Aufgabe 6: Parallelrechner

a) Erklären Sie die Begriffe SIMD, MIMD, UMA, NUMA und NORMA und überlegen Sie, welche Probleme / Vorteile sich aus der jeweiligen Architektur ergeben. Überlegen Sie auch, welchen Restriktionen dem Programmierer durch die Architektur auferlegt werden.

Aufgabe 7: MESI-Protokoll

Gegeben sei ein speichergekoppeltes Multiprozessorsystem mit zwei Prozessoren, die über einen Bus mit einem globalen Speicher verbunden sind. Zur Wahrung der Cache Kohärenz wird das MESI Protokoll verwendet. Die Caches der beiden Prozessoren haben je eine Größe von drei Cache-Lines die genau ein Speicherwort aufnehmen können. Die Caches werden von der niedrigsten Cache-Line aufwärts gefüllt, falls noch freie Lines zur Verfügung stehen. Im Falle eines voll besetzten Caches werden sie nach der Strategie LRU (least recently used) überschrieben. Ergänzen Sie die folgende Tabelle, wobei die Buchstaben die Zustände: M = exclusive modified, E = exclusive unmodified, S = shared unmodified, I = invalid repräsentieren. Die Zahlen hinter den Aktionen und Zuständen bezeichnen Speicheradressen.

CPU	Aktion	Cache1 Line1	Cache1 Line2	Cache1 Line3	Cache2 Line1	Cache2 Line2	Cache2 Line3
		E/8	E/12	I/-	E/6	I/-	I/-
2	read 10						
1	write 8						
1	read 10						
2	read 8						
1	write 8						
1	write 8						
1	read 18						
2	write 10						
2	write 18						

Lösungsvorschlag Aufgabe 6: Parallelrechner

a) Aufgrund der oft unterschiedlichen Struktur von Parallelrechnern ist ein Vergleich sehr schwierig. Die Klassifikation nach Flynn (1966) ermöglicht eine etwas grobe Einteilung. Hierbei wird das Verhältnis Instruktionen zu Daten als Grundlage genommen. Sowohl für Instruktionen als auch für Daten werden je 2 Klassen eingeführt:

SI Single Instruction
MI Multiple Instruction

SD Single Data
MD Multiple Data

Zwei der daraus zu bildenden vier Klassen werden näher betrachtet:

- SIMD = Singe Instruction / Multiple Data: Die gleiche Operation wird auf mehrere Rechenwerken ausgeführt. Geeignet für Verfahren, die gleiche Operationen auf großen Datenmengen durchführen (z.B. auf Vektoren oder Matrizen). Entscheidend für Ergebnisaustausch ist die Konnektivität zwischen den Recheneinheiten.

- MIMD = Multiple Instruction / Multiple Data: Verschiedene Operationen werden auf verschiedenen Rechenwerken durchgeführt. Geeignet für alle parallelisierbaren Aufgaben. Entscheidend für die Effizienz sind auch hier die Verbindungen zwischen den Rechenwerken.

Nun folgen einige MIMD-Spezialfälle:

- UMA: Uniform Memory Access. Alle Rechenwerke haben Zugriff auf einen gemeinsamen Speicher. Der Speicher ist uniform, d.h. jedes Rechenwerk hat die gleiche Zugriffsgeschwindigkeit auf jeden Hauptspeicherteil. Datenaustausch erfolgt über den Hauptspeicher.

  Problem: großer Cache ist wichtig, sonst wird der Hauptspeicher zum Flaschenhals.
  Vorteil: Die Zuordnung von Prozessen auf Rechenwerke zur Laufzeit zu ändern ("dynamische Lastverteilung") ist unproblematisch. Nach kurzer Zeit sind die wichtigsten zum Prozeß gehörenden Daten in den entsprechenden Cache gewandert. Der Zugriff auf die zugehörigen Hauptspeicherteile wird nicht langsamer (uniformer Zugriff). Daten von einem Prozeß zu einem anderen zu transferieren ist unproblematisch, da diese automatisch in dessen Cache wandern und auch danach der Hauptspeicherzugriff genau so schnell wie beim Ursprungsrechenwerk ablaufen.
  Bsp: DEC 2100 4/200, SUN Enterprise 10000

- NUMA: Non Uniform Memory Access. Alle Rechenwerke haben Zugriff auf einen gemeinsamen Speicher - Speicherbereiche sind jedoch einzelnen Prozessoren zugeordnet. Der jeweilige Prozessor hat eine sehr hohe Zugriffsgeschwindigkeit auf seinen eigenen Speicher, kann jedoch auf die Speicher anderer Prozessoren nur sehr langsam zugreifen. Der Datenaustausch erfolgt über den Hauptspeicher.

  Vorteil: Wenn jeder Prozessor nur lokalen Speicher verwendet, kann eine hohe Zugriffsgeschwindigkeit erreicht werden

Problem: Bei Ergebnisaustausch und insbesondere bei dynamischer Lastverteilung werden die Zugriffe sehr teuer. Hier kann nur ein großer Cache helfen.

Bsp: Convex SPP, Cray T3E, KSR 1, HP 9000 V

- Weitere Klassifizierungen:
  ccNUMA: cache coherent NUMA, SGI Origin
  nnNUMA: nearest neighbor NUMA, MEMSY

- NORMA: No Remote Memory Access: Die Rechenwerke haben keinen gemeinsamen Speicher. Der Datenaustausch erfolgt über ein Verbindungsnetzwerk.

Vorteil: billigste Lösung. Beispiel: Workstationcluster. Wenn wenig Daten ausgetauscht werden müssen ist das die effizienteste Lösung

Problem: Datenaustausch ist ineffizient, besonders die Aufsetzzeiten sind groß. Die Programmierung ist schwieriger. Dynamische Lastverteilung ist fast unmöglich.

Bsp: CM5, IBM SP2, nCube 25, PVM Workstation Cluster

Lösungsvorschlag Aufgabe 7: MESI-Protokoll

In der Tabelle werden die sich jeweils ändernden Zustände aufgelistet.

CPU	Aktion	Cache1 Line1	Cache1 Line2	Cache1 Line3	Cache2 Line1	Cache2 Line2	Cache2 Line3
		E 8	E 12	I -	E 6	I -	I -
2	read 10					E 10	
1	write 8	M 8					
1	read 10			S 10		S 10	
2	read 8	S 8					S 8
1	write 8	M 8					I -
1	write 8	M 8					
1	read 18		E 18				
2	write 10			I -		M 10	
2	write 18		I -				M 18