Seminar - Multicore Programmierung

Performanz

Max Binder

Fakultät für Informatik und Mathematik
Universität Passau

10. Februar 2011

Inhaltsverzeichnis

Inhaltsverzeichnis

Hinführung zur Thematik

- Fokusierung auf neue Forschungsgebiete, wie Green IT

Hinführung zur Thematik

- Fokusierung auf neue Forschungsgebiete, wie Green IT
- Single-Core CPU-Leistung stagniert

Hinführung zur Thematik

- Fokusierung auf neue Forschungsgebiete, wie Green IT
- Single-Core CPU-Leistung stagniert
- Multicores selbst in kleinen Geräten (Smartphones)

Inhaltsverzeichnis

Definition von Performanz

Unter dem Oberbegriff Performanz fasst man vier Teilbereiche zusammen, welche die Leistungsfähigkeit eines Datenverarbeitungssystems charakterisieren. Stets dreht es sich dabei um den Anspruch auf Ressourcen des Systems, wie:

- Zeit

Definition von Performanz

Unter dem Oberbegriff Performanz fasst man vier Teilbereiche zusammen, welche die Leistungsfähigkeit eines Datenverarbeitungssystems charakterisieren. Stets dreht es sich dabei um den Anspruch auf Ressourcen des Systems, wie:

- Zeit
- Energie

Definition von Performanz

Unter dem Oberbegriff Performanz fasst man vier Teilbereiche zusammen, welche die Leistungsfähigkeit eines Datenverarbeitungssystems charakterisieren. Stets dreht es sich dabei um den Anspruch auf Ressourcen des Systems, wie:

- Zeit
- Energie
- Prozessor

Definition von Performanz

Unter dem Oberbegriff Performanz fasst man vier Teilbereiche zusammen, welche die Leistungsfähigkeit eines Datenverarbeitungssystems charakterisieren. Stets dreht es sich dabei um den Anspruch auf Ressourcen des Systems, wie:

- Zeit
- Energie
- Prozessor
- Speicher

Inhaltsverzeichnis

Entwicklung dieses Forschungsbereichs

Situation vor Multicore

- Optimierung der Leistungsaufnahme der Hardware

Entwicklung dieses Forschungsbereichs

Situation vor Multicore

- Optimierung der Leistungsaufnahme der Hardware
- Drosselung der Prozessorgeschwindigkeit im idealen Maß

Entwicklung dieses Forschungsbereichs

Situation vor Multicore

- Optimierung der Leistungsaufnahme der Hardware
- Drosselung der Prozessorgeschwindigkeit im idealen Maß
- Single-Core CPUs mit meist nur einer Festplatte

Entwicklung dieses Forschungsbereichs

Situation vor Multicore

- Optimierung der Leistungsaufnahme der Hardware
- Drosselung der Prozessorgeschwindigkeit im idealen Maß
- Single-Core CPUs mit meist nur einer Festplatte

aktuelle Situation

- Komplexere Hardware-Systeme (Multicores, GPGPU)

Entwicklung dieses Forschungsbereichs

Situation vor Multicore

- Optimierung der Leistungsaufnahme der Hardware
- Drosselung der Prozessorgeschwindigkeit im idealen Maß
- Single-Core CPUs mit meist nur einer Festplatte

aktuelle Situation

- Komplexere Hardware-Systeme (Multicores, GPGPU)
- Kaum eine Steigerung bei Akku-Kapazitäten

Entwicklung dieses Forschungsbereichs

Situation vor Multicore

- Optimierung der Leistungsaufnahme der Hardware
- Drosselung der Prozessorgeschwindigkeit im idealen Maß
- Single-Core CPUs mit meist nur einer Festplatte

aktuelle Situation

- Komplexere Hardware-Systeme (Multicores, GPGPU)
- Kaum eine Steigerung bei Akku-Kapazitäten
- Optimierung von Software rückt zunehmend in den Fokus

Inhaltsverzeichnis

Begriffe

Response Time:

- Zeit die ein System benötigt um einen vorliegenden Task auszuführen

Begriffe

Response Time:

- Zeit die ein System benötigt um einen vorliegenden Task auszuführen
- Maßeinheit: Zeiteinheit pro Task

Begriffe

Response Time:

- Zeit die ein System benötigt um einen vorliegenden Task auszuführen
- Maßeinheit: Zeiteinheit pro Task

Throughput:

- Datendurchsatz eines Systems

Begriffe

Response Time:

- Zeit die ein System benötigt um einen vorliegenden Task auszuführen
- Maßeinheit: Zeiteinheit pro Task

Throughput:

- Datendurchsatz eines Systems
- Maßeinheit: Datenmenge pro Zeiteinheit

Gegenüberstellung: Response-Time - Throughput

Es herrscht keine reziproke Verwandtschaft dieser beiden Maße, wie folgendes Beispiel zeigt:

- Datendurchsatz 1000 Tasks pro Sekunde

Gegenüberstellung: Response-Time - Throughput

Es herrscht keine reziproke Verwandtschaft dieser beiden Maße, wie folgendes Beispiel zeigt:

- Datendurchsatz 1000 Tasks pro Sekunde
- Möglich: bis zu 1000 parallele, unabhängige Service-Kanäle bearbeiten je einen Task

Gegenüberstellung: Response-Time - Throughput

Es herrscht keine reziproke Verwandtschaft dieser beiden Maße, wie folgendes Beispiel zeigt:

- Datendurchsatz 1000 Tasks pro Sekunde
- Möglich: bis zu 1000 parallele, unabhängige Service-Kanäle bearbeiten je einen Task
- Folge: Response-Time liegt zwischen 0.001s und 1s

Gegenüberstellung: Response-Time - Throughput

Es herrscht keine reziproke Verwandtschaft dieser beiden Maße, wie folgendes Beispiel zeigt:

- Datendurchsatz 1000 Tasks pro Sekunde
- Möglich: bis zu 1000 parallele, unabhängige Service-Kanäle bearbeiten je einen Task
- Folge: Response-Time liegt zwischen 0.001s und 1s
- Ebenso wenig kann man aus der Response-Time den Throughput berechnen

Inhaltsverzeichnis

Prozentuale Abschätzungen

Die Formulierung von Performanz-Problemen sollte möglichst mit prozentualen Angaben erfolgen.
Negativbeispiel: “Die durchschnittliche Response-Time liegt bei 1 Sekunde oder weniger.”
Besser: “In 99% oder mehr Fällen liegt die Response-Time bei 1 Sekunde oder weniger.”
Gründe:

- Präzisierung durch Einbezug der Varianz

Prozentuale Abschätzungen

Die Formulierung von Performanz-Problemen sollte möglichst mit prozentualen Angaben erfolgen.
Negativbeispiel: “Die durchschnittliche Response-Time liegt bei 1 Sekunde oder weniger.”
Besser: “In 99% oder mehr Fällen liegt die Response-Time bei 1 Sekunde oder weniger.”
Gründe:

- Präzisierung durch Einbezug der Varianz
- Näher an der menschlichen Wahrnehmung

	List A	List B
1	.924	.796
2	.928	.798
3	.954	.802
4	.957	.823
5	.961	.919
6	.965	.977
7	.972	1.076
8	.979	1.216
9	.987	1.273
10	1.373	1.320

- Durchschnittliche Response-Time bei 1s

	List A	List B
1	.924	.796
2	.928	.798
3	.954	.802
4	.957	.823
5	.961	.919
6	.965	.977
7	.972	1.076
8	.979	1.216
9	.987	1.273
10	1.373	1.320

- Durchschnittliche Response-Time bei 1s
- Liste A: geringe Varianz, 10% liegen über erwarteter Response-Time

Prozentuale Abwägungen

	List A	List B
1	.924	.796
2	.928	.798
3	.954	.802
4	.957	.823
5	.961	.919
6	.965	.977
7	.972	1.076
8	.979	1.216
9	.987	1.273
10	1.373	1.320

- Durchschnittliche Response-Time bei 1s
- Liste A: geringe Varianz, 10% liegen über erwarteter Response-Time
- Liste B: hohe Varianz, 40% liegen über erwarteter Response-Time

Inhaltsverzeichnis

Definition: Sequenzdiagramm

- In UML formulierte Interaktionsabläufe

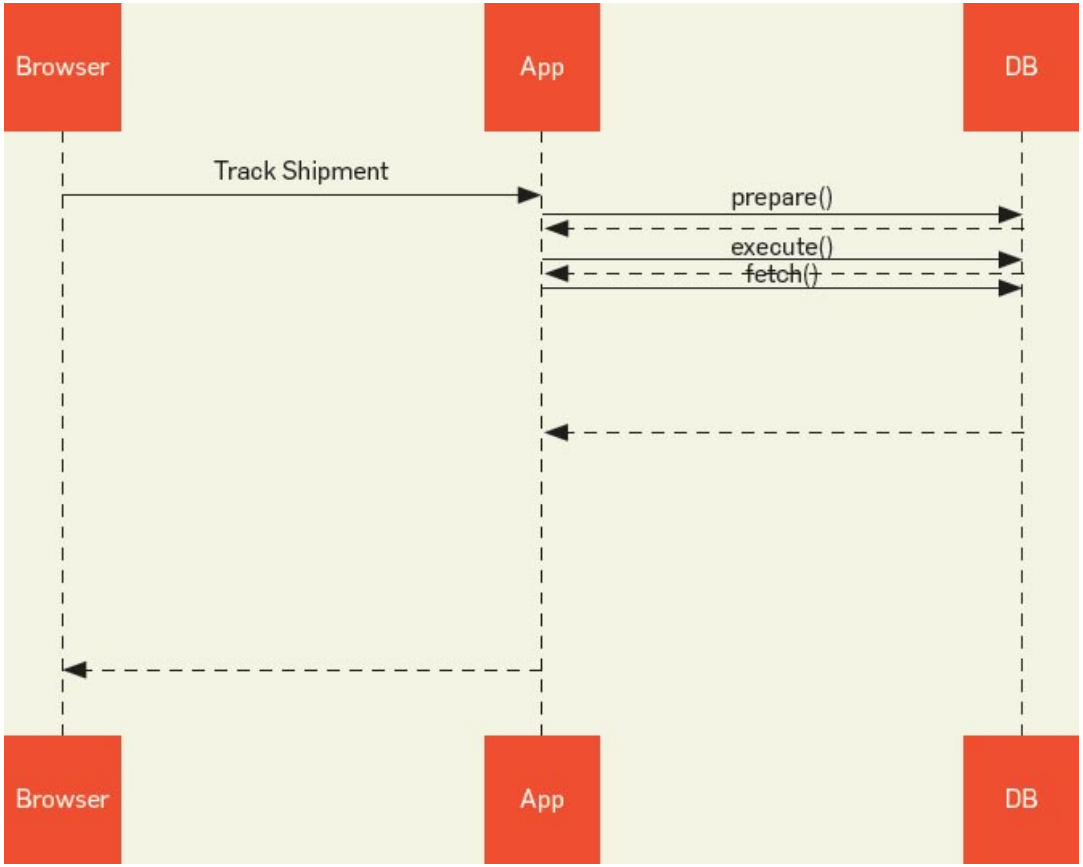
Definition: Sequenzdiagramm

- In UML formulierte Interaktionsabläufe
- Berücksichtigung der zeitlichen Abfolge

Definition: Sequenzdiagramm

- In UML formulierte Interaktionsabläufe
- Berücksichtigung der zeitlichen Abfolge
- Sinnvolle Skalierung erhöht die Aussagekraft

Beispiel: skaliertes Sequenzdiagramm



Inhaltsverzeichnis

Definition: Taskprofil

- Tabellarische Auflistung aller Funktionsaufrufe eines Tasks

Definition: Taskprofil

- Tabellarische Auflistung aller Funktionsaufrufe eines Tasks
- Ermöglicht Aggregation bei hoher Anzahl an Aufrufen

Definition: Taskprofil

- Tabellarische Auflistung aller Funktionsaufrufe eines Tasks
- Ermöglicht Aggregation bei hoher Anzahl an Aufrufen
- Liefert Hinweise auf eine Agenda zur Erhöhung der Performanz

Beispiel: einfaches Taskprofil

	Function Call	R (sec)	Calls
1	DB: fetch()	1,748.229	322,968
2	App: await_db_netIO()	338.470	322,968
3	DB: execute()	152.654	39,142
4	DB: prepare()	97.855	39,142
5	Other	58.147	89,422
6	App: render_graph()	48.274	7
7	App: tabularize()	23.481	4
8	App: read()	0.890	2
	Total	**2,468.000**	

Beispiel: erweitertes Taskprofil

	Potential improvement % and cost of investment	R (sec)	R (%)
1	34.5% super expensive	1,748.229	70.8%
2	12.3% dirt cheap	338.470	13.7%
3	Impossible to improve	152.654	6.2%
4	4.0% dirt cheap	97.855	4.0%
5	0.1% super expensive	58.147	2.4%
6	1.6% dirt cheap	48.274	2.0%
7	Impossible to improve	23.481	1.0%
8	0.0% dirt cheap	0.890	0.0%
	Total	**2,468.000**	

Inhaltsverzeichnis

Power-Down-Mechanismen

- Dienen der Verminderung des Energiebedarfs

Power-Down-Mechanismen

- Dienen der Verminderung des Energiebedarfs
- Beispiele: autom. Ausschalten des Monitors, Stand-By, Display-Helligkeit

Power-Down-Mechanismen

- Dienen der Verminderung des Energiebedarfs
- Beispiele: autom. Ausschalten des Monitors, Stand-By, Display-Helligkeit
- Greifen nach Überschreitung eines Schwellwertes

competitive-analysis

- Berechnung der optimalen Leistungsstufe ist ein online-Problem

competitive-analysis

- Berechnung der optimalen Leistungsstufe ist ein online-Problem
- “competitive-analysis” zur Bestimmung der Güte eines Algorithmus

competitive-analysis

- Berechnung der optimalen Leistungsstufe ist ein online-Problem
- “competitive-analysis” zur Bestimmung der Güte eines Algorithmus
- Vergleich des online-Algorithmus mit einem optimalen offline-Algorithmus

competitive-analysis

- Berechnung der optimalen Leistungsstufe ist ein online-Problem
- “competitive-analysis” zur Bestimmung der Güte eines Algorithmus
- Vergleich des online-Algorithmus mit einem optimalen offline-Algorithmus
- “c-competitive”, falls für jede mögliche Eingabe maximal c-mal Energie des optimalen Algorithmus verbraucht wird

competitive-analysis

- Berechnung der optimalen Leistungsstufe ist ein online-Problem
- “competitive-analysis” zur Bestimmung der Güte eines Algorithmus
- Vergleich des online-Algorithmus mit einem optimalen offline-Algorithmus
- “c-competitive”, falls für jede mögliche Eingabe maximal c-mal Energie des optimalen Algorithmus verbraucht wird
- Liefert starke worst-case Abschätzung, sogar bei “Angriffsszenarien”

Inhaltsverzeichnis

ACPI

- Advanced Configuration and Power Interface

ACPI

- Advanced Configuration and Power Interface
- Offener Industrie-Standard

ACPI

- Advanced Configuration and Power Interface
- Offener Industrie-Standard
- Erschien erstmals im Dezember 1996

ACPI

- Advanced Configuration and Power Interface
- Offener Industrie-Standard
- Erschien erstmals im Dezember 1996
- Ursprünglich entwickelt von Intel, Microsoft, und Toshiba - später auch HP und Phoenix

ACPI

- Advanced Configuration and Power Interface
- Offener Industrie-Standard
- Erschien erstmals im Dezember 1996
- Ursprünglich entwickelt von Intel, Microsoft, und Toshiba - später auch HP und Phoenix
- Definiert plattformunabhängige Interfaces für Konfiguration, Power-Management und Monitoring

ACPI

- Advanced Configuration and Power Interface
- Offener Industrie-Standard
- Erschien erstmals im Dezember 1996
- Ursprünglich entwickelt von Intel, Microsoft, und Toshiba - später auch HP und Phoenix
- Definiert plattformunabhängige Interfaces für Konfiguration, Power-Management und Monitoring
- Beispiele: Systemzustände (S0, ..., S5), Prozessorzustände (C0, ..., C3), Modems (D0, ..., D3)

Prozessorzustände gemäß ACPI

- C0: rechnender Zustand

Prozessorzustände gemäß ACPI

- C0: rechnender Zustand
- C1: Der Prozessor führt aktuell keine Instruktionen aus, kann aber nahezu ohne Zeitaufwand in den Zustand C0 wechseln.

Prozessorzustände gemäß ACPI

- C0: rechnender Zustand
- C1: Der Prozessor führt aktuell keine Instruktionen aus, kann aber nahezu ohne Zeitaufwand in den Zustand C0 wechseln.
- C2: Die Verwaltung der Software wird weiterhin aufrechterhalten, wobei der Wechsel in einen höheren Zustand eine geringe Wake-Up Zeit benötigt. (optional)

Prozessorzustände gemäß ACPI

- C0: rechnender Zustand
- C1: Der Prozessor führt aktuell keine Instruktionen aus, kann aber nahezu ohne Zeitaufwand in den Zustand C0 wechseln.
- C2: Die Verwaltung der Software wird weiterhin aufrechterhalten, wobei der Wechsel in einen höheren Zustand eine geringe Wake-Up Zeit benötigt. (optional)
- C3: In diesem Zustand muss der Prozessor seinen Cache nicht kohärent halten und kann nach einer Wake-Up Zeit in einen höheren Zustand wechseln. (optional)

Modellierung von Systemen mit mehreren Leistungsstufen

- I unterschiedliche Leistungslevel $s_1, ..., s_I$

Modellierung von Systemen mit mehreren Leistungsstufen

- l unterschiedliche Leistungslevel $s_1, ..., s_l$
- r_i sei die Leistungsaufnahme von s_i

Modellierung von Systemen mit mehreren Leistungsstufen

- I unterschiedliche Leistungslevel $s_1, ..., s_I$
- r_i sei die Leistungsaufnahme von s_i
- $r_1 > ... > r_I$, somit ist s_1 der aktivste Zustand

Modellierung von Systemen mit mehreren Leistungsstufen

- I unterschiedliche Leistungslevel $s_1, ..., s_I$
- r_i sei die Leistungsaufnahme von s_i
- $r_1 > ... > r_I$, somit ist s_1 der aktivste Zustand
- β_i sei Transitionsenergie von s_i nach s_1, mit $\beta_1 = 0$

Modellierung von Systemen mit mehreren Leistungsstufen

- I unterschiedliche Leistungslevel $s_1, ..., s_I$
- r_i sei die Leistungsaufnahme von s_i
- $r_1 > ... > r_I$, somit ist s_1 der aktivste Zustand
- β_i sei Transitionsenergie von s_i nach s_1, mit $\beta_1 = 0$
- $\beta_2 < ... < \beta_I$

Modellierung von Systemen mit mehreren Leistungsstufen

- l unterschiedliche Leistungslevel $s_1, ..., s_l$
- r_i sei die Leistungsaufnahme von s_i
- $r_1 > ... > r_l$, somit ist s_1 der aktivste Zustand
- β_i sei Transitionsenergie von s_i nach s_1, mit $\beta_1 = 0$
- $\beta_2 < ... < \beta_l$
- Wechsel zu niedrigwertigeren Zuständen sei kostenlos

Berechnungsformeln und Lower-Envelope

- Wechsel von s_j nach s_i, mit i < j : $\beta_j - \beta_i$

Berechnungsformeln und Lower-Envelope

- Wechsel von s_j nach s_i, mit i < j : $\beta_j - \beta_i$
- Energiekosten von s_i bis zum Zeitpunkt T: $r_i * T + \beta_i$

Berechnungsformeln und Lower-Envelope

- Wechsel von s_j nach s_i, mit i < j : $\beta_j - \beta_i$
- Energiekosten von s_i bis zum Zeitpunkt T: $r_i * T + \beta_i$
- optimale offline-Strategie bei Idle-Phase der Länge T : $OPT(T) = min\{r_i * T + \beta_i\}$, für $1 \leq i \leq I$

Berechnungsformeln und Lower-Envelope

- Wechsel von s_j nach s_i, mit i < j : $\beta_j - \beta_i$
- Energiekosten von s_i bis zum Zeitpunkt T: $r_i * T + \beta_i$
- optimale offline-Strategie bei Idle-Phase der Länge T : $OPT(T) = min\{r_i * T + \beta_i\}$, für $1 \leq i \leq I$
- Liefert Stützstellen t_i für Zustandstransitionen des online-Algorithmus

Berechnungsformeln und Lower-Envelope

- Wechsel von s_j nach s_i, mit i < j : $\beta_j - \beta_i$
- Energiekosten von s_i bis zum Zeitpunkt T: $r_i * T + \beta_i$
- optimale offline-Strategie bei Idle-Phase der Länge T : $OPT(T) = min\{r_i * T + \beta_i\}$, für $1 \leq i \leq I$
- Liefert Stützstellen t_i für Zustandstransitionen des online-Algorithmus
- $r_{i-1} * t + \beta_{i-1} = r_i * t + \beta_i$

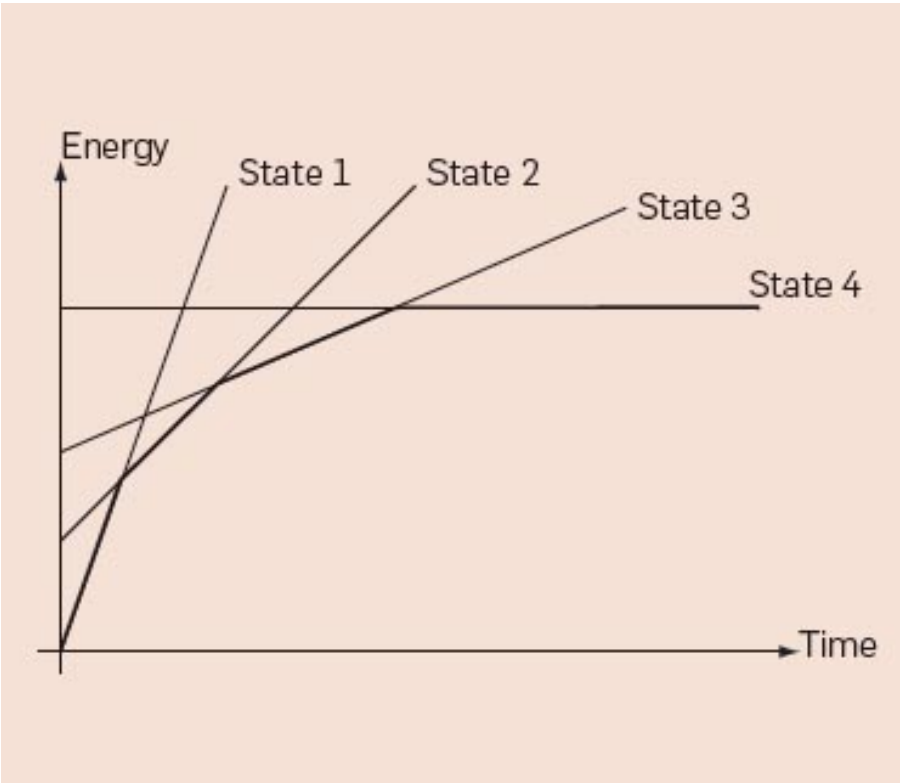
Berechnungsformeln und Lower-Envelope

- Wechsel von s_j nach s_i, mit i < j : $\beta_j - \beta_i$
- Energiekosten von s_i bis zum Zeitpunkt T: $r_i * T + \beta_i$
- optimale offline-Strategie bei Idle-Phase der Länge T : $OPT(T) = min\{r_i * T + \beta_i\}$, für $1 \leq i \leq I$
- Liefert Stützstellen t_i für Zustandstransitionen des online-Algorithmus
- $r_{i-1} * t + \beta_{i-1} = r_i * t + \beta_i$
- Durchlaufen dieser Stützstellen nennt man Lower-Envelope

Berechnungsformeln und Lower-Envelope

- Wechsel von s_j nach s_i, mit i < j : $\beta_j - \beta_i$
- Energiekosten von s_i bis zum Zeitpunkt T: $r_i * T + \beta_i$
- optimale offline-Strategie bei Idle-Phase der Länge T : $OPT(T) = min\{r_i * T + \beta_i\}$, für $1 \leq i \leq I$
- Liefert Stützstellen t_i für Zustandstransitionen des online-Algorithmus
- $r_{i-1} * t + \beta_{i-1} = r_i * t + \beta_i$
- Durchlaufen dieser Stützstellen nennt man Lower-Envelope
- Dieser Algorithmus ist 2-competitive

Grafische Darstellung eines Systems mit vier Leistungsstufen



Inhaltsverzeichnis

Mathematische Modellierung

- Die Jobs $J_1, ..., J_n$ sollen berechnet werden

Mathematische Modellierung

- Die Jobs $J_1, ..., J_n$ sollen berechnet werden
- Job J_i spezifiziert durch Realease-Time r_i, Deadline d_i und Berechnungsvolumen w_i

Mathematische Modellierung

- Die Jobs $J_1, ..., J_n$ sollen berechnet werden
- Job J_i spezifiziert durch Realease-Time r_i, Deadline d_i und Berechnungsvolumen w_i
- Dauer bei konstanter Geschwindigkeit s : w_i/s

Mathematische Modellierung

- Die Jobs $J_1, ..., J_n$ sollen berechnet werden
- Job J_i spezifiziert durch Realease-Time r_i, Deadline d_i und Berechnungsvolumen w_i
- Dauer bei konstanter Geschwindigkeit s : w_i/s
- Preemption sei erlaubt

Scheduling bei festen Deadlines

- Berechnung des Zeitintervalls $I = [t, t']$ mit maximaler Dichte Δ_I

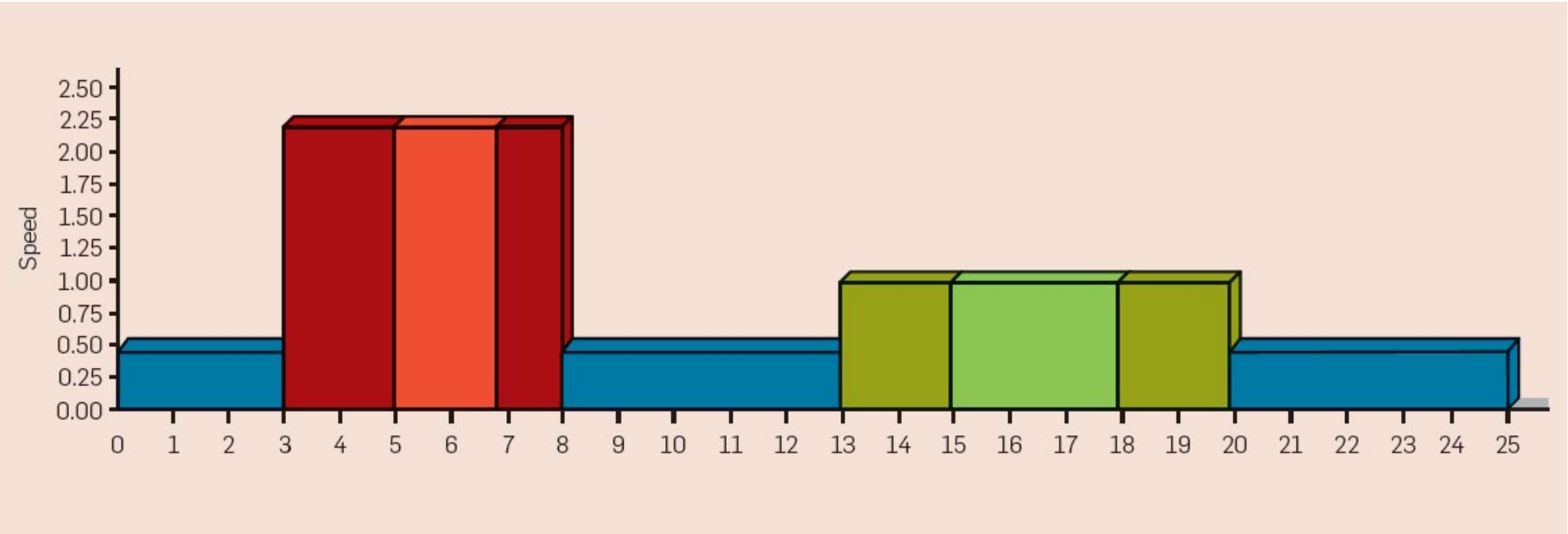
Scheduling bei festen Deadlines

- Berechnung des Zeitintervalls $I = [t, t']$ mit maximaler Dichte Δ_i
- $\Delta_i = \frac{1}{|I|} * \sum_{J_i \in S_I} w_i$

Scheduling bei festen Deadlines

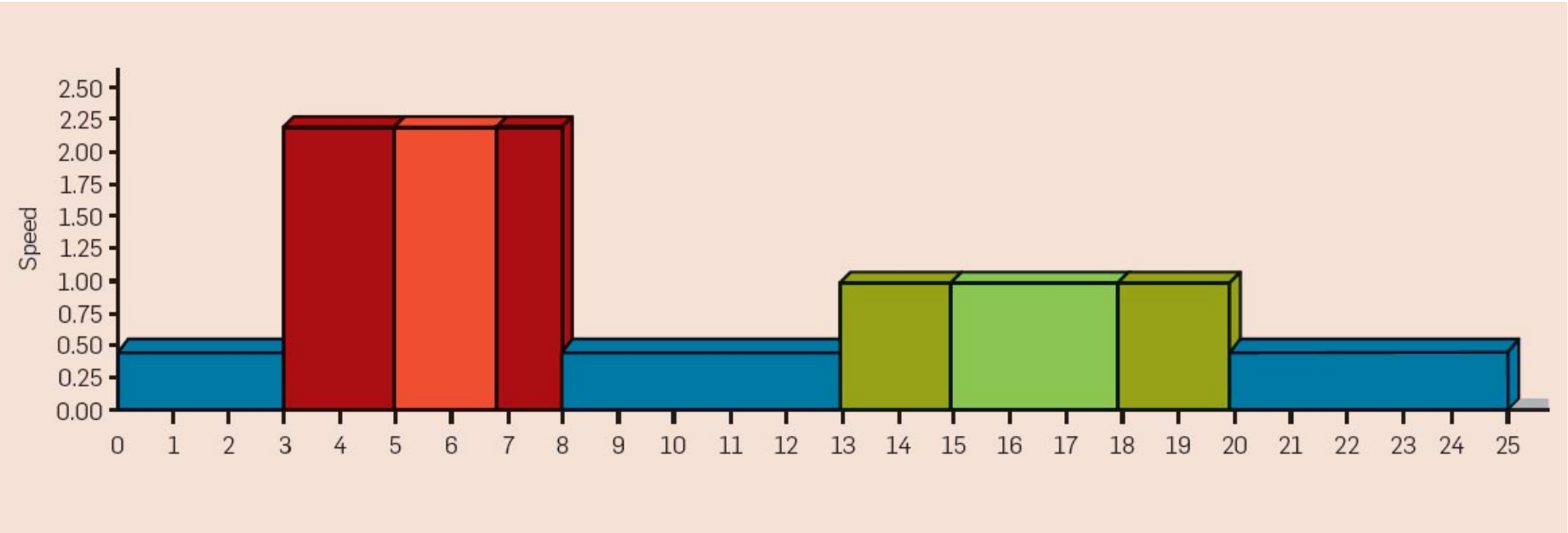
- Berechnung des Zeitintervalls $I = [t, t']$ mit maximaler Dichte Δ_i
- $\Delta_i = \frac{1}{|I|} * \sum_{J_i \in S_I} w_i$
- Erstellung eines partiellen Scheduls nach der Earliest-Deadline-First-Strategie (EDF)

Beispiel: Scheduling mittels YDS-Algorithmus



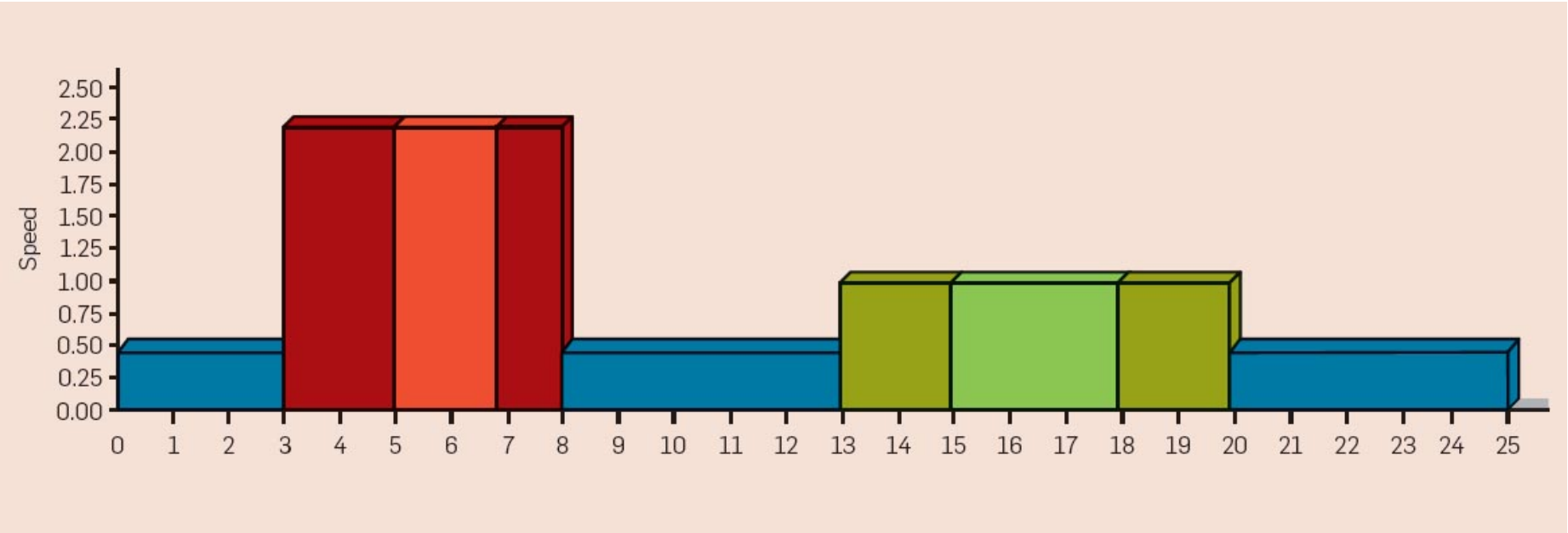
blau $J_1 = (0, 25, 9)$; rot $J_2 = (3, 8, 7)$; orange $J_3 = (5, 7, 4)$;
dunkelgrün $J_4 = (13, 20, 4)$; hellgrün $J_5 = (15, 18, 3)$

Beispiel: Scheduling mittels YDS-Algorithmus



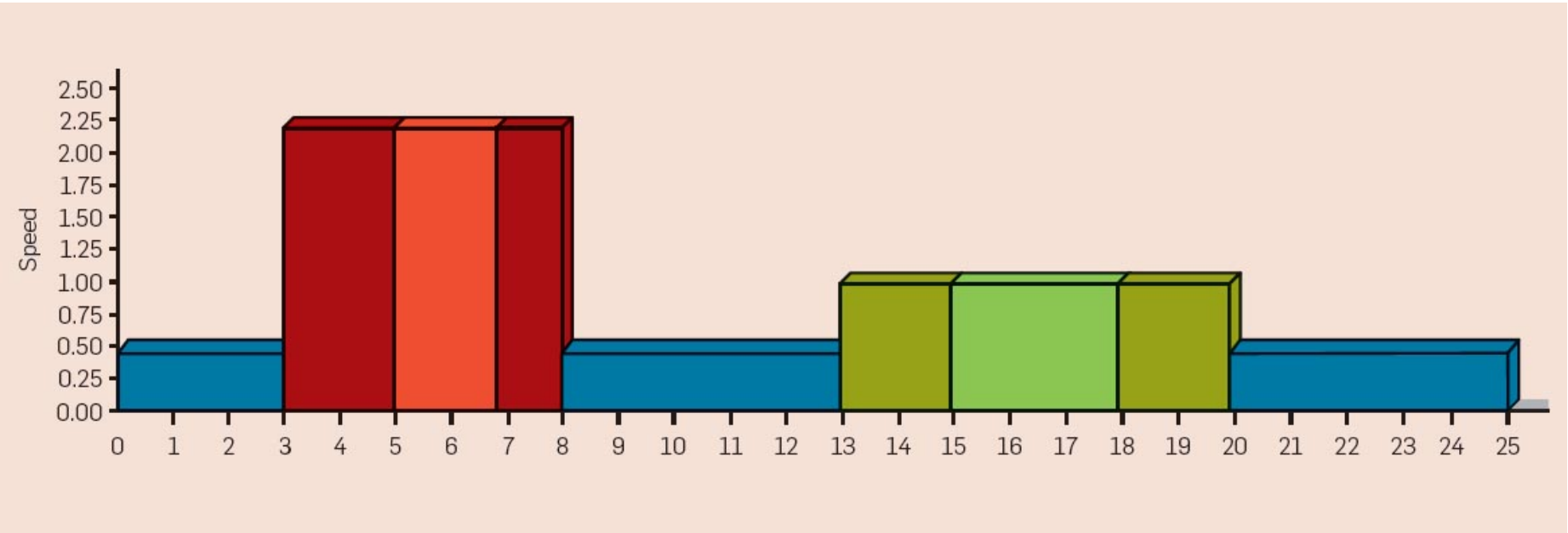
Iteration 1: $I = [3, 8]$, $S = \{J_2, J_3\}$, Preemption von J_2

Beispiel: Scheduling mittels YDS-Algorithmus



Iteration 2: $I = [13, 20]$, $S = \{J_4, J_5\}$, Preemption von J_4

Beispiel: Scheduling mittels YDS-Algorithmus



Iteration 3: Scheduling von J_3 in die übrigen Zeitslots

Fazit

- Energieoptimierungen auf algorithmischem Level wird weiterhin ein zentrales Thema sein

Fazit

- Energieoptimierungen auf algorithmischem Level wird weiterhin ein zentrales Thema sein
- Entwicklung von Modellen, die Zeitverzögerungen bei Zustandstransitionen berücksichtigen

Fazit

- Energieoptimierungen auf algorithmischem Level wird weiterhin ein zentrales Thema sein
- Entwicklung von Modellen, die Zeitverzögerungen bei Zustandstransitionen berücksichtigen
- Erforschung von Speed-Scaling in Multicores

Fazit

- Energieoptimierungen auf algorithmischem Level wird weiterhin ein zentrales Thema sein
- Entwicklung von Modellen, die Zeitverzögerungen bei Zustandstransitionen berücksichtigen
- Erforschung von Speed-Scaling in Multicores
- Einbezug der Laufzeiten der bekannten Optimierungsalgorithmen