

Zentralübung Rechnerstrukturen: Cache-Kohärenz 7. Übungsblatt

Besprechung: 20. Juli 2017

1 Cache: Grundlagen

1.1 Cacheleistung

Bei dem Entwurf eines Systems stehen zwei Entwurfsalternativen zur Auswahl. In beiden Entwurfsalternativen kommt eine zwei-stufige Cache-Hierarchie zum Einsatz. Entwurfsalternative A hat einen kleinen L1-Cache mit einer Zugriffszeit von $t_{A-L1} = 10ns$, sowie einen L2-Cache mit einer Zugriffszeit von $t_{A-L2} = 30ns$. In Entwurfsalternative B kommt ein größerer L1-Cache mit einer Zugriffszeit von $t_{B-L1} = 12ns$ zum Einsatz, sowie ein L2-Cache mit einer Zugriffszeit von $t_{B-L2} = 25ns$. Die Zugriffszeit des Hauptspeichers sei in beiden Entwurfsalternativen gleich und betrage $t_{Mem} = 100ns$.

- a) Geben Sie eine allgemeine Formel zur Berechnung t_a der mittleren Zugriffszeit in einer zwei-stufigen Cache-Hierarchie, bei der Zugriffe auf die nächsten Ebenen parallel stattfinden, an.
- b) Bei der Evaluation beider Entwurfsalternativen wurden folgende Hit-Raten gemessen:
 - Alternative A: $r_{A-L1} = 70\%$, sowie $r_{A-L2} = 40\%$
 - Alternative B: $r_{B-L1} = 75\%$, sowie $r_{B-L2} = 35\%$

Für welche Entwurfsalternative würden Sie sich entscheiden? Begründen Sie ihre Antwort.

1.2 Beweise

Beweisen oder widerlegen Sie folgende Behauptungen:

- a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.
- b) Vollasoziative Caches haben im Vergleich zu satzassoziativen Caches immer eine niedrigere Miss-Rate.

Hinweis: Gehen Sie in ihren Überlegungen davon aus, dass die Caches gemäß Least-Recently-Used-(LRU)-Strategie verdrängen.

1.3 Verständnisfragen

- a) Welche zwei Eigenschaften von Anwendungen werden von Caches ausgenutzt?
- b) Welche Arten von Cache-Misses können unterschieden werden?
- c) Warum wird der Cache-Speicher nicht deutlich größer (Größenbereich 100 MB - mehrere GB) angelegt?

2 Cache-Kohärenzprotokolle

2.1 MESI

Ein Multiprozessorsystem besteht aus drei Prozessoren, die über einen gemeinsamen Speicher gekoppelt sind. Die Caches haben je eine Größe von zwei Cachezeilen, welche je genau ein Speicherwort aufnehmen können. Die Füllung des Caches erfolgt von der niedrigsten Cachezeile an aufwärts, sofern noch freie Zeilen zur Verfügung stehen, andernfalls wird gemäß LRU-Strategie verdrängt. Als Cache-Kohärenzprotokoll komme das MESI-Protokoll zum Einsatz.

- a) Vervollständigen Sie die gegebene Tabelle. Geben Sie jeweils Inhalt der Cachezeile und MESI-Zustand an.

Prozessor	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6						
2	rd 2						
1	rd 4						
3	rd 4						
2	rd 3						
3	wr 7						
1	wr 4						
2	rd 7						
3	wr 5						
1	rd 3						
3	wr 3						
2	wr 7						

2.2 MOESI

Ein Dreiprozessorsystem sei speichergekoppelt. Die Caches haben je eine Größe von zwei Cachezeilen, welche je genau ein Speicherwort aufnehmen können. Die Füllung des Caches erfolgt von der niedrigsten Cachezeile aufwärts, sofern noch freie Zeilen zur Verfügung stehen, andernfalls wird gemäß LRU-Strategie verdrängt. Als Cache-Kohärenzprotokoll komme das MOESI-Protokoll zum Einsatz. Der Cache sei initial leer.

Das MOESI-Protokoll ist ein um einen weiteren Zustand *O* erweitertes MESI-Protokoll. Dieser Zustand beschleunigt den Zugriff auf Daten, die in entfernten Caches bereits als modifiziert gekennzeichnet sind, indem die Daten direkt vom Cache des ersten Prozessorkerns in den Cache des zweiten Prozessorkerns transferiert werden. Dies erspart das Zurückschreiben und das anschließende Lesen der modifizierten Daten aus dem Hauptspeicher.

Wird ein in einem Cache bereits modifiziertes Datum von einem weiteren Cache gelesen, so wechselt der Cache mit dem modifizierten Datum von Zustand *M* in den Zustand *O*. Der lesende Cache übernimmt das Datum aus dem Cache des ersten Prozessors und lagert das Datum, gemäß MESI-Protokoll, als Shared *S* markiert ein. Werden die Daten in Caches mit dem Zustand *S* modifiziert, so wird das Datum in den anderen Caches invalidiert (Zustand *I*) und in den Zustand *M* gewechselt.

- Erweitern Sie den aus der Vorlesung bekannten MESI-Zustandsautomat um die oben beschriebene Erweiterung zum MOESI-Zustandsautomat (sowohl für eigene Aktionen, als auch für durch andere ausgeführte Aktionen).
- Vervollständigen Sie die unten gegebene Tabelle: Geben Sie jeweils Inhalt der Cache-Zeile und MOESI-Zustand an.
- Wie viele Hauptspeicherzugriffe werden durch diese Speicherzugriffsfolge verursacht? Führt hier die Verwendung des MOESI gegenüber des MESI-Protokolls zu einer Leistungssteigerung und wenn ja, warum?

Prozessor	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4						
3	rd 4						
2	rd 3						
2	wr 5						
1	rd 2						
2	wr 4						
2	rd 1						
3	rd 4						
3	rd 3						
1	wr 1						
3	rd 1						

2.3 Verständnisfragen

- Warum existiert im MOESI-Protokoll kein Zustandsübergang vom Zustand *S* nach Zustand *O*?
- Warum läßt sich MESI nicht in Distributed Shared Memory (DSM) Systemen einsetzen?
- Welche Protokolle stellen in DSM-Systemen die Cache-Kohärenz sicher?

3 Klausuraufgaben

In der Übung werden sofern die Zeit ausreicht, zusätzlich Aufgaben der Klausuren des Wintersemesters 2014/15 und Sommersemesters 2014 besprochen. Die Klausuren finden Sie auf der Homepage zur Vorlesung Rechnerstrukturen.

3.1 Wintersemester 2014/15

In der Übung wird die Aufgabe 5 mit den Teilaufgaben a) und b) besprochen.

3.2 Sommersemester 2014

In der Übung wird die Aufgabe 6 mit den Teilaufgaben d) und e) besprochen.