Počítačové systémy

Obsah

1 Paměť počítače

1.1 Základní Metriky Paměti a Přenosu Dat

V počítačových systémech je klíčové porozumět tomu, jak se data ukládají a přenášejí. K tomu slouží standardizované jednotky a metriky.

1.1.1 Jednotky Kapacity Paměti

Základní jednotkou informace v digitálním světě je bit.

- **Bit (b)**: Nezákladnější jednotka informace, která může nabývat pouze dvou stavů: 0 nebo 1 (logická nepravda / logická pravda, vypnuto / zapnuto).
- **Byte (B)**: Skupina bitů, historicky nejčastěji 8 bitů, tvořící nejmenší adresovatelnou jednotku v paměti. Jeden byte se používá k reprezentaci jednoho znaku (např. písmene, číslice) v kódování ASCII. 1 Byte=8 bitu

Větší jednotky se odvozují pomocí binárních nebo dekadických násobků.
Dekadické (SI) Předpony (Desítková Soustava)
Tyto předpony se používají v telekomunikacích, datových přenosech a často (i když ne vždy přesně) pro marketingové kapacity pevných disků (HDD/SSD).

- Kilobyte - KB - 10^3 - 1000
- Megabyte - MB - 10^6 - 1 000 00
- Gigabyte - GB - 10^9 - 1 000 000 000
- Terabyte - TB - 10^12 - 1 000 000 000 000

Binární Předpony (Dvojková Soustava)
Tyto předpony jsou používány pro přesné označení kapacity paměti v operačních systémech a při popisu operační paměti (RAM), kde je výhodné pracovat s mocninami čísla 2. Byly standardizovány komisí IEC.

- Kibibyte - KiB - 2^10 - 1024
- Mebibyte - MiB - 2^20 - 1 048 576
- Gibibyte - GiB - 2^30 - 1 073 741 824
- Tebibyte - TiB - 2^40 - 1 099 511 627 776

1.1.2 Metriky Přenosu Dat

Přenosová rychlost (datový tok) udává, jak rychle se data přenášejí z jednoho místa na druhé. Základní jednotkou je **bit za sekundu** (bit per second). Zde se téměř výhradně používají dekadické předpony, i když je řeč o **bitech**, ne **bytech**.
Bit za sekundu (b/s nebo bps): Udává počet přenesených bitů za jednu sekundu.

- **Kilobit za sekundu (kb/s)**: 103 bitů/s.
- **Megabit za sekundu (Mb/s)**: 106 bitů/s.
- **Gigabit za sekundu (Gb/s)**: 109 bitů/s.

1.1.3 Latence (Odezva)

Zatímco datový tok udává objem dat, latence udává čas.

- **Latence**: Doba zpoždění, která uplyne mezi odesláním požadavku a obdržením první odpovědi (nebo mezi začátkem a koncem přenosu).
- **Jednotky**: Nejčastěji se udává v milisekundách (ms).
- **Význam**: Vysoká latence je typická pro satelitní komunikaci nebo vzdálené servery, zatímco nízká latence je kritická pro online hraní, videokonference a rychlé databázové transakce.

1.1.4 Propustnost (Throughput) a Šířka Pásma (Bandwidth)

Tyto pojmy jsou často zaměňovány:

- **Šířka Pásma (Bandwidth)**: Teoretická maximální přenosová rychlost, kterou daný kanál nebo rozhraní dokáže podpořit (např. 1 Gb/s Ethernetová linka má šířku pásma 1 Gb/s). Udává se v b/s.
- **Propustnost (Throughput)**: Skutečná, efektivní rychlost přenosu dat dosažená za reálných podmínek (včetně režie, chyb, zpoždění). Propustnost je vždy menší nebo rovna šířce pásma. Udává se v b/s nebo B/s.

1.2 Paměťový model

Každý běžící proces v operačním systému (OS) pracuje s vlastním, virtuálním adresovým prostorem. Tento prostor dává procesu iluzi, že má k dispozici souvislý a exkluzivní blok paměti, ačkoli ve skutečnosti je paměť fyzicky rozdělená a sdílená mnoha procesy. Virtuální adresový prostor procesu je obvykle organizován do několika logických segmentů, z nichž každý slouží specifickému účelu.

Paměťový model procesu popisuje, jak je paměť organizována a jak s ní procesy pracují. Obecně se skládá z několika hlavních oblastí, které určují, jak se přistupuje k datům a instrukcím během běhu programu. Proces běžící na moderních systémech má paměť rozdělenou oblasti:

- **Text** - část paměti kam se ukládají instrukce a konstanty programu, Tato oblast paměti je nastavená jako read-only, aby nedošlo k narušení kódu běžícího programu.
- **Data** - část paměti kde jsou uložené globální a statické inicializované proměnné
- **Bss** - část paměti kde jsou uloženy globální a statické neinicializované proměnné (jsou nastaveny na počáteční hodnotu 0)
- **Heap** - část paměti, která slouží k dynamickému přidělování za běhu programu
- **Stack** - část paměti, která slouží k ukládání kontextu volané funkce (lokální proměnné, návratové hodnoty, ukazatel na zásobník)

1.3 Hierarchie Paměti a Cache Paměti

Hierarchie paměti je uspořádání paměťových komponent do pyramidové struktury, která optimalizuje systém tak, aby procesor pracoval co nejčastěji s co nejrychlejší pamětí.

1.3.1 Princip Lokality

Úspěch hierarchie je založen na principu lokality:

- **Časová Lokalita**: Pokud byla data právě použita, je pravděpodobné, že budou použita znovu brzy.
- **Prostorová Lokalita**: Pokud byla data na adrese **A** použita, je pravděpodobné, že budou brzy použita sousední data (A+1, A+2, atd.).

1.3.2 Úrovně Hierarchie

- **L0 - Registry** - přístup ¡1 cyklus CPU - kapacity většinou bitová šířka procesoru
- **L1, L2, L3 - Cache** - přístup 1 až 80 cyklů - kapacita KiB až MiB
- **L4 - Hlavní paměť RAM** - přístup 100+ cyklů - kapacita GiB
- **L5 - Sekundární (trvalé) uložiště** - přístup v řádech milionů cyklů - kapacita TiB

1.3.3 Cache Paměť (L1, L2, L3)

Cache je malá, rychlá SRAM paměť, která slouží jako vyrovnávací paměť mezi CPU a RAM. Paměť cache (mezipaměť) je klíčový prvek v moderních počítačích, který řeší obrovský nepoměr mezi rychlostí CPU a rychlostí operační paměti (RAM). Zatímco se rychlost procesorů dramaticky zvýšila (zdvojnásobení zhruba každých 18 měsíců, známé jako Mooreův zákon), rychlost RAM se tomu nevyrovnala. Cache je malá, ultra-rychlá paměť umístěná uvnitř CPU, která ukládá data nedávno přístupná nebo data, u nichž se očekává, že budou brzy potřeba. Tím se procesoru zkracuje cesta pro data, která by jinak musel složitě a pomaleji čerpat z RAM. Moderní CPU nepoužívají jen jednu úroveň cache, ale hierarchii: L1, L2 a L3

- **L1 Cache**: Nejrychlejší a nejmenší. Cache L1 je obvykle rozdělená na L1i (instrukce) a L1d (data), což odráží zjištění, že oddělené cache fungují lépe, protože oblasti paměti pro kód a data jsou do značné míry nezávislé. Je umístěna přímo uvnitř každého CPU jádra a každé jádro má svou vlastní.
- **L2 Cache**: Větší, pomalejší než L1. Může být dedikovaná nebo sdílená mezi malou skupinou jader. Ukládá často přístupná data, která se nevešla do L1
- **L3 Cache**: Největší a nejsdílenější. Na rozdíl od L1 (a často i L2) je sdílena mezi všemi jádry v rámci jednoho CPU

CPU nepracuje s daty po jednotlivých bajtech, ale v blocích zvaných cache lines (řádky cache). Cache line je základní jednotka přenosu mezi pamětí a cache, přičemž standardní velikost je typicky 64 bajtů (ale závisí na architektuře procesoru).

- **Cache Hit**: Když CPU potřebuje data, nejprve se podívá do L1 cache. Pokud jsou data nalezena (tzv. cache hit), jsou získána téměř okamžitě (během několika nanosekund).
- **Cache Miss**: Pokud data nejsou v L1, CPU se postupně podívá do L2 a L3. Pokud data nejsou nalezena v žádné úrovni cache (cache miss), musí CPU přistoupit až do RAM. Přístup k RAM je 100krát pomalejší než přístup do L1 cache.

1.3.4 Prefetching (Přednačítání)

CPU automaticky předpovídá, jaká data bude program potřebovat, a načítá je do cache dříve, než jsou explicitně vyžádána. **Prefetching** funguje nejlépe, když program přistupuje k datům sekvenčně (např. procházení pole), protože takový přístup je předvídatelný. Při náhodném nebo nepředvídatelném přístupu (např. pointer chasing ve spojovém seznamu) prefetching selhává a program platí plnou cenu latence.

1.3.5 Koherence Cache (Cache Coherency)

V systémech s více procesory (SMP) musí být zajištěno, že všechny procesory vidí stejný obsah paměti. Tato údržba jednotného pohledu na paměť se nazývá **koherence cache**. CPU sledují (snoopují) operace zápisu prováděné ostatními jádry, aby zajistily, že pokud jedno jádro změní data, kopie v cache ostatních jader se označí jako neplatné (Invalid). K řízení tohoto procesu se používají protokoly jako **MESI**.

1.3.6 Falešné sdílení (False Sharing)

Závažný antivzor v multi-vláknovém programování, kdy dvě vlákna modifikují logicky nezávislé proměnné, které se ale nacházejí ve stejné cache line. Hardware považuje celou cache line za jednotku koherence, a tak se jádra neustále přetahují o vlastnictví dané cache line, což vede k neustálé invalidaci a zpomalení výkonu.