

Institut für Betriebssysteme und Rechnerverbund
Übungslösungen zur Vorlesung “Verteilte Systeme”, WS 02/03

<http://www.ibr.cs.tu-bs.de/lehre/ws0203/vs/>

Dozent: Prof. Dr. Stefan Fischer <fischer@ibr.cs.tu-bs.de> · Übungsleiter: Frank Strauß <strauss@ibr.cs.tu-bs.de>

Blätter wie dieses werden **nicht** zu allen Übungen und Aufgaben zur Verfügung gestellt. Sie entbinden **nicht** von zuvor selbst erarbeiteten Lösungen und auch **nicht** von ergänzenden Notizen während der Übungen. Sie dienen lediglich dazu, dass Sie sich das lästige Abschreiben oder Abzeichnen einiger umfangreicherer Lösungsteile ersparen können.

1 Einführung und Grundbegriffe

1.1 Verteilte Systeme und das Internet Protokoll

Übung am 06.11.2002

- IP ist ein offener Standard (Offenheit, Folie 1-18).
- Verbreitung *eines* Standard-Netzwerkprotokolls ist Voraussetzung für Interoperabilität.
- Nicht verbreiteter Kommunikationsmechanismus:
im allgemeinen nur Einsatz in lokal begrenzter Umgebung möglich.
- Offener, verbreiteter Kommunikationsstandard:
Einsatz auch LAN-übergreifend und zwischen Endsystemen, “die sich nicht kennen”, möglich.

1.2 Eng gekoppelte Systeme

Übung am 06.11.2002

- (a) Eng gekoppeltes System.
Lastverbund: SMP-fähige Betriebssysteme können alle existierenden CPUs den rechenbereiten Prozessen bzw. Threads zuordnen. Die Last des (Betriebs-)Systems wird verteilt.
Leistungsverbund: Rechenintensive Programme können parallelisiert werden und so mehrere CPUs nutzen. Die Leistung eines (Programm-)Systems wird erhöht.
Wartungsverbund: Wird die insgesamt benötigte CPU Leistung auf ein SMP-System konzentriert, so wird die Wartung vereinfacht. Es gibt SMP-Systeme mit “hot-pluggable” CPUs. Das Betriebssystem kann bei Ausfall einer CPU weiterlaufen.
- (b) Lose gekoppeltes System.
Informationsverbund: Zentral abgelegte Informationen sind für alle Clients zugänglich.
Wartungsverbund: Zentrale Speicherung der Daten erleichtert Übersicht und Backup.
- (c) Eng gekoppeltes System.
Funktionsverbund: Spezielle Subsysteme erbringen parallel verschiedene Teilfunktionen des Gesamtsystems.
Leistungsverbund: Die CPU wird durch “intelligente” Subsystem-ASICs¹ entlastet.

¹Application-Specific Integrated Circuit.

(a) Bus:

$$\text{links}(n) = n \text{ (zzgl. Bus)}$$

$$\text{maxhops}(n) = 1 \text{ (aber: eine gemeinsame Collision-Domain!)}$$

⊕ geringer Realisierungsaufwand

⊖ große Collision-Domain \implies skaliert sehr schlecht

(b) Ring:

$$\text{links}(n) = n$$

$$\text{maxhops}(n) = n - 1 \text{ (im bidirektionalen Ring: } \frac{n}{2} \text{)}$$

⊕ geringer Realisierungsaufwand

⊖ großes $\text{maxhops}(n) \implies$ skaliert schlecht

(c) Quadratischer Torus ("Donut"):

$$\text{links}(n) = 2n$$

$$\text{maxhops}(n) = \sqrt{n}$$

⊕ Realisierung ebenfalls einfach

⊕ skaliert relativ gut \implies guter Kompromiss

⊖ Verteilung mit Optimierung des Kommunikationsaufwandes relativ aufwändig

(d) Fat-Tree (Tiefe des Baumes: $\log_2(n)$):

$$\text{links}(n) = \sum_{i=1}^{\log_2(n)} 2^i = 2(n-1) \text{ (zzgl. } n-1 \text{ Switches)}$$

$$\text{maxhops}(n) = 2 \log_2(n)$$

⊕ skaliert relativ gut \implies guter Kompromiss

⊖ erfordert Switches

⊖ ⊕ zunehmende Bottlenecks im oberen Bereich des, aber:

⊖ Verteilung mit Optimierung des Kommunikationsaufwandes relativ aufwändig

(e) Vollständiger Graph:

$$\text{links}(n) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

$$\text{maxhops}(n) = 1$$

⊕ optimales $\text{maxhops}(n)$

⊕ Optimierung bei Verteilung von Teilaufgaben entfällt

⊖ hoher Realisierungsaufwand

(f) Hypercube (Stichworte: Dimension 4: vgl. Donut, Hamming-Abstand):

$$\text{links}(n) = \frac{n \log_2(n)}{2}$$

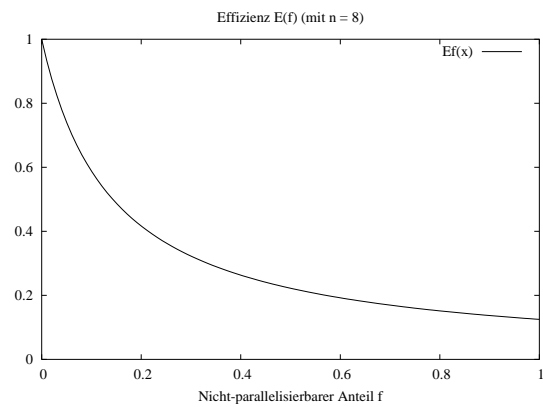
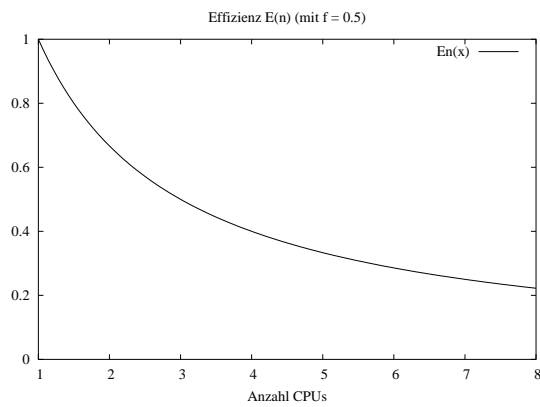
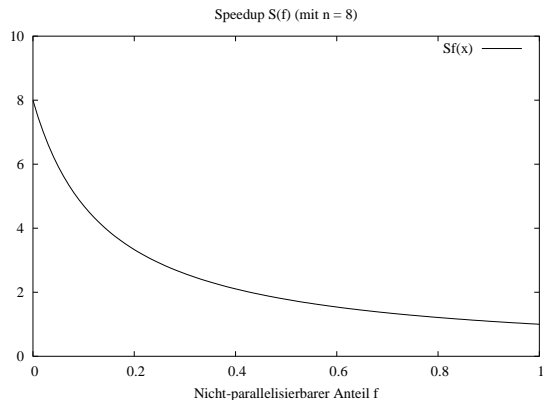
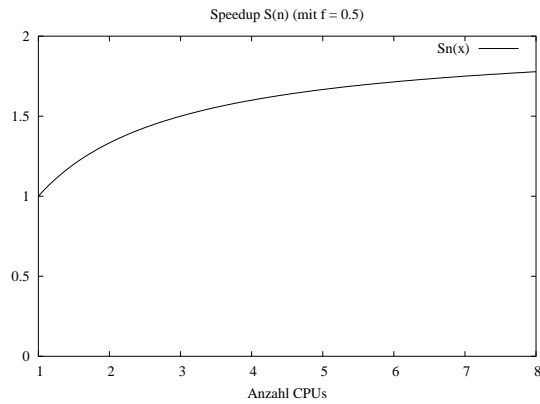
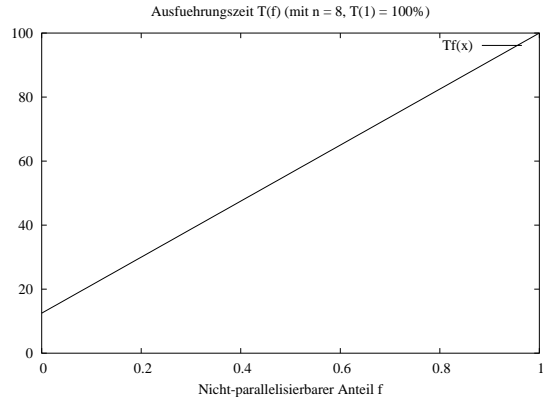
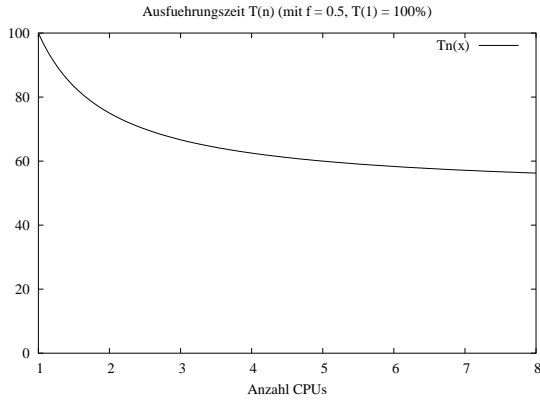
$$\text{maxhops}(n) = \log_2(n)$$

⊕ skaliert gut \implies guter Kompromiss

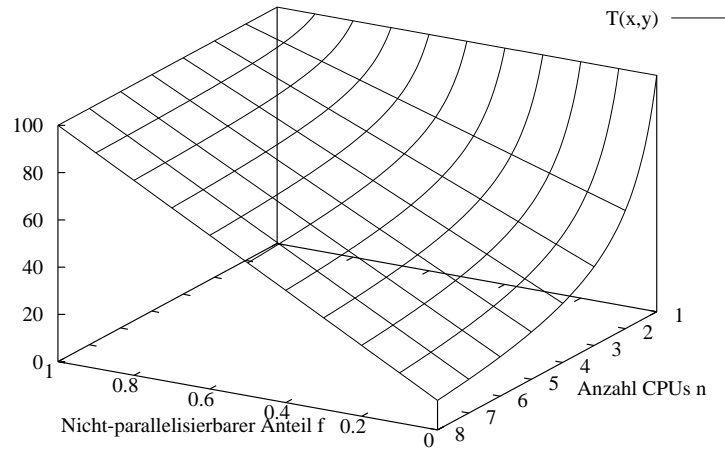
⊕ mehrere gleich gute Pfade, aber:

⊖ effiziente Nutzung erfordert aufwändige Software-Realisierung

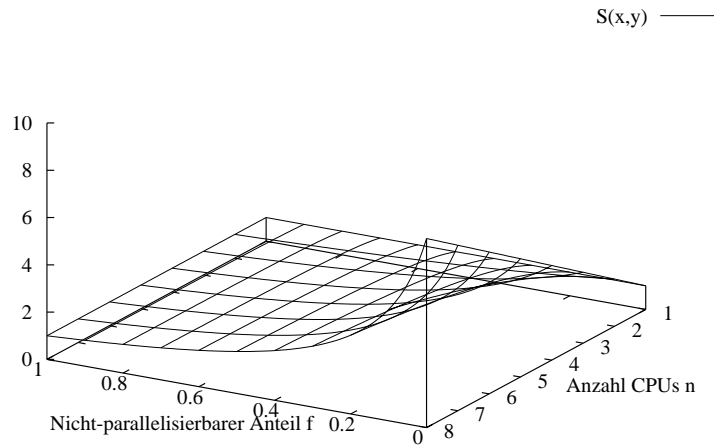
⊖ Hardware-Realisierung bei höherer Dimension kompliziert



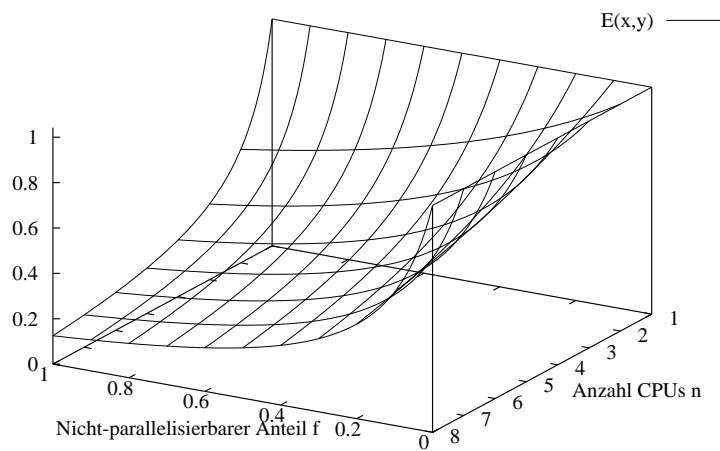
Ausführungszeit $T(f,n)$ ($T(1) = 100\%$)



Speedup $S(f,n)$



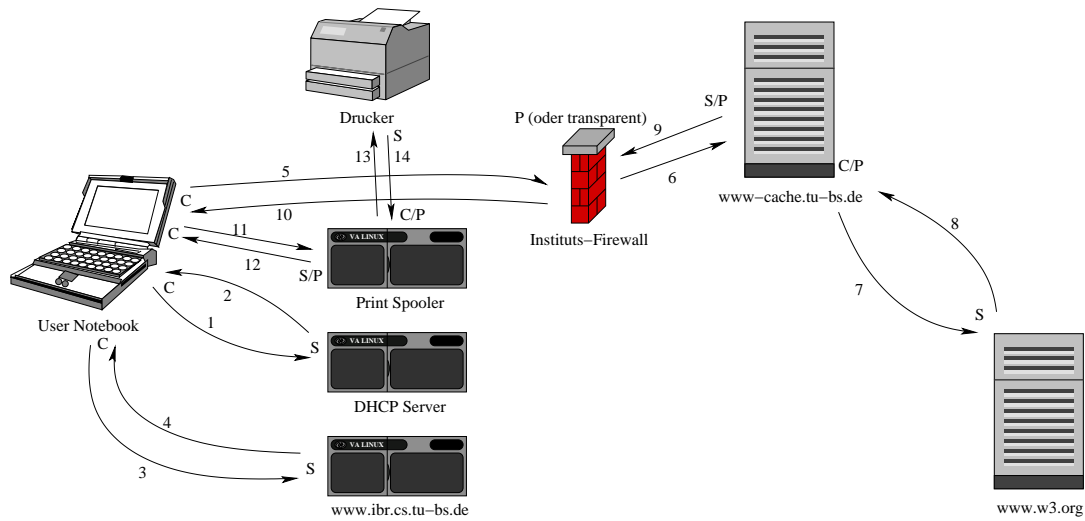
Effizienz $E(f,n)$



2 Systemmodelle

2.1 Zusammenspiel mehrerer Client/Server Beziehungen

Übung am 20.11.2002



2.2 Auswahl von Systemmodellen

Übung am 13.11.2002

Eigenschaft	Service Packs via Gnutella	Urheberrechtlich geschützte Inhalte
Performance	schwankend: von "sehr gut" aufgrund besonders naher Peers bis "unbrauchbar" wegen nicht gefundener Inhalte oder abgebrochener Verbindungen	Server Performance in Verantwortung des Anbieters, Netz-Performance: Best Effort ("wie üblich")
Quality of Service	vgl. Performance: keine Gewährleistung der QoS	vgl. Performance
Caching/Replikation	Da die Quellen nicht verlässlich sind, ist nur eine Prüfung der Integrität nach Empfang möglich.	kein Problem durch zentralen Zugriffspunkt
Verlässlichkeit	schlecht: Durch Peers ausgewiesene Authentizität und Integrität "ist nichts wert"	ist wichtig und kann durch geeignete Mechanismen realisiert werden
Konsequenz	schlecht geeignet	gut geeignet

Aber...