei Ebenen des Impact Monitorings. Die untere Ebene analysiert das Outcome (nicht Output). Die generelle Fragestellung dabei ist, ob

die versprochenen (und auch gelieferten) Leistungen auch die beabsichtigte Wirkung (Nutzen) erzielen. Generell wird dabei impliziert, dass dieser Nutzen positiv sein muss. Aber auch eine gemessene negative Wirkung erzeugt Wissenszuwachs wobei dieser sich meistens darauf reduziert, dass gewählte Aktivitäten oder das Output nicht zu den geplanten/angestrebten Veränderungen geführt haben und dementsprechend geändert werden müssen.

Die obere Ebene des Monitorings soll der Frage nachgehen, in wie weit das Projekt zu einem wie auch immer gearteten übergeordnetem „großem Ganzen“ beigetragen hat. Dabei werden Wirkungsketten aufgestellt anhand derer versucht wird, den Einfluss der Aktivitäten an Veränderungsprozessen für übergeordnete Ziele zu ermitteln.

Zwischenfazit:

Sowohl Evaluation als auch Monitoring sind Prozesse, bei denen Informationen gesammelt, aufbereitet, verdichtet und analysiert werden, um eine fundierte Informationsbasis für Entscheidungsprozesse zu schaffen. Dabei finden Evaluierungen sporadisch, mit Fokus auf die übergeordneten, nicht direkt zuordnungsbaaren Wirkungen und das Hinterfragen der Zielstellung statt. Monitoring ist ein kontinuierlicher Prozess mit Fokus auf den Ablauf, die Zielerreichung und die direkt zuzuordnenden Wirkungen.

4 Aufbau und Gestaltung von IT-Evaluationen

4.1 Definition Benchmark

Bei Computersystemen wird Evaluation als „zielbezogene Beurteilung von Alternativen auf der Grundlage eines Systems von Zielkriterien“ (Heinrich (1995): 101) verstanden. Ziele und demzufolge auch Zielkriterien können dabei unterschiedlich ausgerichtet sein. So können Evaluationen von IT – Systemen darauf ausgerichtet sein, das komplexe Zusammenspiel von unterschiedlichen Hard- und Softwarekomponenten besser zu verstehen um damit mögliche Ansatzpunkte zur Optimierung zu erkennen und zu nutzen. Dabei werden als Evaluationsmethode sogenannte Benchmarks verwendet. Darunter wird nach E. O.Joslin *„a mix (or grouping) of routines to be run on several different computer configurations in order to obtain comparative thrupt performance figures on the capabilities of the various configurations to handle the specific applications“* (zit. n. Heinrich (1995): 88) verstanden. Es werden demnach auf dem zu bewertenden Computersystem Routinen abgearbeitet, welche eine bestimmte Arbeitslast repräsentieren sollen. Dabei wird die Bearbeitungszeit erfasst und empirisch gemessen. Es handelt sich also um die Bewertung des Computersystems bezogen auf tatsächlich vorhandene Routinen.

Eine weiter gefasste und eher an der Zukunft orientierte Definition des Begriffes Benchmark nach W. Buchholz richtet sich nicht an der bisher realisierten Arbeitslast des zu bewertenden Computersystems aus sondern definiert vielmehr ein prognostiziertes, also auf die Zukunft ausgerichtetes Arbeitsspektrum, welches zur Bewertung des Systems herangezogen wird. Dabei handelt es sich um ein Abbild der bisherigen realen Arbeitslast sondern um generierte, synthetische Jobs mit deren Hilfe die Leistungsfähigkeit des Systems ermittelt werden soll. Die synthetischen Jobs werden so definiert, dass sie zum einen Systemunabhängigkeit unter Einbeziehung (Nutzung/Messung) aller Systemteile gewährleisten sollen. Zum anderen sollen sie hochzyklisch sein, wobei sich ihre Laufzeit direkt proportional zur Anzahl der Jobwiederholungen verhalten soll (zit. n. Heinrich (1995): 88 - 89).

In einer weiteren Definition von W. Buchholz wird Arbeitslast und Computersystem als ganzheitliches Bewertungsobjekt betrachtet. Dabei wird die (prognostizierte) Arbeitslast zur empirischen Messung von geeigneten Zielkriterien eines Bewertungsobjektes herangezogen, d. h. die Systemunabhängigkeit als Voraussetzung zur Darstellung der

Arbeitslast wurde aufgegeben, um den Fokus der Bewertung stärker auf das zu messende, individuelle Computersystem und der entsprechenden Arbeitslast zu legen (Heinrich (1995): 89).

4.2 Aufbau und Gestaltung einer Evaluation

Egal in welcher Wissenschaft Evaluationen durchgeführt werden; die Struktur und die Phasen unterscheiden sich nur partiell. Die überwiegend systematische Unterteilung ist dabei vermutlich den Erfahrungen aus den Anfängen der Evaluation geschuldet. Vielfach wurden Daten erhoben, ohne dass sich im Vorfeld der Evaluation Gedanken darüber gemacht wurde, was eigentlich erforscht werden soll und welche Daten geeignet sind, den Forschungsgegenstand näher zu beleuchten und die demnach dazu beitragen, neue Informationen zu gewinnen, um die zu treffenden Entscheidungen auf eine fundierte Informationsbasis stellen.

In der IT werden Evaluationen in drei Phasen unterteilt, welche vorzugsweise sequenziell abgearbeitet werden. Dabei wird unterschieden zwischen:

1. Phase: Planen des Prozesses der Messung

- Festlegung der Messziele
- Bestimmen der Bewertungsobjekte
- Auswahl bzw. Entwicklung von geeigneten Benchmarktests
- Auswahl des Testpersonals
- Auswahl des Testsystems
- Ablaufplanung für die Benchmarktests

2. Phase: Durchführung der Messung

- Abarbeiten der Benchmarks entsprechend der vorher definierten Ablaufplanung
- Messen und beobachten des Prozesses der Messung
- Dokumentation der Messergebnisse, der Messumgebung, des Systemzustandes und der Beobachtungen

3. Phase: Auswertung und Interpretation der Messergebnisse

- Zusammenführung und Verdichtung der Messergebnisse, der Beobachtungen und ggf. der Aufzeichnungen des Beobachtungsgegenstandes
- Analyse der Messergebnisse
- Zusammenfassung der Messergebnisse
- Interpretation der Messergebnisse
- Reporting

Bevor in dieser Arbeit die Evaluation eines Clusters vorgenommen wird, (siehe Kapitel 4) sollen hier generell Aussagen zu den einzelnen Phasen der Evaluation von Computersystemen vorgenommen werden.

4.2.1 Phase 1: Planen des Prozesses der Messung

Die wesentlichen Punkte in dieser Phase ist die Festlegung der eigentlichen Messziele und die, dadurch erforderlich werdende Bestimmung der eigentlichen Bewertungsobjekte. Dem vorgelagert findet im Grunde genommen ein weiterer Schritt statt, bei dem es im Wesentlichen darum geht, den eigentlichen Evaluationszweck und die Evaluationsziele zu bestimmen, da sie erheblichen Einfluss auf Auswahl der Messziele sowie die Bewertungsobjekte, mit denen sich die Messziele realisieren lassen haben. Grundsätzlich sollten Evaluationen immer zielgerichtet sein. Evaluationen nutzen immer auch Ressourcen, welche dem eigentlichen Zweck des evaluierten Systems zum Zeitpunkt der Evaluation entzogen wird. Demzufolge sollte durch Evaluationen auch ein entsprechender Mehrwert generiert werden. Dabei definiert sich dieser Mehrwert nicht über die durch die Evaluation generierte Information sondern vielmehr über den mit ihr einhergehenden Nutzen.

Darüber hinaus ist es sinnvoll, wenn dem Evaluationsteam nicht nur Zweck und Ziel der Evaluation bekannt gemacht wird, sondern auch die Rahmenbedingungen bestimmt werden, wie die gewonnenen Informationen in Wert gesetzt werden sollen. So ist das Design und der Umfang einer Evaluation, welche vorhandene Engpässe eines EDV – Systems identifizieren soll u. a. davon abhängig, wie hoch das für die Beseitigung der Engpässe zur Verfügung stehenden Budget ist.

Ein weiterer, wichtiger Punkt insbesondere für die Ermittlung und Bewertung des Systemverhaltens im Zeitablauf ist das generieren von Base-Line-Daten. Hierbei handelt es sich um Messwerte, die Aussage über das Systemverhalten im Idealzustand geben. Diese Daten können, so sie als Vergleichsbasis von im Zeitverlauf erhobenen Daten herangezogen werden Aussage darüber geben, wie sich das Systemverhalten durch Nutzung im Zeitablauf verändert hat.

Auswahl bzw. Entwicklung von geeigneten Benchmarktests

Zur Evaluation von Computersystemen existieren eine Vielzahl von nicht standardisierten und damit auch hinsichtlich ihrer Messergebnisse nicht vergleichbaren Benchmarktests. Zwar werden in den jeweiligen Tests die gleichen Messgrößen zur Beurteilung des Systems herangezogen, das Zustandekommen der Messwerte, also sowohl die Rahmenbedingungen der Messung als auch die Auswertung sind dabei unterschiedlich. Darüber hinaus können Benchmarktests, angepasst auf die jeweilige Hard- und Softwareumgebung selbst entwickelt werden.

Auswahl des Testpersonals/Testsystems

Evaluationen sind u. a. dadurch gekennzeichnet, dass sie von einem oder mehreren Experten durchgeführt werden. Dabei sollten sich die vorhandenen Kompetenzen sowohl auf den eigentlichen Evaluationsgegenstand (Computersysteme) als auch auf fundierte Methodenkompetenz hinsichtlich des Themas Evaluation erstrecken.

Ablaufplanung der Evaluation

In der empirischen Forschung trägt die systematische Vorgehensweise auf allen Ebenen erheblich zum Gelingen des Forschungsprojektes und damit zum Erlangen von neuen Informationen bei. So können Fehler bei der Auswahl der Stichprobe dazu führen, dass mögliche Aussagen der Untersuchung nicht auf die Grundgesamtheit übertragen werden können (Repräsentativität). Systematische Fehler in der Datenerhebung können das Untersuchungsergebnis wesentlich verändern, wobei damit das eigentliche Erhebungsziel nicht erreicht wird. Demzufolge ist eine detaillierte Ablaufplanung, welche der systematischen Vorgehensweise bei Planung, Durchführung und Auswertung der Evaluation Rechnung trägt und diese auch dokumentiert von erheblicher Bedeutung.

4.2.2 Phase 2: Durchführung der Messung

Nach der Phase der Planung der Evaluation geht es in dieser Phase darum, die eigentliche Datenerfassung durchzuführen. Dabei gilt es auch sicher zu stellen, dass die eigentlichen Messungen valide (wird das gemessen, was gemessen werden soll?) und reliabel (sind die Messergebnisse stabil hinsichtlich Messwiederholungen?) sind.

Abarbeiten der Benchmarks entsprechend der vorher definierten Ablaufplanung

Ein entscheidender Faktor für die Güte von Evaluationen ist die planmäßige Vorgehensweise. Sie trägt zum einen zur Vermeidung von Fehlern bei, zum anderen stellt sie sicher, dass die Evaluation zu einem späteren Zeitpunkt in exakt der gleichen Art und Weise und unter den gleichen Rahmenbedingungen wiederholt werden kann, um so Aussagen über mögliche Entwicklungsprozesse bezüglich des Evaluationsgegenstandes treffen zu können. Eine im Vorfeld der Evaluation entwickelte, detaillierte Ablaufplanung unterstützt die systematische Vorgehensweise allerdings nur dann, wenn dieser Ablauf bei der Durchführung der Datenerhebung auch eingehalten wird.

Messen und beobachten des Prozesses der Messung

Ein wesentlicher Grundsatz, der allen Evaluationen zu Grunde liegt ist der Tatsache geschuldet, dass die Rahmenbedingungen der Datenerhebung wesentlichen Einfluss auf die Güte der Daten haben können. Wie bereits erwähnt, werden Benchmarktests unter vorher definierten Rahmenbedingungen durchgeführt. Dabei reicht es jedoch nicht aus, diese im Vorfeld der Messung herzustellen. Es muss vielmehr im Prozess der Messung sichergestellt werden, dass diese Rahmenbedingungen auch eingehalten werden bzw. müssen gegebenenfalls auftretende Veränderungen der Rahmenbedingungen erfasst und dokumentiert werden. Gegebenenfalls führt eine Änderung der Rahmenbedingungen im Verlauf der Messungen dazu, dass diese wiederholt werden müssen.

Dokumentation der Messergebnisse und der Beobachtungen

Bei Benchmarktests handelt es sich in der Regel um eine Vielzahl von gleichen Operationen, die innerhalb einer vorher definierten Zeiteinheit abgearbeitet werden. Dabei werden die, für jede einzelne Operation verwendete Zeitspanne dokumentiert, die Summe der für die gleichen Operation verwendeten Zeitspannen gebildet und diese durch die Anzahl der durchgeführten Operationen geteilt. Das so berechnete Arithmetische Mittel wird dann als Messwert zur Analyse herangezogen. Dabei können

sowohl einzelne- oder auch eine Vielzahl von Extremwerten den Mittelwert maßgeblich beeinflussen. Dementsprechend ist es erforderlich, nicht nur die sich aus den Messergebnissen ergebenden Mittelwerte sondern auch die einzelnen Messergebnisse (Rohdaten) zu dokumentieren um Extremwerte zu identifizieren und sie in der Analyse der Daten zu berücksichtigen. Gleiches gilt auch für die Beobachtung des Prozesses der Messung. Wenn Rahmenbedingungen für die Messungen definiert wurden, so sollte sowohl das Einhalten dieser, als auch evtl. auftretende Veränderungen entsprechend dokumentiert werden.

4.2.3 Phase 3: Auswertung und Interpretation der Messergebnisse

In der 3. Phase werden die erhobenen Messergebnisse analysiert und in Bezug auf den Evaluationsgegenstand interpretiert. Neben einer ggf. erforderlichen und am Anfang dieser Phase durchzuführende Verdichtung der erhobenen Daten sollte auch deren Güte beurteilt werden, da diese einen wesentlichen Einfluss auf die Interpretation der Messergebnisse hat. Sowohl bei der Verdichtung und Zusammenführung der Daten, als auch bei der Analyse der Messergebnisse bedient man sich statistischer Verfahren sowie geeigneter statistischer Tools oder Programme, die die Auswertung der Daten wesentlich vereinfachen. Wesentliche statistische Grundbegriffe, die für die Auswertung der jeweiligen Messergebnisse in Betracht kommen, werden im Anhang dieser Arbeit näher erläutert (Exkurs Statistische Grundlagen).

Zusammenführung und Verdichtung der Messergebnisse, der Beobachtungen und ggf. der Aufzeichnungen des Beobachtungsgegenstandes

Bei Evaluationen von Computersystemen werden gleiche Operationen in einer vorher definierten Zeitspanne wiederholt ausgeführt, gemessen und die Messergebnisse aufgezeichnet. Häufig wird dann unter sonst gleichen Rahmenbedingungen eine Variable der Operation verändert, um dann die Messung zu wiederholen. Ziel dabei ist es zu überprüfen, welche Auswirkung die geänderte Variable auf das Systemverhalten hat. Handelt es sich bei dem Evaluationsgegenstand um das Computersystem als Ganzes, so wird das Systemverhalten durch die Leistungsfähigkeit der jeweiligen Komponenten bestimmt. Ermittelt wird dieses durch eine Vielzahl von geringfügig veränderten Messungen, welche durch eine, je nach Zeitdauer der einzelnen Messung, unterschiedliche Anzahl von Messergebnissen repräsentiert wird. Demzufolge wird durch die Messung eine Vielzahl von Rohdaten ermittelt, die für eine entsprechende

Analyse zuerst nach schlüssigen Kriterien zusammengefasst und verdichtet werden. Die am häufigsten verwendeten statistische Verfahren zur Datenverdichtung sind die Faktorenanalyse und die Clusteranalyse welche im Anhang dieser Arbeit näher erläutert werden (Exkurs Statistische Grundlagen). Darüber hinaus werden die Messwerte bereinigt. Dieser Schritt umfasst bei der Evaluation von Computersystemen im Wesentlichen die Identifizierung und den Umgang mit Ausreißern (Extremen Messwerten).

Analyse der Messergebnisse

Bei der Analyse der Messwerte werden die Methoden der deskriptiven Statistik auf die Messwerte angewandt. Dabei sollen zum Beispiel mögliche Korrelationen und deren Ausprägung und Richtung aufgedeckt werden. Wesentliche Methoden der deskriptiven Statistik sind Lagemaße, Streuungsmaße und Zusammenhangsmaße (Exkurs Statistische Grundlagen).

Zusammenfassung und Interpretation der Messergebnisse

In der Interpretation der Analyseergebnisse geht es im Wesentlichen darum, die Ergebnisse der statistischen Analyse in den fachlichen Kontext zu bringen. Dabei werden die Zahlenwerte der Analyse in Sprache transferiert, d. h. es werden Ergebnisse der Evaluation formuliert, welche sich auf die Ursprüngliche Fragestellung der Evaluation beziehen.

Vielfach haben die Entwickler des Systems bereits ein Gefühl bzw. eine Vorahnung über die zu erwarteten Ergebnisse, d. h. die Datenerhebung und Auswertung liefert lediglich den Beweis für vorher bereits existierende Vermutungen. Dies ist allerdings auch eine große Gefahr für die Objektivität der Auswertung. Bedingt durch die vorhandenen Vermutungen neigt man bei der Auswertung dazu, die vorhandenen Ergebnisse in die Richtung zu interpretieren, die den Vermutungen entsprechen. Vorhandene Erfahrungen und Insiderwissen werden i. d. R. dazu genutzt, um statistische Abweichungen zu begründen, verhindern jedoch meistens den tatsächlichen Ursachen dieser Abweichung auf den Grund zu gehen.

Reporting

Neben der objektiven Interpretation und Bewertung der Ergebnisse einer Evaluation sollten diese in einer entsprechenden Form dargestellt und präsentiert werden. Sowohl

in dem anzufertigenden Bericht, als auch in der Präsentation sollten dabei nachfolgende Punkte berücksichtigt werden:

- **Vollständigkeit:** Alle relevanten Ergebnisse sollten dokumentiert werden, Stärken und Schwächen angesprochen werden
- **Klarheit und Verständlichkeit:** bei der Sprache und den Präsentationsmethoden
- **Transparenz:** Methoden und Quellen angeben, Bewertungskriterien, Argumentationskette
- **Anpassen an die Zielgruppe:** Präsentation und Kommunikation der Informationen sollten an die Bedürfnisse der Zielgruppe angepasst sein
- **Berücksichtigung der Zeit:** Informationen sollten zu einem notwendigen Zeitpunkt vorliegen
- **Diplomatik:** Probleme und Kritik sollten in einer konstruktiven Art und Weise präsentiert werden (Verbesserungsvorschläge)
- **Anonymität:** Anonymität der Informanten sichern! (Schneider/Meiers (2007): 315-316).

Quellennachweis

1. Balzer, Lars (2005): Wie werden Evaluationsprojekte erfolgreich? - Ein integrierender theoretischer Ansatz und eine empirische Studie zum Evaluationsprozess. Empirische Pädagogik; Auflage: 1., Landau
2. Kromrey, Helmut (2001): Evaluation – ein vielschichtiges Konzept. Begriff und Methodik von Evaluierung und Evaluationsforschung. Empfehlungen für die Praxis.
http://www.bibb.de/dokumente/pdf/a11_vielschichtiges_konzept.pdf
3. Heinrich, Lutz J. et al (1995). Wirtschaftsinformatiklexikon. München, Wien, Oldenbourg: R. Oldenbourg Verlag GmbH, München.
4. Reade, Nicolà (2008): Einführung in Qualitätsmanagement, Monitoring und Evaluation, Osnabrück
5. Schneider, Vera / Meiers, Ralph (2007): Reporting in Handbuch der Evaluation. Hrsg. Reinhard Stockmann. Waxmann Verlag GmbH, Münster
6. Stockmann, Reinhard (2006): Evaluation und Qualitätsentwicklung. Eine Grundlage für wirkungsorientiertes Qualitätsmanagement. Sozialwissenschaftliche Evaluationsforschung Band 5. Waxmann Verlag GmbH, Münster

5 Beispielhafte Evaluation des Testclusters

Im Rahmen dieses Kapitels sollen die in Kapitel 3 dieser Arbeit hergeleiteten Schritte einer professionellen Evaluation in die Praxis umgesetzt werden, d.h. es geht darum, eine qualitativ hochwertige Evaluation des Testclusters vorzunehmen.

5.1 Evaluation des Testclusters mit dem Benchmark „Parabench“

5.1.1 Phase 1: Planung der Messung

Wie in Kapitel 3 bereits beschrieben steht und fällt eine qualitativ hochwertige Evaluation mit der Stringenz seiner Planung. Dabei ist es erforderlich, sowohl das Ziel der Evaluation, als auch dessen Zweck genau zu definieren. Darüber hinaus muss das Evaluationsumfeld (Bewertungsobjekt, Auswahl der Methoden, des Testpersonals etc.) detailliert beschrieben werden. Nachfolgend wird Anhand der in Kapitel 3 beschriebenen Arbeitsschritte die Planungsphase der durchgeführten Testevaluation dargestellt.

Festlegung der Messziele

Zu Beginn haben wir uns darüber Gedanken gemacht, welche Messungen aussagekräftig sind. Aussagekräftige Messungen sind in unserem Fall solche Messungen, die "Ausreißer" und "unkorrekte" Werte hervorbringen.

Dazu musste ein Cluster bestimmt werden, die Anzahl der Knoten sowie die Anzahl an Prozessen. Zudem war es wichtig die CPU-Geschwindigkeit zu bestimmen und die Anzahl der Ein- und Ausgabe-Befehle pro Sekunde festzulegen.

Da es uns schwer fiel, geeignete Werte zu bestimmen haben wir zunächst viel getestet und ausprobiert. Letztendlich haben wir uns darauf geeinigt, dass es sinnvoll wäre, eine Excel-Datei anzulegen, in der wir verschiedene Werte, Anzahl an Prozessen und Knoten gegenüberstellen können.

Des Weiteren führen wir zwei statistische Analysen mit den detaillierten Werten zweier Messungen durch. Diese Messungen führen wir auf jeweils acht Knoten mit zehn Prozessen zu verschiedenen Zeitpunkten zweimal durch.

Anschließend prüfen wir mithilfe statistischer Methoden, ob, wie gewünscht, Diskrepanzen festgestellt werden können.

Begriff	Bedeutung
(W)time	Unter „Time“ lässt sich die Zeit ablesen, in der die Datei geschrieben wird.
(W)loops	Unter „IOops/s“ lässt sich ablesen, wie viele Eingabebefehle und Ausgabebefehle pro Sekunde beim Schreiben der Datei durchgeführt werden.
(W)TP	Unter „TP“ lässt sich ablesen, wie viele Daten pro Sekunde, geschrieben wurden.
(R)time	Unter „Time“ lässt sich die Zeit ablesen, in der die Datei gelesen wird.
(R)loops	Unter „IOops/s“ lässt sich ablesen, wie viele Eingabebefehle und Ausgabebefehle pro Sekunde beim Lesen der Datei durchgeführt werden können.
(R)TP	Unter „TP“ lässt sich ablesen, wie viele Daten pro Sekunde, geschrieben wurden.
(A)time	Unter „Time“ lässt sich die Zeit ablesen, in der die Datei angefügt wird.
(A)loops	Unter „IOops/s“ lässt sich ablesen, wie viele Eingabebefehle und Ausgabebefehle pro Sekunde beim Anfügen der Dateien durchgeführt werden.
(A)TP	Unter „TP“ lässt sich ablesen, wie viele Daten pro Sekunde, angefügt (append) wurden.

Bestimmung der Bewertungsobjekte

Das Bewertungsobjekt ist das Testcluster, das aus verschiedenen Komponenten besteht.

Auswahl bzw. Entwicklung von geeigneten Benchmarktests

Für die Bewertung der Ein- und Ausgabe-Befehle pro Sekunde haben wir uns für den Parabench entschieden. Dieser wurde von Mitarbeitern des WR entwickelt.

Wir haben uns für diese Benchmarks entschieden, da er sich zur Erreichung unseres Ziels sehr gut eignet.

Auswahl des Testpersonals

Bei der Auswahl des reinen Testpersonals haben wir uns gruppenintern darauf geeinigt, wer die Tests in erster Linie durchführt. Sinnvoll ist hierbei unserer Meinung nach, mit mindestens zwei und höchstens drei Personen zu messen. Eventuell ist eine Aufteilung in zwei Gruppen, HPL und Parabench sinnvoll.

Auswahl des Testsystems

Die zu testenden Komponenten sind 10 Knoten aus dem Supercomputer, bei denen wir die CPU-Leistung und die I/O Werte evaluieren. Im Anhang ist ein Knoten beschrieben, der äquivalent zu den anderen 9 ist.

Ablaufplanung für die Benchmarktests

Nachdem wir all die oben genannten Punkte abgearbeitet hatten, mussten wir uns lediglich noch darüber Gedanken machen, wie wir die Benchmarktests letztendlich durchführen wollen.

Nach der Installation von PuTTY (siehe Anhang 4) ist es egal, von wo die Tests durchgeführt werden. Es ist also möglich, sich von jedem Computer einzuloggen und mit den Benchmarks zu arbeiten. Für einen zeitlichen Rahmen haben wir inklusive der Installation und Vorbereitung zwanzig Stunden veranschlagt. Mit in diesen Rahmen eingefasst sind auch schon sehr wahrscheinlich auftretende, unvorhersehbare Probleme.

5.1.2 Phase 2: Durchführung der Messung

Abarbeiten der Benchmarks entsprechend der vorher definierten Ablaufplanung

Um Messungen mit dem Benchmark „Parabench“ durchzuführen, muss man sich zuerst auf dem Testcluster anmelden (Siehe Anhang 4, Schritt 1 - 4).

Schritt 5

Parabench vorinstallieren:

Nun muss Parabench von einem Mitareiter des WR (Wissenschaftliches Rechnen) installiert werden.

Schritt 6

Messung Parabench: Nachdem Parabench von einem Mitarbeiter des WR installiert wurde muss über die Konsole „cd parabench“ eingegeben werden. So gelangt man in den Ordner Parabench. Anschließend „dir“ eingeben und unten erscheint eine Übersicht der vorhandenen Dateien in dem Ordner.

Nun mit einem Editor (Beispielsweise „nano“) ein Jobscript erstellen. Geben Sie „nano“ ein und bestätigen sie mit der Eingabetaste. Es erscheint ein leeres Feld. Hier kopieren

Sie die folgende Vorlage herein. Sie findet sich unter oben genannten Link im Beginners' Guide.

```
Job Managing

A jobscript for MPI applications (mpi.slurm):

#!/bin/sh

# Time limit is one minute.
# See "man sbatch" for other time formats.
#SBATCH --time=1
# Run 10 tasks on 2 nodes.
#SBATCH -N 2 -n 10
# Output goes to job.out, error messages to job.err.
#SBATCH --error=job.err --output=job.out

srun hostname
mpirun ./mpi-application
```

Der rot markierte Bereich zeigt an, wie viele Prozesse auf wie vielen Knoten laufen sollen. In unserem Beispiel laufen 10 Prozesse auf 2 Knoten. Dies kann nach Belieben variiert werden. Anschließend muss der Pfad (grün markierter Bereich) so angepasst werden, dass er zur Pbl-Datei führt, die das eigentliche TestszENARIO enthält. Zu dieser Datei kommen wir im weiteren Verlauf der Dokumentation zurück. Beispiel: „./parabench posixio.pbl“. Die folgende Abbildung zeigt dies als Beispiel

```
vach@cluster: ~/parabench
GNU nano 2.2.6 File: parabench.slurm

#!/bin/sh
# Time limit is one minute.
# See "man sbatch" for other time formats.
#SBATCH --time=1
# Run 10 tasks on 2 nodes.
#SBATCH -N 2 -n 10
# Output goes to job.out, error messages to job.err.
#SBATCH --error=job.err --output=job.out

srun hostname

mpirun ./parabench posixio.pbl -D -e -O details num=1000 env=/mnt/lustre/tmp/test1

cat /proc/cpuinfo
lsb_release -a
lspci
uname -a

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Nun mit „Strg + x“ den Bereich verlassen und mit „y“ speichern und einen Dateinamen vergeben. Beispielsweise „slurm“. Danach geben sie in die Kommandozeile „nano posixio.pbl“ ein, um die Datei im Editor zu editieren.

In den Zeilen der roten Markierung, der nachfolgenden Abbildung, können

- in der ersten Zeile die Anzahl der Dateien, mit denen der Benchmark ausgeführt wird angepasst werden
- und in der zweiten Zeile, in welchem Verzeichnis der Benchmark ausgeführt wird.

In den Zeilen der grünen Markierung kann die Größe der Dateien angepasst werden.

```
vach@cluster: ~/parabench
GNU nano 2.2.6 File: posixio.pbl
**
* POSIX-IO test scenarios for parabench
*/
define param "num" $num "1000"
define param "env" $env "/home/vach/parabench/dat"
$file = "file.dat";
mkdir($env);
#
# POSIX-IO test
#
print "POSIX-IO test START";
dtime["POSIX-IO test"] $i $num {
write("$env/$file-a", 1024);
append("$env/$file-a", 1024);
read("$env/$file-a");
lookup("$env/$file-a");
mkdir("$env/testdir-a");
rmdir("$env/testdir-a");
create("$env/$file-b");
create("$env/$file-c");
delete("$env/$file-c");
stat("$env/$file-a");
}
print "POSIX-IO test STOP";
barrier;
#
# POSIX-IO stresstest (run with -c option)
#
print "POSIX-IO stresstest START";
dtime["POSIX-IO stresstest"] {
dtime["write"] $i $num write("$env/$file-$i", 1024);
dtime["append"] $i $num append("$env/$file-$i", 1024);
dtime["read"] $i $num read("$env/$file-$i");
dtime["lookup"] $i $num lookup("$env/$file-$i");
dtime["mkdir"] $i $num mkdir("$env/dir-$i");
dtime["rmdir"] $i $num rmdir("$env/dir-$i");
dtime["create"] $i $num create("$env/$file-$i");
```

Um die Ansicht zu verlassen „Strg + x“ und „y“ zum Speichern eingeben. Nun können Sie den Benchmark mit dem Kommando „sbatch slurm“ starten.

Messen und beobachten des Prozesses der Messung

Über die Eingabe von „smap“ kann der aktuelle Stand der Messung abgerufen werden. Mit beliebiger Taste kann die Ansicht wieder verlassen werden.

Unter „avgTime“ lässt sich die Durchschnittszeit ablesen, in der die Dateien beigefügt wurden (append). In diesem Beispiel 0.001169s

Unter „minTime“ lässt sich die minimale Zeit ablesen, in der die Dateien gelesen wurden (read). In diesem Beispiel 0.000149s

Unter „maxTime“ lässt sich die maximale Zeit ablesen, in der die Dateien gelesen wurden (read). In diesem Beispiel 0.000312s

Unter „avgTP“ lässt sich ablesen, wie viele Daten pro Sekunde durchschnittlich geschrieben (write), gelesen (read), angefügt (append) werden. In unserem Beispiel: write 1.21 MiB/s

Unter „minTP“ lässt sich ablesen, wie viele Daten pro Sekunde mindestens geschrieben (write), gelesen (read), angefügt (append) werden. In unserem Beispiel: read 6.26 MiB/s

Unter „maxTP“ lässt sich ablesen, wie viele Daten pro Sekunde höchstens geschrieben (write), gelesen (read), angefügt (append) werden. In unserem Beispiel: append 1.77 MiB/s

Unter „IOops/s“ lässt sich ablesen, wie viele Eingabebefehle und Ausgabebefehle pro Sekunde ausgeführt werden können. In unserem Beispiel: write 1239 IOops/s

Wurde der Benchmark nicht ordnungsgemäß durchgeführt, lässt sich über „cat job.err“ ein Fehlerprotokoll aufrufen, mit Hilfe dessen die Fehlerbehebung unterstützt wird.

Schritt 7

Ausgabe der Details in geeignete Grafiken: Um die gewonnenen Daten in ihren Details zu veranschaulichen bietet es sich an, die Daten in eine PDF Datei mit geeigneten Graphen ausgeben zu lassen. Hierzu muss zunächst die „slurm“ Datei angepasst werden. Dies geschieht über die Eingabe von „nano slurm“ in der Befehlszeile. Sie wechseln in die folgende Ansicht.

```
vach@cluster: ~/parabench
GNU nano 2.2.6

#!/bin/sh

# Time limit is one minute.
# See "man sbatch" for other time formats.
#SBATCH --time=1
# Run 1 tasks on 1 nodes.
#SBATCH -N 8 -n 10
# Output goes to job.out, error messages to job.err.
#SBATCH --error=job.err --output=job.out

srun hostname
./parabench posixio.pb1 -D -e -O outputs
```

Hier muss im rot markierten Bereich „-D, -e, -O outputs“ hinzugefügt werden. „outputs“ ist hier variabel, es bezeichnet lediglich den Ordner, in dem die Details gespeichert werden. Dieser Ordner muss im Parabench Ordner extra angelegt werden. Dies geschieht mit folgendem Befehl in der Kommandozeile: „mkdir <Dateiname>“. In unserem Beispiel „outputs“.

Nun verlassen Sie die Ansicht wieder mit „Strg + x“ und „y“ zum Speichern. In die Kommandozeile geben Sie die folgenden Befehlszeilen ein (Info: Kopieren und mit rechter Maustaste einfügen):

CodeI:

```
“./tools/pb2r.py -i posixio.xml -I outputs/posixio/ -g "timeline(id=1003, proc=0, val=Walltime)" -O wallclock-write.pdf”
```

CodeII:

```
“./tools/pb2r.py -i posixio.xml -I outputs/posixio/ -g "boxplot(val=Walltime)" -O boxplot-all.pdf”
```

Erklärung zu der Befehlszeile CodeI:

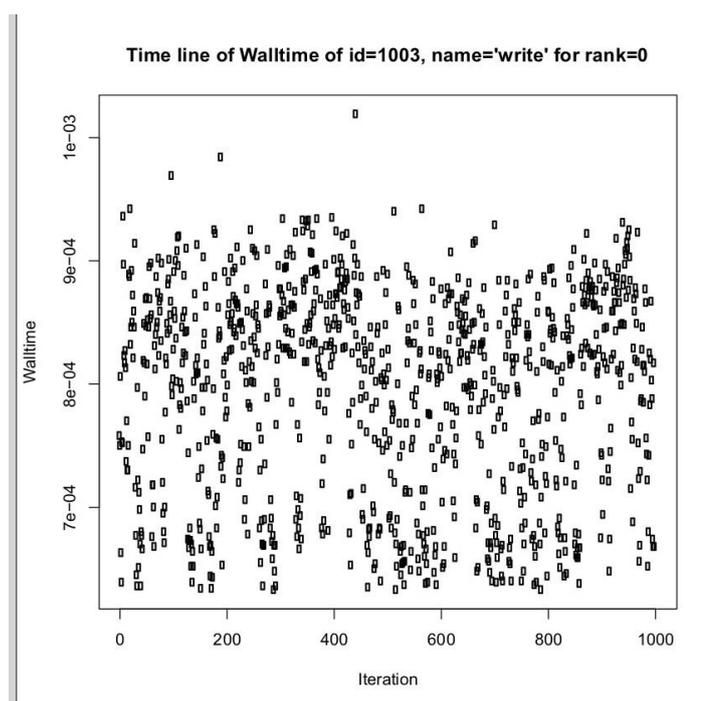
- An der Stelle <outputs> wird angegeben wo die Details gespeichert werden. Es muss also der vorhin erstellte Ordner angegeben werden
- An der Stelle <timeline> wird angegeben, in welcher Form der Graph ausgegeben werden soll. Es gibt verschiedene Ausgabeformen.

- Boxplots
 - Timeline
 - Histogramm
 - Boxscatter
- An der Stelle „id=1003“ wird angegeben, welche Daten ausgegeben werden sollen. Dazu muss man im Benchmark-Ergebnis in der Spalte [#] die entsprechende ID verwenden. In diesem Beispiel: „id=1003“ zeigen die Details des „write“ - Prozesses. „id=2004“ ist „append“.
 - An der Stelle <val=Walltime> lässt sich je nach Wunsch zwischen „avgTime“, „minTime“ und „maxTime“ entscheiden.
 - An der Stelle <wallclock-write.pdf > wird der Name der Datei angegeben, mit dem sie gespeichert wird. Der Name ist frei wählbar.

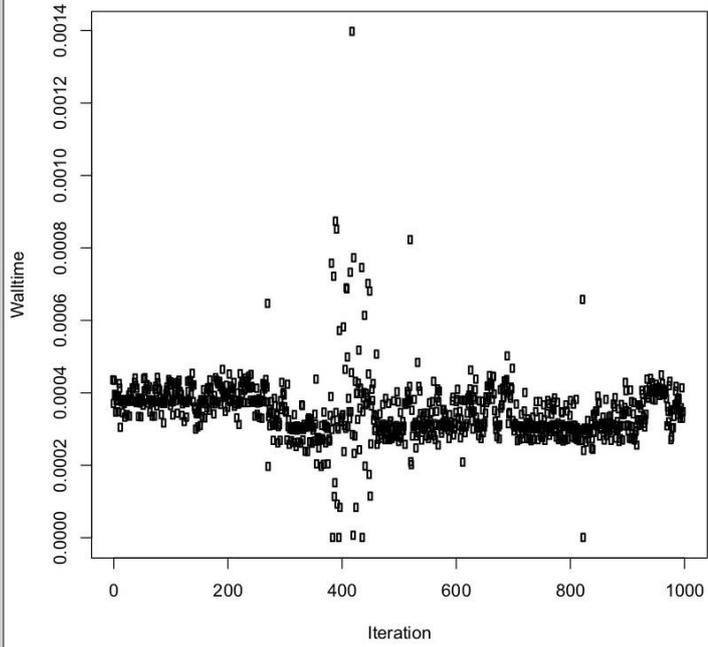
Erklärung zu der Befehlszeile CodeII:

- Ist bis auf eine Ausnahme identisch zu CodeI. Die Ausnahme ist, dass keine ID angegeben ist. Dies hat den Grund, dass die komplette Spalte des Benchmark-Ergebnisses [#] ausgegeben wird. (write, append, read, lookup, mkdir, rmdir, create, stat, delete).

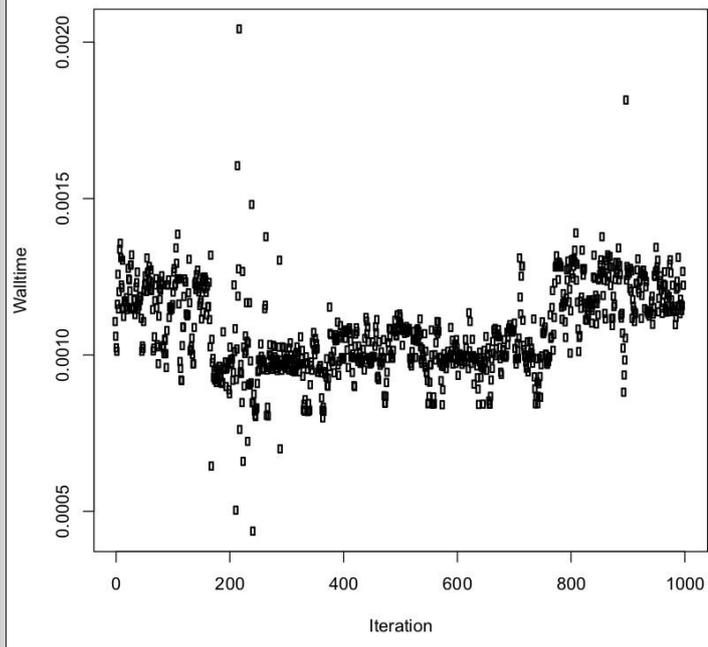
Im Folgenden sind Beispiele zu Code 1 dargestellt:



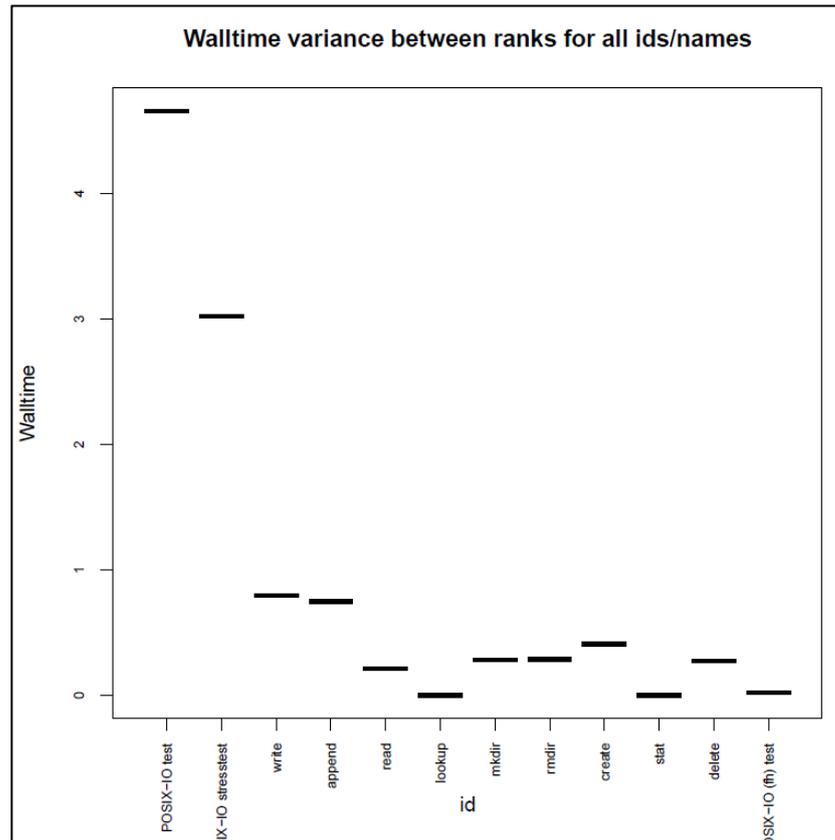
Time line of Walltime of id=3005, name='read' for rank=0



Time line of Walltime of id=2004, name='append' for rank=0



Im Folgenden ist ein Beispiel des CodeII dargestellt.



Weitere Informationen finden Sie unter folgendem Link:

<http://redmine.wr.informatik.uni-hamburg.de/projects/parabench/wiki/PB2R>

Um die erstellten PDF-Dateien, die in dem Ordner Benchmark auf dem Cluster liegen, auf den lokalen Computer zu kopieren, muss man ein FTP-Programm installieren (Bsp.: FileZilla). Dort muss man sich den wieder mit seinem Account an dem Cluster anmelden und kann direkt auf das eigene Homeverzeichnis zugreifen. Dort findet sich unter anderem der Ordner „Parabench“, in dem die erstellen pdf-Dateien liegen.

5.1.3 Phase 3: Ergänzende Auswertung und Interpretation der Messergebnisse

Zusammenführung und Verdichtung der Messergebnisse, der Beobachtungen und ggf. der Aufzeichnungen des Beobachtungsgegenstandes

Die in den vorangegangenen Schritten erhobenen (gemessenen) und mit Parabench ausgewerteten Daten sollen jetzt anhand weiterer statistischer Verfahren analysiert werden. Dabei können andere Tools, die für statistische Auswertungen von Daten geeignet sind verwendet werden. Um ein möglichst breites Spektrum von statistischen Analyseverfah-

ren zur Verfügung zu haben, und um die hier verwendeten Verfahren anschaulich darzustellen, wurde Excel als Basiswerkzeug gewählt. Die Vorteile dieses Werkzeuges liegen dabei auf der Hand. Nahezu jeder Nutzer von Computern ist mit der Verwendung eines Tabellenkalkulationsprogramms vertraut. Auch wenn dabei individuelle Präferenzen vielleicht nicht auf die Microsoft Produktfamilie ausgerichtet ist, so geht der Transfer der individuellen Kenntnisse in der Tabellenkalkulation auf das Produkt Excel relativ schnell und unproblematisch von statten. Für die Aufbereitung und Analyse dient Excel jedoch nur als Basis. Die Funktionalitäten werden durch das statistische Add-In WinSTAT erweitert. Bei WinSTAT handelt es sich um ein Tool, welches statistische Funktionen von Excel bündelt und dabei dem Anwender sämtliche Funktionen unter dem neuen Menüpunkt Add-Ins zur Verfügung stellt. WinSTAT kann als Trial-Version unter <http://www.winstat.de/download/download.htm> heruntergeladen werden. Die Version bietet für 30 Tage den vollen Funktionsumfang. Eine darüber hinausgehende Nutzung ist nur durch Kauf einer Lizenz für ca. 100€ möglich.

Nachfolgend werden die Arbeitsschritte:

- Analyse der Messergebnisse
- Zusammenfassung der Messergebnisse
- Interpretation der Messergebnisse

zusammengefasst. Dies dient im Wesentlichen zur besseren Verständlichkeit einer ziel-führenden Datenauswertung.

Vorbereitung der Datenanalyse und -auswertung

In einem ersten Schritt müssen die von Parabench generierten und gespeicherten Messdaten in eine Form gebracht werden, die mit Hilfe von Excel ausgewertet werden kann. Hierfür wird zuerst das entsprechende Raster der Datentabelle definiert. Eine standardisierte Vorgehensweise ist dabei, dass die einzelnen Spalten jeweils eine der gemessenen Variablen repräsentieren, die einzelnen Zeilen jeweils einem Probanden (einer Messung) entsprechen. Dabei sollte die erste Spalte dazu genutzt werden, die einzelnen Messungen mit einer laufenden Nummer zu versehen. In der hier dargestellten Evaluation sieht die Datentabelle wie folgt aus:

	A	B	C	D	E	F	G	H	I	J
1	ldfNr	(W)time	(W)loops/s	(W)TP	(A)time	(A)loops/s	(A)TP	(R)time	(R)loops/s	(R)TP
2	1									
3	2									
4	3									
5	4									
6	5									
7	6									
8	...									
9	1000									
10										
11										

Abbildung 2: Aufbau der Datentabelle

Die den jeweiligen Spaltenbezeichnungen vorangestellten und in Klammern eingefassten Kürzel (W), (A) und (R) stehen dabei für Write, Append und Read, also für die Prozesse, die gemessen werden sollen.

Das Füllen der Tabelle mit den entsprechenden Datenwerten wird durch einfaches „Copy and Paste“ vorgenommen, d.h. es werden in Parabench die jeweiligen Dateien identifiziert, in denen die einzelnen Messergebnisse abgespeichert wurden. Diese werden geöffnet, die 1000 Datenwerte kopiert und anschließend in die jeweilige Spalte im Excel Arbeitsblatt kopiert. Als nächstes muss gegebenenfalls das Dezimaltrennzeichen in ein für die deutsche Excelversion verständliches Format umgewandelt werden. Parabench verwendet als Dezimaltrennzeichen den Punkt, die deutsche Excelversion das Komma. Hierfür werden die Spalten (W)time, (A)time und (R)time markiert und mit der „Suche – Ersetze“ Funktion von Excel der Dezimalpunkt (.) gesucht und durch das Komma (,) ersetzt. Je nach individuellem Geschmack kann darüber hinaus in den anderen Spalten die 1000-Trennzeichen entfernt werden.

	(W)time	(W)loops/s	(W)TP	(A)time	(A)loops/s	(A)TP
2	0,0008470	868.809.731	1.208.972.845.336	0,001106	904.159.132	925.858.951.175
3	0,0007460	870.322.019	1.372.654.155.496	0,001059	944.287.063	966.949.952.786
4	0,0007050	843.170.320	1.452.482.269.504	0,001022	978.473.581	1.001.956.947.162
5	0,0007060	732.064.422	1.450.424.929.178	0,001011	989.119.683	1.012.858.555.885
6	0,0006820	715.819.613	1.501.466.275.660	0,001161	861.326.443	881.998.277.347
7	0,0007570	708.215.297	1.352.708.058.124	0,001257	795.544.948	814.638.027.049
8	0,0008470	710.227.273	1.208.972.845.336	0,001146	872.600.349	893.542.757.417
9	0,0009060	757.002.271	1.130.242.825.607	0,001201	832.639.467	852.622.814.321
10	0,0007190	727.802.038	1.424.200.278.164	0,001337	747.943.156	765.893.792.072
11	0,0008030	781.250.000	1.275.217.932.752	0,001356	737.463.127	755.162.241.888
12	0,0008800	743.494.424	1.163.636.363.636	0,00131	763.358.779	781.679.389.313
13	0,0008010	797.448.166	1.278.401.997.503	0,001223	817.661.488	837.285.363.859
14	0,0007170	738.552.437	1.428.172.942.817	0,001214	823.723.229	843.492.586.491
15	0,0006690	754.147.813	1.530.642.750.374	0,001302	768.049.155	786.482.334.869
16	0,0006870	801.924.619	1.490.538.573.508	0,00115	869.565.217	890.434.782.609
17	0,0008970	865.800.866	1.141.583.054.627	0,001122	891.265.597	912.655.971.480

Abbildung 3: Gefüllter Datenbankbereich

Zu guter Letzt muss der Datenbereich (also die gesamte Tabelle) nach Fehlern durchsucht werden, die eine statistische Auswertung verhindern würden. Dabei geht es ausschließlich um Anomalien, die sich nicht in extremen Zahlenwerten ausdrücken sondern um die, die eine Fehlermeldung in Textform ausgeben, welche bei den später auszuführenden Analyseverfahren eine entsprechende Fehlermeldung produziert. WinSTAT liefert dabei insbesondere für umfangreichere Datentabelle eine nutzbare Hilfe zur Fehleridentifikation. Auftretende Fehler innerhalb der Datentabelle werden immer mit der genauen Datensatzbezeichnung (also in unserem Fall eine Nummer von 1 bis 1000) und der jeweiligen Spaltenbezeichnung ausgeworfen.

Um WinSTAT zu nutzen muss in einem ersten Schritt der Datenbereich (also der Bereich der Tabelle, welcher die zu analysierenden Daten enthält) festgelegt werden. Hierfür wird der gesamte Bereich der Tabelle inkl. der Spaltenüberschriften aber ohne die Spalte mit den laufenden Nummern markiert. Danach wird der Menüpunkt Add-Ins ausgewählt und dort die Auswahl Datenbankbereich setzen selektiert:

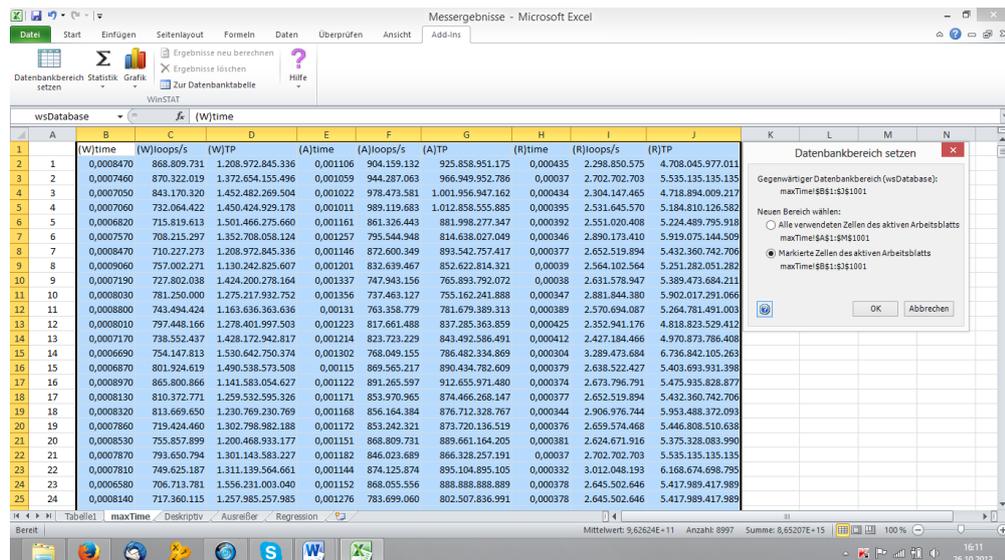


Abbildung 4: Bestimmen des Datenbankbereichs

Im sich öffnenden Kontextmenü wird der Auswahlpunkt <markierte Zellen> des aktiven Arbeitsblattes ausgewählt und mit OK bestätigt.

Sofern die Spaltenüberschriften in der ersten Zeile des markierten Datenbankbereiches stehen und sie nicht aus Zahlen bestehen, werden sie von WinSTAT automatisch als Variablenbezeichnungen benutzt.

Nachdem der Datenbankbereich gesetzt wurde, kann mit der eigentlichen Datenauswertung begonnen werden. Um die einzelnen Arbeitsschritte auch inhaltlich nachvollziehen zu können empfiehlt es sich, parallel zu dieser Anleitung den Exkurs Statistik dieser Hausarbeit zu lesen. Die Auswahl von einzelnen statistischen Verfahren wird hier zwar genannt, aber nicht explizit erklärt.

Anhand der jetzt vorliegenden und vermeintlich korrekten Datentabelle sollen jetzt Verfahren der Statistik angewendet werden, die die vorliegenden Daten näher beschreiben. WinSTAT bietet uns für diesen Bereich die Möglichkeit, eine Vielzahl von unterschiedlichen Parametern (Lage- und Streuungsmaße), die die vorliegenden Daten komprimiert mittels der entsprechender Werte in Tabellenform darstellen. Generell unterscheidet WinSTAT dabei nicht hinsichtlich des Skalenniveaus der vorliegenden Daten, sondern generiert die gesamte Palette der Parameter. Die für die Analyse zu verwendenden Parameter müssen entsprechend ausgewählt werden.

Zur Generierung der deskriptiven Wertetabelle wird im Add-Ins Menü das Untermenü Statistik selektiert und dort der aus dem Menübereich Grundlagen der Bereich deskriptiv ausgewählt.

<Add-Ins> - <Statistik> - <Grundlagen> - <Deskriptiv>

In dem sich öffnenden Fenster lassen sich die jeweiligen Variablen auswählen, die in der Tabelle dargestellt werden sollen. Da es sich in dem hier gewählten Beispiel um eine überschaubare Anzahl von Variablen handelt, können für die Darstellung alle markiert und damit selektiert werden.

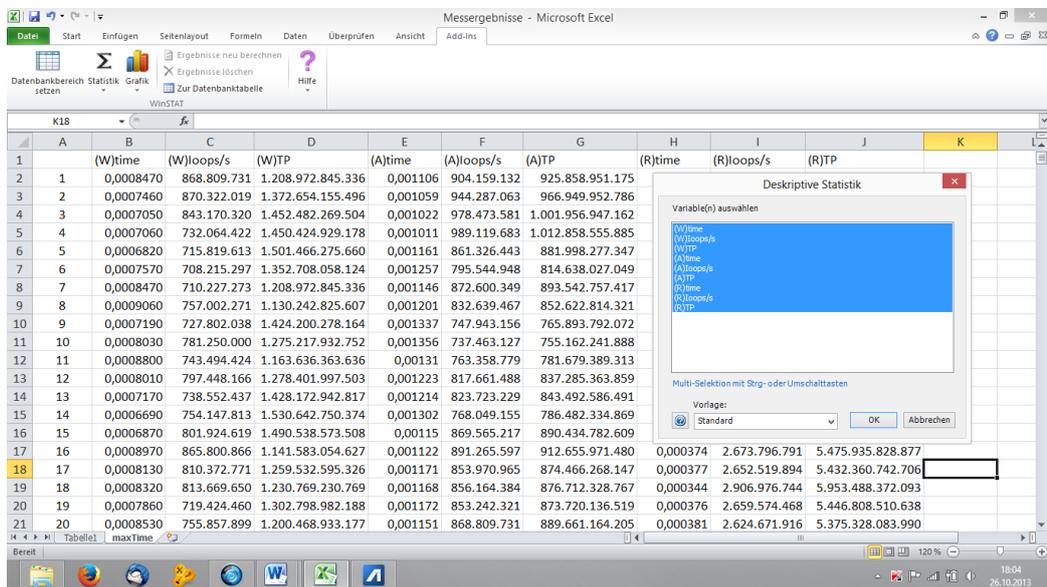


Abbildung 5: Auswahl der darzustellenden Variablen

In diesem konkreten Fall führt die Bestätigung der Auswahl allerdings zu einer Fehlermeldung mit folgendem Inhalt: „Ein nichtnumerischer Wert wurde in Variable '(R)loops/s', Tabellenzeile 387, gefunden.“

Dabei handelt es sich vermutlich um einen Zelleninhalt, der nicht numerisch ist und der damit dazu führt, dass die statistische Analyse bzw. die Verdichtung der Daten nicht ordnungsgemäß durchgeführt werden kann.

	C	D	E	F	G	H	I	J	K
1	(W)loops/s	(W)TP	(A)time	(A)loops/s	(A)TP	(R)time	(R)loops/s	(R)TP	
383	931.966.449	1.262.638.717.633	0,000977	1.023.541.453	1.048.106.448.311	0,000337	2.967.359.050	6.077.151.335.312	
384	934.579.439	1.224.880.382.775	0,00098	1.020.408.163	1.044.897.959.184	0,000389	2.570.694.087	5.264.781.491.003	
385	924.214.418	1.199.063.231.850	0,00098	1.020.408.163	1.044.897.959.184	0,000757	1.321.003.963	2.705.416.116.248	
386	1.008.064.516	1.174.311.926.606	0,001011	989.119.683	1.012.858.555.885	0,000298	3.355.704.698	6.872.483.221.477	
387	967.117.988	1.151.856.017.998	0,000942	1.061.571.125	1.087.048.832.272	0	inf	0.000000	
388	1.111.111.111	1.154.453.213.078	0,001052	950.570.342	973.384.030.418	0,000322	3.105.590.062	6.360.248.447.205	
389	999.000.999	1.236.714.975.845	0,001044	957.854.406	980.842.911.877	0,000721	1.386.962.552	2.840.499.306.519	
390	1.111.111.111	1.365.333.333.333	0,001064	939.849.624	962.406.015.038	0,000113	8.849.557.522	18.123.893.805.310	
391	1.001.001.001	1.416.320.885.201	0,001057	946.073.794	968.779.564.806	0,000151	6.622.516.556	13.562.913.907.285	
392	944.287.063	1.535.232.383.808	0,001046	956.022.945	978.967.495.220	0,000873	1.145.475.372	2.345.933.562.428	
393	755.857.899	1.387.533.875.339	0,001104	905.797.101	927.536.231.884	0,000345	2.898.550.725	5.936.231.884.058	
394	696.864.111	1.372.654.155.496	0,000967	1.034.126.163	1.058.945.191.313	0,000851	1.175.088.132	2.406.580.493.537	
395	1.076.426.265	1.481.910.274.964	0,000934	1.070.663.812	1.096.359.743.041	0,000092	10.869.565.217	22.260.869.565.217	
396	681.198.910	1.560.975.609.756	0,000979	1.021.450.460	1.045.965.270.684	0,000305	3.278.688.525	6.714.754.098.361	
397	937.207.123	1.558.599.695.586	0,00106	943.396.226	966.037.735.849	0,000299	3.344.481.605	6.849.498.327.759	
398	724.112.962	1.448.373.408.769	0,001093	914.913.083	936.870.997.255	0	inf	0.000000	
399	882.612.533	1.354.497.354.497	0,001053	949.667.616	972.459.639.126	0,000571	1.751.313.485	3.586.690.017.513	

Abbildung 6: Fehler im Datenbankbereich

Wenn wir die Datentabelle genauer betrachten, so stellen wir fest, dass dieser Fehler nicht nur einmal, sondern insgesamt viermal auftaucht. Dabei scheinen nicht nur die Datenwerte des (R)loops/s sondern auch die des (R)time und (R)TP des jeweiligen Datensatzes betroffen zu sein. Nun lassen sich diese Daten zwar einfach entfernen, um jedoch statistisch sauber zu arbeiten, müssen geregelte Verfahren zum Umgang mit fehlenden Datenwerten (Missing Values) angewandt werden (siehe Exkurs Statistik)

Im vorliegenden Fall sind von insgesamt 1000 Datensätzen 4 betroffen (0,4%). Sie sind komplett fehlerhaft. Eine komplette Elimination hätte kaum Auswirkungen auf die Analyse und Interpretation der restlichen Datensätze, so dass sie aus dem Datenbankbereich entfernt werden können. Bei einer zweiten Messung muss allerdings darauf geachtet werden, ob diese Form der Fehler erneut auftreten, so dass dahinter eine Systematik vermutet werden kann.

Nachdem die Datenbanktabelle um die fehlerhaften Datensätze bereinigt wurde, soll erneut die deskriptiven Parameter ermittelt werden. die Vorgehensweise wurde bereits beschrieben. Als Ergebnis wird ein zusätzliches Excelsheet mit der Bezeichnung deskriptiv erstellt. Die von WinSTAT erstellte Tabelle gestaltet sich dabei wie folgt:

	(W)time	(W)loops/s	(W)TP	(A)time	(A)loops/s	(A)TP	(R)time	(R)loops/s	(R)TP
Gültige Fälle	1000	1000	1000	1000	1000	1000	996	996	996
Mittelwert	0,000818539	829807109,7	1,2647E+12	0,001071857	948679283,2	9,71448E+11	0,000351467	3142947160	6,43676E+12
Std.Fehler des MW	2,71075E-06	3192543,984	4200228536	4,34127E-06	4075400,225	4173209830	2,61248E-06	166149622,7	3,40274E+11
Varianz	7,34814E-09	1,01923E+16	1,76419E+22	1,88466E-08	1,66089E+16	1,74157E+22	6,79775E-09	2,74953E+19	1,15324E+26
Std.Abweichung	8,57213E-05	100957105,2	1,32823E+11	0,000137283	128875470,9	1,31968E+11	8,24485E-05	5243593647	1,07389E+13
Variationskoeffizient	0,104724769	0,121663341	0,105023202	0,128079582	0,135847249	0,135847249	0,234583947	1,668368375	1,668368375
rel. V.koeffizient(%)	0,331168798	0,384733265	0,332112526	0,405023201	0,429586721	0,429586721	0,743307679	5,286427491	5,286427491
Schiefe	0,474032834	0,368257524	0,401525064	0,521806242	1,935298594	1,935298592	3,405051922	30,55756529	30,55756529
Kurtosis	2,627564544	-0,62459006	0,008975301	3,26063943	16,04184585	16,04184583	32,74714312	950,4634007	950,4634007
Minimum	0,000646	430663221	7,73414E+11	0,000436	489955904	5,01715E+11	0,000006	715819613	1,466E+12
Maximum	0,001324	1165501166	1,58514E+12	0,002041	2293577982	2,34862E+12	0,001397	1,66667E+11	3,41333E+14
Spannweite	0,000678	734837945	8,11725E+11	0,001605	1803622078	1,84691E+12	0,001391	1,65951E+11	3,39867E+14
Summe	0,818539	8,29807E+11	1,2647E+15	1,071857	9,48679E+11	9,71448E+14	0,350061	3,13038E+12	6,41101E+15
1. Perzentil	0,000657	676183185,9	9,94282E+11	0,00080402	744612743,2	7,62483E+11	0,000196	1425346395	2,91911E+12
5. Perzentil	0,000671	706763796,2	1,10108E+12	0,00086605	777604977	7,96267E+11	0,000267	2293577982	4,69725E+12
10. Perzentil	0,000698	719994274	1,14158E+12	0,0009281	793650794	8,12698E+11	0,000274	2375296912	4,86461E+12
25. Perzentil	0,000746	738552437	1,17162E+12	0,00098	857632933	8,78216E+11	0,000305	2585652890	5,29542E+12
Median	0,000837	829875519	1,22342E+12	0,001052	950570342	9,73384E+11	0,000345	2898550725	5,93623E+12
75. Perzentil	0,000874	899280576	1,37265E+12	0,001166	1020408163	1,0449E+12	0,00038675	3278688525	6,71475E+12
90. Perzentil	0,000897	978473581	1,46705E+12	0,00126	1077470213	1,10333E+12	0,000421	3649635036	7,47445E+12
95. Perzentil	0,00093	999000999	1,52608E+12	0,001286	1154667817	1,18238E+12	0,000436	3745318352	7,67041E+12
99. Perzentil	0,00102989	1054830114	1,5586E+12	0,00134298	1243750232	1,2736E+12	0,0007016	5102040816	1,0449E+13
geom. Mittelwert	0,000814103	823770461,5	1,25783E+12	0,001063097	940647914,8	9,63223E+11	0,000342521	2919525432	5,97919E+12

Abbildung 7: Deskriptive Datentabelle für alle Variablen

Um die generierte Tabelle auf den Evaluationsgegenstand anzupassen, werden für die Analyse und Interpretation nicht benötigte Datenwerte eliminiert werden. Dies fördert zum einen die Übersichtlichkeit, zum anderen können für den Report und die Präsentation evtl. erforderliche Grafiken besser generiert werden. Für die Darstellung und Weiterverarbeitung kann auf folgende Daten verzichtet werden:

Geom. Mittelwert, 1. Perzentil, 10. Perzentil, 90. Perzentil, 99. Perzentil, Summe, Kurtosis, Schiefe, rel. V.koeffizient(%) und Std.Fehler des MW. Teile der Parameter werden im Exkurs Statistik nicht beschrieben. Bei Interesse lassen sich Erklärungen der Begriffe im WWW. finden.

Die bereinigte deskriptive Tabelle sieht danach wie folgt aus:

	(W)time	(W)loops/s	(W)TP	(A)time	(A)loops/s	(A)TP	(R)time	(R)loops/s	(R)TP
Gültige Fälle	1000	1000	1000	1000	1000	1000	996	996	996
Mittelwert	0,000818539	829807109,7	1,2647E+12	0,001071857	948679283,2	9,71448E+11	0,000351467	3142947160	6,43676E+12
Varianz	7,34814E-09	1,01923E+16	1,76419E+22	1,88466E-08	1,66089E+16	1,74157E+22	6,79775E-09	2,74953E+19	1,15324E+26
Std.Abweichung	8,57213E-05	100957105,2	1,32823E+11	0,000137283	128875470,9	1,31968E+11	8,24485E-05	5243593647	1,07389E+13
Variationskoeffizient	0,104724769	0,121663341	0,105023202	0,128079582	0,135847249	0,135847249	0,234583947	1,668368375	1,668368375
Minimum	0,000646	430663221	7,73414E+11	0,000436	489955904	5,01715E+11	0,000006	715819613	1,466E+12
Maximum	0,001324	1165501166	1,58514E+12	0,002041	2293577982	2,34862E+12	0,001397	1,66667E+11	3,41333E+14
Spannweite	0,000678	734837945	8,11725E+11	0,001605	1803622078	1,84691E+12	0,001391	1,65951E+11	3,39867E+14
5. Perzentil	0,000671	706763796,2	1,10108E+12	0,00086605	777604977	7,96267E+11	0,000267	2293577982	4,69725E+12
25. Perzentil	0,000746	738552437	1,17162E+12	0,00098	857632933	8,78216E+11	0,000305	2585652890	5,29542E+12
Median	0,000837	829875519	1,22342E+12	0,001052	950570342	9,73384E+11	0,000345	2898550725	5,93623E+12
75. Perzentil	0,000874	899280576	1,37265E+12	0,001166	1020408163	1,0449E+12	0,00038675	3278688525	6,71475E+12
95. Perzentil	0,00093	999000999	1,52608E+12	0,001286	1154667817	1,18238E+12	0,000436	3745318352	7,67041E+12

Abbildung 8: Bereinigte deskriptive Tabelle

Interpretation

Da wir ja schon bei der Datentabelle im Rahmen der Read Daten fehlerhafte Datenwerte feststellen mussten, wird zu aller erst auf die deskriptiven Parameter aus diesem Bereich geachtet ((R)time, (R)loops/s und (R)TP). Auffällig ist dabei die extreme Spannweite bei den Messergebnissen der Read – Operationen. Dies lässt auf Extremwerte sowohl im unteren, als auch im oberen Bereich der Datenwerte schließen. Dies wird bestätigt durch die jeweiligen Minimum- und Maximumwerte.

Bevor die Werte weiter interpretiert werden können, müssen mögliche Ausreißer identifiziert werden. Dabei handelt es sich um Datenwerte, die so weit vom Mittelwert entfernt liegen, dass zu vermuten ist, dass sie nicht repräsentativ für die Grundgesamtheit sind (Eine genauere Beschreibung zur Bestimmung und Definition von Ausreißern liefert die WinSTAT Hilfefunktion). Dadurch können die Parameter der deskriptiven Tabelle nicht nur beschrieben, sondern auch hinsichtlich ihrer Aussagekraft eingeschätzt werden.

WinSTAT bietet hierfür eine eigene Funktion an. Unter

<Add-Ins> - <Statistik> - <Grundlagen> - <Ausreißer>

lässt sich eine Tabelle erstellen, die innerhalb bestimmter Grenzen (vom Nutzer selbst zu definieren), Ausreißer in der Datentabelle identifiziert und in einer Tabelle mit genauen Tabellenbezügen ausgibt.

Variable	Zeile	Wert	n*Sigma	P < 0.05
Wtime	254	0,001173	4,1350396	0,03297605
Wtime	257	0,001251	5,044965	0,000384834
Wtime	258	0,001213	4,601668	0,003742319
Wtime	259	0,001212	4,5900023	0,003962277
Wtime	260	0,001234	4,8466479	0,001091866
Wtime	261	0,001324	5,8965619	2,72396E-06
Atime	214	0,000503	4,1436814	0,031737233
Atime	220	0,002041	7,059454	8,78822E-10
Atime	244	0,000436	4,6317244	0,003226152
Atime	900	0,001814	5,4053353	5,18673E-05
Aloops/s	171	1552735031	4,6875929	0,002446639
Aloops/s	214	1988071571	8,0650901	2,41772E-13
Aloops/s	227	1517450683	4,413341	0,00925033
Aloops/s	244	2293577982	10,435645	6,90909E-24
AJTP	171	1,5901E+12	4,6875929	0,002446639
AJTP	214	2,0358E+12	8,0650901	2,41772E-13
AJTP	227	1,5533E+12	4,413341	0,00925033
AJTP	244	2,3486E+12	10,435645	6,90909E-24
Rtime	385	0,000757	4,3186244	0,000747915
Rtime	389	0,000721	4,4819882	0,006649741
Rtime	392	0,000873	6,3255636	1,66259E-07
Rtime	394	0,000851	6,0587303	9,66069E-07
Rtime	411	0,000689	4,093867	0,039365263
Rtime	412	0,000686	4,0574807	0,04614477
Rtime	418	0,000732	4,6154048	0,003482662
Rtime	421	0,001397	12,681047	5,05563E-37
Rtime	423	0,000006	4,1900936	0,025693291
Rtime	424	0,000772	5,1005562	0,000283837
Rtime	438	0,000745	4,773079	0,001583738
Rtime	449	0,000701	4,2394124	0,020572
Rtime	523	0,000822	5,7069955	8,72467E-06
Rloops/s	423	1,6667E+11	31,185429	0
RJTP	423	3,4133E+14	31,185429	0

Abbildung 9: Ausreißer bei den Rohdaten

Interessant bei dieser Tabelle ist der P-Wert. Er gibt die Wahrscheinlichkeit an, bei einer normalverteilten Stichprobe mindestens einen Wert in diesem Abstand zum Mittelwert zu finden. Die im Kopf der Tabelle mögliche Variation der P-Wert-Grenze ermöglicht demnach, die Wahrscheinlichkeitsgrenze für eine normalverteilte Stichprobe zu variieren.

Interpretation

Über die insgesamt 8988 gültigen Datenwerte in der Datentabelle erfüllen insgesamt 33 Werte das Kriterium, um als Ausreißer zu gelten. Dabei liegt die Wahrscheinlichkeit, dass sie in einer normalverteilten Stichprobe in dieser Form wieder auftreten unter 5%. Generell kann davon ausgegangen werden, dass ihr Einfluss auf die statistische Auswertung vernachlässigbar ist. Jedoch ist auch hier die Häufigkeit der Extremwerte bei den Messwerten für das Read auffällig. Dabei ist die Wahrscheinlichkeit ihres Auftretens in einer normalverteilten Funktion (P-Wert) extrem gering.

Diese Interpretation ließe sich jetzt noch überprüfen, indem die Extremwerte aus der Datentabelle ausgeblendet werden, um danach erneut eine deskriptive Analyse durchzuführen. Dabei sollten die ermittelten Werte nur unwesentlich von denen mit enthaltenen Extremwerten abweichen (Ausgenommen die ermittelten Werte des Minimums und des Maximums).

In einem nächsten Schritt sollen die Mittelwerte einer genaueren Überprüfung unterzogen werden. Dafür stellen wir die Mittelwerte aus den drei Datenreihen der Operation Write denen der Operation Read gegenüber.

In WinSTAT müssen hierfür die Variablen bestimmt werden, für die eine Mittelwertanalyse durchgeführt werden soll.

WinSTAT bietet hierfür eine eigene Funktion an. Unter

<Add-Ins> - <Statistik> - <Grundlagen> - <Mittelwerte>

Im sich öffnenden Kontextmenü muss dann bei 1-n Variablen (W)time und (R)time markiert werden. Es sollte aber immer darauf geachtet werden, dass die Wertedimensionen der Variablen in einem aussagekräftigen Verhältnis stehen. Es macht z.B. wenig Sinn, zwei Variablen zu vergleichen, bei denen die Werte der einen in Millisekunden, die der anderen in Stunden vorliegen.

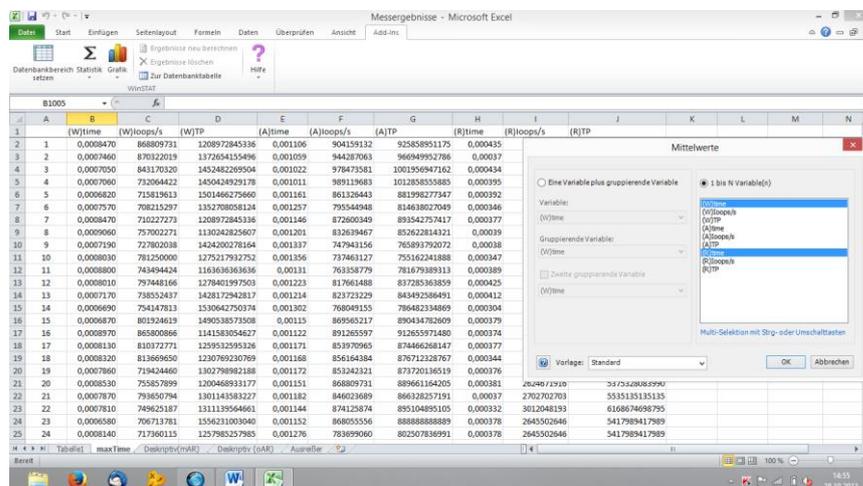


Abbildung 10: Gegenüberstellung ausgewählter Mittelwerte

Nachdem die Auswahl mit OK bestätigt wurde, erhalten wir folgendes Excelsheet:

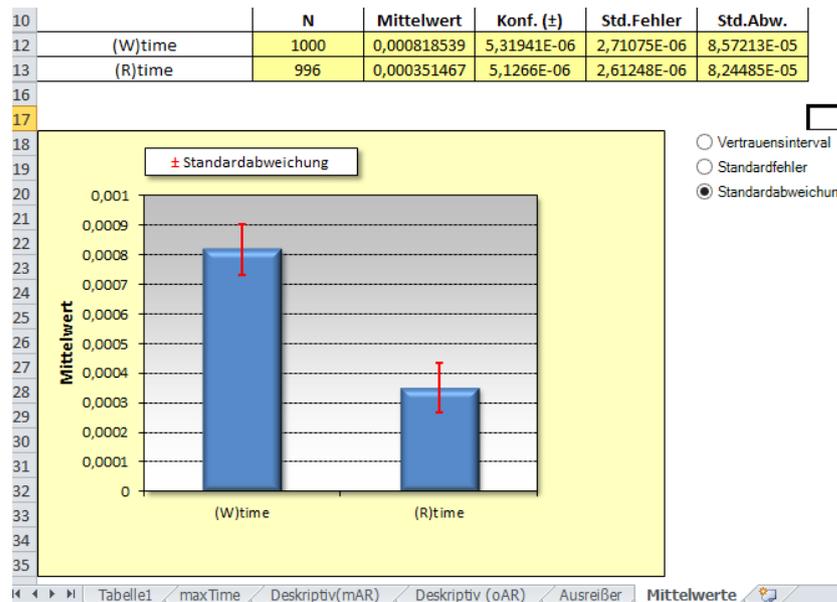


Abbildung 11: Standardabweichungen und Mittelwert von (W)time und (R)time

Der Mittelwert wird in dieser Grafik als Balken abgebildet (nicht gerade die Beste Form zur Darstellung eines solchen Wertes). Die Standardabweichung wird als Spannweite um den Mittelwert dargestellt (rote Linie). Darüber hinaus kann bestimmt werden, ob das Vertrauensintervall oder der Standardfehler anstatt der Standardabweichung dargestellt wird.

Interpretation:

Die dargestellten Mittelwerte zeigen auf, dass der Schreibvorgang im Mittel um den Faktor 2,33 Mal zeitaufwendiger ist als der Lesevorgang ((W)time/(R)time).

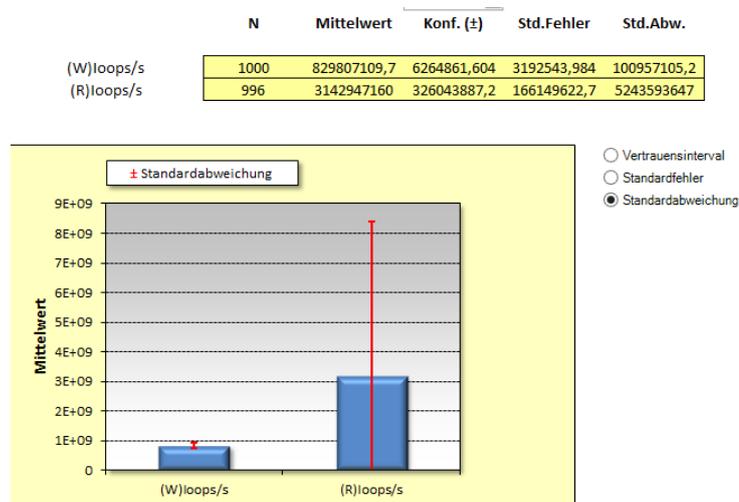


Abbildung 12: Standardabweichung und Mittelwert für (W)loops und (R)loops

Interpretation

Vergleicht man die Mittelwerte so stellt man fest, dass zum einen der Mittelwert vom (R)loops/s um den Faktor 3,75 Mal höher ist als der des (W)loops/s. Dabei ist die Streuung der Datenwerte um den Mittelwert beim den gemessenen Daten für die Lesevorgänge größer als bei den Messdaten für die Schreibvorgänge.

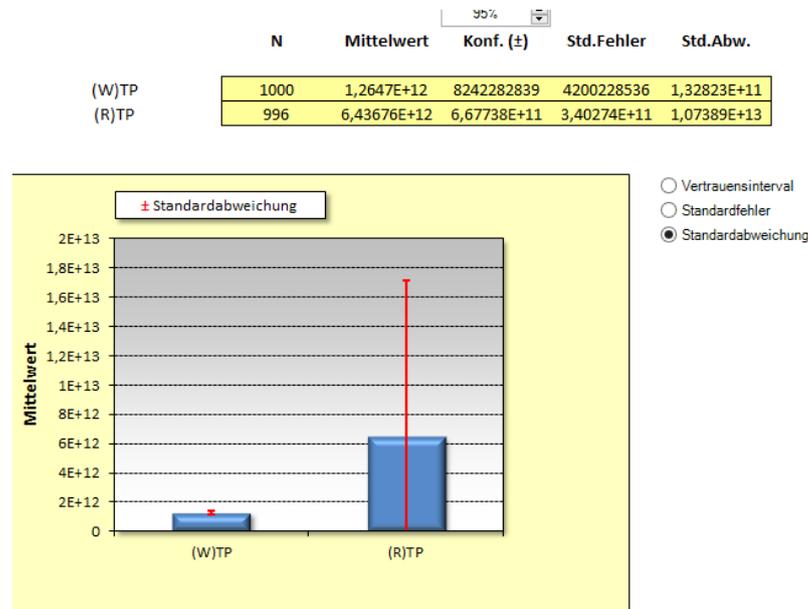


Abbildung 13: Standardabweichung und Mittelwert für (W)TP und (R)TP

Interpretation

Vergleicht man die Mittelwerte so stellt man fest, dass zum einen der Mittelwert vom (R)TP um den Faktor 5,09 Mal höher ist als der des (W)TP. Dabei ist die Streuung der Datenwerte um den Mittelwert beim den gemessenen Daten für die Lesevorgänge größer als bei den Messdaten für die Schreibvorgänge.

Zwischenfazit:

Die Analyse hat bisher lediglich einen bzw. mehrere Hinweise darauf gegeben, dass bei den durchgeführten Messungen der Leseprozesse Probleme aufgetaucht sind. Wenn auch die aufgewendete Zeit für die Lesevorgänge im Mittel um den Faktor 2,33 geringer ist als für die Schreibvorgänge (entspricht im Wesentlichen den Erwartungen der Evaluatoren), so ist das Verhältnis der Anzahl der E/A-Operationen beim Lesevorgang im Mittel um den Faktor 3,75 mal höher als beim Schreibvorgang und der gemessene Datendurchsatz ist bei Lesevorgängen um den Faktor 5,09 mal höher als bei Schreibvor-

gängen. Auffällig ist auch die beobachtete extrem hohe Streuung der gemessenen Datenwerte (R)loops und (R)TP um den Mittelwert.

Setzt man dazu die in der Datenbanktabelle identifizierten und bereits eliminierten 4 fehlerhafte Messwerte sowie die hohe Anzahl von Ausreißern bei den gemessenen Rohdatenwerten der Leseoperationen in Bezug, so ist es empfehlenswert, dort noch etwas tiefer in die Analyse einzusteigen.

Regression

Bei der einfachen Regressionsanalyse wird der Zusammenhang zwischen einer Abhängigen Variablen und einer unabhängigen Variablen aufgezeigt. Betrachten wir die erhobenen Variablen so besteht ein genereller Zusammenhang zwischen der Zeit, die für eine definierte Operation aufgewendet wurde und dem Datendurchsatz. Ist das Datenvolumen fest vorgegeben (bei den durchgeführten Messungen 1024 KB) so kann davon ausgegangen werden, dass der gemessene Datendurchsatz abnimmt wenn der Zeitaufwand für den Lesevorgang zunimmt. Es wird also von den Evaluatoren ein linearer Zusammenhang zwischen dem gemessenen Zeitaufwand und dem Datendurchsatz unterstellt. Diese Unterstellung lässt sich mit einer einfachen linearen Regression beweisen (oder auch widerlegen).

WinSTAT bietet die Möglichkeit, relativ schnell eine entsprechende Regressionsanalyse durchzuführen. Unter

<Add-Ins> - <Statistik> - <Regression> - <Einfache>

öffnet sich nachfolgendes Kontextfenster:

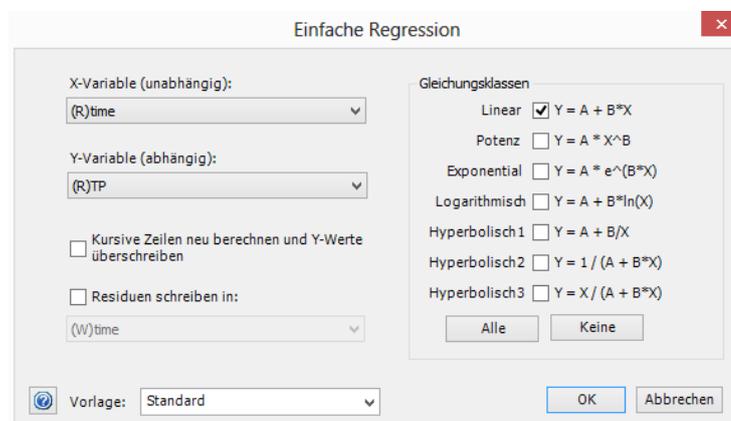


Abbildung 14: Variablenbestimmung für Regression

Als zu untersuchende Variablen bzw. dessen Zusammenhang werden (R)time und (R)TP ausgewählt sowie die Form der Regressionsanalyse (linear) bestimmt. Die von WinSTAT durchgeführte Regressionsanalyse kommt dabei zu folgendem Ergebnis:

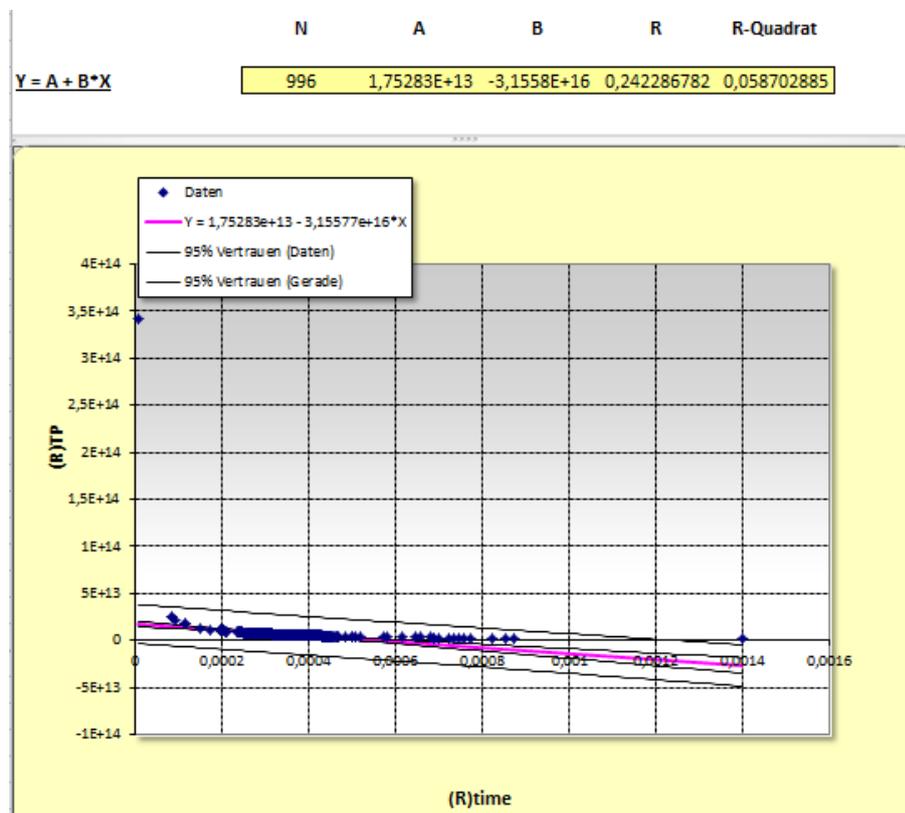


Abbildung 15: Regression (R)time – (R)TP unbereinigt

Interpretation:

Interessant für die Interpretation ist eigentlich nur der R-Quadratwert. Er bestimmt, wie viele der gemessenen Daten durch die Regressionsgrade abgebildet werden. Ein R – Quadrat von 0,058 besagt, dass lediglich 5,8 Prozent der ermittelten Datenwerte durch die Regressionsgerade erklärt werden, d.h. dass der vermutete lineare Zusammenhang zwischen Lesezeit und Datendurchsatz durch die gemessenen Datenwerte so nicht bestätigt wird.

Diese Interpretation, gestützt auf statistische Daten und der eigentlichen Realität widersprechend muss von den Evaluatoren zumindest kritisch hinterfragt bzw. überprüft werden!

Zur Überprüfung soll eine weitere Regressionsanalyse mit bereinigten, also von Extremwerten befreiten Datenwerten durchgeführt werden. Die im Excelsheet <Ausreißer> bereits ermittelten Ausreißer lassen sich dort unterhalb der Tabelle mit den Ausreißerwerten in der Datenbanktabelle aus- bzw. einblenden. Sie werden also erst einmal ausgeblendet. Danach wird die Regressionsanalyse mit der bereinigten Datenbanktabelle erneut durchgeführt. Das Ergebnis sieht wie folgt aus:

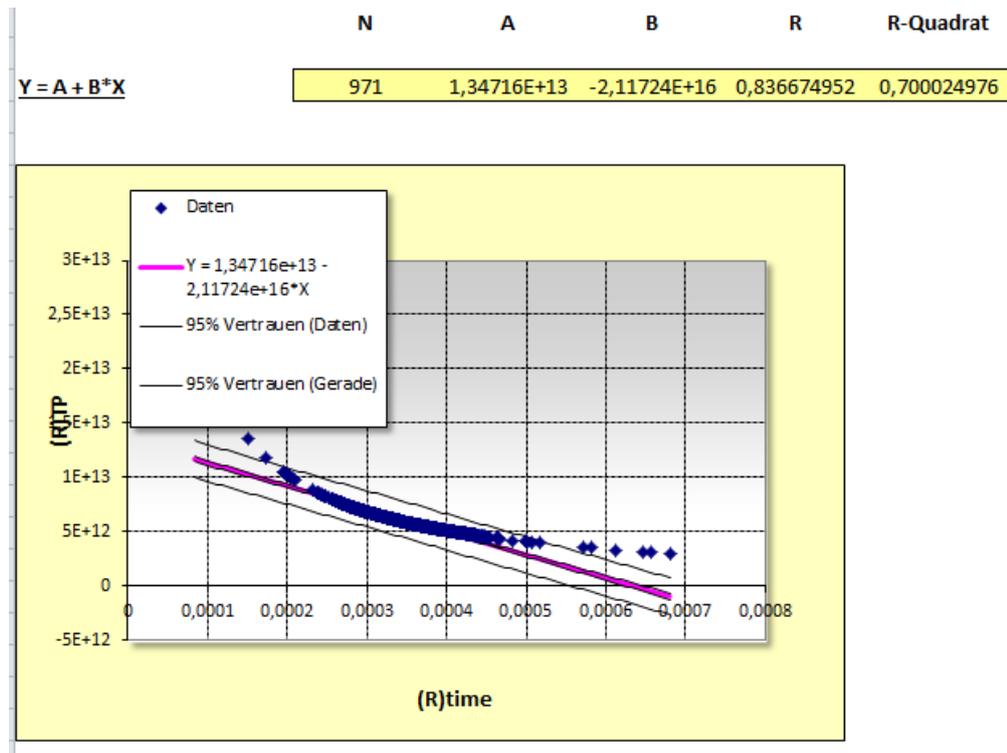


Abbildung 16: Regression (R)time – (R)TP, bereinigt von Ausreißer

Interpretation:
 Wie aus der Tabelle oberhalb der Grafik ersichtlich hat die Anzahl der berücksichtigten Datenpaare (n) im Gegensatz zur vorherigen Regression um 25 Datenpaare abgenommen. Der R – Quadratwert = 0,70, d.h. 70 % der in die Analyse einbezogenen Messwerte werden durch die Regressionsgerade erklärt. Eigentlich ein Grund zur Freude, da zumindest die Gesetzmäßigkeit zwischen Lesezeit und Datendurchsatz wieder in Kraft gesetzt zu sein scheint. Allerdings stellt sich die Frage, ob der Wert von 70 % eigentlich ein „guter Wert“ für Prozesse in der Informationsverarbeitung ist.

Um den ermittelten R - Quadratwert in ein entsprechendes Verhältnis zu setzen, soll sie zuletzt durchgeführte Regression verglichen werden mit einer Regression der entsprechenden Messwerte für das Schreiben. Dabei sollen mögliche Ausreißer auch

nicht berücksichtigt werden. Nachdem die jeweiligen Variablen bestimmt wurden, wird folgende Regressionsgerade ermittelt:

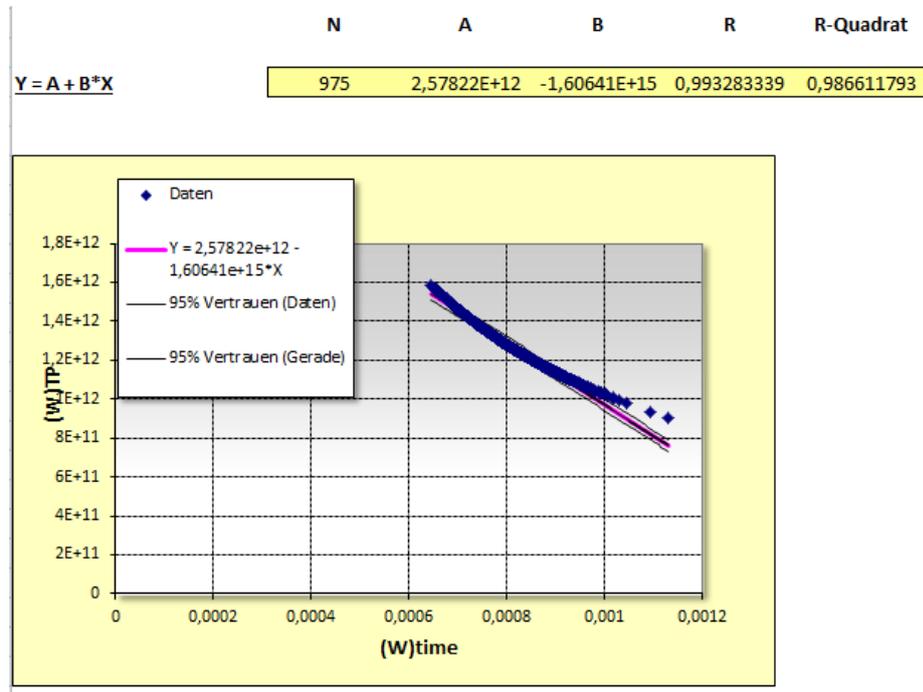


Abbildung 17: Regression (W)time – (W)TP, bereinigt von Ausreißer

Interpretation

Betrachtet man die gemessenen Datenwerte für die Write operation so werden durch die Regressionsgeraden 98,66% der gemessenen Datenwerte erklärt.

Spätestens jetzt ist offensichtlich, dass bei der Read operation anscheinend erhebliche Probleme aufgetreten sind. Die ermittelten Messwerte spiegeln nicht das wieder, was eigentlich erwartet wurde. Um festzustellen, ob es sich dabei um ein systematisch auftretendes Problem handelt, muss die Messung noch einmal wiederholt werden. Dies wurde im Rahmen dieser Evaluation auch gemacht. Es soll an dieser Stelle nicht noch einmal sämtliche Schritte der Datenaufbereitung durchlaufen werden sondern lediglich dessen Ergebnisse.

Bisher wurde von den Evaluatoren ein linearer Zusammenhang zwischen der Variable (R)time und (R)TP unterstellt bzw. wurde angenommen, dass sich der Zusammenhang zwischen den beiden Variablen durch eine lineare Regression abbilden lässt. Tatsache ist jedoch, dass sich der Datendurchsatz i.d.R. nicht linear zur verwendeten Zeit verhält sondern sich vielmehr durch die Funktion:

$$Y = A+B/X$$

darstellen lässt.

die Regressionsbeziehung sieht dann wie folgt aus:

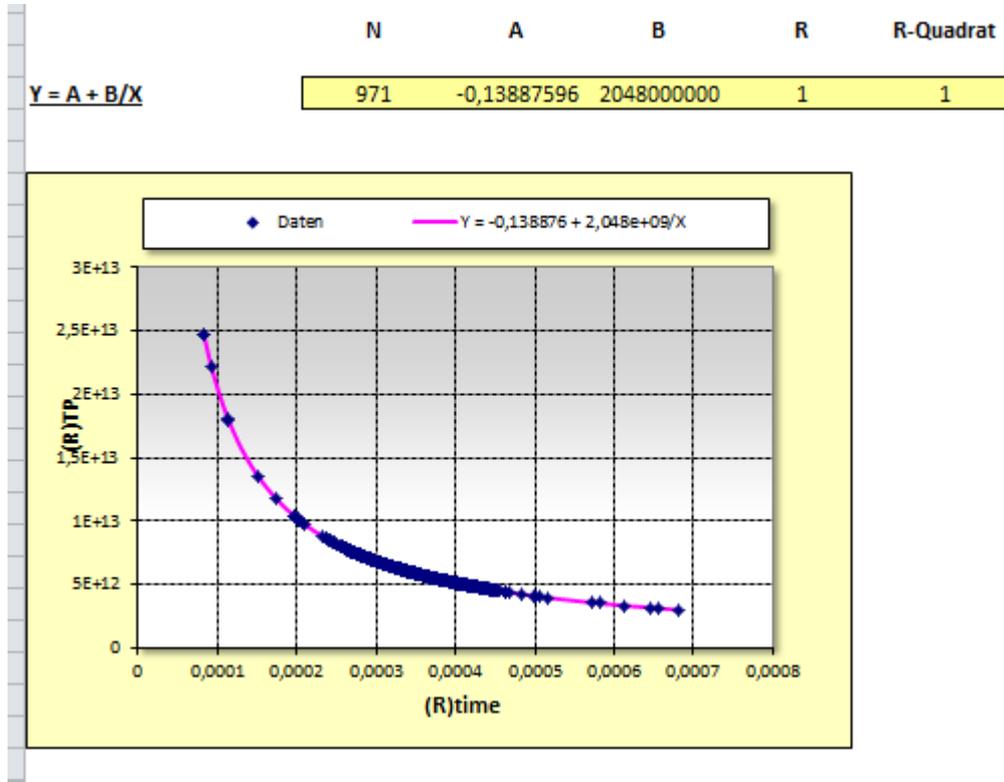


Abbildung 18: Regression (R)time – (R)TP, nicht linear

Interpretation:

Anhand des Graphen ist ersichtlich, dass sämtliche gemessenen Datenwerte auf der Regressionskurve liegen. Das R – Quadrat von 1 besagt, dass sämtliche Datenwerte durch die Regressionsgeraden abgebildet werden (100%). Es ist auch ersichtlich, dass eine Vielzahl von gemessenen Extremwerten ermittelt wurden, die letztendlich dazu geführt haben, dass die ermittelten statistischen Parameter nur eine geringe Aussagekraft haben. Eine der Wesentlichen Fragen, die sich die Evaluatoren stellen sollten ist, ob es sich bei den gemessenen Daten um systematische Extremwerte handelt, d.h. ob sie bei einer weiteren Messung so wieder auftreten werden. Demzufolge wurde eine Kontrollmessung mit den gleichen Rahmenbedingungen vorgenommen.

Kontrollmessung

1. In den ermittelten Rohdaten lagen keine fehlerhaften Daten vor, so dass auch keine Daten eliminiert werden mussten.
2. Mittelwerte wurden nicht weiter betrachtet. Die Regressionsanalyse lieferte folgende Daten und Grafik:

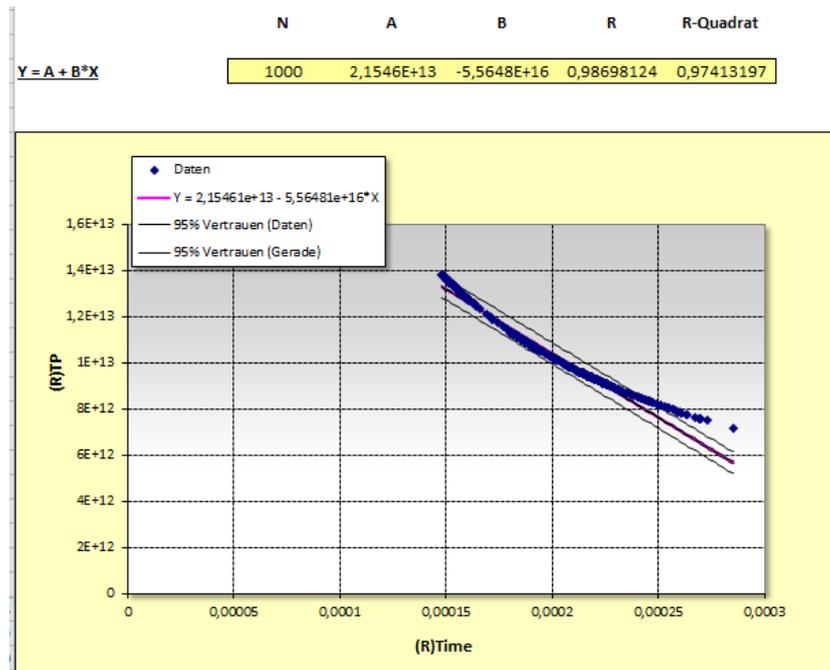


Abbildung 19: Kontrollmessung Regression (R)time – (R)TP, unbereinigt

Interpretation:

Wie schon am r – Quadratwert ersichtlich ist, werden 97,4% der Messdaten durch die Regressionsgerade erklärt, d.h. es gibt einen nachweisbaren Zusammenhang zwischen der Lesezeit und dem Datendurchsatz.

Betrachtet man jedoch die Lage der Datenwerte so fällt auf, dass der Zusammenhang nicht linear ausgeprägt ist.

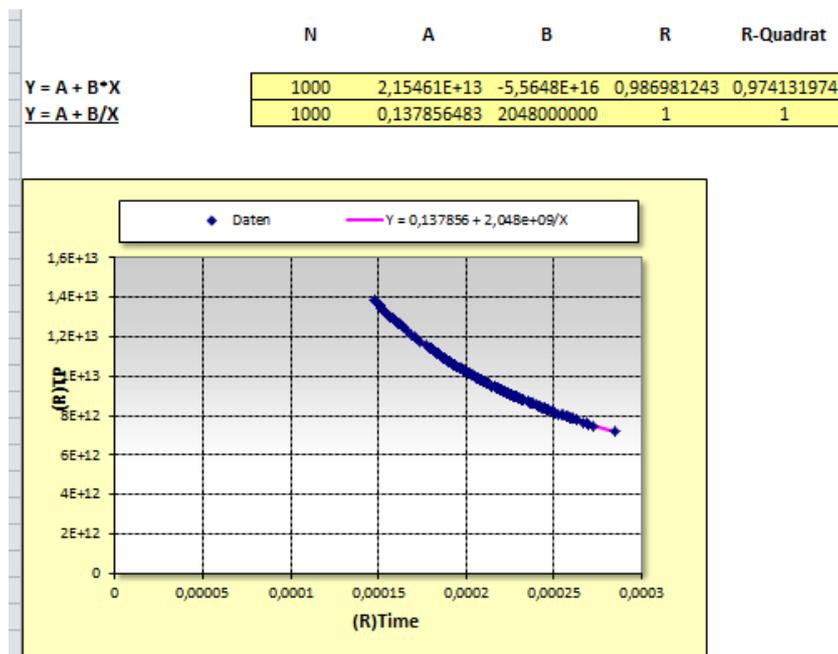


Abbildung 20: Kontrollmessung Regression (R)time – (R)TP, nicht linear

Interpretation

Es ist ersichtlich, dass die gemessenen Datenwerte keine Ausreißer enthalten. Es kann also davon ausgegangen werden, dass die in der ersten Messung ermittelten Ausreißer nicht systematisch auftreten.

5.2 Evaluation des Testclusters mit dem Benchmark „HPL“

5.2.1 Phase 1: Planen des Prozesses der Messung

Die Planungsphase von Benchmarks wurde bereits unter Kapitel 4.1.1 ausführlich für den Benchmark „Parabench“ beschrieben. Bei dem Benchmark „HPL“ ist analog genauso zu verfahren.

5.2.2 Phase 2: Durchführung der Messung

Wie bei dem vorigen Kapitel (Parabench), muss man sich zuerst auf dem Testcluster anmelden (Siehe Anhang 4). Bis einschließlich Schritt 4 ist die Vorgehensweise mit „HPL“ identisch zu „Parabench“.

Schritt 5

HPL vorinstallieren:

Nun muss HPL von einem Mitreiter des WR (Wissenschaftliches Rechnen) installiert werden.

Schritt 6

Dieser Schritt ist für den Anfang identisch mit dem „Parabench“-Kapitel (Schritt 6), es müssen hier lediglich folgende Änderungen vorgenommen werden:

- Der Pfad heißt nicht Parabench, sondern HPL
- Das Jobscript muss wie folgt angepasst werden:

```
#!/bin/sh
# Time limit is one minute.
# See "man sbatch" for other time formats.
#SBATCH --time=1
# Run 10 tasks on 2 nodes.
#SBATCH -N 1 -n 5
# Output goes to job.out, error messages to job.err.
#SBATCH --error=job.err --output=job.out
srun hostname
mpiexec ./xhpl
```

Der Pfad muss hier so angepasst werden, dass er zur "xhpl-Datei" führt. In unserem Beispiel: "mpiexec ./xhpl".

Ab jetzt unterscheidet sich die Vorgehensweise grundsätzlich von Parabench:

Um nun Parameter für die Durchführung des Benchmarks zu variieren, ruft man die HPL.dat mit einem Editor auf und editiert die Parameter. Eine genaue Beschreibung der verschiedenen Parameter findet man unter:

<http://www.netlib.org/benchmark/hpl/tuning.html>

```
PLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6           device out (6=stdout,7=stderr,file)
1           # of problems sizes (N)
829248 NS
1           # of NBs
10 NBs
0          PMAP process mapping (0=Row-,1=Column-major)
3          # of process grids (P x Q)
2 1 4      Ps
2 4 1      Qs
16.0       threshold
3          # of panel fact
0 1 2      PFACTs (0=left, 1=Crout, 2=Right)
2          # of recursive stopping criterium
2 4        NBMINs (>= 1)
1          # of panels in recursion
2          NDIVs
3          # of recursive panel fact.
0 1 2      RFACTs (0=left, 1=Crout, 2=Right)
1          # of broadcast
0          BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1          # of lookahead depth
0          DEPTHS (>=0)
2          SWAP (0=bin-exch,1=long,2=mix)
64         swapping threshold
0          L1 in (0=transposed,1=no-transposed) form
0          U in (0=transposed,1=no-transposed) form
1          Equilibration (0=no,1=yes)
8          memory alignment in double (> 0)
```

Ausführen tut man den Benchmark dann, ähnlich wie beim Parabench, mit "sbatch hpl.slurm" (je nachdem wie man das jobscript benannt hat --> Siehe Schritt 6 beim Parabench). Auch das Aufrufen der Ergebnisse, verläuft ähnlich wie beim Parabench: Einfach im HPL-Verzeichnis "cat job.out" eingeben:

```
=====
T/V          N    NB    P    Q          Time          Gflops
-----
WR00L2L2     8292  10    2    2          44.74          8.498e+00
HPL_pdgesv() start time Thu Oct 17 19:41:45 2013
HPL_pdgesv() end time   Thu Oct 17 19:42:30 2013
=====
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)= 0.0042460 ..... PASSED
```

Das relevante Ergebnis sind hier die Gflops. Je mehr Gflops, desto schneller hat der Supercomputer gearbeitet. Funktioniert der Benchmark durch beispielsweise falsch eingegebene Parameter nicht, kann man auch hier mit "cat job.err" ein Fehlerprotokoll aufrufen.

5.2.3 Phase 3: Auswertung und Interpretation der Messergebnisse

6 Quellenverzeichnis

Balzer, Lars (2005): Wie werden Evaluationsprojekte erfolgreich? - Ein integrierender theoretischer Ansatz und eine empirische Studie zum Evaluationsprozess. Empirische Pädagogik; Auflage: 1., Landau

Ghanbari, S. A. (2002). Einführung in die Statistik für Sozial- und Erziehungswissenschaftler. Springer Verlag, Berlin

Kromrey, Helmut (2001): Evaluation – ein vielschichtiges Konzept. Begriff und Methodik von Evaluierung und Evaluationsforschung. Empfehlungen für die Praxis.

http://www.bibb.de/dokumente/pdf/a11_vielschichtiges_konzept.pdf

Heinrich, Lutz J. et al (1995). Wirtschaftsinformatiklexikon. München, Wien, Oldenbourg: R. Oldenbourg Verlag GmbH, München.

Reade, Nicolà (2008): Einführung in Qualitätsmanagement, Monitoring und Evaluation, Osnabrück

Schneider, Vera / Meiers, Ralph (2007): Reporting in Handbuch der Evaluation. Hrsg. Reinhard Stockmann. Waxmann Verlag GmbH, Münster

Stockmann, Reinhard (2006): Evaluation und Qualitätsentwicklung. Eine Grundlage für wirkungsorientiertes Qualitätsmanagement. Sozialwissenschaftliche Evaluationsforschung Band 5. Waxmann Verlag GmbH, Münster

Anhang 1: Exkurs Supercomputer

Inhaltsverzeichnis

1 Supercomputer bzw. Parallelrechner	57
1.1. Grundlagen	57
1.1.1 Rechner	57
1.1.2 Parallelität	61
1.1.3 Parallelrechner	71
1.2. Cluster Computer	75
1.2.1 Hardware	76
1.2.2 Software	77
1.3. Wie werden Supercomputern miteinander verglichen?	80
Quellenverzeichnis zu Anhang 1	82

1 Supercomputer bzw. Parallelrechner

Als Super- bzw. Parallelrechner werden die schnellsten Rechner unserer Zeit bezeichnet. Sie heißen Parallelrechner, weil Aufträge (engl. Jobs) bzw. Operationen auf mehrere Rechner aufgeteilt und parallel abgearbeitet werden.

In den nachfolgenden Unterkapiteln erläutern wir wesentliche Grundlagen von Supercomputern und beschreiben das Cluster Computing (deut. Haufen bzw. Gruppen rechnen).

1.1. Grundlagen

Um den Begriff Parallelrechner besser zu erläutern werden vorher die allgemeinen Begriffe Rechner und Parallelität beschrieben.

1.1.1 Rechner

Die heutigen Rechner (engl. Computer) sind nach der klassischen Von-Neumann-Architektur (VNA) konstruiert. Die VNA bzw. der Von-Neumann-Rechner (VNR) besteht aus 5 Komponenten. Und zwar aus einem Steuer- und Rechenwerk, die zusammen den Prozessor (engl. Central Processing Unit, kurz CPU) darstellen. Zusätzlich zur CPU besteht der VNR aus einem Speicherwerk und einem Ein-/Ausgabewerk (engl. Input-/Output, kurz I/O). Diese vier Komponenten sind über ein Binary Unit System (kurz BUS) verbunden (Siehe Abbildung 1).

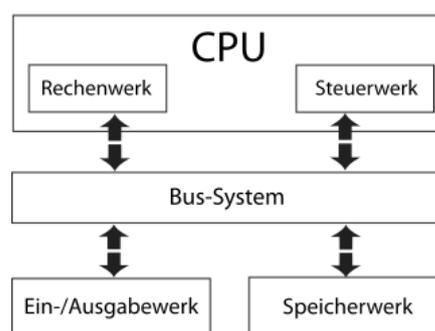


Abbildung 21: Von-Neumann-Architektur

Quelle: http://upload.wikimedia.org/wikipedia/de/d/db/Von-Neumann_Architektur.svg

In den nachfolgenden Abschnitten werden die einzelnen Komponenten näher erläutert.

Prozessor

Der Prozessor (engl. Central Processing Unit, kurz CPU) ist ein Zusammenschluss von zwei Komponenten, dem Steuer- und Rechenwerk. Eine CPU ist in jedem Computer enthalten. Es gibt verschiedene Hersteller von Prozessoren, wie z. B. die Firmen AMD, Apple oder Intel (Siehe Abbildung 3).



Abbildung 22: Intel i7 (links), Apple A5 (mitte), AMD Athlon (rechts)

Quelle: <http://exe.fam-trachsel.ch/InfoGL/prozessor.html>

Steuerwerk

Das Steuerwerk (engl. Control Unit, kurz CU) bekommt Befehle vom Ein-/Ausgabewerk und decodiert diese. Danach leitet das Steuerwerk die übersetzten Befehle weiter an die zuständigen Komponenten, zum Beispiel sollen bei einem Befehl zwei Zahlen miteinander addiert werden, dann schickt das Steuerwerk dem Rechenwerk ein Steuersignal mit den genauen Anweisungen und dem Speicherwerk ein Datensignal, mit den benötigten Daten für die Operation. Allgemein betrachtet ist das Steuerwerk die Leitungseinheit eines Prozessors, die den Ablauf der Befehlsverarbeitung steuert. Für eine detailliertere Beschreibung Siehe http://tams-www.informatik.uni-hamburg.de/applets/baukasten/DA/VNR_Einleitung.html#Steuerwerk.

Rechenwerk

Das Rechenwerk (engl. Arithmetical Logical Unit, kurz ALU) führt Operationen durch und speichert zwischendurch die Ergebnisse in einem Register ab, bis der Befehl, zuvor vom Steuerwerk geschickt, ausgeführt wurde. Dieses Register ist ein interner Speicher im Prozessor, der eine Größe von 64 Bit aufweisen kann. Danach leitet das Rechenwerk das Endergebnis an das Speicherwerk weiter und schickt eine Rückmeldung an das Steuerwerk. Das Rechenwerk ist somit die ausführende Einheit eines Prozessors. Für eine detailliertere Beschreibung Siehe http://tams-www.informatik.uni-hamburg.de/applets/baukasten/DA/VNR_Einleitung.html#Steuerwerk.

BUS-Einheit

Ein BUS (engl. Binary Unit System, kurz BUS) beinhaltet immer einen Steuer-, Adress- und Datenbus (engl. Control-, Adress- und Data-BUS).

- Der Steuer-BUS koordiniert Lese- und Schreibweisungen zwischen den Komponenten (WER greift auf welche Daten zu?).
- Der Adress-BUS überträgt ausschließlich Speicheradressen (Wohin soll was gespeichert bzw. geschickt werden?).
- Der Daten-BUS übermittelt die Daten zwischen den einzelnen Komponenten (Was soll geschickt werden?).

Wie dieser BUS arbeitet sieht man in der Abbildung 4, dort schickt die CPU einen Befehl über den Steuerbus an das Ein-/Ausgabewerk oder an das Speicherwerk. Genauso auch beim Adressbus, schickt die CPU die Adressierung an die beiden Komponenten. Der Datenbus übermittelt dann die gewünschten Daten zwischen den Komponenten hin und her.

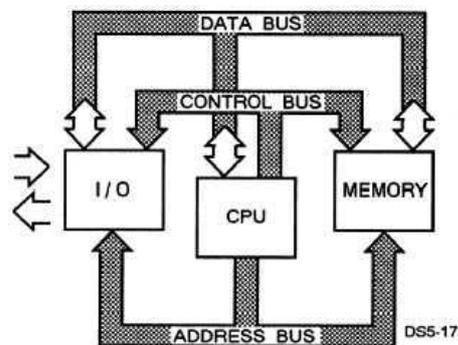


Abbildung 23: Steuer-, Adress- und Datenbus

Quelle: <http://exe.fam-trachsel.ch/InfoGL/bussystem.html>

Für eine detailliertere Beschreibung Siehe <http://exe.fam-trachsel.ch/InfoGL/bussystem.html>.

Speicherwerk

Das Speicherwerk (Siehe Abbildung 5) bzw. der Arbeitsspeicher (engl. Random Access Memory, kurz RAM) speichert Daten für die Dauer der Ausführung und hält diese für

das Rechenwerk bereit. Die Daten können von den anderen Komponenten überschrieben, beschrieben oder nur gelesen werden.

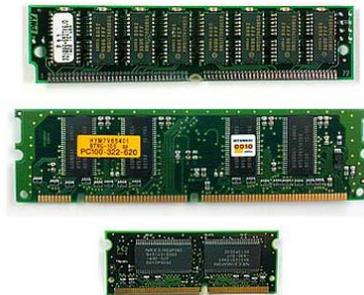


Abbildung 24: Arbeitsspeicher (Random Access Memory)

Quelle: <http://exe.fam-trachsel.ch/InfoGL/ram.jpg>

Für eine detailliertere Beschreibung Siehe http://exe.fam-trachsel.ch/InfoGL/interne_speicher.html.

Eingabe-/Ausgabewerk

Das Eingabe- und Ausgabewerk (engl. Input/Output Unit, kurz I/O) sind die Schnittstellen zwischen der Zentraleinheit und den Peripheriegeräten, über diese Anschlüsse werden Daten ausgetauscht. Im Eingabewerk werden Befehle eingetragen (z. B. über eine Tastatur), die sequentiell abgearbeitet und an das Steuerwerk übergeben werden. Nach der Bearbeitung eines Befehls wird das Ergebnis an das Ausgabewerk geschickt und das Ausgabewerk stellt das Ergebnis z. B. optisch dar (z. B. über einen Bildschirm). Für eine detailliertere Beschreibung Siehe <http://exe.fam-trachsel.ch/InfoGL/einausgabeeinheit.html> und <http://exe.fam-trachsel.ch/InfoGL/peripheriegerate.html>.

1.1.2 Parallelität

Speedup, Zeitgewinn, Geschwindigkeitsgewinn

Speedup ist ein allgemeines Maß zur Bestimmung, in welchem Umfang die Leistungsfähigkeit von Computersystemen durch eine bestimmte Optimierung verbessert wird. In unserem Falle sprechen wir von Parallelisierung als Optimierung. In aller Regel wird der Begriff „Speedup“ im Zusammenhang mit der Verkürzung der Antwortzeiten angewandt. Somit kann die Effektivität einer Parallelisierung bestimmt werden, indem festgestellt wird, wie stark sich die Antwortzeit einer ausgewählten Transaktion oder Operation durch Parallelisierung verkürzt hat. Vereinfacht ausgedrückt beschreibt es den Zusammenhang zwischen der seriellen und parallelen Ausführungszeit eines Programmteiles.

Formale Darstellung:

$$\text{Zeitgewinn} = \frac{\text{ursprüngliche Laufzeit}}{\text{neue Laufzeit}} = \frac{1}{(1 - P) + \frac{P}{N}}$$

Abbildung 25: Formel für den Zeitgewinn

Strong Scaling (starke Skalierung)

Der Begriff „Strong Scaling“ ist so definiert, dass eine Problemgröße, wie beispielsweise der Durchlauf eines bestimmten Programmes, immer fest, also gleich groß bleibt. Hier sprechen wir von einer festen Problemgröße. Durch eine zunehmende Anzahl an Prozessoren wird die Durchlaufzeit des Programmes verkürzt. Der Speedup steigt. Dieser Zusammenhang wird im weiteren Verlauf noch näher erläutert (Amdahlsches Gesetz).

Merke: Feste Problemgröße, zunehmende Anzahl an Prozessoren. Programm läuft immer schneller.

Weak Scaling (schwache Skalierung)

Der Begriff „Weak Scaling“ ist so definiert, dass eine Problemgröße, wie beispielsweise der Durchlauf eines bestimmten Programmes zunehmend größer wird. Die Zeit, in der das Programm durchgeführt werden soll ist allerdings fest begrenzt. Durch die steigende

Anzahl an Prozessoren können also größere Problemgrößen in gleichbleibender Zeit abgehandelt werden.

Dieser Zusammenhang wird im weiteren Verlauf noch näher erläutert (Gustafsonsches Gesetz).

Merke: Wachsende Problemgröße, mehr Prozessoren. Löst wachsende Probleme in einem festen Zeitbudget.

In folgender Grafik ist das typische Skalierungsmuster von „Strong Scaling“ und „Weak Scaling“ veranschaulicht.

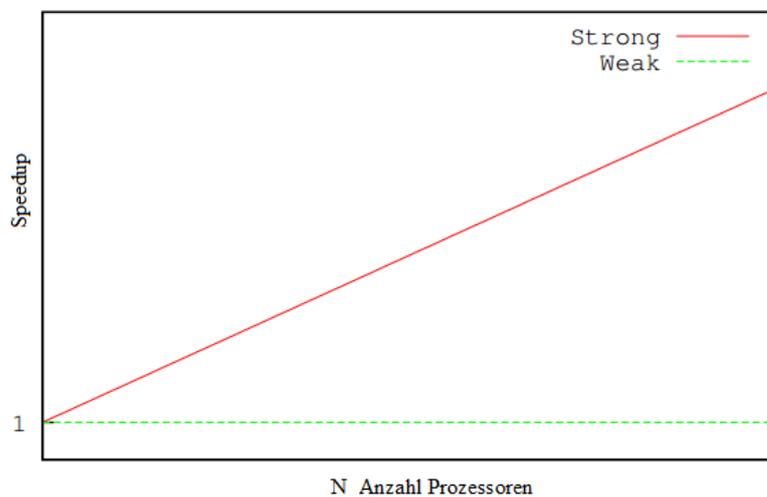


Abbildung 26: Strong und Weak Scaling

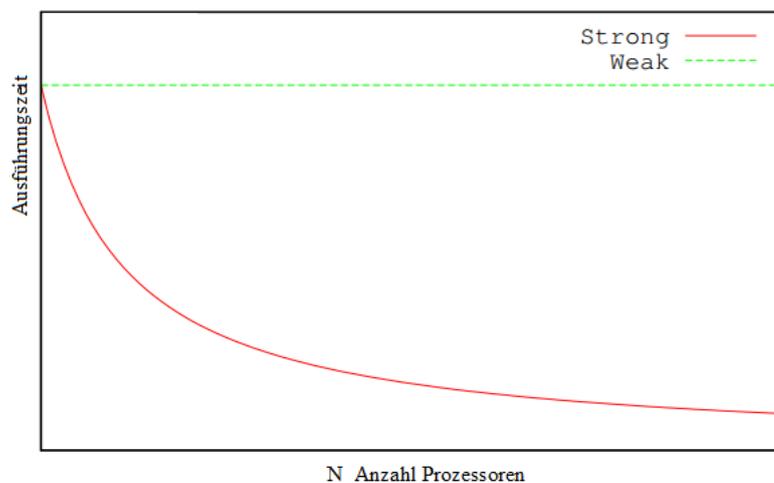


Abbildung 27: Strong und Weak Scaling

Parallelität

Arbeitsabläufe bzw. deren Einzelschritte heißen parallel, wenn sie gleichzeitig und voneinander unabhängig durchgeführt werden können. Im Vergleich zur seriellen Verarbeitung erfordert die parallele Verarbeitung größeren technischen Aufwand bezüglich der Hardware (z.B. mehrere Prozessoren, gleichzeitiger Zugriff auf Speicher) und der Software (aufwendigere Verwaltungsprogramme, Überwachungs- und Synchronisierungsaufgaben usw.). Im engeren Sinne versteht man unter paralleler Verarbeitung die gleichzeitige Durchführung von Tätigkeiten, sprich Nebenläufigkeit.

Im Inneren von Parallelcomputern, Rechnersystemen und von Mikroprozessoren werden möglichst viele Tätigkeiten parallel ausgeführt, vorwiegend um die Verarbeitungsgeschwindigkeit zu erhöhen und um die einzelnen Komponenten besser auszulasten. Die Benutzerprogramme werden auf die Hauptprozessoren verteilt.

Um die Leistungsfähigkeit eines Parallelrechners richtig ausnutzen zu können, muss die Programmierung genau auf die verteilten Hauptprozessoren zugeschnitten werden. Hier sprechen wir von paralleler Programmierung. Prinzipiell handelt es sich dabei um ein logistisches Problem. Es gilt die knappen Ressourcen – Rechenzeit, Speicherzugriffe, Datenbusse – effizient auszunutzen. Stets sollte der sequentielle Programm-Overhead minimal sein (im späteren Verlauf kommen wir darauf noch einmal detaillierter zu sprechen. Stichpunkt: Amdahlsches Gesetz).

Je größer die Anzahl der Hauptprozessoren ist, desto schneller steigt der Kommunikationsaufwand in einer Parallelrechnerumgebung überproportional an. Der Zusammenhang ist nicht linear.

Im Endeffekt muss das Optimum zwischen Anzahl der Hauptprozessoren, Kommunikationsaufwand und Programmierung gefunden werden.

Das folgende Schaubild verdeutlicht die Funktionsweise Serieller und Paralleler Betriebsarten in vereinfachter Darstellung.

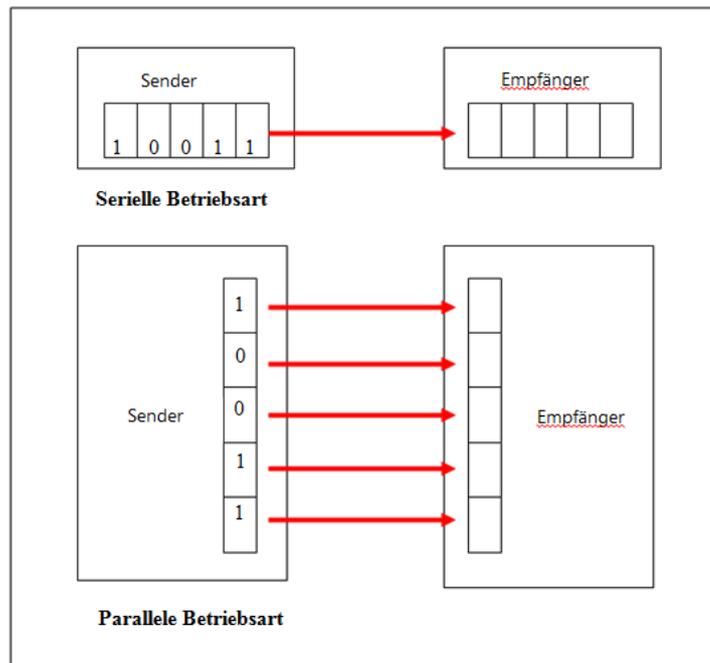


Abbildung 28: Serielle und Parallele Betriebsart

Amdahlsches und Gustafsonsches Gesetz

Es gibt viele theoretische Untersuchungen zur parallelen Programmierung. Doch zwei von ihnen haben besonderes Aufsehen erregt. Das Gesetz von Amdahl und das Gesetz von Gustafson.

Diese beiden Theorien werden uns dabei helfen, den so genannten „Speedup“ oder Zeitgewinn, also den Zusammenhang zwischen der seriellen und parallelen Ausführungszeit eines Programmteils bestimmen und nachvollziehen zu können.

Im Grunde sind beides sehr einfache Theorien, die eng miteinander zusammenhängen.

Im folgenden Abschnitt werden wir uns die beiden Modelle genauer ansehen. Wir werden einfache Folgerungen ziehen und die Grenzen der Modelle beleuchten.

Das Amdahlsche Gesetz

Das Amdahlsche Gesetz beschäftigt sich mit der Frage, wie stark man ein bestimmtes Programm durch Parallelisierung beschleunigen kann (Strong Scaling). Hierbei geht es von einem sehr einfachen Modell des zu parallelisierenden Programmcodes aus. Es nimmt an, dass jedes Programm in zwei Teile aufzuteilen ist. Zum einen der Teil, der beliebig stark parallelisierbar ist, und zum anderen, der Teil der sequenziell ablaufen muss und somit überhaupt nicht parallelisiert werden kann.

Wir starten unser Programm beispielsweise auf einem System mit nur einem Prozessor und gehen von einer Laufzeit von 1 aus. Nun lassen wir das Programm laufen und schauen zunächst auf den ersten, parallelisierbaren Teil. Den zeitlichen Anteil, den das Programm zum Durchlaufen dieses Teiles benötigt nennen wir P. Daraus folgt, dass der zweite, sequentielle Teil mit (1-P) bezeichnet werden kann.

Nun können wir uns die Frage stellen, wie schnell das Programm auf N Prozessoren läuft. Die Laufzeit des sequentiellen Teiles kann sich nicht verändern. Der parallelisierbare Teil jedoch, wird optimal auf alle Prozessoren verteilt und läuft daher N-mal so schnell.

Daraus ergibt sich folgende Formel:

$$\underbrace{(1 - P)}_{\text{sequenziell}} + \underbrace{\frac{P}{N}}_{\text{parallel}}$$

Wie viel schneller das Programm also nun geworden ist lässt sich mit folgender Formel feststellen:

$$\text{Zeitgewinn} = \frac{\text{ursprüngliche Laufzeit}}{\text{neue Laufzeit}} = \frac{1}{(1 - P) + \frac{P}{N}}$$

Zusammengefasst ist N die Anzahl der Prozessoren und P der Laufzeit-Anteil des parallelisierbaren Programmcodes.

In der Literatur findet man an Stelle des Begriffes „Zeitgewinn“ auch den Begriff „speedup“ oder „Geschwindigkeitsgewinn“.

Das eben aufgezeigte ist im Grunde schon das ganze Amdahlsche Gesetz. Einfach dargestellt zeigt es auf, wie stark sich ein bestimmtes Programm durch Parallelisierung beschleunigen lässt.

Jedoch stößt es schnell an seine Grenzen. Schon bald stellte sich heraus, dass ein Programm durch Parallelisierung nur bis zu einer bestimmten Grenze hin beschleunigt werden kann.

Dies beleuchtet die folgende Ungleichung:

$$\text{Zeitgewinn} < \frac{1}{1 - P}$$

Der Grund dafür ist, dass der sequentielle Anteil des Programmes der ist, der niemals verschwindet. $(1-P)$ bleibt also immer bestehen und somit ist es unmöglich durch immer mehr Parallelisierung unendlichen Zeitgewinn zu erzielen.

Wir können die Grenze auch dadurch herleiten, dass wir im Amdahlschen Gesetz die Anzahl N der Prozessoren gegen ∞ laufen lassen.

$$\lim_{N \rightarrow \infty} \text{Zeitgewinn} = \lim_{N \rightarrow \infty} \frac{1}{(1 - P) + \frac{P}{N}} = \frac{1}{1 - P}$$

Man sieht, dass man selbst mit Tausenden Prozessoren ein zu 95% parallelisierbares Programm maximal um den Faktor 20 beschleunigen kann.

$$\left(\frac{1}{1 - P} = \frac{1}{1 - 0,95} = 20 \right)$$

Das folgende Diagramm veranschaulicht dies für verschieden gut parallelisierbare Programme. Dabei ist die N -Achse (Anzahl der Prozessoren) logarithmisch eingeteilt, das heißt, die Anzahl der Prozessoren wird immer verdoppelt.

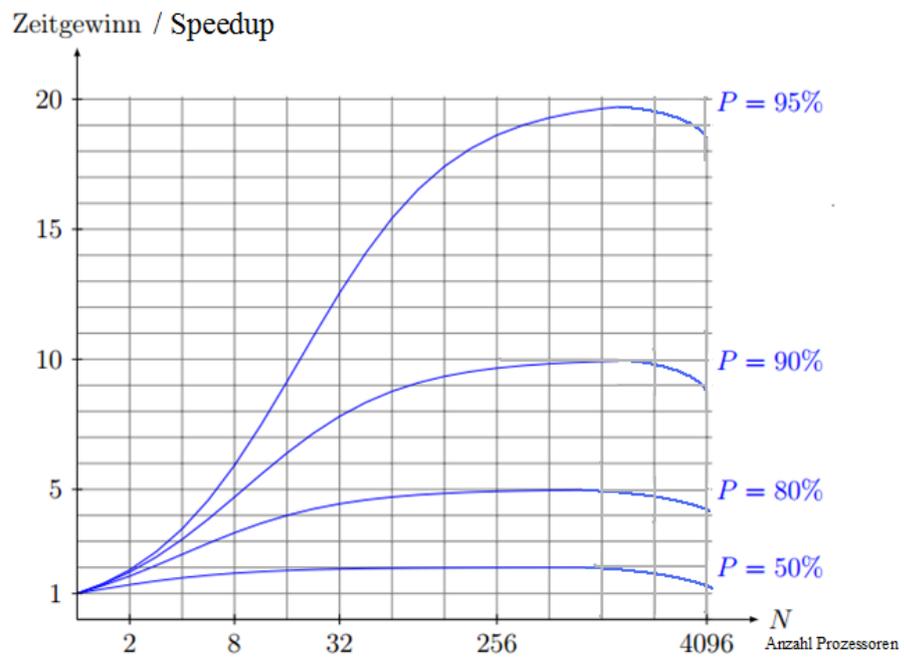


Abbildung 29: Maximale Speedup

Aufgrund der vermehrten Kommunikation der einzelnen Prozessoren fällt die Kurve zum Ende hin typischerweise jeweils wieder ab.

Bei einem Programm optimaler Parallelisierbarkeit von 100% würde die Kurve komplett linear verlaufen wie folgendes Schaubild verdeutlicht.

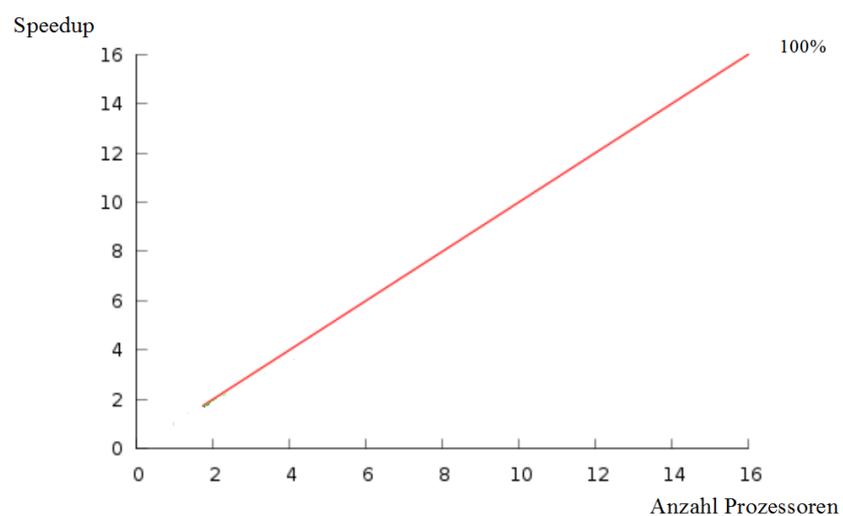


Abbildung 30: Optimaler Speedup

Neben dieser soeben beleuchteten Grenze des Amdahlschen Gesetzes, gibt es darüber hinaus weitere Punkte, die von der Theorie nicht beleuchtet werden. Ohne weiter darauf eingehen zu wollen, sollten sie doch nicht unerwähnt bleiben.

1. Je mehr Prozessoren an einem Problem arbeiten, desto aufwändiger wird die Kommunikation zwischen ihnen. Dieser zusätzliche Verwaltungsaufwand senkt nicht zwingend, aber in der Regel die zu erwartenden Zeitgewinne durch Parallelisierung nochmals.
2. Neben der Anzahl der Prozessoren wachsen auch weitere Ressourcen mit der Parallelisierung. So bringt jeder zusätzliche Prozessor normalerweise auch Cache mit, sodass insgesamt für die Berechnungen viel mehr Prozessor-Cache zu Verfügung steht.
3. Oft wollen wir unsere Ergebnisse gar nicht schneller haben, sondern lieber größere Probleme in der bisherigen Zeit berechnen. Diese Idee führt uns direkt zum Gustafsonschen Gesetz, welches wir folgend etwas näher beleuchten werden.

Das Gustafsonsche Gesetz

Das Gustafsonsche Gesetz zeigt auf, dass sich massive Parallelisierung doch lohnt. Zwar nicht, um beliebig schneller zu werden, jedoch um in gleicher Zeit beliebig große Probleme zu lösen (Weak Scaling). Gustafson zeigt damit einen Aspekt auf, der von Amdahl übersehen wurde.

Der wichtigste Gedanke von Gustafson ist die Einbeziehung der Problemgröße in das Modell. Er spricht von Eingangsdaten, deren Größe wir mit K bezeichnen.

Für unser Beispiel nehmen wir den Größenfaktor 1. Also gilt $K = 1$. Als Laufzeit des Programms, in einem System mit einem Prozessor, gilt auch der Faktor 1. Der parallelisierte Programmanteil soll bei Problemgröße $K = 1$ einen Anteil von P' an der Laufzeit haben.

Nun stellen wir uns die Frage, wie sich die Laufzeit ändert, wenn wir die Problemgröße erhöhen.

Laut Gustafson bekommt hier nur der parallelisierte Programmteil mehr zu tun, da dieser typischerweise aus vielen Operationen besteht deren Dimension sich direkt aus der Größe der Eingabedaten ergibt. Der sequentielle Programmteil hingegen besteht überwiegend aus Initialisierungsschritten, die unabhängig von der Problemgröße sind. Die daraus folgende Laufzeit kann formal so dargestellt werden:

$$\underbrace{(1 - P')}_{\text{sequenziell}} + \underbrace{K \cdot P'}_{\text{parallel}}$$

Nun stellen wir uns wiederum die Frage, wie viel schneller das Programm auf einem N -Prozessor-System läuft. Ähnlich wie bei Amdahl ergibt sich hier die folgende Formel.

$$\underbrace{(1 - P')}_{\text{sequenziell}} + \underbrace{\frac{K \cdot P'}{N}}_{\text{parallel}}$$

Der Zeitgewinn lautet entsprechend:

$$\text{Zeitgewinn} = \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}}$$

Im Prinzip haben wir dem Amdahlschen Gesetz lediglich den Faktor Problemgröße K zugefügt. Somit lässt sich der Zeitgewinn in Abhängigkeit von der Problemgröße ermitteln.

Setzen wir in der Formel K=1 ein, so finden wir exakt das Amdahlsche Gesetz wieder.

Etwas komplizierter, jedoch als Beweis der Ähnlichkeit zu Amdahl wichtig, ist die Herleitung der Formel aus dem Amdahlschen Gesetz. Sie lässt sich folgendermaßen darstellen:

$$P = \frac{\text{parallelisierbarer Teil der Laufzeit}}{\text{gesamte Laufzeit}} = \frac{K \cdot P'}{(1 - P') + K \cdot P'}$$

Zunächst muss P (der Anteil des parallelisierbaren Teils an der Gesamtlaufzeit) ermittelt werden.

$$\begin{aligned} \text{Zeitgewinn} &= \frac{1}{(1 - P) + \frac{P}{N}} = \frac{1}{\left(1 - \frac{K \cdot P'}{(1 - P') + K \cdot P'}\right) + \frac{\frac{K \cdot P'}{(1 - P') + K \cdot P'}}{N}} \\ &= \frac{(1 - P') + K \cdot P'}{\left((1 - P') + K \cdot P' - K \cdot P'\right) + \frac{K \cdot P'}{N}} = \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}} \end{aligned}$$

Anschließend kann die Formel P in das Amdahlsche Gesetz anstelle des P's eingesetzt werden. Somit sind die Herleitung und der Beweis vollendet. Es zeigt sich, dass wir dieselbe Formel erhalten.

1.1.3 Parallelrechner

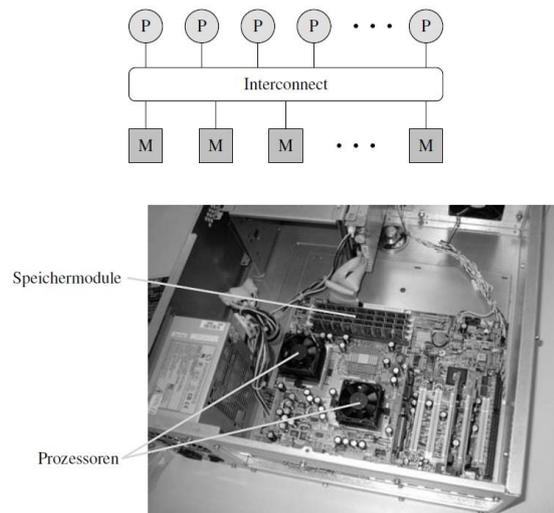
Parallelrechner werden heutzutage als Skalar- oder Vektorrechner konzipiert. Sie basieren auf unterschiedlichen Prozessorarchitekturen. Vektorprozessoren (auch Array-Prozessoren genannt) führen eine Berechnung gleichzeitig auf vielen Daten (in einem Vektor bzw. Array) aus. Skalarprozessoren können hingegen nur ein Operandenpaar pro Befehl bearbeiten. Skalarrechner basieren daher oft auf Tausenden von Standardprozessoren, die miteinander verbunden sind, dies wird als Rechnerverbund (engl. Computercluster, auf Deutsch übersetzt = Rechnerhaufen) bezeichnet. Siehe <http://de.wikipedia.org/wiki/Supercomputer>

Gemeinsamer oder verteilter Speicher

Nach dem von-Neumann-Modell besteht ein Rechner aus einem Prozessor und einem Speicher. Bei Parallelrechnern gibt es zwei grundsätzlich verschiedene Varianten: Bei Systemen mit gemeinsamem Speicher teilen sich alle Prozessoren einen Speicher, bei Systemen mit verteiltem Speicher verfügt jeder Prozessor über einen eigenen Speicher. (Bauke & Mertens, 2005, S. 21)

Gemeinsamer Speicher

Rechner mit einem gemeinsamen Speicher (engl. shared memory) stellen mehrere Speichermodule zu einem einheitlichen Adressraum dar, auf den alle Prozessoren zugreifen können (Siehe Abbildung 11). Der Zugriff auf die Speichermodule erfolgt über eine Verbindung untereinander (engl. Interconnect oder Interconnection). Rechner mit einem gemeinsamen Speicher werden Multiprozessoren genannt. (Bauke & Mertens, 2005, S. 21)



**Abbildung 31: Parallelrechner mit gemeinsamem Speicher (shared memory).
Abstraktes Modell (oben) und realer Rechner (unten)**

Quelle: (Bauke & Mertens, 2005, S. 22)

Verteilter Speicher

Bei Rechnern mit einem verteilten Speicher (engl. distributed memory) hat jeder Prozessor einen eigenen lokalen Speicher, auf den er zugreifen kann (Siehe Abbildung 12). Die Kommunikation zwischen den Prozessoren erfolgt über ein Netzwerk. Der Datentransport über Netzwerke ist in der Regel viel langsamer als der Datentransport zwischen CPU und Speicher. Je ein Prozessor und sein lokaler Speicher bilden einen Knoten (engl. node) des Parallelrechners. Da ein Knoten nach dem von-Neumann'schen Sinn ein Rechner ist, nennt man Rechner mit einem verteilten Speicher auch Multicomputer. (Bauke & Mertens, 2005, S. 23)

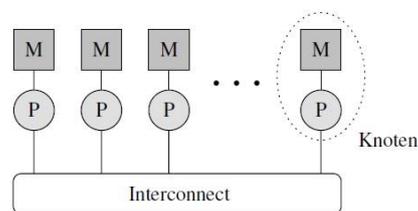


Abbildung 32: Parallelrechner mit verteiltem Speicher (distributed memory)

Quelle: (Bauke & Mertens, 2005, S. 23)

Damit Multicomputer eine hohe Rechenleistung erreichen, benötigt man also ein Hochleistungsnetzwerk, damit die Knoten sehr schnell miteinander kommunizieren können. Im Nachfolgenden Abschnitt wird das Netzwerk kurz beschrieben.

Netzwerk

Bei Parallelrechnern mit einem verteilten Speicher ist das Netzwerk eine Leistungsbestimmende Größe. Das Netzwerk ist Abhängig von der Bandbreite (engl. Bandwidth), Latenz (engl. Latency) und der Topologie bzw. Struktur des Netzwerks. (Bauke & Mertens, 2005, S. 24)

1. Bandbreite

Die Bandbreite gibt die Rate an, mit der Daten von einem Knoten zu einem anderen übertragen werden können. Die Angabe erfolgt in Bit/s oder Byte/s (Bauke & Mertens, 2005, S. 24)

2. Latenz

Die Latenz ist die Zeit, die vergeht, bis das erste Bit vom Empfänger zum Sender gelangt. (Bauke & Mertens, 2005, S. 24)

3. Topologie

Die Topologie bzw. Struktur eines Netzwerks gibt Aufschluss über die Verbindung zwischen den Knoten. Bauke und Mertens haben dazu in Ihrem Buch einige Topologien dargestellt (Siehe Abbildung 13), wie ein solches Netzwerk aufgebaut werden kann. (Bauke & Mertens, 2005, S. 24-25)

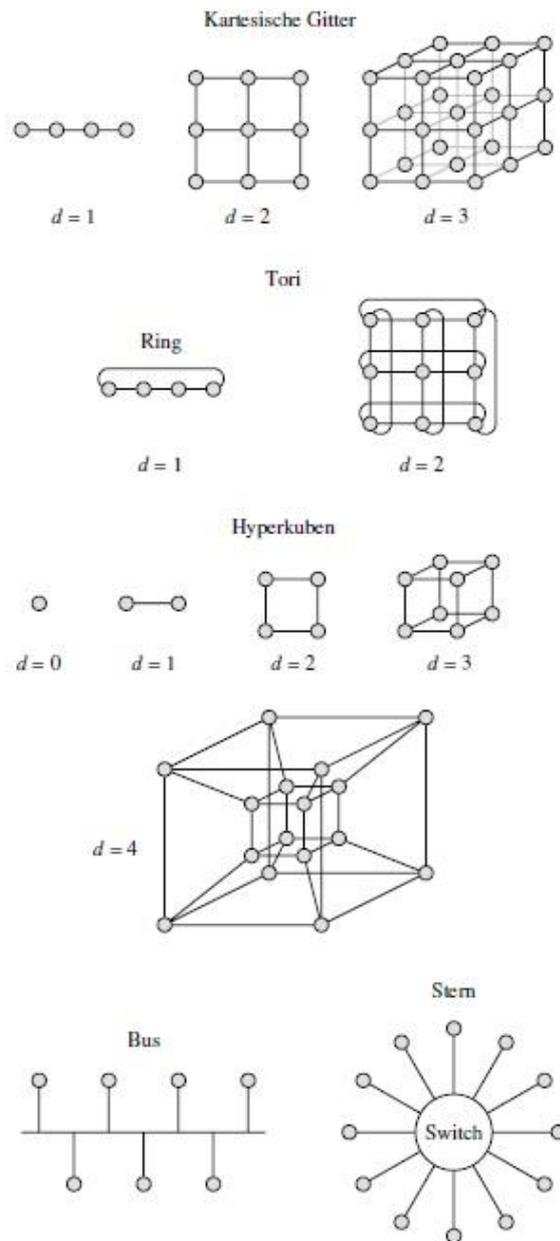


Abbildung 33: Topologien für Netzwerke

Quelle: (Bauke & Mertens, 2005, S. 25)

Laut Bauke und Mertens ist die Sternform am häufigsten verbreitet:

„Die meisten modernen lokalen Vernetzungen von PCs sind switched Ethernets, ganz gleich, ob es sich dabei um das Netz handelt, das Bürocomputer mit dem Abteilungsdrucker verbindet oder um das Netz, das aus einem Haufen Workstations einen parallelen Supercomputer macht.“ (Bauke & Mertens, 2005, S. 27)

1.2. Cluster Computer

Nach Bauke und Mertens ist ein Cluster oberflächlich betrachtet lediglich eine Ansammlung von Standard-Rechnern. Werden diese Rechner jedoch durch ein Netzwerk miteinander verbunden und mit entsprechender Cluster-Software ausgestattet, dann wird daraus ein Parallelrechner von Supercomputerformat. Dass die Bezeichnung Supercomputer keine Übertreibung ist, zeigt ein Blick in die aktuelle Top-500-Liste (Siehe <http://www.top500.org/>) der schnellsten Computer der Welt. (Bauke & Mertens, 2005, S. 27)

Aufgrund dessen, dass heutzutage überwiegend Cluster-Computer eingesetzt werden, wie aus dem Diagramm der Top-500-Liste hervorgeht (Siehe Abbildung 14), betrachten wir den Parallelrechner mit einem verteilten Speicher (Cluster-Computer) noch genauer und gehen nicht weiter auf den Parallelrechner mit einem gemeinsamen Speicher ein.

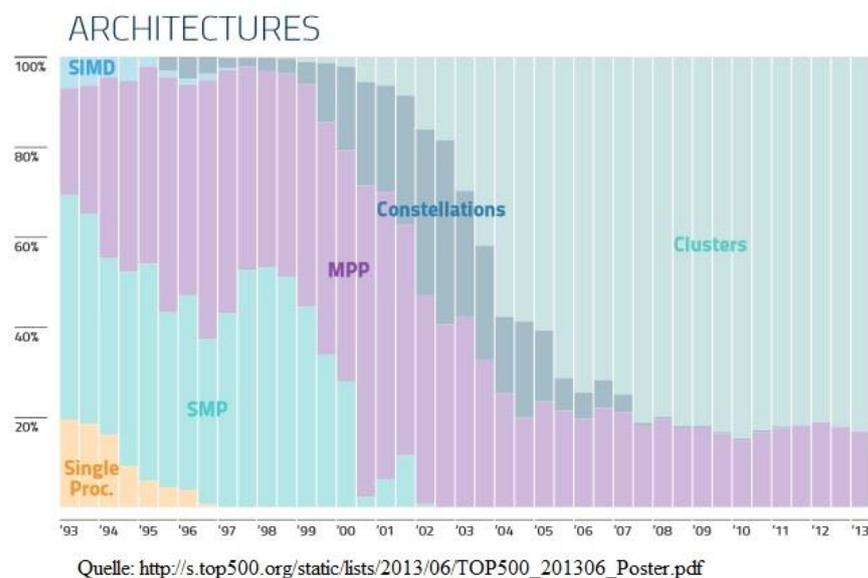


Abbildung 34: Cluster Architektur wird immer wichtiger

Cluster-Computer sind deshalb so beliebt, weil sie sehr kostengünstig sind in ihrer Anschaffung. Man benötigt für einen Cluster-Computer lediglich ein paar normale Rechner, dazu ein geeignetes Netzwerk und die entsprechende Software und schon besitzt man einen eigenen Parallelrechner (Cluster-Computer). Welche genauen Hardwarekomponenten man für einen Cluster-Computer benötigt und welche Software geeignet ist wird in den beiden nachfolgenden Kapiteln erläutert.

1.2.1 Hardware

Cluster-Computer sind sehr unterschiedlich, doch die Grundkomponenten sind bei allen gleich. Ein Cluster-Computer besteht aus Knoten (engl. Nodes), jeder Knoten besitzt eine CPU und ein paar dazugehörige Speicher. Die Knoten sind über ein Hochgeschwindigkeitsnetzwerk miteinander verbunden. (Bauke & Mertens, 2005, S. 51)

CPU und Speicher

Jede CPU hat drei (meistens) interne Speicher, die „Cache 1, Cache 2 und Cache 3“ genannt werden und zusätzlich noch einen, außerhalb der CPU angebrachten, Arbeits- bzw. Hauptspeicher, der wenige Gigabyte beträgt. Diese Speicher sind, in der eben genannten Reihenfolge, hierarchisch angeordnet. Die Hierarchie sagt aus, dass falls eine CPU nur den Cache benötigt schneller rechnen kann als wenn der Cache vollständig genutzt wird und die CPU zusätzlich auf den Arbeitsspeicher zugreifen muss. Aufgrund dessen, dass der Arbeitsspeicher nicht in der CPU enthalten ist, benötigt die CPU länger, um auf den Arbeitsspeicher zuzugreifen. (Bauke & Mertens, 2005, S. 62)

Als letzter Speicher in der Hierarchie befindet sich der Massen- bzw. Festplattenspeicher oder auch als externer Speicher bekannt. Dieser Massenspeicher hat in der Regel eine sehr große Speicherkapazität bis in den Terrabyte oder Petabyte Bereich hinein. Aufgrund der externen Massenspeicherlagerung benötigt die CPU eine wesentlich höhere Zeit, um gewisse Daten in den virtuellen Speicher (engl. virtual memory) auszulagern, deshalb ist es wichtig Cache und Arbeitsspeicher mit großem Speichervolumen zu verwenden, damit die CPU ihre Berechnungen mit der höchsten Leistung umsetzen kann. (Bauke & Mertens, 2005, S. 62)

Hochleistungsnetzwerk

Für ein Hochleistungsnetzwerk wird meistens ein Switch (allg. engl. Network device) bzw. Switch-Netzwerk (engl. Switched Network) mit einer Stern-Topologie eingesetzt. Aber für ein Netzwerk benötigt man nicht nur einen Switch, sondern auch Netzwerkkarten (engl. Interface), wovon mindestens eine Netzwerkkarte in einem Knoten enthalten ist und Netzwerkkabel (engl. Network links). (Bauke & Mertens, 2005, S. 54)

Zugehörig zu den Hardware-Netzwerkkomponenten benötigt man auch bestimmte Netzwerktechnologien, um die Daten zwischen verschiedenen Komponenten auszutauschen. Als Beispiel nennen wir fünf relevante Netzwerktechnologien:

- Gigabit Ethernet
- InfiniBand
- SCI
- Myrinet-2000
- QsNet

Für eine detailliertere Beschreibung bzw. Erläuterung der einzelnen Netzwerktechnologien sehen Sie sich Kapitel 4.3 aus dem Buch von Bauke und Mertens an. (Bauke & Mertens, 2005, S. 53)

Knoten

Knoten unterteilt man in zwei Arten, zum Einen gibt es Server Knoten (engl. Server Nodes bzw. front end node) und zum Anderen berechnende Knoten (engl. Compute Nodes). Die compute nodes sind die Arbeiter eines Clusters, auf ihnen werden sämtliche Operationen ausgeführt. Die server nodes hingegen stellen eine für den Clusterbetrieb notwendige Infrastruktur zur Verfügung. Über den server node meldet man sich im Cluster an und dieser leitet dann die Operationen über den Switch weiter an die compute nodes (Siehe Abbildung 15). (Bauke & Mertens, 2005, S. 51)

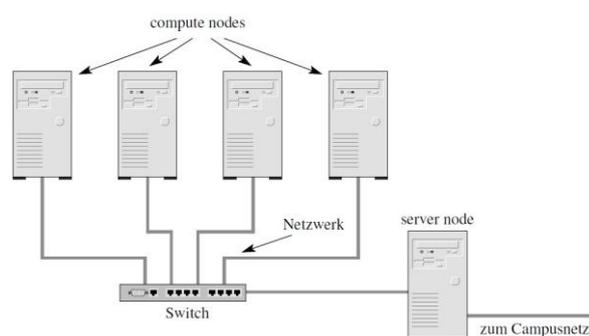


Abbildung 35: Grundlegender Aufbau eines kleinen Clusters

Quelle: (Bauke & Mertens, 2005, S. 52)

1.2.2 Software

Nachdem die Hardware für einen Cluster Computer dargestellt wurde, gehen wir nun auf die benötigte Software ein. Aufgrund dessen, dass es sehr viel Software auf dem

Markt gibt, gehen wir hier auf die für uns relevante Software ein. Wir werden das Dateisystem, die Benutzerverwaltung, das Batch-System und das Scheduling näher betrachten.

Dateisystem

Ein Dateisystem (engl. File System) in einem Cluster ist dafür zuständig, dem Nutzer, der gerade auf das Cluster zugriff hat, eine transparente Datensicht zu gewährleisten. Dies bedeutet, dass der Nutzer, der zum Beispiel auf den Datenbestand von Knoten Nr. 1 zugreift, auf den gleichen Datenbestand von zum Beispiel Knoten Nr. 5 zugriff hat. Falls Änderungen an einer Datei vorgenommen wurden, sind diese sofort auf allen anderen Knoten sichtbar (transparent). Außerdem ermöglichen Clusterdateisysteme den parallelen Anwendungen effizient Daten zu lesen und zu schreiben. Zusätzlich gewährleisten Dateisysteme ein hohes Maß an Redundanz und/oder besitzen weitere spezielle Funktionen. (Bauke & Mertens, 2005, S. 64)

Nachfolgend werden kurz vier Dateisysteme vorgestellt:

- NFS (Network File System)
- PVFS (Parallel Virtual File System)
- Lustre (Clusterdateisystem)
- InterMezzo (Clusterdateisystem)

Für detailliertere Erläuterungen zu den oben genannten Dateisystemen finden Sie im Kapitel 4.6.2 aus dem Buch von Bauke und Mertens. (Bauke & Mertens, 2005, S. 64-65)

Batch-System

Ein Batch-System (Batch lautet auf Deutsch Stapel, Partie oder Ladung) hat die Aufgabe, Aufträge bzw. Operationen (auch „Jobs“ genannt) anzunehmen und diese auf einen oder mehrere Knoten des Clusters zu starten, aber nur wenn genügend Ressourcen auf dem Cluster zur Verfügung stehen. (Bauke & Mertens, 2005, S. 151)

Ein sehr gutes analoges Beispiel für ein Batch-System geben Bauke und Mertens:

„In der Schalterhalle einer (altmodischen) Bank warten zahlreiche Kunden. Jeder dieser Kunden möchte eine andere Dienstleistung in Anspruch nehmen. Ein Kunde

möchte eine Überweisung tätigen, ein anderer sich Geld auszahlen lassen, ein weiterer Kunde hat einen Termin für ein Beratungsgespräch mit seinem Kundenberater. An den Schaltern werden unterschiedliche Dienste angeboten. An einigen Schaltern können alle möglichen Bankgeschäfte abgewickelt werden, am Expressschalter werden nur Auszahlungen getätigt und ein Schalter ist besonders wichtigen Kunden mit einer gewissen Mindestanlage vorbehalten. Würde ein Batch-System die Schalterhalle verwalten, so liefе dies in etwa wie folgt ab:

- *Wenn ein Kunde die Schalterhalle betritt, so muss er sich ausweisen und angeben, welche Art von Bankgeschäft er tätigen möchte. Das Batch-System vermerkt diese Angaben zusammen mit dem Zeitpunkt, an dem der Kunde die Bank betrat.*
- *Jedem Kunden wird eine Priorität zugewiesen. Je länger der Kunde schon wartet, desto höher seine Priorität. Aber auch persönliche Merkmale können seine Priorität beeinflussen. Zum Beispiel könnte eine Bank Geschäftskunden eine höhere Priorität als Privatkunden einräumen.*
- *Sobald ein Schalter frei ist, wird ihm vom Batch-System der Kunde mit der größten Priorität zugewiesen, dessen Bedürfnissen der Schalter gerecht wird.*

In dieser Analogie entsprechen die Kunden den Jobs und die Schalter den Knoten eines Clusters. Die Jobs können verschiedene Ansprüche an Arbeitsspeicher, Anzahl der CPUs oder andere Merkmale der Hard- oder Softwarekonfiguration der Knoten haben.“ (Bauke & Mertens, 2005, S. 151-152)

Es ist noch zu erwähnen, dass die Jobs, die an das Batch-System geschickt werden, in eine Warteschlange (engl. Queue genannt) gepackt werden, wie die Warteschlange vor den Bankschaltern. (Bauke & Mertens, 2005, S. 152)

Scheduling

Beim Scheduling (deutsch Ablaufplanung) entscheidet der Scheduler (deutsch Ablaufplaner bzw. Zeitplaner) wann, welcher Job auf welchem Knoten des Clusters gestartet wird. Es gibt dabei mehrere Strategien, damit das Cluster optimal ausgenutzt wird. Diese Strategien werden hier kurz beschrieben. (Bauke & Mertens, 2005, S. 152)

First-Come-First-Served

Die „First-Come-First-Served“ Strategie (deutsches Sprichwort: „Wer zuerst kommt, malt zuerst“) ist die einfachste Scheduling-Strategie. Bei dieser Strategie werden die Jobs der Reihe nach abgearbeitet. (Bauke & Mertens, 2005, S. 152)

Backfill

Die Backfill-Strategie (deutsch Hinterfüllung) versucht die Auslastung des Clusters zu optimieren, indem freie CPUs auch mit kleineren Jobs belegt werden. (Bauke & Mertens, 2005, S. 152)

Fairshare

Bei der Fairshare-Strategie (deutsch fairer Anteil) wird der Ressourcenverbrauch der Vergangenheit mit berücksichtigt, das bedeutet, wenn ein Nutzer (engl. User) sehr oft einen Job in die Queue stellt, dann muss er irgendwann seine anderen User-Kollegen vorlassen, die in der Vergangenheit weniger Ressourcen verbraucht haben als er selber. (Bauke & Mertens, 2005, S. 152)

Exclusive

Die Exclusive-Strategie (deutsch Exklusiv oder ausschließlich) gewährt pro CPU nur einen Job, um die Durchlaufzeit zu minimieren. (Bauke & Mertens, 2005, S. 153)

Preemption

Eine Preemption-Strategie (deutsch Vorkauf) besitzt die Fähigkeit, Jobs für kurze Zeit zu beenden, um zwischenzeitlich andere Jobs durchlaufen zu lassen. Danach werden die beendeten Jobs wieder gestartet. Eine Grundvoraussetzung für Preemption ist das Checkpointing, für nähere Details zum Checkpointing sehen Sie im Buch von Bauke und Mertens nach (Kapitel 6.2.3). (Bauke & Mertens, 2005, S. 153)

Nach Bauke und Mertens werden Batch-Systeme in der Praxis meistens kombiniert. (Bauke & Mertens, 2005, S. 153)

1.3. Wie werden Supercomputern miteinander verglichen?

Es stellt sich die Frage, wie die Leistungsfähigkeit eines Supercomputers untersucht werden kann. Hierfür haben sich standardisierte Messverfahren etabliert, die zum einen Auskunft über das Leistungsvermögen eines Systems geben, zum anderen als Messgröße für ein Ranking, der weltweit im Betrieb befindlichen Supercomputer, dienen. Früher wurden die Supercomputer anhand der Messgröße OPS, Operationen bzw. Befehle pro Sekunde, bewertet, heutzutage hingegen in FLOPS, Gleitkommaoperationen pro Sekunde. Die Maßeinheit FLOPS unterteilt sich dabei wie folgt:

Abkürzung	Name	FLOPS
KFLOPS	KiloFLOPS	1.000
MFLOPS	MegaFLOPS	1.000.000
GFLOPS	GigaFLOPS	1.000.000.000
TFLOPS	TeraFLOPS	1.000.000.000.000
PFLOPS	PetaFLOPS	1.000.000.000.000.000
EFLOPS	ExaFLOPS	1.000.000.000.000.000.000
ZFLOPS	ZettaFLOPS	1.000.000.000.000.000.000.000
YFLOPS	YottaFLOPS	1.000.000.000.000.000.000.000.000

Tabelle 2: Einheiten der Gleitkommarechenleistung mit Vorsätzen nach SI

Das Ranking der Supercomputer findet Ausdruck in einer, seit 1978 erstellten und ständig aktualisierten, Weltrangliste der 500 Leistungsstärksten Supercomputer (<http://www.top500.org>). Zusätzlich gibt es noch eine Liste der ökologischsten Computer (Green 500), diese Liste wurde im Jahre 2005 das erste Mal erstellt und fortlaufend angepasst (<http://www.green500.org>).

Quellenverzeichnis zu Anhang 1

Bauke, H., & Mertens, S. (2005). Cluster Computing - Praktische Einführung in das Hochleistungsrechnen auf Linux-Clustern. Magdeburg: Springer.

Claus, Prof. Dr. Volker, D. A. (2. vollständig überarbeitete und erweiterte Auflage). DUDEN Informatik / Ein Sachlexikon für Studenten und Praxis. Herausgeber: Lektorat des B.I.-Wissenschaftsverlags: DUDENVERLAG Mannheim, Leipzig, Wien, Zürich.

Claus, Prof. Dr. Volker, D.-I. A. (kein Datum). SCHÜLERDUDEN - Die Informatik. Bibliographisches Institut Mannheim/Wien/Zürich Dudenverlag.

Fischer, Peter, P. H. (14. überarbeitete Auflage). Lexikon der Informatik. Springer.

Grundlagen der Informatik (Letzter Aufruf am 29.10.2013, um 16:37 Uhr)

<http://exe.fam-trachsel.ch/InfoGL/bussystem.html>

Sciengines (Letzter Aufruf am 29.10.2013, um 16:30 Uhr)

<http://www.sciengines.com/copacobana/index.html>

Uni Leipzig (Letzter Aufruf am 29.10.2013, um 16:27 Uhr)

<http://dbs.uni-leipzig.de/buecher/mrddb/index.html>

Wikipedia/Parallelrechner (Letzter Aufruf am 29.10.2013, um 16:22 Uhr)

<http://de.wikipedia.org/wiki/Parallelrechner>

Wikipedia/Supercomputer (Letzter Aufruf am 29.10.2013, um 16:37 Uhr)

<http://de.wikipedia.org/wiki/Supercomputer>

Anhang 2: Exkurs Benchmark

Inhaltsverzeichnis

1 Benchmark.....	84
1.1 Anwendungsbenchmarks	87
1.2 Leistungsdaten.....	87
1.3 Messung der Leistung	89
1.4 Komponenten- und Systemleistung	89
1.5 Abgrenzung Lasttest und Benchmark	90
Quellenverzeichnis zu Anhang 2	91

1 Benchmark

Benchmarks sind genormte Mess- und Bewertungsverfahren, mit denen man die Rechenleistungen von EDV-Systemen ermitteln kann. Die Ergebnisse der verschiedenen Systeme kann man dann miteinander vergleichen. Meistens werden die Benchmarks zur Ermittlung der Hardwareleistung durchgeführt. Es gibt allerdings auch so genannte Softwarebenchmarks, mit deren Hilfe man die Leistung von Programmiersprachen, mit Interpretern sowie Compilern und deren Laufzeitsystemen auf derselben Hardware vergleichen kann. Man kann sagen, dass ein Benchmark ein Programm ist, das ein meistens aus der Mathematik stammendes, kompliziertes Problem löst. Beispiele sind das Lösen linearer Gleichungssysteme oder die Berechnung von Mersenne-Primzahlen.

Beispiele für Benchmarks sind:

- LINPACK
- Whetstone
- Dhrystone
- Standard Performance Evaluation Corporation
- HPCC

Bei sogenannten I/O Benchmarks wird die I/O Performance gemessen. Hierzu zählen beispielsweise die Schreib –und Leseperformance von SSDs, Flashspeicher oder mechanischen Festplatten. Hierbei kann die Messung auch mit einem RAID-Controller durchgeführt werden. Um die Tests von I/O Benchmarks nachvollziehen zu können, sollte man die Testumgebung genau dokumentieren. Hierzu zählen die relevanten Hardwarekomponenten (Mainboard inkl. BIOS-Version, CPU, Arbeitsspeicher, RAID-Controller inkl. Firmware-Version, Speichermedien (Festplatten, SSDs, ...) inkl. Firmware-Versionen etc.) sowie relevante Softwarekomponenten (BIOS-Konfiguration (am besten Load Defaults und anschließend etwaige Konfigurationsänderungen dokumentieren), Betriebssystem mit genauen Infos zur Installation, Software-Versionen, etc., genaue Konfiguration des RAID-Controllers (RAID-Level, Cache-Konfiguration, ...), Testtools mit genauer Versionsangabe ect.).

Bei der Messung der I/O Performance gibt es einige Dinge, die man beachten sollte.

Bei der Messung der **SSD** und **Flashspeicher** Performance:

Gleicher Ausgangszustand:

Je gefüllter eine SSD-Platte oder ein Flashspeicher ist, desto unterschiedlicher ist die Leistung dieser Speicher. Eine leere SSD ist meistens langsamer als eine fast Vollgeschriebene. Daher sollte man vor der Messung darauf achten, dass die Speicher vorher vollständig gelöscht wurden oder zuvor mit nachvollziehbaren Datenmengen beschrieben wurden.

Gleiche Cache-Einstellungen:

Beim Vergleich mehrere SSD muss man drauf achten, dass die Cache Einstellungen der SSDs möglichst gleich ist.

Mit Random-Daten testen:

Manche SSD-Controller haben eine Datenkompression integriert. Ein Testen mit NULL-Daten kann dabei zu sehr hohen Messwerten führen, die mit echten Daten aber nicht erreicht werden. Daher sollte man den Test mit zufälligen Daten ausführen.

Bei der Messung **mechanischer Festplatten**:

Gleiche Bereiche von Festplatten vergleichen:

Man sollte darauf achten, bei Vergleich mehrerer Festplatten immer die gleichen Bereiche der Festplatten zu vergleichen, da bei konstanter Rotationsgeschwindigkeit der Datendurchsatz am äußeren Rand höher ist als am Inneren. Da Festplatten von außen nach innen beschrieben werden, ist der Durchsatz am Beginn des Datenbereichs somit höher als am Ende des Datenbereichs der Festplatte.

Gleiche Festplatten Schreibcache-Einstellungen:

Beim Vergleich mehrere Festplatten muss man drauf achten, dass die Cache Einstellungen der Platten möglichst gleich ist.

Keine laufenden SMART-Tests:

Man sollte darauf achten, dass während des I/O Benchmarks keine Selbsttests der Festplatte durchgeführt werden, da sie die Performance beeinflussen können und somit das Ergebnis verfälscht werden könnte.

LINPACK:

LINPACK ist im eigentlichen Sinne eine numerische Programmbibliothek zum Lösen von Gleichungssystemen. Im späteren Verlauf wurde der Name allerdings auch für ein Programm verwendet, welches die Geschwindigkeit eines Computers ermittelt (=> Benchmark).

Da bei Supercomputern und den dort häufig implementierten impliziten Lösungsverfahren, das Lösen von linearen Gleichungssystemen eine zentrale Rolle spielt wird LINPACK zur Leistungsmessung von Supercomputern eingesetzt. Das Ergebnis wird in Floating Point Operations Per Second angegeben und in der TOP500-Liste der schnellsten Supercomputer veröffentlicht. Fortran ist die Sprache, die meistens verwendet wird, um LINPACK Programme zu schreiben, es gibt aber auch Versionen in den Programmiersprachen C, C++, Pascal, Java und anderen.

Ein LINPACK Benchmark besteht aus drei Teilen:

Der erste Teil besteht aus einer 100x100 Matrix, wobei die benötigte Zeit für das LGS gemessen wird und der Lösungsalgorithmus vorgegeben ist. Der zweite Teil besteht aus einer 1000x1000 Matrix, wobei die Flop/s gemessen werden. Beim zweiten Teil darf außerdem der Lösungsalgorithmus optimiert werden und die Lösung muss die vorgeschriebene Genauigkeit einhalten. Den dritten Teil nennt man HPL (Highly Parallel Linpack) Benchmark. Hier gibt es eine beliebige Problemgröße, wobei der Lösungsalgorithmus optimiert werden darf. Die Anzahl der Operationen muss $\frac{2}{3}n^3 + O(n^2)$ Flops entsprechen. Die Standard Operationenanzahl muss dem einer LU Faktorisierung entsprechen, wofür der Strassen-Algorithmus ausgeschlossen ist. Die Lösung muss die vorgeschriebene Genauigkeit einhalten. Dieser Benchmark wird für die TOP500 benutzt.

Kritik an LINPACK

Vor allem Systeme, die mit relativ geringen Speicherbandbreiten und –Kapazitäten ausgestattet sind, sind dank ihrer reinen CPU-Leistung, oft an der Spitze der Liste zu finden. Allerdings sind sie nur für spezielle Probleme, nicht aber für umfangreiche, realitätsnahe Simulationen verwendbar.

1.1 Anwendungsbenchmarks

Eine weitere Form von Benchmarks stellen die Anwendungsbenchmarks dar. Hier wird die Leistungsfähigkeit von Rechnersystemen beim Zusammenspiel aller Hardware- und Softwarekomponenten beurteilt. Bei diesen Benchmarks werden auch Größen wie Energieverbrauch und Akkulaufzeit bei mobilen Geräten ermittelt und verglichen. Um Anwendungsbenchmarks durchzuführen werden realitätsnahe Anwendungsszenarien erstellt, wobei die Laufzeit dieser Szenarien einen einheitlichen Vergleich verschiedener Rechnersysteme erlaubt. Beispiele hierfür sind:

- SysMark
- MobileMark
- EEcoMark
- 3DMark
- PCMark
- PowerMark

1.2 Leistungsdaten

1. Instructions per cycle

Bezeichnet die Anzahl der von einem Prozessor in einem Taktzyklus ausführbaren Befehle. Dadurch, dass die Anzahl der ausgeführten Befehle pro Taktzyklus bei den meisten Prozessorarchitekturen variiert, handelt es sich hierbei in der Regel um einen Mittelwert. Einfache Additionen können z. B. schneller ausgeführt werden als eine Menge von Gleitkommabefehlen. Zum Berechnen wird meist eine große Anzahl von Befehlen ausgeführt und durch die benötigten Taktzyklen, welche sich aus der Prozessortaktung und der benötigten Zeit ergeben dividiert.

2. Instructions per second

Die Einheit gibt an, wie viele Maschinenbefehle (Instruktionen) ein Mikroprozessor pro Sekunde ausführen kann. 1 MIPS bedeutet, er kann eine Million Maschinenbefehle pro Sekunde ausführen.

3. Floating Point Operations Per Second

Floating Point Operations Per Second (kurz FLOPS; englisch für Gleitkommaoperationen pro Sekunde) bezeichnet die Anzahl der Gleitkommazahl-

Operationen (Additionen oder Multiplikationen), die von einem Computer pro Sekunde ausgeführt werden kann. Diese Leistungsgröße wird insbesondere für Supercomputer verwendet, da im Einsatzgebiet Hochleistungsrechnen diese Operationen eine wichtige Rolle spielen.

4. Datenübertragungsrate

Die Datenübertragungsrate gibt die Gesamtmenge der Daten (Nutzdaten und Steuerdaten) pro Zeit an, die übertragen werden kann.

5. Datendurchsatz

Der Datendurchsatz gibt die Menge der Nutzdaten pro Zeit an, die übertragen werden kann.

6. Antwortzeit

Die Antwortzeit ist die Zeitspanne zwischen dem Absenden einer Nachricht am Computer und dem Empfang des ersten Zeichens der zugehörigen Antwort am Terminal. Die Antwortzeit setzt besteht aus der Datenübertragungsrate in den Übertragungsmedien der Wartezeit und der Bearbeitungszeit der Nachricht.

Die Antwortzeit kann durch folgende Faktoren beeinflusst werden:

- Bei der Datenübertragung: Netzleistung (Effizienz) und Netzlast
- Bei der Wartezeit: Rückstau bei hoher Anfragerate
- Bei der Bearbeitungszeit: Leistung der Rechner und des Datenbanksystems, Qualität des Datenbankdesigns und der Programmierung.

7. Response ratio

Verhältnis von Bearbeitungszeit zur Antwortzeit.

Response ratio = Antwortzeit / Dauer für Funktionsbearbeitung = (Wartezeit + Dauer für Funktionsbearbeitung) / Dauer für Funktionsbearbeitung

8. Frames per second

Wird als Leistungskennzahl von Grafikkarten verwendet für die Anzahl der ausgegebenen Bilder pro Sekunde.

9. Latenzzeit

Ausdruck für Reaktionszeit (Auftragsbearbeitungszeit nach einem Interrupt) mit der Nebenbedingung, dass ein bestimmter oberer Wert nicht überschritten wird.

10. Zugriffszeit

Bei einem Speichermedium ist die Zugriffszeit die Zeit zwischen dem Eintreffen eines Schreib- oder Lesebefehls und dem Beginn des entsprechenden Vorgangs.

1.3 Messung der Leistung

Es wird zwischen Software-Messung und Hardware-Messung unterschieden.

Hardware-Messung

Anbringung von Messfühlern direkt ans Messobjekt, welche die entsprechenden Daten übertragen. Hier entsteht keine Beeinträchtigung des Ablaufs des Objektrechners (Rechner, bei dem die Leistung gemessen wird).

Software-Messung

Hier wird ein Messprogramm auf dem Rechner installiert, welches die gewünschten Informationen über eine Standardschnittstelle überträgt. Da die Messprogramme unabhängig von der Hardware arbeiten, sind nur minimale Kenntnisse des Objektrechners nötig und die Messprogramme sind auf nahezu allen Rechnern lauffähig. Es werden allerdings zusätzliche Ressourcen verbraucht, da der Programmablauf auf dem Rechner verändert wird. Das dynamische Verhalten des Objektrechners wird verfälscht.

1.4 Komponenten- und Systemleistung

Komponentenleistung

Bei der Komponentenleistung geht es um die einzelnen Komponenten eines Computers, wie beispielsweise der CPU, den Hauptspeicher oder der Grafikkarte.

Systemleistung

Mit Systemleistung wird die Leistung eines kompletten DV-Systems bezeichnet, das aus einer Vielzahl von Komponenten bestehen kann. Elemente des Systems können beispielsweise einzelne Softwarekomponenten, Rechnernetze oder spezielle Geräte sein.

1.5 Abgrenzung Lasttest und Benchmark

Während das Ziel des Lasttest es ist, einen Nachweis zu erbringen ob die zu erwartende Last in geforderter Zeit abgearbeitet werden kann, ist es das Ziel des Benchmarks eine Kennzahl zu ermitteln die zwischen verschiedenen Systemen verglichen werden kann.

Quellenverzeichnis zu Anhang 2

Wikipedia/Benchmark (Letzter Zugriff am 29.10.2013, um 16:41 Uhr)

[http://de.wikipedia.org/wiki/Benchmark_\(Computer\)](http://de.wikipedia.org/wiki/Benchmark_(Computer))

Wikipedia/Lasttest (Letzter Zugriff am 29.10.2013, um 16:42 Uhr)

[http://de.wikipedia.org/wiki/Lasttest_\(Computer\)](http://de.wikipedia.org/wiki/Lasttest_(Computer))

Wikipedia/Rechenleistung (Letzter Zugriff am 29.10.2013, um 16:44 Uhr)

<http://de.wikipedia.org/wiki/Rechenleistung>

Wikipedia/LINPACK (Letzter Zugriff am 29.10.2013, um 16:46 Uhr)

<http://de.wikipedia.org/wiki/LINPACK>

TK Server (Letzter Zugriff am 29.10.2013, um 16:48 Uhr)

http://www.thomas-krenn.com/de/wiki/Messen_von_I/O_Performance

Anhang 3: Hardwareliste eines Knoten des Testclusters

```
west1
  description: System
  product: X8DTL (To Be Filled By O.E.M.)
  vendor: Supermicro
  version: 1234567890
  serial: 1234567890
  width: 64 bits
  capabilities: smbios-2.6 dmi-2.6 vsyscall32
  configuration: boot=normal chassis=server family=Server sku=To Be Filled
By O.E.M. uuid=534D4349-0002-0290-2500-029025006F47
*-core
  description: Motherboard
  product: X8DTL
  vendor: Supermicro
  physical id: 0
  version: 1234567890
  serial: 1234567890
  slot: To Be Filled By O.E.M.
*-firmware
  description: BIOS
  vendor: American Megatrends Inc.
  physical id: 0
  version: 2.1a
  date: 12/30/2011
  size: 64KiB
  capacity: 4032KiB
  capabilities: isa pci pnp upgrade shadowing escd cdboot bootselect
socketedrom edd int13floppy1200 int13floppy720 int13floppy2880 int5printscreen
int9keyboard int14serial int17printer int10video acpi usb ls120boot zipboot
biosbootspecification
*-cpu:0
  description: CPU
  product: Intel(R) Xeon(R) CPU X5650 @ 2.67GHz
  vendor: Intel Corp.
  physical id: 4
  bus info: cpu@0
  version: Intel(R) Xeon(R) CPU X5650 @ 2.67GHz
  serial: To Be Filled By O.E.M.
  slot: CPU 1
  size: 1600MHz
  capacity: 1600MHz
  width: 64 bits
  clock: 133MHz
  capabilities: x86-64 fpu fpu_exception wp vme de pse tsc msr pae mce
cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss
ht tm pbe syscall nx pdpe1gb rdtscp constant_tsc arch_perfmon pebs bts
rep_good nopl xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64 monitor
ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 popcnt aes
lahf_lm ida arat epb dtherm tpr_shadow vnmi flexpriority ept vpid cpufreq
  configuration: cores=6 enabledcores=6 threads=12
*-cache:0
  description: L1 cache
  physical id: 5
  slot: L1-Cache
  size: 384KiB
  capacity: 384KiB
  capabilities: internal write-through instruction
*-cache:1
  description: L2 cache
  physical id: 6
  slot: L2-Cache
  size: 1536KiB
  capacity: 1536KiB
  capabilities: internal write-through unified
*-cache:2
  description: L3 cache
  physical id: 7
```

```

        slot: L3-Cache
        size: 12MiB
        capacity: 12MiB
        capabilities: internal write-back unified
*-cpu:1
  description: CPU
  product: Intel(R) Xeon(R) CPU X5650 @ 2.67GHz
  vendor: Intel Corp.
  physical id: 8
  bus info: cpu@1
  version: Intel(R) Xeon(R) CPU X5650 @ 2.67GHz
  serial: To Be Filled By O.E.M.
  slot: CPU 2
  size: 1600MHz
  capacity: 1600MHz
  width: 64 bits
  clock: 133MHz
  capabilities: x86-64 fpu fpu_exception wp vme de pse tsc msr pae mce
cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss
ht tm pbe syscall nx pdpe1gb rdtscp constant_tsc arch_perfmon pebs bts
rep_good nopl xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64 monitor
ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 popcnt aes
lahf_lm ida arat epb dtherm tpr_shadow vnmi flexpriority ept vpid cpufreq
  configuration: cores=6 enabledcores=6 threads=12
*-cache:0
  description: L1 cache
  physical id: 9
  slot: L1-Cache
  size: 384KiB
  capacity: 384KiB
  capabilities: internal write-through instruction
*-cache:1
  description: L2 cache
  physical id: a
  slot: L2-Cache
  size: 1536KiB
  capacity: 1536KiB
  capabilities: internal write-through unified
*-cache:2
  description: L3 cache
  physical id: b
  slot: L3-Cache
  size: 12MiB
  capacity: 12MiB
  capabilities: internal write-back unified
*-memory:0
  description: System Memory
  physical id: 2d
  slot: System board or motherboard
*-bank:0
  description: DIMM 1333 MHz (0.8 ns)
  product: NT2GC72B8PA0NL-CG
  vendor: Nanya
  physical id: 0
  serial: 1408C69C
  slot: P2_DIMM1A
  size: 2GiB
  width: 64 bits
  clock: 1333MHz (0.8ns)
*-bank:1
  description: DIMM 1333 MHz (0.8 ns)
  product: NT2GC72B8PA0NL-CG
  vendor: Nanya
  physical id: 1
  serial: 1508C1C5
  slot: P2_DIMM2A
  size: 2GiB
  width: 64 bits
  clock: 1333MHz (0.8ns)
*-bank:2
  description: DIMM 1333 MHz (0.8 ns)

```

```

        product: NT2GC72B8PA0NL-CG
        vendor: Nanya
        physical id: 2
        serial: 1408C128
        slot: P2_DIMM3A
        size: 2GiB
        width: 64 bits
        clock: 1333MHz (0.8ns)
*-memory:1
    description: System Memory
    physical id: 35
    slot: System board or motherboard
*-bank:0
    description: DIMM 1333 MHz (0.8 ns)
    product: NT2GC72B8PA0NL-CG
    vendor: Nanya
    physical id: 0
    serial: 1308B5C5
    slot: P1_DIMM1A
    size: 2GiB
    width: 64 bits
    clock: 1333MHz (0.8ns)
*-bank:1
    description: DIMM 1333 MHz (0.8 ns)
    product: NT2GC72B8PA0NL-CG
    vendor: Nanya
    physical id: 1
    serial: 1108C6AC
    slot: P1_DIMM2A
    size: 2GiB
    width: 64 bits
    clock: 1333MHz (0.8ns)
*-bank:2
    description: DIMM 1333 MHz (0.8 ns)
    product: NT2GC72B8PA0NL-CG
    vendor: Nanya
    physical id: 2
    serial: 1108A682
    slot: P1_DIMM3A
    size: 2GiB
    width: 64 bits
    clock: 1333MHz (0.8ns)
*-memory:2 UNCLAIMED
    description: Flash Memory
    physical id: 3d
    slot: System board or motherboard
    capacity: 4MiB
*-bank UNCLAIMED
    description: FLASH Non-volatile 33 MHz (30.3 ns)
    product: 26DF321
    vendor: ATMEL
    physical id: 0
    slot: BIOS
    size: 4MiB
    width: 8 bits
    clock: 33MHz (30.3ns)
*-memory:3 UNCLAIMED
    physical id: 1
*-memory:4 UNCLAIMED
    physical id: 2
*-pci:0
    description: Host bridge
    product: 5500 I/O Hub to ESI Port
    vendor: Intel Corporation
    physical id: 100
    bus info: pci@0000:00:00.0
    version: 22
    width: 32 bits
    clock: 33MHz
*-pci:0
    description: PCI bridge

```

```

        product: 5520/5500/X58 I/O Hub PCI Express Root Port 1
        vendor: Intel Corporation
        physical id: 1
        bus info: pci@0000:00:01.0
        version: 22
        width: 32 bits
        clock: 33MHz
        capabilities: pci msi pciexpress pm normal_decode bus_master
cap_list
        configuration: driver=pcieport
        resources: irq:0
*-pci:1
        description: PCI bridge
        product: 5520/5500/X58 I/O Hub PCI Express Root Port 3
        vendor: Intel Corporation
        physical id: 3
        bus info: pci@0000:00:03.0
        version: 22
        width: 32 bits
        clock: 33MHz
        capabilities: pci msi pciexpress pm normal_decode bus_master
cap_list
        configuration: driver=pcieport
        resources: irq:0
*-pci:2
        description: PCI bridge
        product: 5520/5500/X58 I/O Hub PCI Express Root Port 7
        vendor: Intel Corporation
        physical id: 7
        bus info: pci@0000:00:07.0
        version: 22
        width: 32 bits
        clock: 33MHz
        capabilities: pci msi pciexpress pm normal_decode bus_master
cap_list
        configuration: driver=pcieport
        resources: irq:0
*-pci:3
        description: PCI bridge
        product: 7500/5520/5500/X58 I/O Hub PCI Express Root Port 9
        vendor: Intel Corporation
        physical id: 9
        bus info: pci@0000:00:09.0
        version: 22
        width: 32 bits
        clock: 33MHz
        capabilities: pci msi pciexpress pm normal_decode bus_master
cap_list
        configuration: driver=pcieport
        resources: irq:0
*-generic:0 UNCLAIMED
        description: PIC
        product: 7500/5520/5500/X58 I/O Hub I/OxAPIC Interrupt Controller
        vendor: Intel Corporation
        physical id: 13
        bus info: pci@0000:00:13.0
        version: 22
        width: 32 bits
        clock: 33MHz
        capabilities: pm io_x_apic bus_master cap_list
        configuration: latency=0
        resources: memory:fec8a000-fec8afff
*-generic:1
        description: PIC
        product: 7500/5520/5500/X58 I/O Hub System Management Registers
        vendor: Intel Corporation
        physical id: 14
        bus info: pci@0000:00:14.0
        version: 22
        width: 32 bits
        clock: 33MHz

```

```

capabilities: pciexpress 8259 cap_list
configuration: driver=i7core_edac latency=0
resources: irq:0
*-generic:2 UNCLAIMED
description: PIC
product: 7500/5520/5500/X58 I/O Hub GPIO and Scratch Pad Regis-
ters
vendor: Intel Corporation
physical id: 14.1
bus info: pci@0000:00:14.1
version: 22
width: 32 bits
clock: 33MHz
capabilities: pciexpress 8259 cap_list
configuration: latency=0
*-generic:3 UNCLAIMED
description: PIC
product: 7500/5520/5500/X58 I/O Hub Control Status and RAS Regis-
ters
vendor: Intel Corporation
physical id: 14.2
bus info: pci@0000:00:14.2
version: 22
width: 32 bits
clock: 33MHz
capabilities: pciexpress 8259 cap_list
configuration: latency=0
*-generic:4 UNCLAIMED
description: PIC
product: 7500/5520/5500/X58 I/O Hub Throttle Registers
vendor: Intel Corporation
physical id: 14.3
bus info: pci@0000:00:14.3
version: 22
width: 32 bits
clock: 33MHz
capabilities: 8259
configuration: latency=0
*-generic:5
description: System peripheral
product: 5520/5500/X58 Chipset QuickData Technology Device
vendor: Intel Corporation
physical id: 16
bus info: pci@0000:00:16.0
version: 22
width: 64 bits
clock: 33MHz
capabilities: msix pciexpress pm bus_master cap_list
configuration: driver=ioatdma latency=0
resources: irq:43 memory:fbef8000-fbefbfff
*-generic:6
description: System peripheral
product: 5520/5500/X58 Chipset QuickData Technology Device
vendor: Intel Corporation
physical id: 16.1
bus info: pci@0000:00:16.1
version: 22
width: 64 bits
clock: 33MHz
capabilities: msix pciexpress pm bus_master cap_list
configuration: driver=ioatdma latency=0
resources: irq:44 memory:fbef4000-fbef7fff
*-generic:7
description: System peripheral
product: 5520/5500/X58 Chipset QuickData Technology Device
vendor: Intel Corporation
physical id: 16.2
bus info: pci@0000:00:16.2
version: 22
width: 64 bits
clock: 33MHz

```

```

capabilities: msix pciexpress pm bus_master cap_list
configuration: driver=ioatdma latency=0
resources: irq:45 memory:fbef0000-fbef3fff
*-generic:8
description: System peripheral
product: 5520/5500/X58 Chipset QuickData Technology Device
vendor: Intel Corporation
physical id: 16.3
bus info: pci@0000:00:16.3
version: 22
width: 64 bits
clock: 33MHz
capabilities: msix pciexpress pm bus_master cap_list
configuration: driver=ioatdma latency=0
resources: irq:46 memory:fbec000-fbeeffff
*-generic:9
description: System peripheral
product: 5520/5500/X58 Chipset QuickData Technology Device
vendor: Intel Corporation
physical id: 16.4
bus info: pci@0000:00:16.4
version: 22
width: 64 bits
clock: 33MHz
capabilities: msix pciexpress pm bus_master cap_list
configuration: driver=ioatdma latency=0
resources: irq:43 memory:fbee8000-fbeebfff
*-generic:10
description: System peripheral
product: 5520/5500/X58 Chipset QuickData Technology Device
vendor: Intel Corporation
physical id: 16.5
bus info: pci@0000:00:16.5
version: 22
width: 64 bits
clock: 33MHz
capabilities: msix pciexpress pm bus_master cap_list
configuration: driver=ioatdma latency=0
resources: irq:44 memory:fbee4000-fbee7fff
*-generic:11
description: System peripheral
product: 5520/5500/X58 Chipset QuickData Technology Device
vendor: Intel Corporation
physical id: 16.6
bus info: pci@0000:00:16.6
version: 22
width: 64 bits
clock: 33MHz
capabilities: msix pciexpress pm bus_master cap_list
configuration: driver=ioatdma latency=0
resources: irq:45 memory:fbee0000-fbee3fff
*-generic:12
description: System peripheral
product: 5520/5500/X58 Chipset QuickData Technology Device
vendor: Intel Corporation
physical id: 16.7
bus info: pci@0000:00:16.7
version: 22
width: 64 bits
clock: 33MHz
capabilities: msix pciexpress pm bus_master cap_list
configuration: driver=ioatdma latency=0
resources: irq:46 memory:fbedc000-fbedffff
*-usb:0
description: USB controller
product: 82801JI (ICH10 Family) USB UHCI Controller #4
vendor: Intel Corporation
physical id: 1a
bus info: pci@0000:00:1a.0
version: 00
width: 32 bits

```

```

clock: 33MHz
capabilities: uhci bus_master cap_list
configuration: driver=uhci_hcd latency=0
resources: irq:16 ioport:cc00(size=32)
*-usb:1
description: USB controller
product: 82801JI (ICH10 Family) USB UHCI Controller #5
vendor: Intel Corporation
physical id: 1a.1
bus info: pci@0000:00:1a.1
version: 00
width: 32 bits
clock: 33MHz
capabilities: uhci bus_master cap_list
configuration: driver=uhci_hcd latency=0
resources: irq:21 ioport:c880(size=32)
*-usb:2
description: USB controller
product: 82801JI (ICH10 Family) USB UHCI Controller #6
vendor: Intel Corporation
physical id: 1a.2
bus info: pci@0000:00:1a.2
version: 00
width: 32 bits
clock: 33MHz
capabilities: uhci bus_master cap_list
configuration: driver=uhci_hcd latency=0
resources: irq:19 ioport:c800(size=32)
*-usb:3
description: USB controller
product: 82801JI (ICH10 Family) USB2 EHCI Controller #2
vendor: Intel Corporation
physical id: 1a.7
bus info: pci@0000:00:1a.7
version: 00
width: 32 bits
clock: 33MHz
capabilities: pm debug ehci bus_master cap_list
configuration: driver=ehci_hcd latency=0
resources: irq:18 memory:fbeda000-fbeda3ff
*-pci:4
description: PCI bridge
product: 82801JI (ICH10 Family) PCI Express Root Port 1
vendor: Intel Corporation
physical id: 1c
bus info: pci@0000:00:1c.0
version: 00
width: 32 bits
clock: 33MHz
capabilities: pci pciexpress msi pm normal_decode bus_master
cap_list
configuration: driver=pcieport
resources: irq:65 ioport:1000(size=4096) memory:c0000000-c01fffff
ioport:c0200000(size=2097152)
*-pci:5
description: PCI bridge
product: 82801JI (ICH10 Family) PCI Express Root Port 5
vendor: Intel Corporation
physical id: 1c.4
bus info: pci@0000:00:1c.4
version: 00
width: 32 bits
clock: 33MHz
capabilities: pci pciexpress msi pm normal_decode bus_master
cap_list
configuration: driver=pcieport
resources: irq:66 ioport:d000(size=4096) memory:fbcb0000-fbcbffff
ioport:c0400000(size=2097152)
*-network
description: Ethernet interface
product: 82574L Gigabit Network Connection

```

```

        vendor: Intel Corporation
        physical id: 0
        bus info: pci@0000:06:00.0
        logical name: eth0
        version: 00
        serial: 00:25:90:02:47:6e
        size: 1Gbit/s
        capacity: 1Gbit/s
        width: 32 bits
        clock: 33MHz
        capabilities: pm msi pciexpress msix bus_master cap_list
ethernet physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
        configuration: autonegotiation=on broadcast=yes driver=e1000e
driverversion=2.0.0-k duplex=full firmware=1.8-0 ip=10.0.0.1 latency=0
link=yes multicast=yes port=twisted pair speed=1Gbit/s
        resources: irq:16 memory:fbce0000-fbcfffff
ioport:dc00(size=32) memory:fbcdc000-fbcdffff
    *-pci:6
        description: PCI bridge
        product: 82801JI (ICH10 Family) PCI Express Root Port 6
        vendor: Intel Corporation
        physical id: 1c.5
        bus info: pci@0000:00:1c.5
        version: 00
        width: 32 bits
        clock: 33MHz
        capabilities: pci pciexpress msi pm normal_decode bus_master
cap_list
        configuration: driver=pcieport
        resources: irq:67 ioport:e000(size=4096) memory:fbd00000-fbdfffff
ioport:c0600000(size=2097152)
    *-network
        description: Ethernet interface
        product: 82574L Gigabit Network Connection
        vendor: Intel Corporation
        physical id: 0
        bus info: pci@0000:07:00.0
        logical name: eth1
        version: 00
        serial: 00:25:90:02:47:6f
        size: 1Gbit/s
        capacity: 1Gbit/s
        width: 32 bits
        clock: 33MHz
        capabilities: pm msi pciexpress msix bus_master cap_list
ethernet physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
        configuration: autonegotiation=on broadcast=yes driver=e1000e
driverversion=2.0.0-k duplex=full firmware=1.8-0 ip=10.0.1.1 latency=0
link=yes multicast=yes port=twisted pair speed=1Gbit/s
        resources: irq:17 memory:fbde0000-fbdfffff
ioport:ec00(size=32) memory:fbddc000-fbddffff
    *-usb:4
        description: USB controller
        product: 82801JI (ICH10 Family) USB UHCI Controller #1
        vendor: Intel Corporation
        physical id: 1d
        bus info: pci@0000:00:1d.0
        version: 00
        width: 32 bits
        clock: 33MHz
        capabilities: uhci bus_master cap_list
        configuration: driver=uhci_hcd latency=0
        resources: irq:23 ioport:c480(size=32)
    *-usb:5
        description: USB controller
        product: 82801JI (ICH10 Family) USB UHCI Controller #2
        vendor: Intel Corporation
        physical id: 1d.1
        bus info: pci@0000:00:1d.1
        version: 00
        width: 32 bits

```

```

clock: 33MHz
capabilities: uhci bus_master cap_list
configuration: driver=uhci_hcd latency=0
resources: irq:19 ioport:c400(size=32)
*-usb:6
description: USB controller
product: 82801JI (ICH10 Family) USB UHCI Controller #3
vendor: Intel Corporation
physical id: 1d.2
bus info: pci@0000:00:1d.2
version: 00
width: 32 bits
clock: 33MHz
capabilities: uhci bus_master cap_list
configuration: driver=uhci_hcd latency=0
resources: irq:18 ioport:c080(size=32)
*-usb:7
description: USB controller
product: 82801JI (ICH10 Family) USB2 EHCI Controller #1
vendor: Intel Corporation
physical id: 1d.7
bus info: pci@0000:00:1d.7
version: 00
width: 32 bits
clock: 33MHz
capabilities: pm debug ehci bus_master cap_list
configuration: driver=ehci_hcd latency=0
resources: irq:23 memory:fbed8000-fbed83ff
*-pci:7
description: PCI bridge
product: 82801 PCI Bridge
vendor: Intel Corporation
physical id: 1e
bus info: pci@0000:00:1e.0
version: 90
width: 32 bits
clock: 33MHz
capabilities: pci subtractive_decode bus_master cap_list
resources: memory:faf00000-fb7fffff
ioport:f9000000(size=16777216)
*-display UNCLAIMED
description: VGA compatible controller
product: MGA G200eW WPCM450
vendor: Matrox Electronics Systems Ltd.
physical id: 1
bus info: pci@0000:08:01.0
version: 0a
width: 32 bits
clock: 33MHz
capabilities: pm vga_controller bus_master cap_list
configuration: latency=64 maxlatency=32 mingnt=16
resources: memory:f9000000-f9ffffff memory:faffc000-faffffff
memory:fb000000-fb7fffff
*-isa
description: ISA bridge
product: 82801JIR (ICH10R) LPC Interface Controller
vendor: Intel Corporation
physical id: 1f
bus info: pci@0000:00:1f.0
version: 00
width: 32 bits
clock: 33MHz
capabilities: isa bus_master cap_list
configuration: driver=lpc_ich latency=0
resources: irq:0
*-storage
description: SATA controller
product: 82801JI (ICH10 Family) SATA AHCI Controller
vendor: Intel Corporation
physical id: 1f.2
bus info: pci@0000:00:1f.2

```

```

        version: 00
        width: 32 bits
        clock: 66MHz
        capabilities: storage msi pm ahci_1.0 bus_master cap_list
        configuration: driver=ahci latency=0
        resources: irq:71 ioport:b480(size=8) ioport:c000(size=4)
ioport:bc00(size=8) ioport:b880(size=4) ioport:b800(size=32) memory:fbed6000-
fbed67ff
    *-serial UNCLAIMED
        description: SMBus
        product: 82801JI (ICH10 Family) SMBus Controller
        vendor: Intel Corporation
        physical id: 1f.3
        bus info: pci@0000:00:1f.3
        version: 00
        width: 64 bits
        clock: 33MHz
        configuration: latency=0
        resources: memory:fbed4000-fbed40ff ioport:400(size=32)
    *-pci:1
        description: Host bridge
        product: Xeon 5600 Series QuickPath Architecture Generic Non-core
Registers
        vendor: Intel Corporation
        physical id: 101
        bus info: pci@0000:fe:00.0
        version: 02
        width: 32 bits
        clock: 33MHz
    *-pci:2
        description: Host bridge
        product: Xeon 5600 Series QuickPath Architecture System Address De-
coder
        vendor: Intel Corporation
        physical id: 102
        bus info: pci@0000:fe:00.1
        version: 02
        width: 32 bits
        clock: 33MHz
    *-pci:3
        description: Host bridge
        product: Xeon 5600 Series QPI Link 0
        vendor: Intel Corporation
        physical id: 103
        bus info: pci@0000:fe:02.0
        version: 02
        width: 32 bits
        clock: 33MHz
    *-pci:4
        description: Host bridge
        product: Xeon 5600 Series QPI Physical 0
        vendor: Intel Corporation
        physical id: 104
        bus info: pci@0000:fe:02.1
        version: 02
        width: 32 bits
        clock: 33MHz
    *-pci:5
        description: Host bridge
        product: Xeon 5600 Series Mirror Port Link 0
        vendor: Intel Corporation
        physical id: 105
        bus info: pci@0000:fe:02.2
        version: 02
        width: 32 bits
        clock: 33MHz
    *-pci:6
        description: Host bridge
        product: Xeon 5600 Series Mirror Port Link 1
        vendor: Intel Corporation
        physical id: 106

```

```

        bus info: pci@0000:fe:02.3
        version: 02
        width: 32 bits
        clock: 33MHz
*-pci:7
    description: Host bridge
    product: Xeon 5600 Series QPI Link 1
    vendor: Intel Corporation
    physical id: 107
    bus info: pci@0000:fe:02.4
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:8
    description: Host bridge
    product: Xeon 5600 Series QPI Physical 1
    vendor: Intel Corporation
    physical id: 108
    bus info: pci@0000:fe:02.5
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:9
    description: Host bridge
    product: Xeon 5600 Series Integrated Memory Controller Registers
    vendor: Intel Corporation
    physical id: 109
    bus info: pci@0000:fe:03.0
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:10
    description: Host bridge
    product: Xeon 5600 Series Integrated Memory Controller Target Ad-
dress Decoder
    vendor: Intel Corporation
    physical id: 10a
    bus info: pci@0000:fe:03.1
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:11
    description: Host bridge
    product: Xeon 5600 Series Integrated Memory Controller RAS Registers
    vendor: Intel Corporation
    physical id: 10b
    bus info: pci@0000:fe:03.2
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:12
    description: Host bridge
    product: Xeon 5600 Series Integrated Memory Controller Test Regis-
ters
    vendor: Intel Corporation
    physical id: 10c
    bus info: pci@0000:fe:03.4
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:13
    description: Host bridge
    product: Xeon 5600 Series Integrated Memory Controller Channel 0
Control
    vendor: Intel Corporation
    physical id: 10d
    bus info: pci@0000:fe:04.0
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:14

```

```

description: Host bridge
product: Xeon 5600 Series Integrated Memory Controller Channel 0
Address
vendor: Intel Corporation
physical id: 10e
bus info: pci@0000:fe:04.1
version: 02
width: 32 bits
clock: 33MHz
*-pci:15
description: Host bridge
product: Xeon 5600 Series Integrated Memory Controller Channel 0
Rank
vendor: Intel Corporation
physical id: 10f
bus info: pci@0000:fe:04.2
version: 02
width: 32 bits
clock: 33MHz
*-pci:16
description: Host bridge
product: Xeon 5600 Series Integrated Memory Controller Channel 0
Thermal Control
vendor: Intel Corporation
physical id: 110
bus info: pci@0000:fe:04.3
version: 02
width: 32 bits
clock: 33MHz
*-pci:17
description: Host bridge
product: Xeon 5600 Series Integrated Memory Controller Channel 1
Control
vendor: Intel Corporation
physical id: 111
bus info: pci@0000:fe:05.0
version: 02
width: 32 bits
clock: 33MHz
*-pci:18
description: Host bridge
product: Xeon 5600 Series Integrated Memory Controller Channel 1
Address
vendor: Intel Corporation
physical id: 112
bus info: pci@0000:fe:05.1
version: 02
width: 32 bits
clock: 33MHz
*-pci:19
description: Host bridge
product: Xeon 5600 Series Integrated Memory Controller Channel 1
Rank
vendor: Intel Corporation
physical id: 113
bus info: pci@0000:fe:05.2
version: 02
width: 32 bits
clock: 33MHz
*-pci:20
description: Host bridge
product: Xeon 5600 Series Integrated Memory Controller Channel 1
Thermal Control
vendor: Intel Corporation
physical id: 114
bus info: pci@0000:fe:05.3
version: 02
width: 32 bits
clock: 33MHz
*-pci:21
description: Host bridge

```

```

Control      product: Xeon 5600 Series Integrated Memory Controller Channel 2
            vendor: Intel Corporation
            physical id: 115
            bus info: pci@0000:fe:06.0
            version: 02
            width: 32 bits
            clock: 33MHz
*-pci:22
            description: Host bridge
            product: Xeon 5600 Series Integrated Memory Controller Channel 2
Address      vendor: Intel Corporation
            physical id: 116
            bus info: pci@0000:fe:06.1
            version: 02
            width: 32 bits
            clock: 33MHz
*-pci:23
            description: Host bridge
            product: Xeon 5600 Series Integrated Memory Controller Channel 2
Rank        vendor: Intel Corporation
            physical id: 117
            bus info: pci@0000:fe:06.2
            version: 02
            width: 32 bits
            clock: 33MHz
*-pci:24
            description: Host bridge
            product: Xeon 5600 Series Integrated Memory Controller Channel 2
Thermal Control
            vendor: Intel Corporation
            physical id: 118
            bus info: pci@0000:fe:06.3
            version: 02
            width: 32 bits
            clock: 33MHz
*-pci:25
            description: Host bridge
            product: Xeon 5600 Series QuickPath Architecture Generic Non-core
Registers    vendor: Intel Corporation
            physical id: 119
            bus info: pci@0000:ff:00.0
            version: 02
            width: 32 bits
            clock: 33MHz
*-pci:26
            description: Host bridge
            product: Xeon 5600 Series QuickPath Architecture System Address De-
coder       vendor: Intel Corporation
            physical id: 11a
            bus info: pci@0000:ff:00.1
            version: 02
            width: 32 bits
            clock: 33MHz
*-pci:27
            description: Host bridge
            product: Xeon 5600 Series QPI Link 0
            vendor: Intel Corporation
            physical id: 11b
            bus info: pci@0000:ff:02.0
            version: 02
            width: 32 bits
            clock: 33MHz
*-pci:28
            description: Host bridge
            product: Xeon 5600 Series QPI Physical 0
            vendor: Intel Corporation

```

```

    physical id: 11c
    bus info: pci@0000:ff:02.1
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:29
    description: Host bridge
    product: Xeon 5600 Series Mirror Port Link 0
    vendor: Intel Corporation
    physical id: 11d
    bus info: pci@0000:ff:02.2
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:30
    description: Host bridge
    product: Xeon 5600 Series Mirror Port Link 1
    vendor: Intel Corporation
    physical id: 11e
    bus info: pci@0000:ff:02.3
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:31
    description: Host bridge
    product: Xeon 5600 Series QPI Link 1
    vendor: Intel Corporation
    physical id: 11f
    bus info: pci@0000:ff:02.4
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:32
    description: Host bridge
    product: Xeon 5600 Series QPI Physical 1
    vendor: Intel Corporation
    physical id: 120
    bus info: pci@0000:ff:02.5
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:33
    description: Host bridge
    product: Xeon 5600 Series Integrated Memory Controller Registers
    vendor: Intel Corporation
    physical id: 121
    bus info: pci@0000:ff:03.0
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:34
    description: Host bridge
    product: Xeon 5600 Series Integrated Memory Controller Target Ad-
address Decoder
    vendor: Intel Corporation
    physical id: 122
    bus info: pci@0000:ff:03.1
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:35
    description: Host bridge
    product: Xeon 5600 Series Integrated Memory Controller RAS Registers
    vendor: Intel Corporation
    physical id: 123
    bus info: pci@0000:ff:03.2
    version: 02
    width: 32 bits
    clock: 33MHz
*-pci:36
    description: Host bridge

```

```

        product: Xeon 5600 Series Integrated Memory Controller Test Regis-
ters
        vendor: Intel Corporation
        physical id: 124
        bus info: pci@0000:ff:03.4
        version: 02
        width: 32 bits
        clock: 33MHz
*-pci:37
        description: Host bridge
        product: Xeon 5600 Series Integrated Memory Controller Channel 0
Control
        vendor: Intel Corporation
        physical id: 125
        bus info: pci@0000:ff:04.0
        version: 02
        width: 32 bits
        clock: 33MHz
*-pci:38
        description: Host bridge
        product: Xeon 5600 Series Integrated Memory Controller Channel 0
Address
        vendor: Intel Corporation
        physical id: 126
        bus info: pci@0000:ff:04.1
        version: 02
        width: 32 bits
        clock: 33MHz
*-pci:39
        description: Host bridge
        product: Xeon 5600 Series Integrated Memory Controller Channel 0
Rank
        vendor: Intel Corporation
        physical id: 127
        bus info: pci@0000:ff:04.2
        version: 02
        width: 32 bits
        clock: 33MHz
*-pci:40
        description: Host bridge
        product: Xeon 5600 Series Integrated Memory Controller Channel 0
Thermal Control
        vendor: Intel Corporation
        physical id: 128
        bus info: pci@0000:ff:04.3
        version: 02
        width: 32 bits
        clock: 33MHz
*-pci:41
        description: Host bridge
        product: Xeon 5600 Series Integrated Memory Controller Channel 1
Control
        vendor: Intel Corporation
        physical id: 129
        bus info: pci@0000:ff:05.0
        version: 02
        width: 32 bits
        clock: 33MHz
*-pci:42
        description: Host bridge
        product: Xeon 5600 Series Integrated Memory Controller Channel 1
Address
        vendor: Intel Corporation
        physical id: 12a
        bus info: pci@0000:ff:05.1
        version: 02
        width: 32 bits
        clock: 33MHz
*-pci:43
        description: Host bridge

```

```

Rank      product: Xeon 5600 Series Integrated Memory Controller Channel 1
          vendor: Intel Corporation
          physical id: 12b
          bus info: pci@0000:ff:05.2
          version: 02
          width: 32 bits
          clock: 33MHz
*-pci:44
          description: Host bridge
          product: Xeon 5600 Series Integrated Memory Controller Channel 1
Thermal Control
          vendor: Intel Corporation
          physical id: 12c
          bus info: pci@0000:ff:05.3
          version: 02
          width: 32 bits
          clock: 33MHz
*-pci:45
          description: Host bridge
          product: Xeon 5600 Series Integrated Memory Controller Channel 2
Control
          vendor: Intel Corporation
          physical id: 12d
          bus info: pci@0000:ff:06.0
          version: 02
          width: 32 bits
          clock: 33MHz
*-pci:46
          description: Host bridge
          product: Xeon 5600 Series Integrated Memory Controller Channel 2
Address
          vendor: Intel Corporation
          physical id: 12e
          bus info: pci@0000:ff:06.1
          version: 02
          width: 32 bits
          clock: 33MHz
*-pci:47
          description: Host bridge
          product: Xeon 5600 Series Integrated Memory Controller Channel 2
Rank
          vendor: Intel Corporation
          physical id: 12f
          bus info: pci@0000:ff:06.2
          version: 02
          width: 32 bits
          clock: 33MHz
*-pci:48
          description: Host bridge
          product: Xeon 5600 Series Integrated Memory Controller Channel 2
Thermal Control
          vendor: Intel Corporation
          physical id: 130
          bus info: pci@0000:ff:06.3
          version: 02
          width: 32 bits
          clock: 33MHz
*-scsi
          physical id: 3
          logical name: scsi0
          capabilities: emulated
*-disk
          description: ATA Disk
          product: ST3250318AS
          vendor: Seagate
          physical id: 0.0.0
          bus info: scsi@0:0.0.0
          logical name: /dev/sda
          version: CC38
          serial: 5VMD72FJ

```

```

size: 232GiB (250GB)
capabilities: partitioned partitioned:dos
configuration: ansiversion=5 signature=0006be37
*-volume:0
  description: EXT4 volume
  vendor: Linux
  physical id: 1
  bus info: scsi@0:0.0.0,1
  logical name: /dev/sda1
  logical name: /
  version: 1.0
  serial: 3bf2ab63-fb49-4487-a3c3-63639fa3cc88
  size: 18GiB
  capacity: 18GiB
  capabilities: primary journaled extended_attributes
large_files huge_files dir_nlink recover extents ext4 ext2 initialized
  configuration: created=2013-10-09 15:10:10 filesystem=ext4
label=root lastmountpoint=/ modified=2013-10-09 15:13:42 mount.fstype=ext4
mount.options=rw,relatime,data=ordered mounted=2013-10-15 14:41:59
state=mounted
  *-volume:1
    description: EXT4 volume
    vendor: Linux
    physical id: 2
    bus info: scsi@0:0.0.0,2
    logical name: /dev/sda2
    version: 1.0
    serial: c2d54d19-add4-4ddc-8804-f88bc4540976
    size: 74GiB
    capacity: 74GiB
    capabilities: primary journaled extended_attributes
large_files huge_files dir_nlink extents ext4 ext2 initialized
  configuration: created=2013-10-09 15:10:09 filesystem=ext4
label=100G modified=2013-10-09 15:10:09 state=clean
  *-volume:2
    description: EXT4 volume
    vendor: Linux
    physical id: 3
    bus info: scsi@0:0.0.0,3
    logical name: /dev/sda3
    logical name: /tmp
    version: 1.0
    serial: 19071d44-b815-4f94-86b0-697ab60b1335
    size: 139GiB
    capacity: 139GiB
    capabilities: primary journaled extended_attributes
large_files huge_files dir_nlink recover extents ext4 ext2 initialized
  configuration: created=2013-10-09 15:10:06 filesystem=ext4
label=tmp lastmountpoint=/tmp modified=2013-10-15 14:42:00 mount.fstype=ext4
mount.options=rw,relatime,data=ordered mounted=2013-10-15 14:42:00
state=mounted

```

Anhang 4: Exkurs Erste Schritte mit parabench und HPL

Schritt 1

Account WR: Zunächst ist ein Account des WR einzurichten, um überhaupt mit Clustern des WR arbeiten zu können. Dieser Account ist von einem Mitarbeiter des WR einzurichten. Es werden ein Username und ein zugehöriges Passwort generiert.

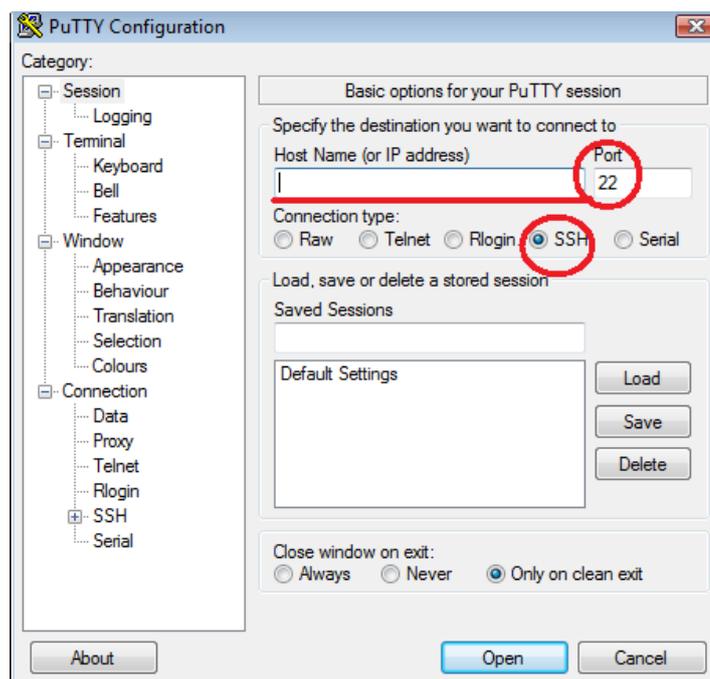
Schritt 2

PuTTY installieren: PuTTY ist ein konsolenbasiertes Programm, mit dem man auf Cluster des WR zugreifen kann. Unter folgendem Link findet ihr die Downloaddatei.

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Schritt 3

Über PuTTY einloggen: Ist PuTTY vollständig installiert erscheint folgende Ansicht:



Die rot markierten Punkte sind entsprechend der Abbildung einzustellen. Den Host Name (or IP address) findet man unter folgendem Link.

http://wr.informatik.uni-hamburg.de/teaching/ressourcen/beginners_guide

Die folgende Abbildung zeigt die oben verlinkte Beginners' Guide. Der rot umkreiste Bereich muss in die PuTTY Maske kopiert werden. Anschließend ist der „<name>“

entsprechend dem erhaltenen Benutzernamen zu ersetzen. Zudem ist der Anfang der Zeile „ssh“ zu entfernen.

Beginners' Guide

You can access the cluster using [SSH](#). The default shell is Bash ([Quick Reference](#)). A listing of the common commands can be found at the [Unix/Linux Command Reference](#) from [FOSSWire](#). A more detailed introduction to Linux/Unix usage can be found in this [script](#).

Table of Contents
•Beginners' Guide
•User ID
•Login
•Changing the Password
•Public Key Authentication
•Copying Files
•Job Managing
•Compiling
•POSIX Threads
•OpenMP
•MPI
•Modules

User ID

Login

To login you have to authenticate yourself on the master node with SSH at `cluster.wr.informatik.uni-hamburg.de`.

On *nix-like operation systems you can simply open a shell and use the following command (where <name> is your user name on the cluster):

```
ssh <name>@cluster.wr.informatik.uni-hamburg.de
```

To use GUI applications, you need X forwarding.

```
ssh -X <name>@cluster.wr.informatik.uni-hamburg.de
```

To view PDFs use zathura:

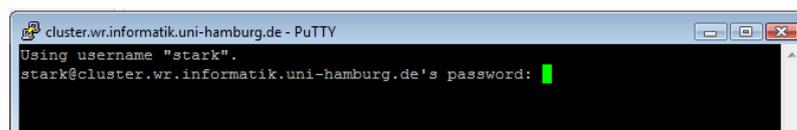
```
zathura your_pic.pdf
```

If you are running Windows it is recommended to use a graphical SSH client like [PuTTY](#). The leading X-Server for Windows is [Xming](#). Do not forget to enable [X11 forwarding](#).

To transere data between cluster and your local Windows-System use [WinSCP](#).

Warning: If you try to login too often within two minutes your login gets blocked for two minutes. Should you try to login again with these two minutes the block will be extended automatically.

Sind die Eingaben vollständig klicken Sie auf „Open“. Eine DOS-Box erscheint. Das Beispiel führen wir mit dem Namen „stark“ oder „vach“ aus. (Siehe nachfolgende Abbildung)



Hier geben Sie ihr erhaltenes Passwort ein und bestätigen mit der Eingabetaste.

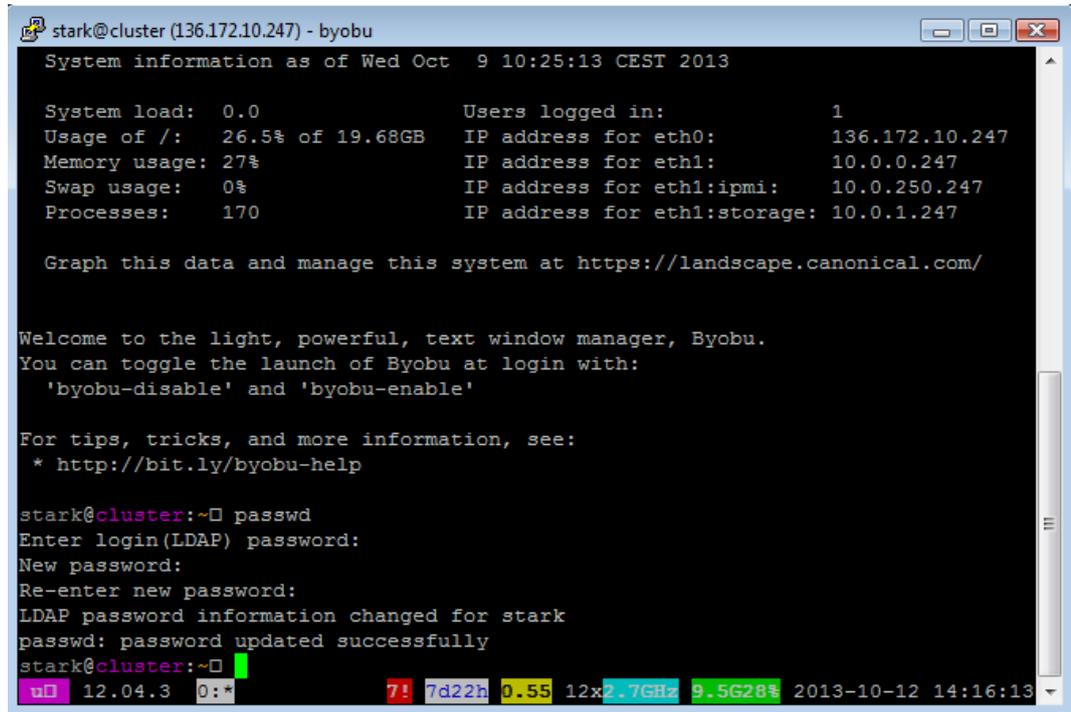
Schritt 4

Passwort ändern:

Geben Sie in die Kommandozeile „passwd“ ein und bestätigen dies mit der Eingabetaste. Anschließend geben Sie das alte Passwort einmal ein. Bestätigen mit Eingabe Taste.

Nun geben Sie Ihr neues Passwort 2 x ein und bestätigen jeweils mit der Eingabetaste.

Die folgende Abbildung sollte erscheinen.



```
stark@cluster (136.172.10.247) - byobu
System information as of Wed Oct  9 10:25:13 CEST 2013

System load:  0.0          Users logged in:      1
Usage of /:   26.5% of 19.68GB  IP address for eth0:  136.172.10.247
Memory usage: 27%          IP address for eth1:  10.0.0.247
Swap usage:   0%           IP address for eth1:ipmi: 10.0.250.247
Processes:   170          IP address for eth1:storage: 10.0.1.247

Graph this data and manage this system at https://landscape.canonical.com/

Welcome to the light, powerful, text window manager, Byobu.
You can toggle the launch of Byobu at login with:
  'byobu-disable' and 'byobu-enable'

For tips, tricks, and more information, see:
  * http://bit.ly/byobu-help

stark@cluster:~$ passwd
Enter login(LDAP) password:
New password:
Re-enter new password:
LDAP password information changed for stark
passwd: password updated successfully
stark@cluster:~$
```

12.04.3 0:~ 7! 7d22h 0.55 12x2.7GHz 9.5G28% 2013-10-12 14:16:13

Anhang 5: Exkurs Statistische Grundlagen

Inhaltsverzeichnis

1. Statistische Grundlagen.....	113
1.1 Skalenniveaus	114
1.2 Deskriptive Statistik	115
1.3 Verfahren der deskriptiven Statistik.....	117
1.3.1 Häufigkeitsauszählung	117
1.3.2 Lageparameter.....	118
1.3.3 Streuungsparameter.....	119
1.3.4 Zusammenhangsmaße	122
Quellenverzeichnis zu Anhang 5	129

1. Statistische Grundlagen

Von Statistiken wird gefordert, dass sie „objektiv“ (unabhängig vom Standpunkt des Erstellers der Statistik), „reliabel“ (verlässlich), „valide“ (überkontextuell gültig), „signifikant“ (bedeutend) und „relevant“ (wichtig) sind. Dabei werden die vorliegenden Daten in geeigneter Weise beschrieben, aufbereitet, zusammengefasst, analysiert und interpretiert.. Mit statistischen Methoden und Instrumenten verdichtet man quantitative Daten zu Tabellen, graphischen Darstellungen und Kennzahlen.

Nachfolgend wird ein kurzer Abriss über die einzelne Teilbereiche der Statistik (induktive-, explorative- und deskriptive Statistik) und deren Ziele gegeben, bevor für die deskriptive Statistik wichtige Methoden und Kennzahlen näher beschrieben werden. Die Beschränkung auf die ausschließliche Darstellung der deskriptiven Statistik begründet sich dabei aus der Tatsache, dass zur Auswertung von Messdaten aus der Evaluation von Computersystemen ausschließlich Methoden und Instrumente dieses Teilbereiches der Statistik herangezogen werden.

Teilbereiche der Statistik

Ziel der **deskriptiven** (beschreibenden) Statistik ist es, die in gezielten, systematisch verlaufenden Untersuchungen gesammelten Daten so aufzubereiten und durch Tabellen, Kennzahlen und Grafiken darzustellen, dass ihr Informationsgehalt sinnvoll und leicht überblickt werden kann.

Im Gegensatz zur deskriptiven Statistik befasst sich die **explorative** (erkundende) mit bisher unbekanntem. Sie ist darauf ausgerichtet, Zusammenhänge zwischen den, in der Regel in Stichproben erhobenen Daten zu identifizieren und aus diesem Kontext neue Hypothesen zu generieren.

Die **induktive** Statistik untersucht mittels wahrscheinlichkeitstheoretischer Methoden die Allgemeingültigkeit der, auf Stichprobendaten basierenden Hypothesen.

Da die deskriptive Statistik sowie die Auswahl von Kennzahlen und Parametern im Wesentlichen vom verwendeten Skalenniveau abhängt, werden diese zuerst erklärt, bevor darauf folgend die deskriptive Statistik näher erläutert wird.

1.1 Skalenniveaus

Die Wahl von zur Erklärung geeigneter Parameter hängt dabei maßgeblich von der Skalierung der zu erklärenden Daten ab. Dabei wird in der deskriptiven Statistik differenziert zwischen:

Nominalskala

Eine Zuordnung der erhobenen Daten zu einer Kategorie der Skala erfolgt nach dem Kriterium der Gleichheit, d.h. Nur wenn die jeweils erhobenen Daten exakt dem Definitionskriterium einer Kategorie entsprechen, werden sie diesem auch zugeordnet. Die Zuordnung zu mehr als einer Kategorie darf dabei ebenso wenig vorkommen, wie eine nicht Zuordnung zu einer der angebotenen Kategorien.

Tabelle 3: Beispiele von nominalskalierten Merkmalen

Merkmal	Geschlecht	Haarfarbe	Nationalität
Kategorien	männlich, weiblich	braun, schwarz, blond	Deutsch, russisch,
Zulässige math./log. Operationen	$= / \neq$		
Lageparameter	Modus		

Quelle: eigene Darstellung

Ordinalskala

Zwischen den einzelnen Kategorien der Skala besteht eine eindeutige Rangfolge, so dass die Zuordnung der einzelnen Daten zu einer Kategorie bereits eine Bewertung darstellt. Die Einzelnen Daten werden in der Ordinalskala im Vergleich zu den anderen Daten positioniert.

Tabelle 4: Beispiele für ordinal skalierte Merkmale

Merkmal	Einkommen	Schulnoten
Kategorien	hoch > mittel > niedrig	sehr gut > gut > befriedigend > ausreichend > mangelhaft > ungenügend
Zulässige math./log. Operationen	$= / \neq, < / >$	
Lageparameter	Median	

Quelle: eigene Darstellung

Intervallskala

Hierbei handelt es sich um metrische Skalen, wobei die Abstände zwischen den einzelnen Kategorien quantifizierbar und gleich sein müssen.

Tabelle 5: Beispiele für intervallskalierte Merkmale

Merkmal	Zeitpunkte	Temperatur in Grad Celsius
Kategorien	Startzeit, Zielzeit	°C um 06:00 Uhr, °C um 12:00 Uhr
Zulässige math./log. Operationen	$=/\neq, </>, +/-,$	
Lageparameter	Arithmetisches Mittel	

Quelle: eigene Darstellung

Verhältnisskala

Wie Intervallskala (metrische Skala mit quantifizierbaren und gleichen Abständen zwischen den Kategorien), wobei die Verhältnisskala durch einen absoluten Nullpunkt gekennzeichnet ist, wodurch die Bildung von Verhältnissen zwischen einzelnen Merkmalsausprägungen möglich ist.

Tabelle 6: Beispiele für verhältnisskalierte Merkmale und dessen Nullpunkt

Merkmal	Temperatur in Kelvin	Preis	Prozentsätze
Kategorien	Absoluter Nullpunkt	kostenlos	0 %
Zulässige math./log. Operationen	$=/\neq, </>, +/-, \times/\div$		
Lageparameter	Geometrisches Mittel		

Quelle: eigene Darstellung

1.2 Deskriptive Statistik

Methoden der deskriptiven Statistik

Da die deskriptive Statistik sich mit der Aufbereitung und Darstellung von einer Vielzahl unterschiedlicher Datenwerte beschäftigt, nutzt sie im Wesentlichen 3 Methoden:

- **Tabellen** (Daten werden in einer Matrix erfasst und dargestellt. Die Zeile entspricht dabei üblicherweise den Daten eines Probanden, die Spalte gibt die für eine Variable erhobenen Datenwerte wieder.)

- **Diagramme** (Darstellung von i.d.R. zusammengefassten Daten in Grafiken oder Diagrammen. Die Zusammenfassung kann dabei zu Informationsverlusten führen.)
- **Parameter** (Aggregation der Daten bzw. eines Aspektes der Daten zu einer einzigen Zahl (z.B. Mittelwert). Um den Informationsverlust der Aggregation auszugleichen, wird die aggregierte Zahl durch weitere, berechnete Parameter näher spezifiziert (z.B. Fehler des Mittelwertes, Streuung etc.)

In der deskriptiven Statistik werden die Parameter hinsichtlich Ihrer Art unterschieden, wobei differenziert wird zwischen:

- **Lagemaße** (Häufigkeitsauszählungen von beobachteten/gemessenen Ausprägungen der jeweiligen Variablen zur Bestimmung der Lageparameter (Modus, Median, arithmetisches Mittel, geometrisches Mittel)
- **Streuungsmaße** (zur Bestimmung der Variabilität der Häufigkeitsverteilung anhand von zu berechnenden Streuungsparametern (Spannweite, Interquartilsabstand, Varianz, Standardabweichung, Variationskoeffizient)
- **Zusammenhangsmaße** (zur Ermittlung des Zusammenhangs zwischen 2 Variablen anhand zu berechnender Zusammenhangsparametern (Korrelation, Regressionsanalyse, Konfidenzintervall, Hypothesentests)

Um Parameter korrekt zu ermitteln und damit aussagekräftig interpretieren zu können, muss die Datentabelle frei von Fehlern sein. Ein wesentlicher Aspekt ist dabei, der Umgang mit Missing Values, also mit Fragebögen bei denen zu einzelnen Variablen keine oder fehlerhafte Angaben gemacht wurden.

Die Statistik unterscheidet dabei folgende Verfahren im Umgang mit Missing Values:

- Die Elimination des ganzen Datensatzes
- Die Elimination des betroffenen Datenfeldes (also des fehlenden Wertes) für den betroffenen Datensatz. Die Werte der anderen Variablen des Datensatzes bleiben als Teil des Datenbestandes erhalten

- Das Auffüllen der betroffenen Datenfelder mit dem Mittelwert (führt zu einer gleichbleibenden und vollständigen Datensatzanzahl). Das Auffüllen von leeren oder fehlerhaften Datenfeldern mit dem Mittelwert führt dazu, dass sie Wertneutral hinsichtlich der Lageparameter und Streuungsmaße sind.
- Nearest Neighbour Methode, d.h. es wird in allen Datensätzen nach dem Datensatz gesucht, der hinsichtlich aller einzelnen Werte für die jeweiligen Variablen am ähnlichsten ist zu den restlichen Werten des fehlerhaften Datensatzes. Die fehlenden Datenwerte werden durch die des identifizierten „nearest Neighbour“ aufgefüllt.

1.3 Verfahren der deskriptiven Statistik

Erhobene Daten stellen Träger von unterschiedlichen Informationen dar. Die Verfahren der deskriptiven Statistik sollen die Vielzahl von Informationen auf wenige, für die jeweiligen Variablen aussagekräftige Werte komprimieren.

Nachfolgend werden die typischen Verfahren der deskriptiven Statistik erläutert.

1.3.1 Häufigkeitsauszählung

Dieses Verfahren wird üblicherweise als erstes Verfahren einer deskriptiven Analyse von Daten angewendet. Dabei wird gezählt, wie häufig die jeweilig möglichen Ausprägungen einer Variablen beobachtet wurden.

Üblicherweise wird Häufigkeitsauszählung nur für nominal- und ordinalskalierte Variablen verwendet. Bei metrisch skalierten Variablen führt diese Form der Darstellung bedingt durch die Fülle von möglichen unterschiedlichen Wertausprägungen nicht zu zusätzlichen Informationen. Häufig wird deshalb die Häufigkeitsauszählung von metrischen Variablen anhand vorher festgelegter Klassen vorgenommen. Dadurch wird gewährleistet, dass die möglichen Merkmalsausprägungen auf ein übersichtliches Maß reduziert werden.

Will man zum Beispiel die Haushalte der Bundesrepublik Deutschland hinsichtlich der Variable „zur Verfügung stehendes Jahreseinkommen“ erfassen und darstellen, so kann

man sich gut vorstellen, dass es eine Vielzahl von unterschiedlichen Merkmalsausprägungen dieser Variable gibt. Dementsprechend teilt man es in entsprechende Klassen ein (< 15.000,-€, von 15.000,-€ bis 20.000,-€, über 20.000,-€ bis 30.000,-€ usw.), wobei die Klassenbreiten durchaus unterschiedlich sein können.

1.3.2 Lageparameter

Ein Lageparameter beschreibt, wo das gesamte Datenmaterial einer Variablen im Mittel lokalisiert ist. Dabei werden sämtliche erhobenen/gemessenen Datenwerte zu einem Lageparameter zusammengefasst. Typische Lageparameter sind Modus, Median, arithmetisches Mittel und geometrisches Mittel. Welcher von ihnen zur Beschreibung der erhobenen Daten dient ist dabei abhängig vom Skalenniveau der Daten.

Modus

Beim Modus handelt es sich um die Ausprägung einer Variablen, die den erhobenen/gemessenen Daten am häufigsten vorkommt. Von bimodalen bzw. multimodalen Häufigkeitsverteilung spricht man, wenn zwei oder mehrere Ausprägungen in den erhobenen Daten mit der gleichen Häufigkeit vorkommen. Der Modus wird als Lageparameter für nominal skalierte Variablen verwendet.

Median

Der Median ist der Stichprobenwert, der in der Verteilung so liegt, dass 50% der anderen Werte oberhalb, und 50% unterhalb liegen. Er wird üblicherweise als Lageparameter bei ordinal skalierten Variablen verwendet, kann allerdings auch zur Beschreibung metrischer Variablen angewendet werden. Für nominal skalierte Variablen findet er keine Anwendung, da diese nicht nach ihrer Größe geordnet werden können.

Arithmetisches Mittel

Das arithmetische Mittel (auch Durchschnitt) ist ein rechnerisch bestimmter Mittelwert. Es handelt sich dabei um den Punkt, bei dem sich die Merkmalsausprägungen einer Variablen im Mittel befinden. Er berechnet sich durch die Addition sämtlicher Merkmalsausprägungen einer Variablen geteilt durch die Anzahl der Merkmalsausprägungen.

Die mathematische Formel fürs arithmetische Mittel lautet:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

n = Anzahl der Befragten; x_i = Antwort der Person i

Bei dem arithmetischen Mittel handelt es sich um den gebräuchlichsten Lageparameter für metrisch skalierte Variablen. Häufig wird es auch für ordinal skalierte Variablen verwendet, wobei diese dann metrisch interpretiert werden.

Geometrisches Mittel

Beim geometrischen Mittel handelt es sich um einen Mittelwert von Größen, bei denen das Produkt und nicht die Summe interpretierbar ist. Dementsprechend wird es zur Interpretation von Wachstumsraten oder von Verhältnissen verwendet.

Die mathematische Formel für das geometrische Mittel lautet:

$$\bar{x} = \sqrt[n]{\prod_{i=1}^n x_i} = \sqrt[n]{x_1 * x_2 \cdots * x_n}$$

Neben der Skalierung entscheidet darüber hinaus die Fragestellung der Evaluation darüber, welche Lageparameter zur Beschreibung der erhobenen Daten verwendet werden sollen. Dabei reagiert der Median im Vergleich zum arithmetischen Mittel weniger stark auf Ausreißer.

Standardfehler des Mittelwertes

Ist ein Maß für die Unsicherheit in der Schätzung. Je größer die Stichprobe gewählt wird, desto sicherer kann man sein, dass der Mittelwert der betrachteten Schätzung dem wahren Mittelwert der Grundgesamtheit entspricht.

1.3.3 Streuungsparameter

Die Streuungsparameter beschreiben die Verteilung der Merkmalsausprägungen einzelner Variablen. Dabei sind sie komplementäre Maße zu den Lageparametern, d.h. sie geben Auskunft darüber, wie die erhobenen/gemessenen Datenwerte im Mittel um

die dazugehörigen Lageparameter streuen. Damit stellen sie wichtige Analyseinstrumente der deskriptiven Statistik dar. Nachfolgend werden einige Streuungsparameter näher erläutert.

Spannweite

Bei der Spannweite handelt es sich um die Differenz zwischen dem höchsten Wert (Maximum) und dem niedrigsten Wert (Minimum) der erhobenen Merkmalsausprägungen einer Variablen. Da sie bei ihrer Berechnung nur die jeweils extremsten Werte der Merkmalsausprägungen berücksichtigt ist sie extrem anfällig für Ausreißer.

Interquartilsabstand

Er gibt den Wertebereich an, in dem sich die mittleren 50% der Merkmalsausprägungen einer Variablen konzentrieren. Zu seiner Berechnung wird der gesamte Wertebereich der ermittelten Merkmalsausprägungen in vier gleich große Wertebereiche (Quartil) unterteilt. Das Quartil ist dabei der Grenzwert zwischen den jeweiligen Wertebereichen, d.h. das 1. Quartil ist der Wert, der die untersten 25% der Messwerte von den restlichen Messwerten abgrenzt. Das 2. Quartil ist der Stichprobenwert, der in der Verteilung so liegt, dass 50% der anderen Werte oberhalb, und 50% unterhalb liegen. Er entspricht damit dem Median.

Da der Interquartilsabstand ist der Abstand zwischen dem 2. und 3. Quartils, d.h. dass die Daten der unteren und oberen 25% der Messwerte unberücksichtigt bleiben und damit Ausreißer in seine Berechnung nicht mit einfließen. Anwendung findet er insbesondere bei der Beschreibung von ordinal skalierten Variablen.

Varianz

Die Varianz ist definiert als die durchschnittliche quadrierte Abweichung aller Merkmalsausprägungen einer metrisch skalierten Variablen von ihrem Mittelwert. Die mathematische Formel zur Berechnung der Varianz lautet:

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

Die Varianz ist extrem anfällig für Ausreißer, da eine Verdoppelung der Abweichung quadriert in den Varianzterm eingehen würde.

Standardabweichung

Die Standardabweichung ist die Quadratwurzel der Varianz. Die mathematische Formel lautet:

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

Im Gegensatz zur Varianz reagiert die Standardabweichung im gleichen Verhältnis auf eine Verdoppelung der Abweichungen. Sie ist damit weniger anfällig für Extremwerte und damit besser geeignet für die Beschreibung des Streuverhaltens der Merkmalsausprägungen einer metrisch skalierten Variablen.

Wird die Standardabweichung interpretiert, so kann grob davon ausgegangen werden, dass ca. $\frac{2}{3}$ der erhobenen Merkmalsausprägungen in dem Bereich der Standardabweichung um den Mittelwert liegen. Dabei wird die Standardabweichung als Wurzelausdruck sowohl positiv als auch negativ interpretiert, so dass ca. $\frac{1}{3}$ in dem durch die Standardabweichung definierten Bereich unterhalb des Mittelwertes und $\frac{1}{3}$ in dem durch die Standardabweichung definierten Bereich oberhalb des Mittelwertes liegen. Eine niedrige Standardabweichung kann darüber hinaus als Hinweis auf homogene Merkmalsausprägungen der metrisch skalierten Variablen angesehen werden, da die Datenwerte nur wenig um den errechneten Mittelwert streuen.

Bei der Standardabweichung handelt es sich darüber hinaus um ein absolutes Streuungsmaß, welches demnach nur als Vergleichsmaßstab von Variablen mit identischen Skalenniveaus herangezogen werden kann. Zum Vergleich der Streuung von Variablen, bei denen die Merkmalsausprägungen mit verschiedenen Maßeinheiten ermittelt wurden, muss auf einen maßstabslosen (dimensionslosen) Streuungsparameter zurückgegriffen werden.

Variationskoeffizient

Der Variationskoeffizient wird dadurch ermittelt, dass die Standardabweichung geteilt wird durch das arithmetische Mittel. Die mathematische Formel lautet:

$$V = \frac{S}{\bar{x}} * 100$$

Er wird als Maßzahl der relativen Streuung um den Mittelwert interpretiert.

1.3.4 Zusammenhangsmaße

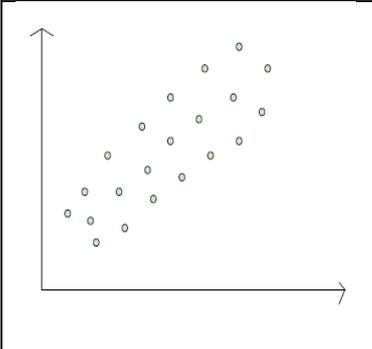
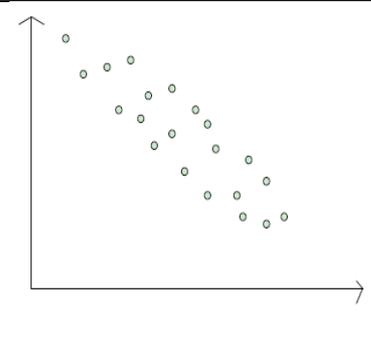
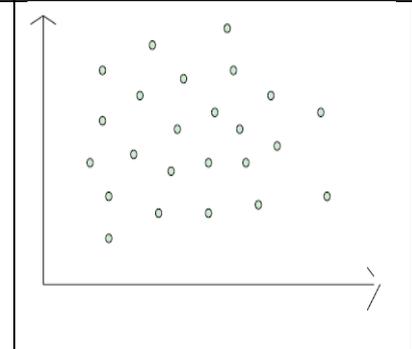
Neben der reinen Beschreibung von Messergebnissen ist es häufig sinnvoll, diese in einen, wie auch immer gearteten Zusammenhang zu bringen um so Aussagen darüber treffen zu können, ob möglicherweise Ursachen - Wirkungsbeziehungen vorliegen und damit Kausalzusammenhänge zwischen den Variablen nachgewiesen werden können. In der Statistik wird dabei differenziert zwischen den unabhängigen Variablen (exogene oder erklärende Variable) und der abhängigen Variable (endogene oder zu erklärende Variable). Nachfolgend werden einige Verfahren der Statistik vorgestellt, die es ermöglichen, das Verhalten von Variablen zueinander zu untersuchen.

Streudiagramme

Streudiagramme stellen die Beziehung von zwei Variablen anhand von beobachteten Wertepaaren in grafischer Form dar. Sie liefern einen ersten visuellen Eindruck über einen eventuell vorliegenden Zusammenhang zwischen den jeweiligen Wertepaaren.

Nachfolgende Streudiagramme zeigen mögliche Beziehungen zwischen 2 Variablen auf:

Tabelle 7: Streudiagramme

Positiver Zusammenhang	Negativer Zusammenhang	Kein Zusammenhang
		
Steigt die eine Variable, dann nimmt auch die andere zu	Steigt die eine Variable, dann nimmt die andere Variable ab	Es besteht kein Zusammenhang zwischen den Variablen

Quelle: Eigene Zusammenstellung in Anlehnung an http://www.uni-due.de/imperia/md/content/soziologie/stein/datenanalyse_vi.pdf

Korrelation

Korrelation bezeichnet den numerischen Zusammenhang zwischen zwei oder mehreren Variablen. Dabei lässt die Korrelation allerdings keine Kausalaussagen zu. Kausalzusammenhänge begründen sich auf sachliche Zusammenhänge der Variablen und nicht auf deren numerischen Zusammenhang.

Ein immer wieder zitiertes Beispiel stellt dabei die positive Korrelation zwischen der Anzahl von Störchen zur Anzahl der Geburten dar, die, sofern sie als Kausalzusammenhang interpretiert wird impliziert, dass Störche doch die Kinder bringen. Ursächlich für diese Korrelation ist jedoch die Drittvariable „ländlicher Raum“, indem sowohl die Anzahl der Störche, als auch die der Geburten höher ist als in der Stadt.

Die unterschiedlichen statistischen Verfahren, die zur Berechnung der Korrelation herangezogen werden richten sich nach den jeweiligen Skalenniveaus der betrachteten Variablen. Zur Berechnung werden üblicherweise Statistikprogramme herangezogen, so dass eine mathematische Herleitung hier nicht vorgenommen werden soll. Von entscheidender Bedeutung ist aber die Interpretation des ermittelten Korrelationskoeffizienten. Dieser bewegt sich in einer Spanne von -1 (perfekter negativer Zusammenhang) bis +1 (perfekter positiver Zusammenhang), wobei Werte nahe 0 bedeuten, dass ein Zusammenhang der Variablen ausgeschlossen werden kann, sie sich also nicht gegenseitig beeinflussen.

Regressionsanalyse

Die Regressionsanalyse untersucht den Einfluss einer oder mehrerer unabhängigen Variablen auf eine abhängige Variable und ist damit ausgerichtet auf Ursache – Wirkungsfragestellungen. Darüber hinaus findet sie Anwendungen sowohl in Ursachenanalysen (wie stark ist der Einfluss der unabhängigen Variablen), in Wirkungsprognosen (Zukunftsentwicklung der abhängigen Variablen bei Variation der unabhängigen Variablen), Zeitreihenanalysen (Veränderung der abhängigen Variablen im Zeitablauf) und Querschnittsanalysen (Einfluss der unabhängigen Variable auf die abhängige Variable bei unterschiedlichen Bedingungen).

Das Verfahren der Regressionsanalyse basiert auf der Methode der kleinsten Quadrate, d.h. aus den erhobenen Datenpaaren und dem sich daraus ergebenden Streudiagramm

wird genau die Regressionsgrade berechnet, bei der die Summe der quadrierten Abweichungen der Beobachtungen zur Regressionsgrade minimiert sind.

Histogramme

Die sogenannten Histogramme sind eine spezielle Form der Balkendiagramme zur Darstellung von intervallskalierten Variablen mit vielen Werten, wie etwa das Alter oder das Jahreseinkommen. Bei Histogrammen werden alle Daten häufig in unterschiedlich große oder gleich große Klassen unterteilt, womit sie sich besonders für die Visualisierung vieler Ausprägungen oder große Datenmengen eignen. (Kuckartz, Rädiker, Ebert, & Schehl, 2013, S. 45)

Bei der Interpretation von Histogrammen ist nicht die Höhe oder Breite der Rechtecke entscheidend, sondern die Fläche der Balken bzw. Rechtecke wird zur Interpretation herangezogen. Um Aussagen zu den Balken zu treffen ist es entscheidend, ob die Breite (Klassengröße) der Balken gleich groß sind oder nicht, denn nur wenn die Klassen gleich groß gewählt wurden, ist ein optischer Vergleich der Balken möglich, ansonsten würden die Balken bei unterschiedlicher Klassengröße unterschiedlich dünn werden bzw. unterschiedlich hoch sein und rein optisch gesehen, würde ein Histogramm vielleicht falsch interpretiert werden. Deswegen sollte man immer die Fläche der Balken miteinander vergleichen und dazu Aussagen treffen. Anhand des nachfolgenden Diagramms wird dies deutlicher. Im Diagramm (Abbildung 22) sieht man das Einkommen von 83 Studenten pro Monat. Die Klassen wurden jeweils gleich groß gewählt und demnach ist die Anzahl an Studenten, die zwischen 600 und 900 € pro Monat verdienen mit 21 Studenten am höchsten bzw. Häufigsten. So sieht man nun auf den ersten Blick, bei gleicher Klassengröße, dass die meisten Studenten zwischen 600 und 900 € verdienen.

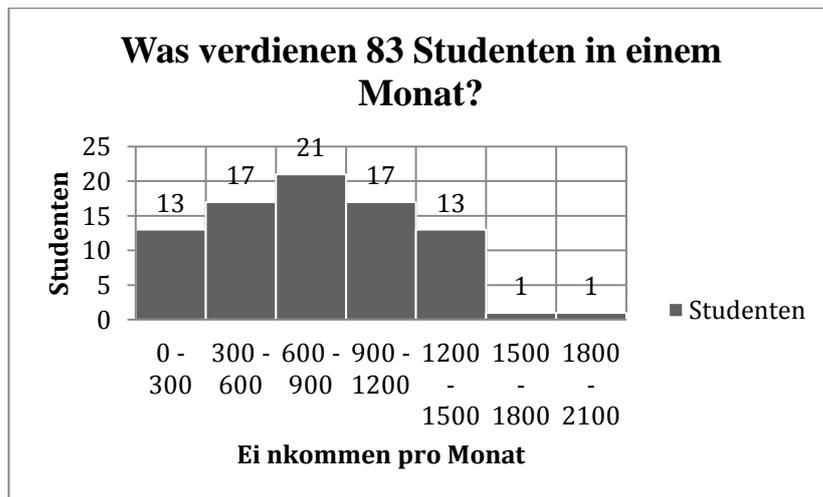


Abbildung 36: Histogramm (Beispiel)

Quelle: Eigene beispielhafte Darstellung

Bei Histogrammen kann man zusätzlich noch eine Linie einzeichnen, die durch die obersten Punkte der Balken gezogen werden kann (immer in der Mitte von der jeweiligen Klassengröße, ein sogenannter Polygonzug, Siehe Abbildung 23).

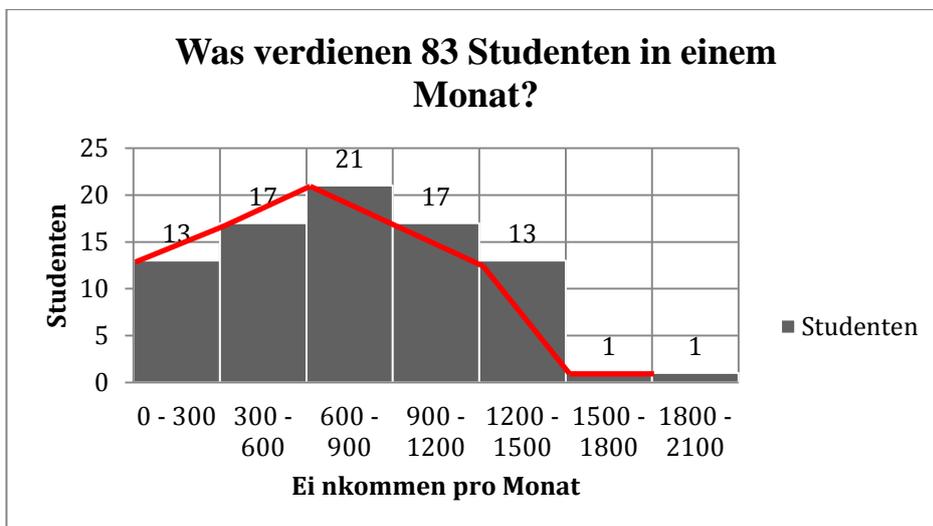


Abbildung 37: Histogramm mit Polygonzug (Beispiel)

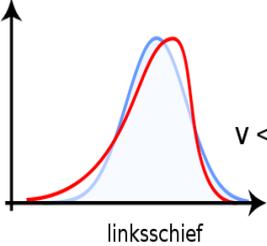
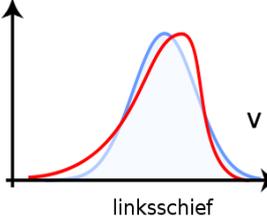
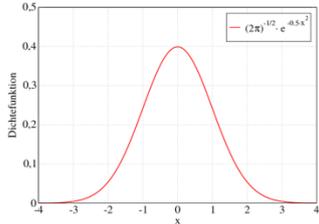
Quelle: Eigene beispielhafte Darstellung

Diese kann dann auch noch geglättet werden, damit eine Kurve dargestellt werden kann. Aus dieser geglätteten Kurve oder den Balken kann man dann Aussagen zur Schiefe und Wölbung treffen, die im nächsten Abschnitt näher beschrieben werden.

Prüfung auf Normalverteilung

Schiefe (v)

Das Maß der Schiefe überprüft, in wie weit die Verteilung der erhobenen Datenwerte symmetrisch zum ermittelten Mittelwert verteilt sind. Schiefe ist damit ein Maß, welches die Symmetrie der betrachteten Verteilung zur Gaußschen Normalverteilung analysiert. Folgende Ausprägungen können auftreten:

$v < 0$	betrachtete Verteilung ist linksschief bzw. rechtssteil, d.h. bei der Verteilung der ermittelten Daten sind höhere Werte als die des Mittelwertes häufiger zu beobachten. Der Gipfel der Verteilung befindet sich dementsprechend rechts vom Mittelwert.	
$v > 0$	betrachtete Verteilung ist rechtsschief bzw. linkssteil, d.h. bei der Verteilung der ermittelten Daten sind kleinere Werte häufiger zu beobachten als größere. Der Gipfel der Verteilung befindet sich links vom Mittelwert.	
$v = 0$	betrachtete Verteilung entspricht der Normalverteilung	

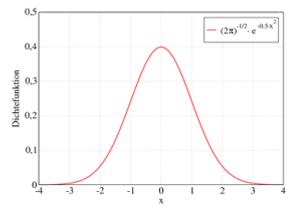
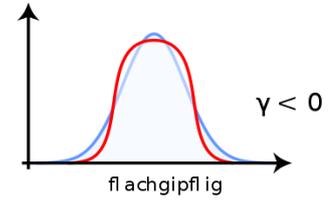
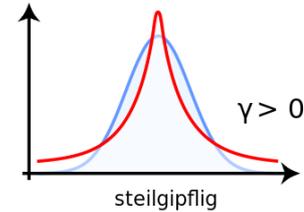
Quelle: eigene Darstellung, Grafiken aus <http://de.wikipedia.org/wiki/Normalverteilung> und http://de.wikipedia.org/wiki/Schiefe_%28Statistik%29

Kurtosis (Wölbung)

Maß für die Steilheit oder Spitzigkeit einer statistischen Dichtefunktion oder Häufigkeitsverteilung. Um das Ausmaß der Wölbung einer betrachteten Verteilung besser einschätzen zu können, wird sie zur Wölbung einer Normalverteilung ins Verhältnis gesetzt. Dabei wird allerdings nicht das Maß der Wölbung sondern der Exzess herangezogen.

Exzess = Wölbung - 3

Folgende Ausprägungen sind möglich:

Exzess = 0	betrachtete Verteilung entspricht der Normalverteilung	
Exzess < 0	Betrachtete Verteilung ist flachgipflig und subgaußförmig., d.h. eine abgeflachte Verteilung mit vielen Datenwerten, die nur gering um den Mittelwert streuen.	
Exzess > 0	betrachtete Verteilung ist steilgipflig bzw. supergaußförmig, d.h. bei der betrachteten Verteilung handelt es sich um Datenwerte mit vielen maximalen Ausprägungen.	

Quelle: Eigene Darstellung, Grafiken aus <http://de.wikipedia.org/wiki/Normalverteilung> und http://de.wikipedia.org/wiki/W%C3%B6lbung_%28Statistik%29

Zeitreihen

Eine **Zeitreihe** ist eine zeitabhängige Folge von Datenpunkten (meist aber keine Reihe im mathematischen Sinne). Typische Beispiele für Zeitreihen sind Börsenkurse, Arbeitslosenquote oder Wetterbeobachtungen.

(Wikipedia: <http://de.wikipedia.org/wiki/Zeitreihenanalyse>)

Die **Zeitreihenanalyse** ist die Disziplin, die sich mit der mathematisch-statistischen Analyse von Zeitreihen und der Vorhersage (Trends) ihrer künftigen Entwicklung beschäftigt. Sie ist eine Spezialform der Regressionsanalyse.

(Wikipedia: <http://de.wikipedia.org/wiki/Zeitreihenanalyse>)

Auf der nachfolgenden Abbildung sieht man eine solche Zeitreihe bzw. -serie.



Abbildung 38: Arbeitslosenquote in den USA seit 1980 (als Zeitreihe dargestellt)

Quelle: http://blog.zeit.de/herdentrieb/files/2013/09/US_Arbeitslosenquote_seit_1980.gif

Für weiterführende Erläuterungen Siehe Zucchini, Nenadic, 2007.

Quellenverzeichnis zu Anhang 5

Kuckartz, Udo; Rädiker, Stefan; Ebert, Thomas; Schehl, Julia (2013) Statistik -
Eine verständliche Einführung. Marburg: Springer VS

Wikipedia/Zeitreihenanalyse (Letzter Zugriff am 29.10.2013, um 16:50 Uhr)

<http://de.wikipedia.org/wiki/Zeitreihenanalyse>

Zucchini, Prof. Dr. W., Nenadic, O. (2007) Zeitreihenanalyse -

Teil II: Lineare Zeitreihenmodelle. Göttingen: Skript.

Anhang 6: Exkurs Evaluationsstandards

Inhaltsverzeichnis

1. Evaluationsstandards.....	131
1.1 Nützlichkeit.....	132
1.2 Durchführbarkeit.....	134
1.3 Korrektheit.....	134
1.4 Genauigkeit.....	135

1. Evaluationsstandards

Standards für Evaluationen dienen als Bewertungsgrundlage für die Professionalität einer Evaluation. Dabei definieren sie die grundsätzlichen Qualitätsansprüche an Evaluationen sowie ethische Grundsätze, die im Wesentlichen aus der Entstehung der Evaluation aus den Sozialwissenschaften geschuldet sind. In sozialwissenschaftlichen Evaluationen steht der Mensch bzw. die Interaktion von Menschen in sozialen Systemen im Fokus bzw. beeinflussen Menschen den im Fokus stehenden Evaluationsgegenstand. Daten, die zur Bewertung herangezogen werden, spiegeln dabei die individuelle Meinung bzw. Bewertung des Evaluationsgegenstandes wieder. In kritischen Systemen ist es daher erforderlich, dass die Datenherkunft anonymisiert wird wobei der Evaluator sicherstellen muss, dass dieses zu 100% gewährleistet ist. So wird sichergestellt, dass die Probanden ihre tatsächliche Bewertung abgeben können, ohne Repressalien befürchten zu müssen.

In technischen Systemen spielt dies eine eher untergeordnete Rolle. Auch wenn Komponenten eines Computersystem miteinander agieren und sich innerhalb des Systems beeinflussen, so ist diese Interaktion jedoch wertfrei.

Während Evaluationen in sozialen Systemen über entsprechende mehrfach überarbeitete und auch über Landesgrenzen hinweg allgemein anerkannte Qualitätsstandards verfügen, existieren diese in der Informatik in dieser Form nicht. Auch wenn es Empfehlungen für die systematische und planvolle Planung, Durchführung und Analyse von Computersystemevaluationen gibt und dementsprechend ein Gefühl für eine qualitativ hochwertige Evaluation existiert, so lassen sich durchgeführte Evaluationen jedoch nicht hinsichtlich ihrer Qualität anhand allgemeingültiger Standards kontrollieren, bewerten und/oder klassifizieren. Darüber hinaus sind allgemeingültige Standards die Voraussetzung für Metaevaluationen, d.h. die Zusammenfassung und Analyse von mehreren, ähnlich gelagerten Evaluationen. Entsprechen diese nicht einheitlichen Standards, so lassen sich aus ihrer Gegenüberstellung keine zusätzlichen Informationen generieren.

Um den Mangel an geeigneten Standards in der Informatik entgegen zu treten, werden hier die Standards der Gesellschaft für Evaluation (DeGEval) herangezogen, um aus ihnen die auf Evaluationen von Computersystemen anwendbaren Standards zu identifizieren und gegebenenfalls anzupassen. Dabei handelt es sich lediglich um einen Ver-

such, für zukünftige Evaluationen einen Rahmen zu definieren, der die Qualität der Evaluationen sicherstellen soll. Bei den nachfolgend aufgeführten Standards handelt es sich um eine erste, nicht abschließende Auflistung und Beschreibung sondern vielmehr um einen ersten Draft, der zur Diskussion, Überarbeitung und Ergänzung anregen soll.

Nach DeGVal werden für Evaluationen vier Qualitätsstandards definiert:

Nützlichkeit: Evaluationen sollen einen Nutzen haben, d.h. sie sollen an den jeweiligen Informationsbedürfnissen der jeweiligen Nutzer ausgerichtet sein.

Durchführbarkeit: Evaluationen sollen die Realität widerspiegeln. Sie sollen dabei gut durchdacht und kostenbewusst durchgeführt werden.

Korrektheit: Evaluationen sollen korrekt ablaufen. Dabei sollen die Evaluationsergebnisse entsprechend ihres Entstehens Wahrheitsgetreu, Unverfälscht und Unbereinigt wiedergegeben werden.

Genauigkeit: Evaluationen sollen Informationen über die Güte des Evaluationsgegenstandes generieren und diese den jeweiligen Nutzern in angemessener Form vermitteln.

Zur besseren Darstellung und umfassenderen Spezifizierung werden die vier Standards der DeGEval in Unterpunkte unterteilt. Nachfolgend werden diese hinsichtlich des Evaluationsgegenstandes Computersysteme angepasst bzw. um diejenigen bereinigt, die sich ausschließlich auf Evaluationen von Sozialsystemen beziehen. Die kompletten Evaluationsstandards der DeGEval befinden sich in der Anlage dieses Exkurses.

1.1 Nützlichkeit

Evaluationen sollen einen Nutzen haben, d.h. sie sollen an den jeweiligen Informationsbedürfnissen der jeweiligen Nutzer ausgerichtet sein.

N1 Identifizierung der zu evaluierenden Komponenten, Prozesse und Nutzer der generierten Informationen

Das Informationsinteresse der Nutzer der Evaluation ist zu identifizieren und genau zu spezifizieren. Es ist die Basis zur Spezifizierung Des Evaluationsgegenstandes sowie zu dessen eindeutiger Abgrenzung vom Gesamtsystem. Es ist zu gewährleisten, dass die so gewonnenen Informationen in die Planung der Evaluation einfließen.

N2 Klärung des Zwecks der Evaluation

Der mit der Evaluation verfolgte Zweck muss eindeutig und transparent bestimmt und kommuniziert werden, damit der Arbeitsauftrag des Evaluationsteams klar und eindeutig formuliert sowie strukturiert und zielführend abgearbeitet werden kann.

N3 Glaubwürdigkeit und Kompetenz des Evaluators / der Evaluatorin

„Wer Evaluationen durchführt, soll persönlich glaubwürdig sowie methodisch und fachlich kompetent sein, damit bei den Evaluationsergebnissen ein Höchstmaß an Glaubwürdigkeit und Akzeptanz erreicht wird.“ (Standards der DeGEval)

N4 Auswahl und Umfang der Informationen

Die zu untersuchenden Fragestellungen stehen im Fokus der Evaluation. Die Auswahl und der Umfang der zu erfassenden Informationen sowie die zu verwendenden Messverfahren und Analysemethoden sollen dabei sicherstellen, dass die Fragestellungen umfassend beantwortet werden und den Informationsbedarf der Nutzer befriedigen.

N5 Transparenz von Werten

Die den Bewertungen zugrundeliegenden Messdaten müssen eindeutig dem Evaluationsgegenstand (Komponenten oder Prozessen) zugeordnet und beschrieben werden, um so sicher zu stellen, dass die Grundlagen der Bewertung klar und eindeutig nachvollziehbar dargestellt werden.

N6 Vollständigkeit und Klarheit der Berichterstattung

„Evaluationsberichte sollen alle wesentlichen Informationen zur Verfügung stellen, leicht zu verstehen und nachvollziehbar sein.“ (Standards der DeGEval)

N7 Rechtzeitigkeit der Evaluation

„Evaluationsvorhaben sollen so rechtzeitig begonnen und abgeschlossen werden, dass ihre Ergebnisse in anstehende Entscheidungsprozesse bzw. Verbesserungsprozesse einfließen können.“ (Standards der DeGEval)

N8 Nutzung und Nutzen der Evaluation

Aufbau, Struktur, Ergebnisse sowie die Präsentation der selbigen sollen die Bereitschaft und Motivation der Evaluationsnutzer fördern, Ergebnisse und Handlungsempfehlungen zur Kenntnis zu nehmen und diese in ihre zukünftige Arbeit einfließen zu lassen.

1.2 Durchführbarkeit

Evaluationen sollen die Realität widerspiegeln. Sie sollen dabei gut durchdacht und kostenbewusst durchgeführt werden.

D1 Angemessene Verfahren

Auswahl bzw. die entwickelten und verwendeten Messverfahren sind so zu gestalten, das die durch sie erzeugte Belastung des Evaluationsgegenstandes Verhältnismäßig zu dem zu erwartenden Nutzen der Evaluation steht.

D2 Akzeptanz schaffen

Aufbau, Struktur, Ergebnisse sowie die Präsentation der selbigen sollen eine hohe Akzeptanz der Evaluationsnutzer hinsichtlich Durchführung und Ergebnisse der Evaluation gewährleisten.

D3 Effizienz von Evaluation

„Der Aufwand für Evaluation soll in einem angemessenen Verhältnis zum Nutzen der Evaluation stehen.“ (Standards der DeGEval)

1.3 Korrektheit

Evaluationen sollen korrekt ablaufen. Dabei sollen die Evaluationsergebnisse entsprechend ihres Entstehens Wahrheitsgetreu, Unverfälscht und Unbereinigt wiedergegeben werden.

F1 Transparente und nachvollziehbare Vereinbarungen

Die Beteiligten einer Evaluation (Auftraggeber, Evaluationsteam und ggf. Evaluationsnutzer) sollen Gegenstand, Struktur, zu verwendende Methoden sowie die aus der Evaluation zu erwartenden Informationen für alle transparent (schriftlich) festhalten. Gegebenenfalls erforderliche Änderungen sollten gemeinsam ausgehandelt werden.

F2 Vollständige und faire Überprüfung

„Evaluationen sollen die Stärken und die Schwächen des Evaluationsgegenstandes möglichst vollständig und fair überprüfen und darstellen, so dass die Stärken weiter ausgebaut und die Schwachpunkte behandelt werden können.“ (Standards der DeGEval)

F3 Unparteiische Durchführung und Berichterstattung

„Die Evaluation soll unterschiedliche Sichtweisen von Beteiligten und Betroffenen auf Gegenstand und Ergebnisse der Evaluation in Rechnung stellen. Berichte sollen ebenso wie der gesamte Evaluationsprozess die unparteiische Position des Evaluationsteams erkennen lassen. Bewertungen sollen fair und möglichst frei von persönlichen Gefühlen getroffen werden.“ (Standards der DeGEval)

F4 Offenlegung der Ergebnisse

„Die Evaluationsergebnisse sollen allen Beteiligten und Betroffenen soweit wie möglich zugänglich gemacht werden.“ (Standards der DeGEval)

1.4 Genauigkeit

Evaluationen sollen Informationen über die Güte des Evaluationsgegenstandes generieren und diese den jeweiligen Nutzern in angemessener Form vermitteln.

G1 Beschreibung des Evaluationsgegenstandes

„Der Evaluationsgegenstand soll klar und genau beschrieben und dokumentiert werden, so dass er eindeutig identifiziert werden kann.“ (Standards der DeGEval)

G2 Kontextanalyse

„Der Kontext des Evaluationsgegenstandes soll ausreichend detailliert untersucht und analysiert werden.“ (Standards der DeGEval)

G3 Beschreibung von Zwecken und Vorgehen

„Gegenstand, Zwecke, Fragestellungen und Vorgehen der Evaluation, einschließlich der angewandten Methoden, sollen genau dokumentiert und beschrieben werden, so dass sie identifiziert und eingeschätzt werden können.“ (Standards der DeGEval)

G4 Angabe von Informationsquellen

„Die im Rahmen einer Evaluation genutzten Informationsquellen sollen hinreichend genau dokumentiert werden, damit die Verlässlichkeit und Angemessenheit der Informationen eingeschätzt werden kann.“ (Standards der DeGEval)

G5 Valide und reliable Informationen

„Die Verfahren zur Gewinnung von Daten sollen so gewählt oder entwickelt und dann eingesetzt werden, dass die Zuverlässigkeit der gewonnenen Daten und ihre Gültigkeit bezogen auf die Beantwortung der Evaluationsfragestellungen nach fachlichen Maßstäben sichergestellt sind. Die fachlichen Maßstäbe sollen sich an den Gütekriterien quantitativer und qualitativer Sozialforschung orientieren.“ (Standards der DeGEval)

G6 Systematische Fehlerprüfung

„Die in einer Evaluation gesammelten, aufbereiteten, analysierten und präsentierten Informationen sollen systematisch auf Fehler geprüft werden.“ (Standards der DeGEval)

G7 Analyse qualitativer und quantitativer Informationen

„Qualitative und quantitative Informationen einer Evaluation sollen nach fachlichen Maßstäben angemessen und systematisch analysiert werden, damit die Fragestellungen der Evaluation effektiv beantwortet werden können.“ (Standards der DeGEval)

G8 Begründete Schlussfolgerungen

„Die in einer Evaluation gezogenen Folgerungen sollen ausdrücklich begründet werden, damit die Adressatinnen und Adressaten diese einschätzen können.“ (Standards der DeGEval)

G9 Meta-Evaluation

„Um Meta-Evaluationen zu ermöglichen, sollen Evaluationen in geeigneter Form dokumentiert und archiviert werden.“ (Standards der DeGEval)