

Otázky k lekci č. 12: Vývoj výpočetního modelu

1. Co se rozumí pod pojmem "výpočetní model"? Uveďte příklady

- ucelená představa
 - kde jsou aplikace uchovávány, jako programy a kde skutečně běží
 - zda a jak jsou aplikace rozděleny na části a jak tyto části spolupracují
 - kde a jak se uchovávají a zpracovávají data
 - kde se nachází uživatel, kdy, jak a jakým způsobem komunikuje se svými aplikacemi
- neustále se vyvíjí
- některé modely nepočítají s existencí sítě (modely pro dávkové zpracování)
- jiné počítají (modely pro klient/server aplikace)
- další nutně vyžadují (modely pro distribuované zpracování ...)

2. Charakterizujte dávkové zpracování (batch processing)

- nejstarší výpočetní model
- byl vynucen dobou
 - nedokonalostí tech. základny (málo výkonný HW)
 - malými schopnostmi SW (nebyl multitasking)
 - vysokými náklady
 - potřebou kolektivního využití dostupné výpočetní techniky
- ještě není mrtvý
- princip: zájemce si dopředu připravil výpočet (program, vstupní data, pokyny pro zpracování), vše zabalil do jednoho celku (dávky), ta se přenesla k počítači a zařadila do fronty čekajících dávek a když na ni přišla řada, zpracovala se, pak vznikl výstup (např. tisk), na nějž autor dávky mohl reagovat (opravou chyby, změnou vstupních dat)
- **nevýhody**
 - uživatel nemá bezprostřední kontakt se svou úlohou
 - doba obrátky (zpracování jedné dávky) trvala dlouho
- **výhody**
 - dokáže relativně dobře vytižit dostupné zdroje
 - nutí programátory programovat hlavou a ne rukama
- **později**
 - v prostředí se používalo tzv vzdálené zpracování úloh
 - uživatel na jednom uzlu připravil dávku, poslal ke zpracování na jiný (sám určoval kam dávku pošle!!!)
- **dnes**
 - modernější alternativa RJE (distribuovaná aplikační platforma) -> určuje sama, kam pošle dávku na zpracování
- **do budoucna**
 - model autonomních agentů
 - samostatní agenti(programy) dostanou určité zadání a to v prostředí plní samostatně, autonomně

3. Charakterizujte výpočetní model host/terminál

- vznikl jako reakce na neinteraktivnost dávk. zpracování
 - dokáže uživatelům zajistit přímý kontakt s jejich úlohami a interaktivní způsob práce
 - umí obsloužit více uživatelů současně
- vznik umožněn zdokonalením SW a HW, hlavně existencí uživatelských pracovišť (terminálů)

host

- počítač, který je hostitelem systémových zdrojů(processor, paměť, I/O zařízení), programů, dat ...

podstata modelu host/terminál

- vše je na jedné hromadě (programy běží na hostu, data se zpracovávají na místě, kde se nachází
- mezi hostem a terminály se přenáší jen výstupy na obrazovku uživatele, vstup z klávesnice
- terminály mohou být v libovolné vzdálenosti (osobní, vzdálené ...)
- hostitelský počítač je role, ve které nějaký konkrétní počítač vystupuje
- střediskový počítač, mainframe atd... jsou kategorie počítačů
- mainframe může fungovat : dávkově nebo jako hostitelský počítač
- jako hostitelský počítač může fungovat například PC s unixem (rozhodující je charakter OS!!!)
- **výhody**: centralizovaný charakter, relativně snadná implementace, nízké nároky na přenos dat
- **nevýhody**: uživatel má iluzi, že má host. počítač výhradně ke své dispozici, už. komfort relativně nízký
 - není to vina výpočetního modelu, ale jeho využití (dnes již existují GUI based terminály)

4. Jaké byly důvody pro vznik prvních sítí LAN a WAN?

LAN

- řeší především potřebu sdílení (souborů, periférií ...)
- uživatel nesmí sdílení poznat (pozorovat výrazný rozdíl v rychlosti přístupu ke svým a sdíleným objektům), neměl by si sdílení uvědomovat -> mechanismy sdílení musí být implementovány transparentně
- nutné dostatečně rychlé přenos. technologie (např. 10 Mbps Ethernet)
- vše realizováno jako lokální síť (LAN - local area network)
- LAN řešeny tak, aby "je nebylo vidět" (aby na nich mohli pracovat aplikace neuzpůsobené síti)
- teprve později se mohou stát viditelné (pro aplikace, které se sítí přímo počítají)

WAN

- potřeba překlenout vzdálenost pro potřeby komunikace, sdílení výpočetní kapacity, dat, vzdál. přístupu
- z toho důvodu začaly vznikat první rozlehlé sítě
- kvůli přenos. možnostem (rychlostí nelze dosáhnout potřebné transparentnosti, proto řešeno netransparentně (uživatel si uvědomuje rozdíl mezi místním a vzdáleným))

5. Charakterizujte model file server / pracovní stanice

- nový výpočetní model pro LAN
- snaží se vyjít vstříc potřebám sdílení v sítích LAN (aplikace a data umístěny centrálně na fileserveru, jako soubory)
- aplikace a data se zpracovávají lokálně na pracovních stanicích
- pro aplikace neviditelný (plná transparentnost) -> použitelný i pro aplikace neuvědomující si existenci sítě
- umožňuje sdílení dat i programů
- umožňuje centrální správu
- někdy neefektivní (zbytečný přenos, zahlcení sítě)
 - protože data zpracovávána jinde, než umístěna
 - podobně pro programy

6. Charakterizujte model klient/server,

- myšlenka data zpracovávat, kde se nachází, výstupy pro uživatele generovat, kde se nachází uživatel
- musí dojít na rozdělení na serverovou a klientskou část
- klient a server si posílají data v podobě dotazů a odpovědí a pokud se dobře dohodnou, mohou minimalizovat objem přenášených dat
- klient i server mohou stát na různých platformách
- server pasivně čeká na dotaz, komunikaci iniciuje klient
- musí být definována komunikace (např. protokol HTTP)
- příklady k/s služeb: WWW (server/browser + HTTP), email (mail server/klient + SMTP POP3/IMAP)...
- **nevýhody**
 - klient není univerzální (pro různé aplikace třeba různých klientských částí)
 - s vývojem aplikace dochází i k vývoji klientské části (nutno instalovat nové a nové verze)
 - důsledek: nárůst nákladů
 - možné řešení - rozdělit aplikace na části (prezenční/aplikační/datovou)
 - přínos - lze použít univerzálního klienta

7. Charakterizujte 3-úrovňový model klient/server

- klasické klient/server dělí aplikaci na 2 části a vzniká dvouvrstvá architektura
- nové řešení - 3 části - **prezentační/aplikační/datová**
- lze implementovat jako 3 úrovňové i dvouúrovňové řešení (celkem 5 možností)
- **výhody**
 - klient může být velmi univerzální (browser)
 - vše specifické je před uživateli schováno
 - WWW server (i DB server) se mohou nacházet kdekoli
- **důsledky**
 - původně samostatné 2úrovňové aplikace přechází do podoby nesamostatných služeb, charakteru nadstavby nad WWW (eventuelně el. poštou)
- **příklady**
 - vyhledávání - původně samostatně, dnes skrze WWW
 - webmail - práce skrz webové rozhraní
 - ...

8. Charakterizujte pojem NC (Network Computer) a koncept Network-Centric Computing

- cesta snižování nákladů na provoz
- klasické **PC je tlusté** - programy nainstalované kvůli potenciální potřebě - potřebuje více HW prostředků
- návrh zavádět programy až v okamžiku potřeby
- důsledek - počítač (terminál, koncové zařízení ...) stačí vybavit minimalisticky tím, co potřebuje ke stažení toho, co potřebuje - tenké PC (thin client)
- představa, že programy si bude thin client stahovat ze sítě (např aplety Java)
- modelu fungování thin clientů se začalo říkat **Network-Centric Computing**
- pro tenkého klienta se vžil název **Network Computer (NC)**
- v praxi se myšlenka NC neujala (nedostatečná kapacita sítě, nepřipravenost aplikací a SW, malý cenový rozdíl mezi PC a NC)
- NC však našly uplatnění (v rámci intranetů, pro specializované aplikace, pro jednoúčelová nasazení)

9. Charakterizujte model "Server-Based Computing"

- cesta snižování nákladů na provoz skrze NC se ukázala neschůdná
- další pokus: návrat k plné centralizaci (k modelu host/terminál)
- snaha umožnit použití i jiných zařízení, než jen PC
- tech. předpoklady: našla se řešení, která umožňují vzdálený terminálový přístup v GUI při únosných nárocích na přenosovou kapacitu (např X Window)
- aplikace běží na aplikačním serveru, své graf. výstupy generuje na aplikačním serveru
- problém v neúnostnosti velikosti generovaných grafických dat vyžadující velkou přenos. kapacitu
- řešení je přemístit část generující graf. data do terminálu (musí se udělat s ohledem na minimalizaci přenášených dat a možnost implementace na platformě terminálu)
- problém s rozdílnými zobraz. schopnostmi různých terminálů

10. Charakterizujte koncept webových služeb a architektury orientované na služby architektura orientovaná na služby

- dochází k úplnému oddělení způsobu provozování a fungování aplikací a efektu, který to přináší
- kdokoli (jakýkoli agent) může nabízet poskytovanou službu
- kdokoli (jakýkoli agent) může poskytovanou službu využívat
- komunikace bezstavová charakteru "požadavek/odpověď"
- musí být vyřešeno
 - jak se agenti dozvědí o poskytovaných službách
 - jak budou agenti vzájemně komunikovat
 - jak budou formulovány požadavky a odpovědi

webové služby

- to, co musí být zajištěno řešeno pomocí technologií WWW a nadstaveb nad WWW
 - **UDDI** (universal description, discovery and integration) - pro zveřejnění popisu služby v rámci adresáře, pro vyhledání služeb
 - **WSDL** (web services description language - pro popis poskytovaných služeb)
 - **SOAP** (simple object access protocol - pro zabalení požadavků a odpovědí do jednoho celku (XML based))
 - **HTTP** (a TCP/IP) pro přenos dat
- jsou určeny pro vzájemnou komunikaci programů, nepředpokládá se, že přímým uživatelem bude člověk
- v praxi se využívají spíše uvnitř firemních subjektů

11. Charakterizujte koncepty ASP a Utility Computing

ASP

- spíše ekonomický model, než výpočetní
- oddělení vlastníka od uživatele
- vlastník si pořizuje SW, stará se o něj, nese náklady na provoz, aktualizuje ...
- uživatel pouze používá funkce (odpadají mu náklady na pořízení)
- přínosy
 - malým uživatelům se nevyplatí koupit velký SW, ani ho následně spravovat
 - zákazník se neupisuje na dlouhou dobu (když přestane vyhovovat, přestane využívat)
 - dostupnost může být smluvně zajištěna

utility computing

- v hostingových centrech dostupné vše (konektivita, prostor, výpočetní kapacita) v takové míře, v jaké zákazník požaduje bez pořizovacích poplatků, jen s platbami za využití služby
- jednotlivé zdroje (výp. kapacita, paměť, konektivita) jsou tzv. virtualizovány (odděleny od své podstaty a nabízeny jako škálovatelná služba)

- uživatel může průběžně konzumovat zdroje v míře, jaká odpovídá jeho momentálním potřebám stylem, jako když spotřebovává vodu, plyn
- utility computing je výp. model, kdy zákazník konzumuje zdroje na principu utility

12. Charakterizujte pojmy hosting, housing, aplikační hosting, naznačte jejich výhody a nevýhody

hosting

- umístění dat a aplikací zákazníků na zařízeních v prostorách poskytovatele včetně OS a standardních aplikací a utilit
- uživatel plní server svými daty (Web hosting -> vystavuje zde své WWW stránky atp..)

housing

- umístění celých zařízení zákazníku v prostorách poskytovatele (lepší konektivita, server stále patří uživateli, se server se stará jeho vlastníh/uživatel)

aplikační hosting

- poskytovatel se stará o server, jež mu patří
- uživatel si na serveru provozuje své aplikace