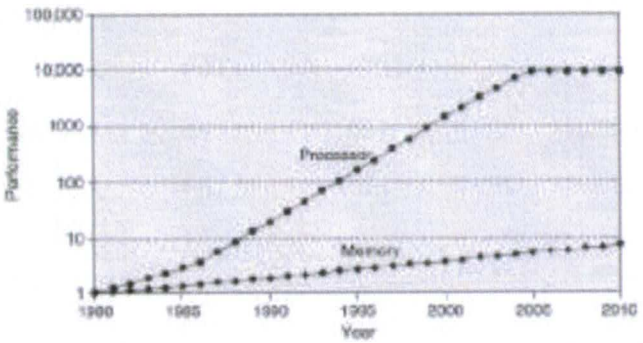
4. Architektury se sdílenou pamětí UMA a NUMA, zajištění lokality dat.



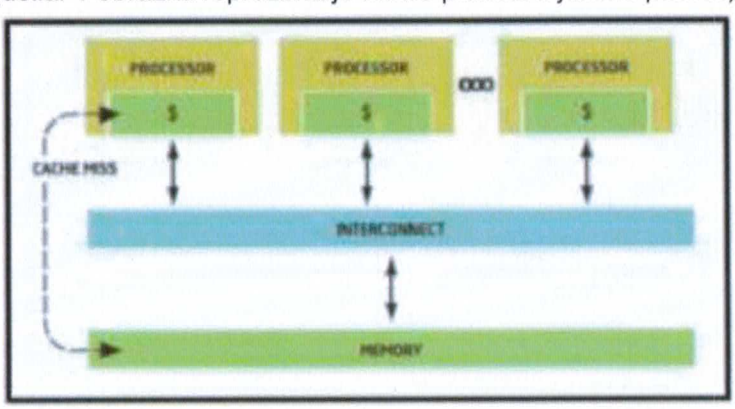
V průběhu vývoje se **rychlost procesorů zvyšovala**, ale **rychlost přístupu do pamětí nerostla** dostatečně rychle. Pokud tedy pracujeme s více procesory paralelně, práce s pamětí bude hlavním důvodem zpoždění.

Existují **architektury s distribuovanou pamětí**. Jedná se o clustery na serverech, kde jsou paměti spojeny přes síťové propojení (paměť je distribuovaná mezi více fyzických hostů v síti), typicky se zde neřeší koherence pamětí cache.

V rámci jednoho stroje však vznikly dvě základní **architektury se sdílenou pamětí**: UMA a NUMA.

UMA (Uniform Memory Access)

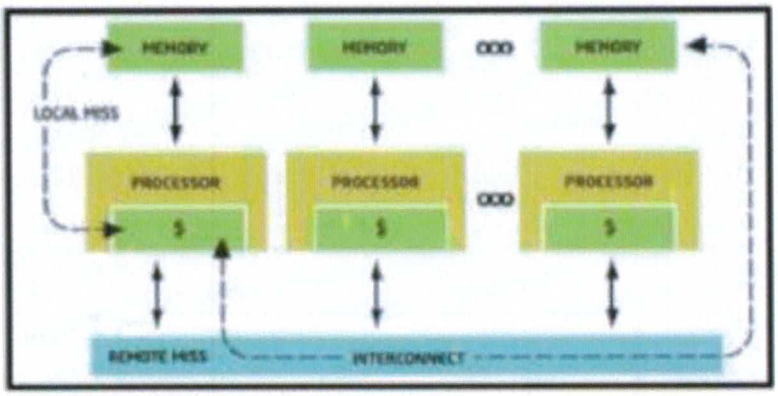
(dollar v obrázku reprezentuje cache paměti... jakože peníze)



- Existuje jedna hlavní paměť, sdílená mezi všemi procesory, které k ní přistupují přes sdílenou datovou sběrnici.
- Architektura tedy poskytuje uniformní dobu přístupu do hlavní paměti do všech částí paměti (paměť není distribuovaná).

- Sběrnice je zde úzké hrdlo, neboť při souběžném pokusu o čtení více procesory nebo zápisu více procesorů dojde k přetížení sběrnice a procesory mohou operace provádět pouze pomalu, postupně.
- Implementováno pomocí výkonné sběrnice, zajišťující případně i malé množství souběžných přenosů.
- Jakmile přeroste počet procesorů určité množství, přístup UMA přestává škálovat.
- UMA architektura je jednoduchá na implementaci a levná na realizaci (je potřeba jediná sdílená sběrnice a jediný řadič paměti). UMA je vhodná například pro zařízení pro koncové uživatele (desktop, notebook, tablet, mobil aj.), ale nevhodná pro velké serverové clustery s mnoha procesory.

NUMA (NonUniform Memory Access)



- Cache paměti u každého procesoru nejsou dostatečně velké, aby ukládaly takové množství dat, s jakým pracují dnešní procesory lokálně. Jako řešení problémů škálovatelnosti byla zavedena architektura NUMA.
- Neexistuje jedna centrální paměť. Místo toho, hlavní paměť je fyzicky distribuovaná mezi více menších pamětí.
- Každý procesor má přidělenou blízko sobě vlastní část paměti (lokální paměť), ke které přistupuje výrazně rychleji než k ostatním.
- K ostatním pamětím přiděleným dalším procesorům (vzdálené paměti) musí procesor přistupovat draho přes sběrnici.
- Vytvořením topologií nad dvojicemi (procesor, paměť) jsou určeny vlastnosti konkrétního NUMA systému (hustě propojená síť procesorů a pamětí má lepší rychlost přenosu než lineární propojení, ale je zase výrazně dražší; naopak třeba topologie hvězda nebo strom strojí někde mezi: ne příliš drahé, ale ani příliš rychlé ve všech případech).
- Architektura tedy poskytuje neuniformní dobu přístupu do hlavní paměti (např. lokální a vzdálený výpadek v cache).
- Typickou aplikací architektury NUMA jsou servery, kde jednotlivé uzly v clusteru vykonávají specifické operace nad vlastními daty. Zde se výhody NUMA architektury projevují nejvíce.

- NUMA architekrura je komplexní architektura drahá na implementaci.

Lokalita dat

Problémem v NUMA systémech je udržení dat blízko k procesorům, které s nimi pracují (v lokálních pamětích ideálně). Návrh architektury a topologie tedy potřebuje zajistit, aby nejvíce využívaná data byla vždy co nejblíže procesorům, které je využívají.

Například data po vytvoření budou zapsána do lokální paměti, ale když dojde k přepnutí kontextu procesorů a k pozastavenému procesu se dostane jiný procesor, najednou jsou pro něj tato data na vzdálené paměti a dochází ke zpomalení. Když scheduler NUMA architektury dojde k situaci, že více dat než nějaká mezní hranice se nachází na vzdálené paměti, je možné přesunout data do lokální paměti pro zajištění lepší lokality dat.

First Touch

Do které lokální paměti se umístí stránka se rozhoduje pomocí pravidla First Touch – stránka se načte do lokální paměti procesoru, který na daná data přistoupí jako první. Problémem tohoto přístupu je, že přístup k datům v rámci programu pak musí odpovídat tomuto vzoru, aby program byl efektivní.

Bindování procesů na NUMA domény

Moderní NUMA architektury rozdělují CPU do tzv. NUMA domény, tedy tříd ekvivalence určujících, do které části RAM mají daná CPU jak daleko. Pokud vlákno běží na procesory v NUMA doméně 1, je důležité, aby OS scheduler nenaplánoval vlákno po přepnutí kontextu do jiné NUMA domény, která má do této části RAM daleko.

Pomocí explicitního bindovaní procesů do určité NUMA domény můžeme definovat, do kterých domén se mohou vlákna naplánovat, například v OpenMP pomocí OMP_PLACES (zamkne vlákno pro pouze dané domény, OMP_PROC_BIND umožní nastavit distribuci vláken na daném OMP místě).

NUMA FIRST TOUCH - První přístup do stránky rozhodne o umístění do paměti, která je nejblíže jádru které způsobilo výpadek