Architektury multiprocesorových počítačů. Architektury symetrických multiprocesorových počítačů (SMP). Způsoby zajištění koherence v SMP. Pravidla pro provádění paměťových operací, zajištění sekvenční konzistence, slabší modely paměťové konzistence. Architektury s distribuovanou sdílenou pamětí, zajištění koherence a konzistence. Vícevláknové procesory. (A4M36PAP)

Architektury multiprocesorových počítačů

Multiprocesorové systémy, systémy s více procesory. Ale mohou to být i procesory „jen" s více jádry.

Paměťová koherence - pravidla pro přístupy k jednotlivým paměťovým místům

Paměťová konzistence - pravidla pro všechny paměťové operace

Hardwarový pohled:

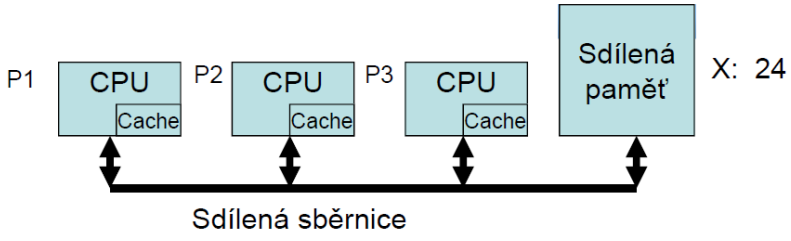
- Systémy se sdílenou fyzickou pamětí (společný fyzický adresový prostor) – **SMP**
- Systémy s oddělenou fyzickou pamětí (každý uzel má nezávislý fyzický adresový prostor) – **UMA**, **NUMA**, ale i Clustery, NOW, metapočítače.

Softwarový pohled:

- Procesy se sdílenou pamětí (shared memory) - na počítači běží jedna instance OS (společné plánování), Standard Open MP.
    - Výhody: snazší programování, nižší latence, HW podpora konzistence skryté paměti.
- Procesy s oddělenou pamětí: komunikují pomocí předávání zpráv – message passing. Na každém uzlu (procesoru) jiná instance OS. Síťové protokoly, RPC, Standard MPI.
    - Výhody: méně potřebného HW, snazší škálování.

Systémy se sdílenou fyzickou pamětí (SMP)

Přirozené rozšíření jednoprocesorových počítačů přidáním dalších procesorů (či jader).



Problém spočívá v paměťové nekoherenci - cache jednotlivých procesorů může obsahovat jiná data.

Definice: Řekneme že multiprocesorový paměťový systém je koherentní jestliže výsledek jakéhokoli provádění programu je takový, že pro každé paměťové místo je možné sestavit myšlené sériové pořadí čtení a zápisů k tomuto paměťovému místu a platí:

- 1. Paměťové operace k danému paměťovému místu pro každý proces se provedou v pořadí, ve kterém byly tímto procesem spuštěny .
- 2. Hodnoty vrácené každou operací čtení jsou hodnotami naposledy provedené operace zápis do daného paměťového místa s ohledem na sériové pořadí.
- Metody pro zajištění koherence nazýváme koherenční protokoly.

ZAJIŠTĚNÍ KOHERENCE – KOHERENČNÍ PROTOKOLY

Přímý zápis – **Write-through**:

- současný zápis slova do paměťového bloku skryté paměti i na odpovídající místo v hlavní paměti.
- jednoduchý, ale pomalý způsob udržování shodného obsahu cache a paměti.
- zatěžuje komunikaci s pamětí, vyžaduje zapisovací vyrovnávací paměť – Write Buffer.
- protokol Write-through je pro rozsáhlejší systémy prakticky nepoužitelný, není škálovatelný.
- reálně používané protokoly jsou MESI, MSI a MOESI.

Write-back je jiný protokol k zajištění téhož. Do hlavní paměti se zapisuje až tehdy, kdy by se o blok ze SP mohlo přijít. Je z principu rychlejší

SEKVENČNÍ KONZISTENCE

- Každý procesor P(i) spouští paměťové operace v programovém pořadí.
- Procesor P(i), který spustí operaci Write, nespustí další paměťovou operaci dříve, než se tato dokončí.
- Procesor P(i), který spustí operaci Read, nespustí další paměťovou operaci dříve, než se tato dokončí a než se dokončí operace Write, jejíž hodnotu vrací operace Read.

Snoopy protokol:

- Se zneplatněním, Write-back skryté paměti.
- Každý paměťový blok je v jednom ze stavů:
    - Je ve cache (alespoň jedné), je aktualizován v paměti (Shared)
    - Nebo je Dirty právě v jedné cache (Exclusive)
    - Nebo v žádné cache není.
- Každý blok SP je právě v jednom stavu:
    - Shared : blok se může číst
    - nebo Exclusive : cache je v jediné kopii, je zapisovatelná a umazaná (dirty)
    - Nebo Invalid : neobsahuje data
- Výpadek čtení (Read miss) způsobí, že všechny cache začnou slídit (snoop) na sběrnici.
- Zápisy související s čištěním linky se zpracují jako výpadky.

MESI Protokol

Každý řádek SP může být v jednom z těchto 4 stavů (kóduje se 2 bity)

- M – Modified. Obsah řádku ve cache se liší od obsahu paměti, (jinak odpovídá běžnému Dirty)
- E – Exclusive. Je jedině v této cache a je stejný, jako v paměti
- S – Shared. Je stejný, jako v paměti, ale nemusí být jen v této SP
- I – Invalid. Obsah řádku neplatí

ARCHITEKTURY S DISTRIBUOVANOU SDÍLENOU PAMĚTÍ

Distribuovaná sdílená paměť (DSM) je abstrakce používaná pro sdílení dat mezi procesy v počítačích, kde není sdílená fyzická paměť. Uživatelé nemusí explicitně řešit problém přenosu dat mezi uzly. Základním problémem je dobrá propustnost komunikačního subsystému, která by neměla záviset na rozlehlosti systému a počtu procesorů.

DSM je prostředek pro řešení paralelních aplikací, distribuovaných aplikací nebo skupinových aplikacích, ve kterých jsou individuální sdílené datové položky přístupné přímo.

Data jsou vyměňována komunikačním systémem. V každém uzlu je z důvodu rychlého přístupu lokální kopie dat, systém musí zajistit konzistentnost dat v jednotlivých uzlech.

Řešení problému souvisí s realizací replik i cacheováním sdílených souborů.

Z technického hlediska můžeme problém řešit v prostředí těsně vázaných multiprocesorů se společnou (sdílenou) pamětí, nebo volně vázaných multiprocesorů komunikujících posíláním zpráv.

Těsně vázané multiprocesory se sdílenou pamětí – propojení pomocí sběrnice dovoluje propojit max. 10 až 20 procesorů. Realizace – architektura NUMA (Non-Uniform Memory Access) s max. 64 procesory – hierarchická architektura, kde jsou procesorové desky se 4 procesory propojeny rychlou sběrnicí nebo přepínačem. Procesory vidí jeden adresní prostor obsahující všechny paměti na všech deskách. Doba přístupu k lokální paměti je však menší než doba přístupu ke vzdálené paměti – nejedná se o transparentní pohled na paměť.

Volně vázané multiprocesory s distribuovanou (sdílenou) pamětí nemají společný paměťový prostor, ale jsou propojeny velmi rychlou sítí s přepínači. Počet procesorů může být vyšší než 64.

Non-uniform Memory Access (NUMA):

- Procesor může přímo pracovat s lokálními i vzdálenými paměťovými místy
- Bez podpory programového vybavení
- Pracovní stanice na síti
- Mohou pracovat pouze s lokální pamětí
- Cíl distribuované sdílené paměti
- Přidat software aby umožnil pracovat s multiprocesorovým kódem
- Zjednodušit programování

ZAJIŠTĚNÍ KOHERENCE A KONZISTENCE

Sekvenční konzistentnost:

- Pouze jedna kopie každé stránky
  - Konzistentnost je zaručena (triviální)
- Replikované stránky
  - Read/only – v pořádku
  - Read/write
    - Operace čtení – instalace lokální kopie, nastavena na R/O
    - Operace zápisu – oprava nebo zneplatnění ostatních kopií
- Typický protokol
  - R(readable), W(writable and readable) stránky
  - Každá stránka má vlastníka: proces, který zapisoval do stránky naposledy

Sdílené proměnné v distribuované sdílené paměti

- Není nutné sdílet celý adresní prostor
- Sdílení jednotlivých proměnných
- Větší vůle v algoritmech pro opravování replikovaných proměnných
- Příležitost eliminovat falešné sdílení
- **Munin**
  - Používá MMU: umístění každého sdíleného objektu ve zvláštní stránce
  - Explicitní deklarace sdílených proměnných
    - Klíčové slovo shared
    - Překladač ukládá proměnné do zvláštních stránek
  - Synchronizace:
    - Uzamykání proměnných
    - Bariéry
    - Podmíněné proměnné
  - Uvolňovací konzistentnost

VÍCEVLÁKNOVÉ PROCESORY

Snižují latenci podporou několika souběžně běžících nezávislých vláken.

Dlouhotrvající operace:

- přístup do paměti
- vzdálené čtení
- synchronizační operace

Pokud se vyskytne dlouhotrvající operace v jednom vlákně, spustí se další vlákno

Modely:

- Coarse-grained (block interleaving)
- Fine-grained (cycle-by-cycle interleaving)
- Multiple-issue (simultaneous)