Script **generated by TTT**

Title: Grundlagen_Betriebssysteme (09.12.2015)

Date: Wed Dec 09 13:16:01 CET 2015

Duration: 45:31 min

Pages: 17

Speicherhierarchie / Caches

Einige fundamentale Eigenschaften von Hardware und Software

schnelle Speichertechnologien kosten mehr Geld pro Byte und haben kleinere Kapazitäten.

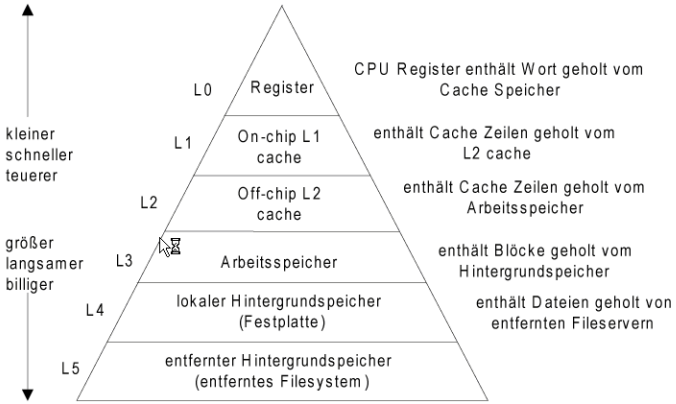
Lücke zwischen CPU und Arbeitsspeichergeschwindigkeit wächst.

gut geschriebene Programme tendieren dazu, gute Lokalität zu haben

zeitliche Lokalität: kürzlich referenzierte Objekte werden in naher Zukunft wieder referenziert.

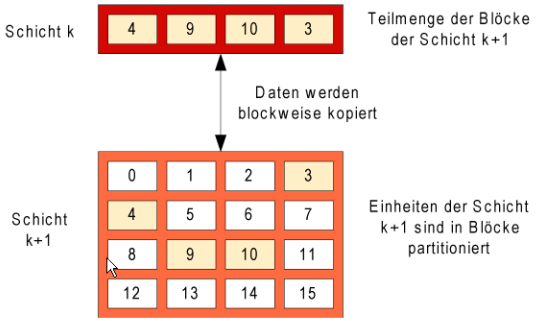
räumliche Lokalität: Objekte mit beieinander liegenden Adressen, tendieren dazu, ungefähr zur gleichen Zeit referenziert zu werden.

Speicherhierarchie - Beispiel



Caching in der Speicherhierarchie

kleinere, schnellere, teuere Einheiten auf Schicht k cachen Daten von größeren, langsameren, preiswerteren Einheiten der Schicht k + 1.



Programm braucht Objekt d in Block b

Cache-Treffer (hit): Programm findet b im Cache der Schicht k, z.B. in Block 10

Cache-Fehler (miss): b ist nicht auf Schicht k; Cache muss b von Schicht k+1 holen (z.B. Block 8).

wenn Schicht k Cache voll, muss ein Block entfernt werden, z.B. nach LRU.

Typen von Cache Problemen

Es können unterschiedliche Arten von Problemen unterschieden werden:

kalter Cache: der Cache ist leer

Kapazitätsfehler: resultieren daraus, dass die Summe der aktiven Cache Blöcke größer ist als der Cache

Konfliktfehler

Caches limitieren meist Platzierungsmöglichkeiten der Blöcke der Schicht k+1 auf eine 1-elementige Teilmenge der Blöcke der Schicht k,

z.B. Block i muss auf Block i mod 4 platziert werden

Konfliktfehler entstehen, wenn der Cache groß genug ist, aber mehrere Datenobjekte auf denselben Block der Schicht k abgebildet werden.

z.B. Zugriff auf Blöcke 0, 8, 0,8, 0, 8, ... jedesmal ein Fehler

Beispiele aus der Cache Hierarchie

Caches werden auf unterschiedlichen Ebenen in modernen Rechensystemen genutzt.

Cache Typ	was wird gecached	wo wird gecached	kontrolliert
Register	4 Byte Wort	CPU Register	Compiler
TLB	Address Translation	On-Chip TLB	Hardware
L1 Cache	32 Byte Block	On-Chip L1	Hardware
L2 Cache	32 Byte Block	Off-Chip L2	Hardware
virtual memory	4 KB page	Arbeitsspeicher	Hardware + BS
buffer cache	Teile von Dateien	Arbeitsspeicher	BS
network buffer cache	Teile von Dateien	lokale Platte	NFS client
Browser cache	Web Seiten	lokale Platte	Web Browser
Web cache	Web Seiten	entfernte Server Platte	Web Proxy Server

Realisierung von Caches

Obwohl je nach der Ebene der Cache Hierarchie unterschiedliche Varianten der Umsetzung existieren, folgen diese doch gemeinsamen allgemeinen Prinzipien.

Allgemeine Cache Organisation

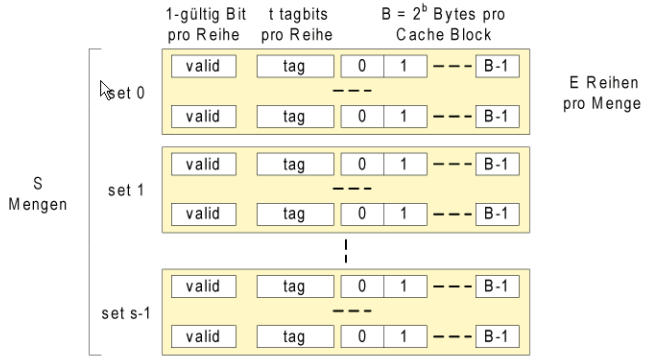
Angenommen, das Rechensystem hat Arbeitsspeicheradressen mit m bits ⇒ $M = 2^m$ eindeutige Adressen.

Cache ist ein Array von Mengen (S = Anzahl von Mengen).

jede Menge enthält eine oder mehrere Reihen (E = Anzahl von Reihen pro Menge).

jede Reihe enthält einen Datenblock (B ist Blockgröße pro Reihe).

Cachegröße = B * E * S Bytes.



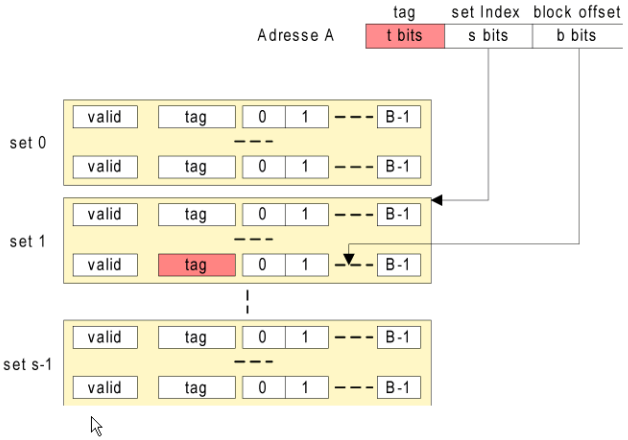
Adressierung von Caches

Adressierung von Caches

Die Cache Organisation bedingt eine Strukturierung der Speicheradresse

das Wort an Adresse A ist im Cache, falls die Tagbits einer der gültigen Reihen der Menge "set Index" den Tag "tag" besitzen

der Wortinhalt beginnt am Offset "block offset" Bytes vom Blockanfang



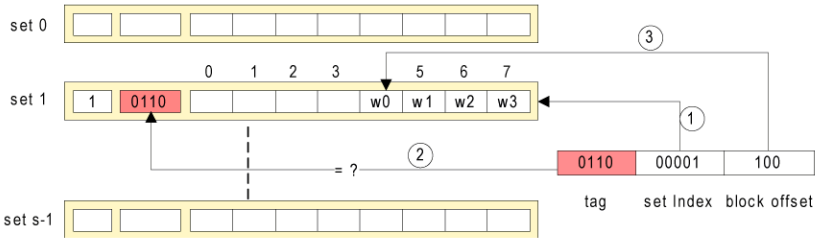
Direct-Mapped Cache

Einfachste Art des Caches mit genau einer Reihe pro Menge; Zugriff erfolgt in der Reihenfolge

Mengenselektion: Benutzung der "set Index" bits zum Bestimmen der relevanten Menge.

Reihenabgleich: Finden der gültigen Reihe in der selektierten Menge mit dem richtigen Tag.

Wortselektion: Benutzung des block offsets zum Bestimmen des relevanten Wortes.



Generated by Targeteam

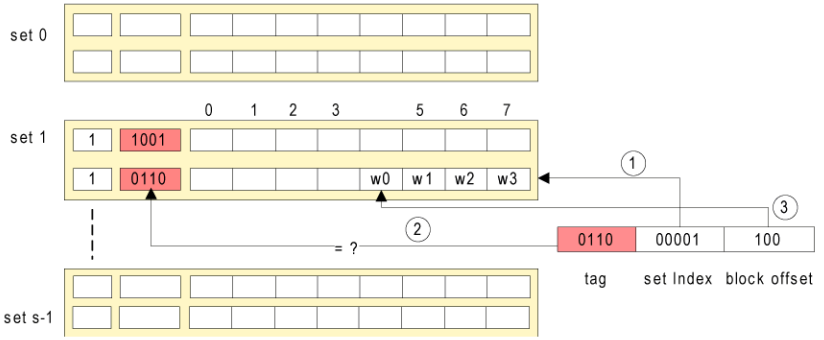
Mengen Assoziativer Cache

Mehr als eine Reihe pro Menge, d.h. E > 1;

Mengenselektion: identisch zu direct-mapped Cache.

Reihenabgleich: Vergleich der Tags für jede gültige Reihe in der selektierten Menge.

Wortselektion: identisch zu direct-mapped Cache.



Generated by Targeteam

Speicherhierarchie / Caches

Einige fundamentale Eigenschaften von Hardware und Software

schnelle Speichertechnologien kosten mehr Geld pro Byte und haben kleinere Kapazitäten.

Lücke zwischen CPU und Arbeitsspeichergeschwindigkeit wächst.

gut geschriebene Programme tendieren dazu, gute Lokalität zu haben

zeitliche Lokalität: kürzlich referenzierte Objekte werden in naher Zukunft wieder referenziert.

räumliche Lokalität: Objekte mit beieinander liegenden Adressen, tendieren dazu, ungefähr zur gleichen Zeit referenziert zu werden.

Generated by Targeteam

Grundlagen: Betriebssysteme und Systemsoftware (GBS)

- Prof. J. Schlichter
  - Lehrstuhl für Angewandte Informatik / Kooperative Systeme, Fakultät für Informatik, TU München
  - Boltzmannstr. 3, 85748 Garching

    Email: schlichter@in.tum.de

    Tel.: 089-289 18654

    URL: http://www11.informatik.tu-muenchen.de/

Generated by Targeteam

Prozesskommunikation

Disjunkte Prozesse, d.h. Prozesse, die völlig isoliert voneinander ablaufen, stellen eher die Ausnahme dar. Häufig finden Wechselwirkungen zwischen den Prozessen statt ⇒ Prozesse interagieren. Die Unterstützung der Prozessinteraktion stellt einen unverzichtbaren Dienst dar.

Fragestellungen

Dieser Abschnitt beschäftigt sich mit den Mechanismen von Rechensystemen zum Austausch von Informationen zwischen Prozessen.

Kommunikationsarten.

nachrichtenbasierte Kommunikation, insbesondere Client-Server-Modell.

Netzwerkprogrammierung auf der Basis von Ports und Sockets.

Einführung

Nachrichtenbasierte Kommunikation

Client-Server-Modell

Netzwerkprogrammierung

Einführung

Prozessinteraktion kann Rechner-lokal und Rechner-übergreifend stattfinden. Prozesse können auf vielfältige Weise Informationen austauschen.

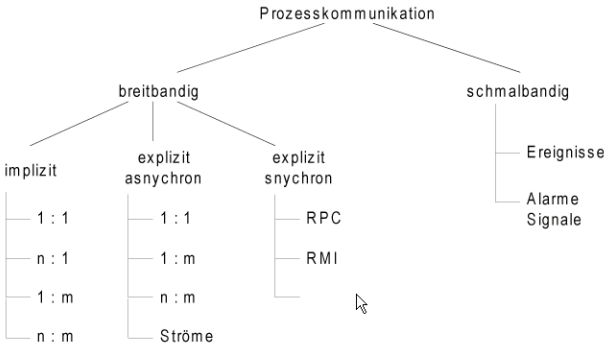
Kommunikationsarten

Verteilte Systeme

Bisher haben wir uns mit systemnahen Konzepten von zentralen Systemen beschäftigt. Seit Ende der 80er Jahre haben jedoch verteilte Systeme rapide an Bedeutung gewonnen.

Ansätze zur Kopplung von Recheneinheiten

Kommunikationsarten



Die Bandbreite des Kommunikationskanals bestimmt die Datenrate, in der Daten zwischen Prozessen ausgetauscht werden können.

Schmalbandige Kanäle

Implizite Kommunikation

Explizite Kommunikation

Schmalbandige Kanäle

Schmalbandige Kanäle werden im Betriebssystem zum Melden von Ereignissen oder für die Synchronisation unterstützt. Übertragung von wenigen Bits an Information, z.B. Setzen von Flags

Dienste des Betriebssystems

Melden von Ereignissen,

Warten auf Ereignisse,

Ereignisverwaltung.

Beim Ablauf von Prozessen können Alarme entstehen (z.B. arithmetische Alarme).

Die Alarme werden über Namen identifiziert, die im BS vereinbart sind. Das BS stellt Dienste zur Zustellung von Alarmen zur Verfügung.

Implizite Kommunikation ist eine breitbandige Kommunikationsform. Die Kommunikation erfolgt über einen gemeinsamen Speicher (Dateien, Register, Datenstrukturen).

Die Kommunikation findet ohne direkte Unterstützung und ohne Kenntnis des BS statt.

Vorteil: einfach und schnell (kein Kopieren zwischen Adressräumen).

Nachteil:

a) gemeinsame Bereiche sind nicht immer vorhanden: z.B. in physisch verteilten, vernetzten Systemen gibt es i.d.R. keinen gemeinsamen Speicher.

b) gegebenenfalls aufwendiges busy waiting ⇒ Mischform: Ereigniszustellung, d.h. schmalbandige Kommunikation, die das Vorhandensein von Daten signalisiert.

Implizite Kommunikationsformen

Implizite Kommunikation

Implizite Kommunikation ist eine breitbandige Kommunikationsform. Die Kommunikation erfolgt über einen gemeinsamen Speicher (Dateien, Register, Datenstrukturen).

Die Kommunikation findet ohne direkte Unterstützung und ohne Kenntnis des BS statt.

Vorteil: einfach und schnell (kein Kopieren zwischen Adressräumen).

Nachteil:

a) gemeinsame Bereiche sind nicht immer vorhanden: z.B. in physisch verteilten, vernetzten Systemen gibt es i.d.R. keinen gemeinsamen Speicher.

b) gegebenenfalls aufwendiges busy waiting ⇒ Mischform: Ereigniszustellung, d.h. schmalbandige Kommunikation, die das Vorhandensein von Daten signalisiert.

Implizite Kommunikationsformen

Generated by Targeteam