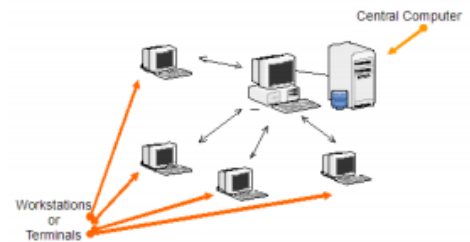


Databázové architektury (11)

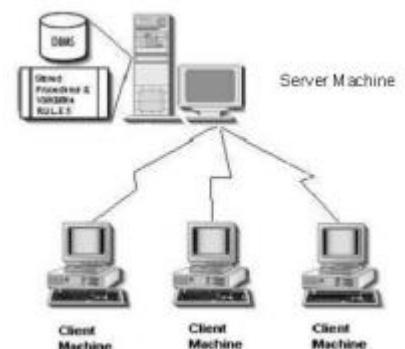
DBS: Centralizované

- běží na jednom počítačovém systému
- neinteragují s dalšími počítačovými systémy
- **general-purpose computer system**
 - o jeden či více CPU a několik ovládacích zařízení
 - o propojené přes common bus (poskytuje přístup do sdílené paměti)
- **single-user system**
 - o např. osobní PC či pracovní stanice
 - o většinou jedno či více CPU a jeden či dva pevné disky
 - o OS může podporovat jen jednoho uživatele
- **multi-user system**
 - o více disků, více pamětí, více CPU, vícero uživatelů v OS
 - o obsluhuje větší množství uživatelů, kteří jsou k systému připojeni přes terminály



DBS: Client-server

- servery uspokojují požadavky generované v m klientstých systémech
- výhody nahrazení sálových počítačů sítěmi stanic nebo osobních PC připojených k backendu:
 - o lepší funkce za danou cenu
 - o flexibilita při hledání zdrojů a rozšiřování
 - o lepší UI
 - o snadnější údržba



Front-end vs. Back-end

- rozhraní mezi FE a BE – SQL či API
- **back-end**
 - o spravuje přístupové struktury, vyhodnocuje a optimalizuje dotazy, řízení konkurence a obnovy
- **front-end**
 - o nástroje typu formuláře, reportovací nástroje, grafická uživatelská rozhraní,...

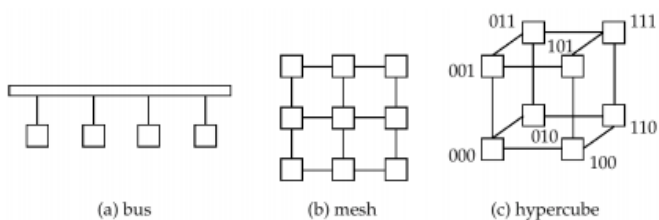
DBS: Paralelní

- skládá se z více procesorů a disků spojených rychlou vnitřní sítí
- **coarse-grain** (hrubozrná) **parallel machine** – skládá se z menšího množství výkonných CPU
- **fine-grain** (jemnozrná) **parallel machine** – tisíce menších CPU

- dva hlavní indikátory výkonu:
 - o **propustnost** – množství úloh, které mohou být splněny v daném časovém intervalu
 - o **odezva (/latence)** – jak dlouho trvá dokončit jednu úlohu od jejího odeslání
- **speed-up** – problém o fixní velikosti vykonávaný na malém systému je dán systému, který je N-krát větší (efektivnější)
- **scale-up** – zvýšení velikosti jak problému, tak systému
- oboje často sublineární kvůli:
 - o **počáteční ceně** (cena startu vícera procesů > čas výpočtu)
 - o **rušení** – procesy přistupují ke sdíleným zdrojům a soutěží mezi sebou (a čekají)
 - o **zkreslení** – větší stupeň paralelizace → větší odchylka mezi vykonanými úlohami

Architektury propojení

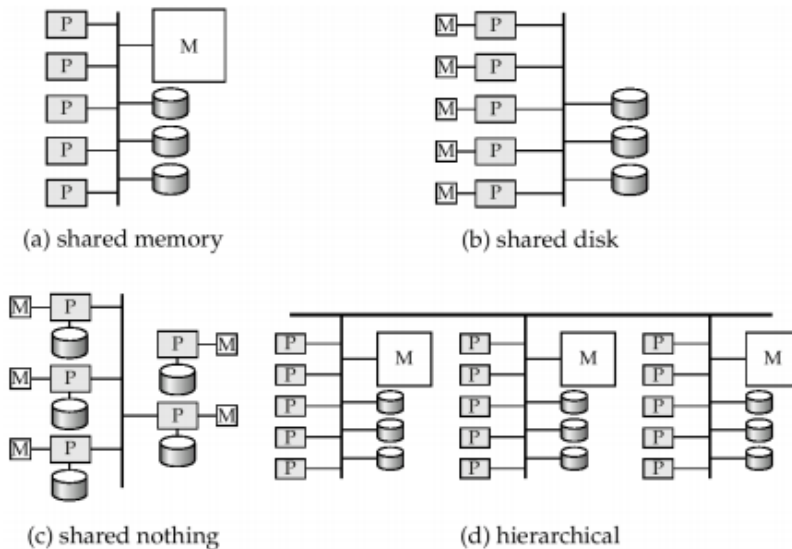
- **bus**
 - o komponenty posílají data na (/získávají z) jednoho komunikačního busu
 - o nevýhoda – neškáluje dobře se zvyšujícím se paralelismem
- **mesh (smíšená topologie)**
 - o komponenty jsou uspořádány jako nody v mřížce
 - o každá komponenta je spojena s vedlejší komponentou
 - o výhoda – lepší škálování (propojení roste s množstvím komponent)
 - o nevýhoda – může vyžadovat hodně skoků při poslání zprávy nějakému nodu
- **hyperkostka**
 - o komponenty jsou očíslovány v binární soustavě
 - o jsou na sebe napojené, pokud se jejich binární reprezentace liší přesně v jednom bitu
 - o n komponent je napojeno na $\log(n)$ dalších komponent, nejdelší spojení je $\log(n)$
 - o redukuje komunikační prodlevy



Paralelní databázové architektury

- **sdílená paměť**
 - o procesory sdílí společnou paměť
 - o efektivní komunikace mezi nimi, ale není moc škálovatelná
- **sdílený disk**
 - o procesory sdílí společný disk
 - o určitý stupeň tolerance chyb (pokud procesor vypadne, jiný vezme jeho úlohu)
 - o data jsou sdílena všemi procesory

- **shared nothing**
 - o procesory nesdílí ani paměť, ani disk
 - o komunikují mezi sebou pomocí propojené sítě
 - o nevýhoda – cena komunikace, přístup k nelokálnímu disku
- **hierarchická**
 - o kombinace předchozích architektur
 - o top-level je shared nothing
 - o každý node systému může být subsystémem sdílené paměti

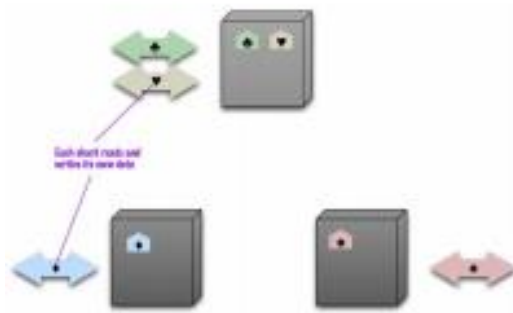


DBS: Distribuované

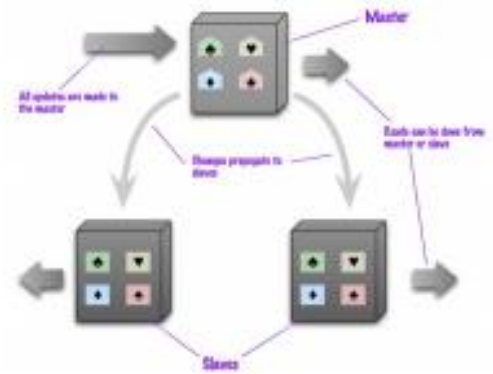
- **scale-out** – data jsou distribuována (šířena) přes několik zařízení (nodů)
- data jsou replikována (systém může pracovat, i když node vypadne)
- **homogenní distribuované DB**
 - o stejný SW/schéma na všech nodech, data se mohou rozdělit mezi nody
 - o cílem je poskytnout pohled na jednu DB, skrýváje detaily distribuce
- **heterogenní distribuované DB**
 - o rozdílný SW/schéma na různých nodech
 - o cílem je integrovat existující DB pro poskytnutí užitečné funkcionality

Modely distribuce

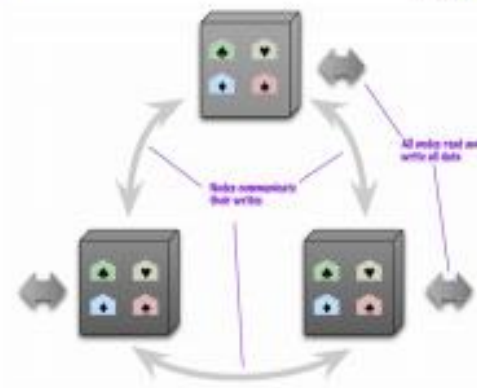
- **single server** – bez distribuce
- **sharding** – rozdělení různých částí dat mezi různé servery (ale může být příliš na jednom)
- **master/slave replikace** – master poskytuje R/W, slave poskytuje čtení (ale bez škálování W)
- **peer-to-peer replikace** – všechny repliky mají ekvivalentní váhu (každý node je master)
- častá kombinace shardingu a replikace



sharding = **distribution**



master/slave **replication**



peer-to-peer **replication**

Typy dotazů (queries)

- **deklarativní**
 - popisujeme požadovaná data, ale ne jak je získáme
 - např. DRC, TRC
- **procedurální**
 - poskytujeme jak získáme chtěná data
 - např. relační algebra (částečně)
- SQL umožňuje obojí
- **QBE** (Query By Example)
 - grafický dotazovací jazyk (ze 70. let)
 - mnoho grafických FE pro DB dnes užívají ideu