Zugriffskontrolle in Mehrkern-Systemen

Eduard Potwigin

19.01.2016

Gliederung

Motivation

Prioritätsumkehr und Blockierung

Protokolle

Praxisvergleich
- Fallstudie 1
- Fallstudie 2

Zusammenfassung

Gliederung

Motivation

Prioritätsumkehr und Blockierung

Protokolle

Praxisvergleich
- Fallstudie 1
- Fallstudie 2

Zusammenfassung

Zugriffskontrolle

- **Überwachung und Steuerung gleichzeitiger Zugriffe auf gemeinsame aber unteilbare Ressourcen**
- **Wer** will auf **was** zugreifen?
  - Wer: **Arbeitsauftrag (Job)**
  - Was: **Betriebsmittel**

Aber warum überhaupt Zugriffskontrolle?

Zugriffskontrolle

Problem: Unteilbarkeit

Beispiel

Zugriffskontrolle

Abbildung 1: Beispiel für unkontrollierte Prioritätsumkehr. Quelle: FAU, Vorlesung Echtzeitsysteme, Kapitel 7 Zugriffskontrolle

- Prioritäten: $J_h > J_m > J_l$ (Gilt für den gesamten Vortrag)
- J_h verpasst seine Frist, weil J_m J_l verdrängt
- $\rightarrow$ Ziel: Minimierung der unkontrollierten Prioritätsumkehr

Gliederung

Motivation

Prioritätsumkehr und Blockierung

Protokolle

Praxisvergleich
- Fallstudie 1
- Fallstudie 2

Zusammenfassung

Definition

Arbeitsaufträge sind im **Streit** (contention) um ein Betriebsmittel, wenn einer das Betriebsmittel anfordert, das ein anderer bereits besitzt

- der anfordernde Job **blockiert** und wartet auf die Freigabe des Betriebsmittels durch den Job, der das Betriebsmittel belegt
- der das Betriebsmittel belegende Job löst den auf die Freigabe des Betriebsmittels wartenden Job aus, d.h. er deblockiert ihn wieder

Definition

- Sind zwei Arbeitsaufträge im Streit, so wird einer blockiert
  - Werkzeug für die Blockierung ist die Semaphore
- Hat der blockierte Job dabei eine höhere Priorität, dann spricht man von **Prioritätsumkehr**

Streit/Blockierung + falsche Priorität $\rightarrow$ Prioritätsumkehr

Lösungsansätze

- **Verdrängungssteuerung**
  - Arbeitsaufträge in kritischen Abschnitten können nicht verdrängt werden
  - Verklemmungsfrei
  - höher Priorisierte Aufträge werden manchmal unnötig blockiert
- **Prioritätsvererbung**
  - werden höher priorisierte Aufträge blockiert, dann vererben sie ihre Priorität weiter
  - verringert Blockierungszeit
  - nicht Verklemmungsfrei

Lösungsansätze

- **Verdrängungssteuerung**
  - Arbeitsaufträge in kritischen Abschnitten können nicht verdrängt werden
  - Verklemmungsfrei
  - höher Priorisierte Aufträge werden manchmal unnötig blockiert
- **Prioritätsvererbung**
  - werden höher priorisierte Aufträge blockiert, dann vererben sie ihre Priorität weiter
  - verringert Blockierungszeit
  - nicht Verklemmungsfrei
- **Prioritätsobergrenzen** $\rightarrow$ PCP
  - Obergrenzen werden den Betriebsmittel entsprechend der maximalen Priorität aller Tasks, die das Betriebsmittel verwenden, zugewiesen
  - Verklemmungsfrei
  - Erweiterung der Prioritätsvererbung

Lösungsansätze

- **Verdrängungssteuerung**
  - Arbeitsaufträge in kritischen Abschnitten können nicht verdrängt werden
  - Verklemmungsfrei
  - höher Priorisierte Aufträge werden manchmal unnötig blockiert
- **Prioritätsvererbung**
  - werden höher priorisierte Aufträge blockiert, dann vererben sie ihre Priorität weiter
  - verringert Blockierungszeit
  - nicht Verklemmungsfrei
- **Prioritätsobergrenzen** $\rightarrow$ PCP
  - Obergrenzen werden den Betriebsmittel entsprechend der maximalen Priorität aller Tasks, die das Betriebsmittel verwenden, zugewiesen
  - Verklemmungsfrei
  - Erweiterung der Prioritätsvererbung

Blockierungsarten

- **Direkte Blockierung**
  - J_h fordert ein gesperrtes Betriebsmittel an, das von J_l belegt ist
- **Blockierung durch Vererbung**
  - J_l erbt Priorität x von J_h. Nun wird J_m blockiert, obwohl J_m das Betriebsmittel nicht anfordert

Blockierungsarten

- **Direkte Blockierung**
  - J_h fordert ein gesperrtes Betriebsmittel an, das von J_l belegt ist
- **Blockierung durch Vererbung**
  - J_l erbt Priorität x von J_h. Nun wird J_m blockiert, obwohl J_m das Betriebsmittel nicht anfordert
- **Blockierung durch Prioritätsobergrenze**
  - Ein Job wird von der Obergrenze eines Betriebsmittels blockiert

Blockierungsarten

- **Direkte Blockierung**
  - J_h fordert ein gesperrtes Betriebsmittel an, das von J_l belegt ist
- **Blockierung durch Vererbung**
  - J_l erbt Priorität x von J_h. Nun wird J_m blockiert, obwohl J_m das Betriebsmittel nicht anfordert
- **Blockierung durch Prioritätsobergrenze**
  - Ein Job wird von der Obergrenze eines Betriebsmittels blockiert
- **Entfernte Blockierung**
  - Ein Job wartet auf die Ausführung eines anderen Jobs, der auf einem anderen Prozessorkern ausgeführt wird

Blockierungsarten

- **Direkte Blockierung**
  - J_h fordert ein gesperrtes Betriebsmittel an, das von J_l belegt ist
- **Blockierung durch Vererbung**
  - J_l erbt Priorität x von J_h. Nun wird J_m blockiert, obwohl J_m das Betriebsmittel nicht anfordert
- **Blockierung durch Prioritätsobergrenze**
  - Ein Job wird von der Obergrenze eines Betriebsmittels blockiert
- **Entfernte Blockierung**
  - Ein Job wartet auf die Ausführung eines anderen Jobs, der auf einem anderen Prozessorkern ausgeführt wird

Gliederung

Motivation

Prioritätsumkehr und Blockierung

Protokolle

Praxisvergleich
- Fallstudie 1
- Fallstudie 2

Zusammenfassung

Multiprocessor Priority Ceiling Protocol

- **MPCP** erweitert PCP für den Multikernbetrieb mittels **globaler** Obergrenzen
  - Prioritätsobergrenzen über Prozessorkerngrenzen hinweg müssen größer als alle anderen Obergrenzen sein
- Wartet ein Job auf ein globales Betriebsmittel, dann wird er in eine **priorisierte Warteschlange** eingefügt
- Ist J_h blockiert, belegt aber kein Betriebsmittel, dann darf J_l ausgeführt werden

Multiprocessor Stack Resource Policy

- **MSRP** verwendet **Verdrängungswerte** anstatt Prioritäten
  - Basieren auf relativen Fristen
  - Je kürzer die Frist desto höher der Verdrängungswert
- Ein Job verdrängt einen laufen Job wenn
  - die absolute Frist kürzer ist und
  - der Verdrängungswert höher als die aktuelle Obergrenze ist
- Wartet ein Job auf ein globales Betriebsmittel, dann wird er in eine **FCFS Warteschlange** eingefügt
- belegt ein Job ein globales Betriebsmittel, dann wird er für diesen Abschnitt nicht verdrängbar

Vergleich

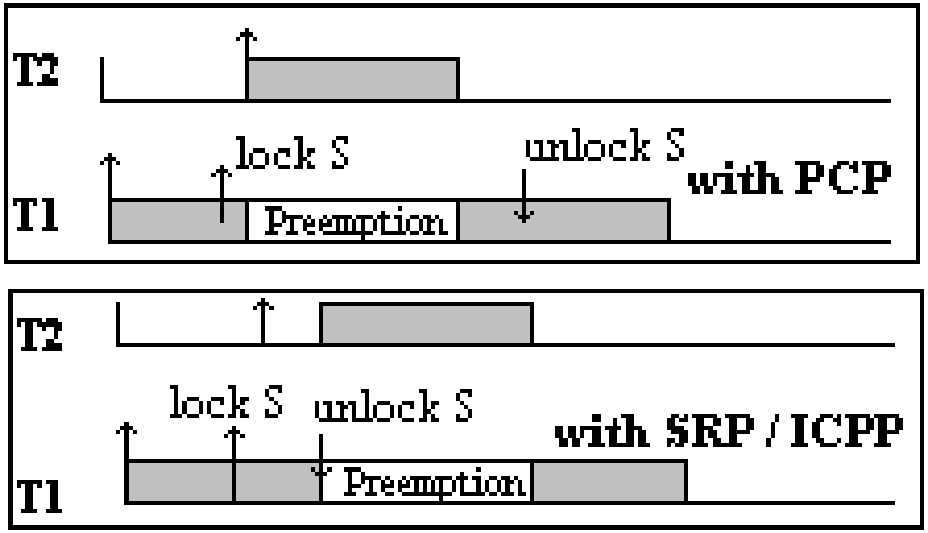

Abbildung 2: Vergleich von PCP mit SRP. Quelle: (*)

- Prioritäten: T2 >T1 und S >T2
- Wesentlicher Unterschied: **Einlastungszeitpunkt** von T2

Gliederung

Motivation

Prioritätsumkehr und Blockierung

Protokolle

Praxisvergleich
- Fallstudie 1
- Fallstudie 2

Zusammenfassung

Fallstudie 1

- Vergleich von PCP und SRP im **Einkernbetrieb**
- Fokus auf genaue **Antwortzeiten**
- Anwendung des GAP (General Avionics Platform) Tasksystems (siehe Ausarbeitung)
  - 16 Tasks sind periodisch
  - 1 Task (Task 11) ist sporadisch

$\rightarrow$ PCP schneidet besser ab

PCP vs SRP

Task	Period T	Deadline D	WCET C	WCRT PCP	WCRT SRP
1	200000	5000	3000	4180	4180
2	25000	25000	2100	6480	6480
3	25000	25000	4200	13180	14930
4	40000	40000	1000	14280	16030
5	50000	50000	3000	17580	19330
6	50000	50000	5000	23344	24694
7	59000	59000	8000	38848	40198
8	80000	80000	9000	49648	50998
9	80000	100000	2000	40780	43630
10	99995	115000	5000	98954	122604
11	200000	200000	1000	138794	140144
12	200000	200000	1000	138860	158510
13	200000	200000	1000	139926	159576
14	200000	200000	3000	144540	144540
15	199995	200000	3100	146488	146488
16	1000000	1000000	1000	147554	147554
17	1000000	1000000	1000	148620	148620
			Total:	1312056	1388506

Abbildung 3: Vergleich von PCP mit SRP bezüglich der ungünstigsten Antwortszeit (WCRT). Quelle: (*)

PCP vs SRP

Task	Period T	Deadline D	WCET C	ACRT PCP	ACRT SRP
1	200000	5000	3000	3236	3236
2	25000	25000	2100	2704	2704
3	25000	25000	4200	7804	7834
4	40000	40000	1000	4207	5214
5	50000	50000	3000	11959	12018
6	50000	50000	5000	17764	18199
7	59000	59000	8000	18798	19681
8	80000	80000	9000	29628	33424
9	80000	100000	2000	19763	40260
10	99995	115000	5000	84008	89773
11	200000	200000	1000	115190	119060
12	200000	200000	1000	117776	122786
13	200000	200000	1000	118842	123852
14	200000	200000	3000	107736	115676
15	199995	200000	3100	122894	122894
16	1000000	1000000	1000	147554	147554
17	1000000	1000000	1000	148620	148620
			Total:	1078483	1132785

Abbildung 4: Vergleich von PCP mit SRP bezüglich der Durchschnittsantwortszeit (ACRT). Quelle: (*)

Fallstudie 2

- **Echtzeit-Datenerfassung und -Verarbeitung** von Sensorwerten und Informationen von verteilten Steuerungsrechnern in einem **Stahlwerk**
- ausgiebige Tests mit Variationen in
  - Taskanzahl: 10 bis 100
  - Periodendauer: 50 bis 1000 Millisekunden
  - Anzahl der Betriebsmittelanforderungen pro Task: 0 bis 8
  - Dauer des Gebrauchs von Betriebsmitteln: 0,02 bis 0,2 Millisekunden

Fallstudie 2

- Verwendete Schedulingverfahren: **EDF** und **RM**
- Die gemessenen Eigenschaften umfassen:
  1. verpasste **Frist**
  2. **Wartezeit** für eine Betriebsmittelanforderung
  3. **Prioritätsänderung**
  4. **Kontextwechsel**
- Zum Vergleich wird das **FCFS**-Verfahren gegenübergestellt

Verpasste Fristen

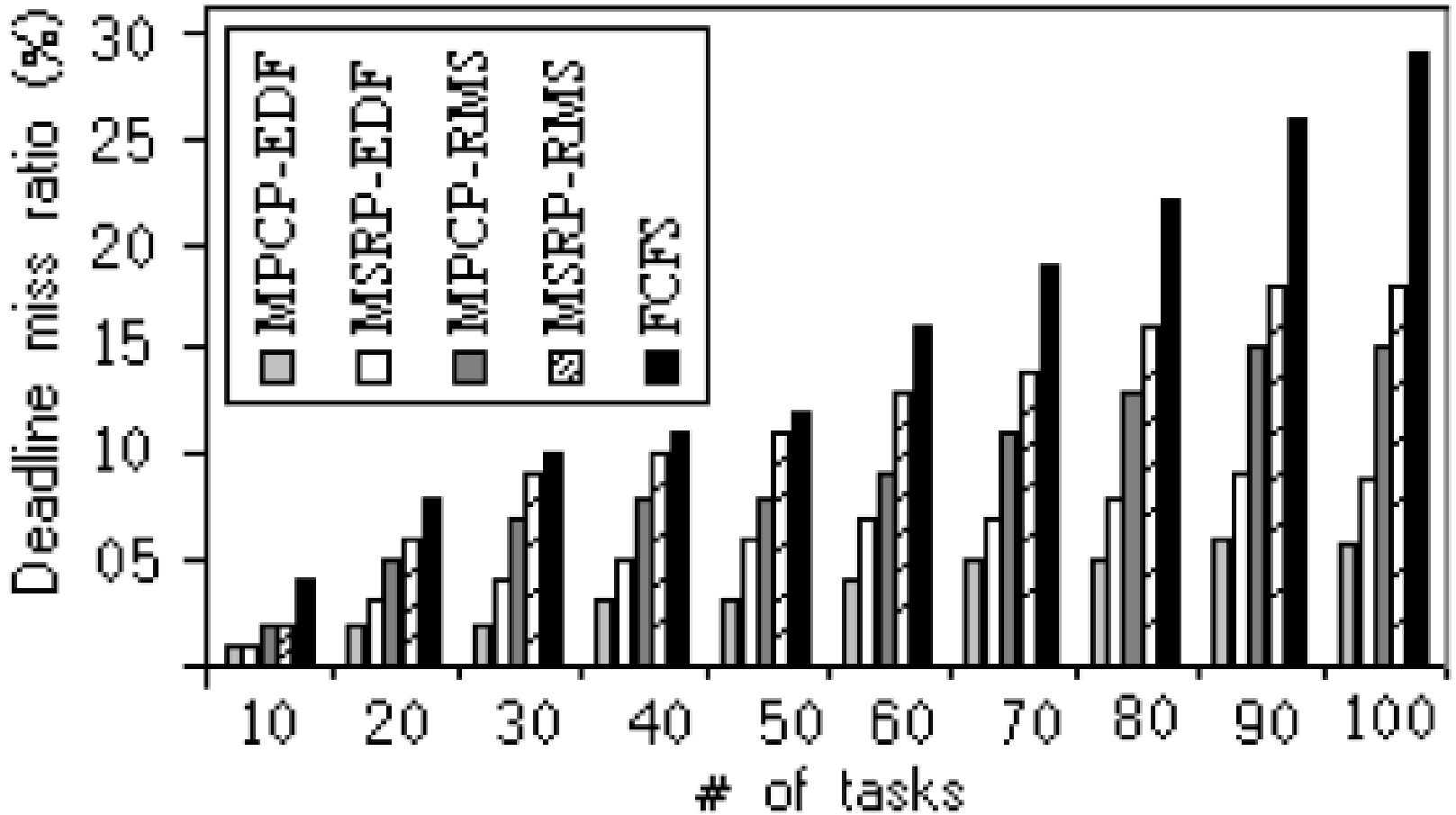

Abbildung 5: Vergleich der Anzahl der verpassten Fristen. Quelle: (*)

Wartezeiten

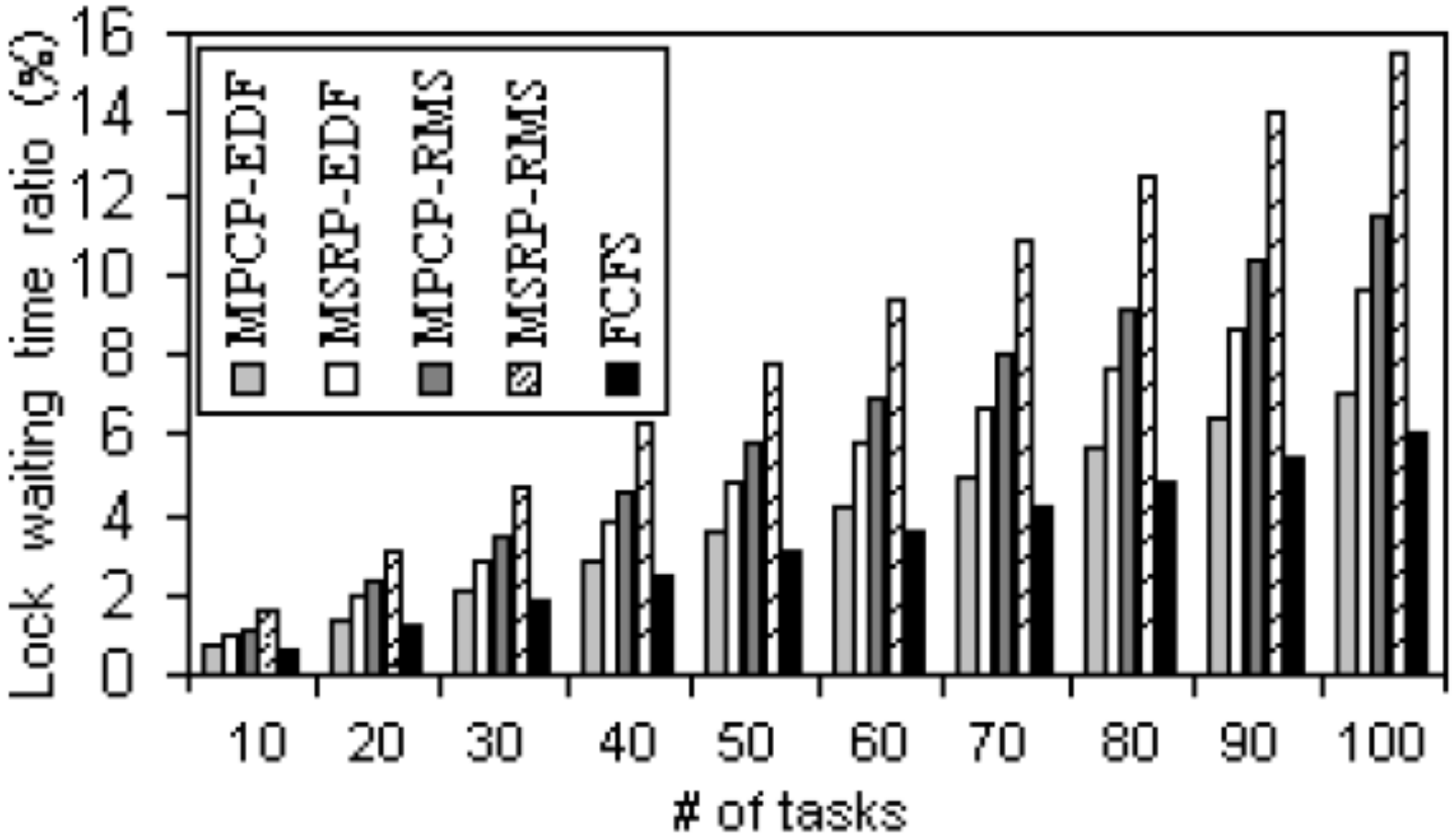

Abbildung 6: Vergleich der Wartezeiten. Quelle: (*)

Prioritätsänderungen

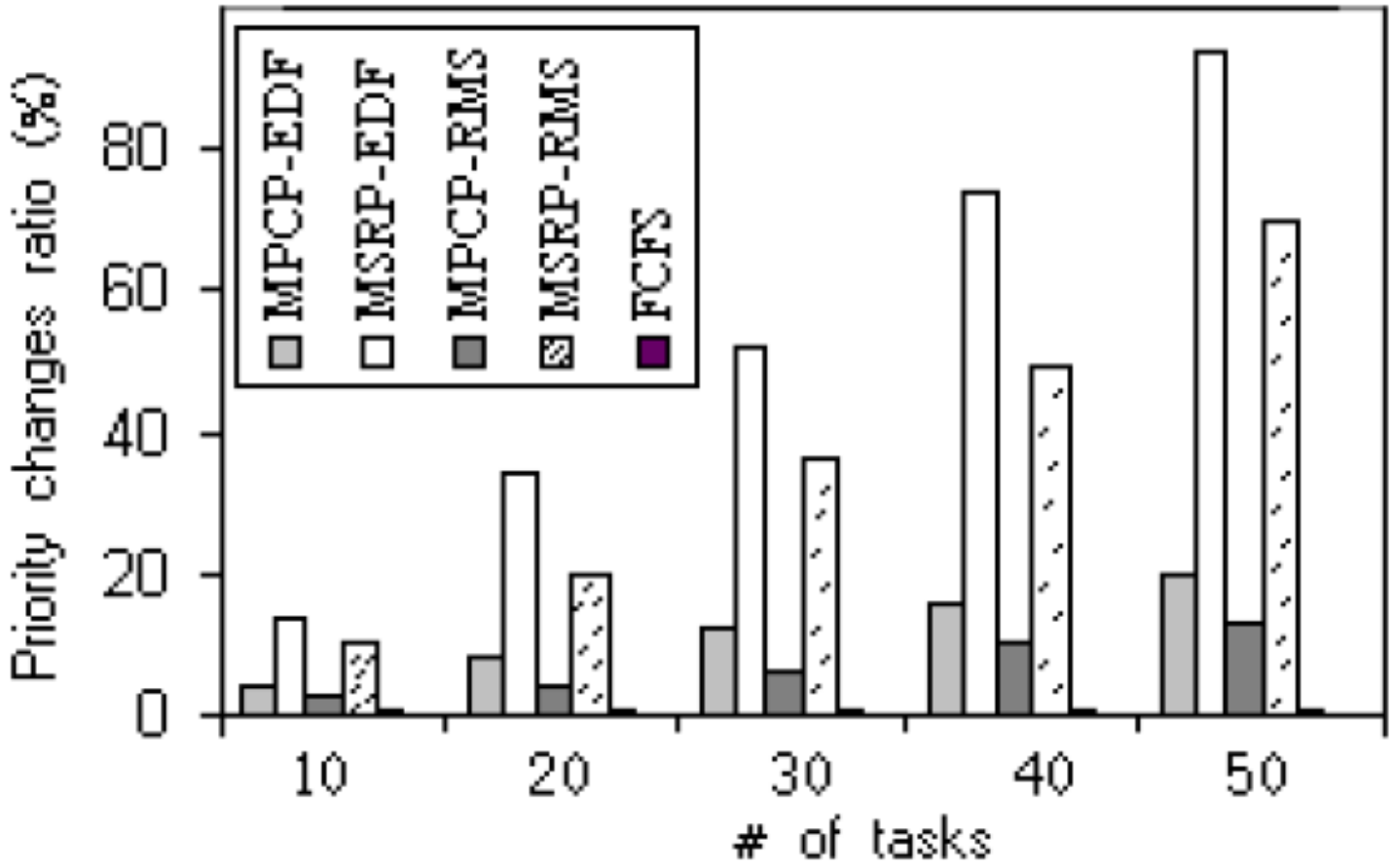

Abbildung 7: Vergleich der Anzahl der Prioritätsänderungen. Quelle: (*)

Kontextwechsel

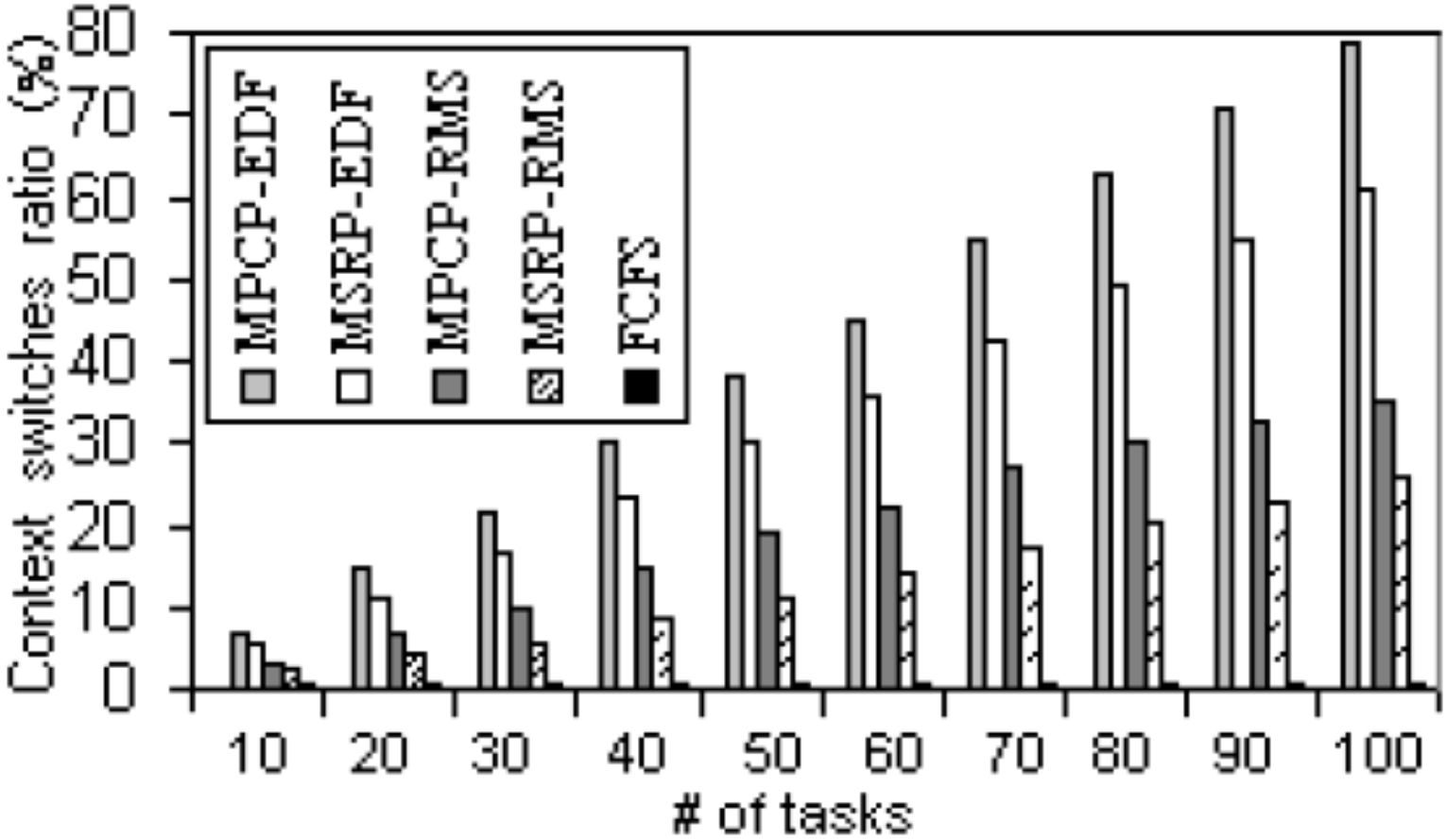

Abbildung 8: Vergleich der Anzahl der Kontextwechsel. Quelle: (*)

Gliederung

Motivation

Prioritätsumkehr und Blockierung

Protokolle

Praxisvergleich
- Fallstudie 1
- Fallstudie 2

Zusammenfassung

Zusammenfassung

- Zugriffskontrolle ist wichtig
- Prioritätsumkehr wird es immer geben, wenn unteilbare Betriebsmittel von mehreren Tasks benötigt werden
- MPCP schneidet leicht besser ab als MSRP, wobei MSRP in manchen Fällen auch besser sein kann
- MPCP ändert die Priorität erst, wenn es nötig ist
  + weniger Prioritätsumkehr
  - Entscheidung verursacht Overhead
- MSRP blockiert direkt bei Belegung eines Betriebsmittels
  + weniger Kontextwechsel
  - mehr Prioritätsumkehr

Quelle

- *(*)* J. Ras and A. M. Cheng. An evaluation of the dynamic and static multiprocessor priority ceiling protocol and the multiprocessor stack resource policy in an smp system. In Real-Time and Embedded Technology and Applications Symposium, 2009. RTAS 2009. 15th IEEE, pages 13-22. IEEE, 2009.

Danke für die Aufmerksamkeit!

Noch Fragen?