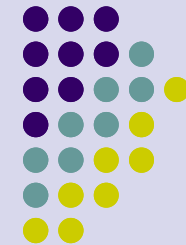


# Distribuované souborové systémy (DFS)

---



Přednášky z Distribuovaných systémů  
Ing. Jiří Ledvina, CSc.



# Přístup k souborům

- FTP, telnet
  - Explicitní přístup
  - Spojení ke vzdáleným zdrojům přímo řídí uživatel
  - Např. editování souboru
    - Přenos na lokální disk
    - Modifikace
    - Uložení na původní místo
- Existuje však požadavek větší transparentnosti
  - Přistupovat ke vzdáleným zdrojům jako k lokálním
- V oblasti přístupu k souborům
  - Souborové systémy, síťové souborové systémy, distribuované souborové systémy



# Souborový systém

- Organizace souborů
  - Vyhledávání
  - Sdílení
  - Uchovávání
  - Označování
  - Ochrana
- Organizace adresářů
  - Adresářové služby
  - Spojení jména souboru s indentifikátorem souboru
  - Kontrola přístupových práv
- Operace na nižší úrovni
  - Diskové operace, využití vyrovnávacích pamětí, ...



# Klasický systém ovládání souborů

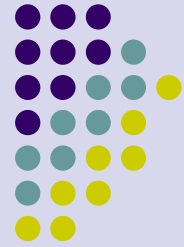
- Symbolic File system
  - Převod jména souboru na jeho jednoznačný identifikátor
  - Dovoluje jednoduše označit jeden fyzický soubor více jmény
- Basic File System
  - Zpřístupnění adresářové položky pro zadaný identifikátor souboru
  - Každý soubor je reprezentován pouze jednou, zde je uloženo kde se nachází a jaká jsou k němu přístupová práva
- File Access Control
  - Kontrola přístupových práv k souboru (přístupová matice, přístupový vektor, capabilities)



# Klasický systém ovládání souborů

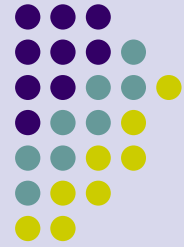
- Logical File System
  - Převod pozice v souboru na adresu první slabiky v souboru a výpočet čísla logického bloku včetně posunutí
- Physical File System
  - Převod logického čísla bloku a posunutí na číslo fyzického bloku a posunutí v bloku
- Device Access
  - Převod fyzického čísla bloku a posunutí na skutečné umístění bloku na vnějším médiu (sektor, stopa, cylindr)
- Memory Management
  - Organizace paměti na vnějším médiu

# Vlastnosti distribuovaného souborového systému



- Transparentnost přístupu
  - Klienti nevnímají, že soubory jsou na vzdálených serverech
- Transparentnost umístění
  - Konzistentní prostor jmen souborů (stejně označení lokální i vzdálené soubory)
- Transparentnost souběžných přístupů
  - Koherentní (vnitřně provázané) modifikace
- Transparentnost chyb
  - Schopnost klienta fungovat správně i po chybě serveru
- Heterogenita
  - Schopnost zajistit operace se soubory na různých programových i technických platformách

# Vlastnosti distribuovaného souborového systému



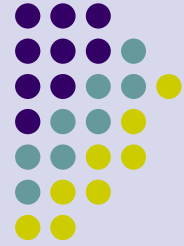
- Škálovatelnost
  - Schopnost spolupracovat s (libovolným – od několika po několik tisíců) počtem klientů bez omezení
- Transparentnost replikací
  - Klienti nerozpoznají replikace
  - Systém udržuje koherentní (konzistentní) data
- Transparentnost migrace
  - Soubory se mohou přemísťovat bez vědomí klientů
- Jemná distribuce dat
  - Snaha o umístění objektů blízko procesů, které je využívají



# Souborový systém - pojmy

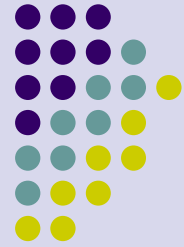
- Souborová služba
  - Specifikace co souborový systém nabízí klientům
- Soubor
  - Objekt s vlastnostmi jako jméno, atributy, data
- Immutable file – neměnitelný soubor
  - Soubor, který nelze měnit jakmile byl vytvořen
  - Zavádí se kvůli jednoduché replikovatelnosti a jednoduchému zpracování ve vyrovnávacích pamětech
- Ochrana souborů
  - Přístupová matice (soubory x uživatelé x přístupová práva)
  - Přístupový seznam (access control list)
  - Capabilities (schopnost, způsobilost)





# Typy služeb

- Model upload/download
  - Model s přesunem souboru
  - Čtení souboru – kopírování souboru ze serveru na klienta
  - Zápis do souboru – kopírování souboru z klienta na server
- Výhoda
  - Jednoduchá manipulace
- Nevýhoda
  - Konzistentnost – souběžná modifikace souboru více klienty
  - Efektivnost přenosu – co když je soubor příliš velký, co když klient nemá dostatek místa pro uložení souboru
  - Neúsporný – co když chce klient pracovat pouze s malým kouskem souboru



# Typy služeb

- Model vzdáleného přístupu
  - Souborový systém zajišťuje rozhraní pro provedení standardních funkcí
  - Create, delete, open, close, read, write, link, ...
  - Pracuje s jednotlivými bloky souboru metodou požadavek/odpověď
- Výhody
  - Klient požaduje jen to, co potřebuje
  - Server může zajistit konzistentní pohled na souborový systém (sdílený přístup, ...)
- Nevýhody
  - Možnost zahlcení serveru i sítě – server je požadován klientem mnohonásobně, mohou být požadována opakovaně tatáž data



# Problémy se jmény

- Mapování vzdálených souborových systémů může být realizováno na každém z klientů jinak
  - Nestejné připojení týchž podadresářů
  - Z hlediska programů, které zpracovávají data nevhodné
  - Mapování závislé na klientovi
- Logický požadavek identického mapování na všech klientech – transparentní připojení vzdáleného souborového systému
- Nejjednodušší řešení
  - Přejít od lokálního souborového systému k síťovému přidáním jména stroje k cestě k souboru



# Transparentnost umístění

- V přístupové cestě k souboru zadané jméno souborového serveru
  - Např. `//server1/dir/file`
  - Problém při přesunutí souborového systému na jiný počítač
  - `//server2/dir/file` – klient musí přepsat všechny odkazy
  - Není transparentní
- V přístupové cestě nesmí být uvedeno jméno souboru
  - Mapování na zvláštní svazek nebo do podadresáře
  - Jméno souborového serveru musí být uživatelskému programu skryto
  - Konfigurační soubor – není odolné vůči přemístění
  - Adresářový server – modifikace jména serveru mimo klienta – transparentní umístění

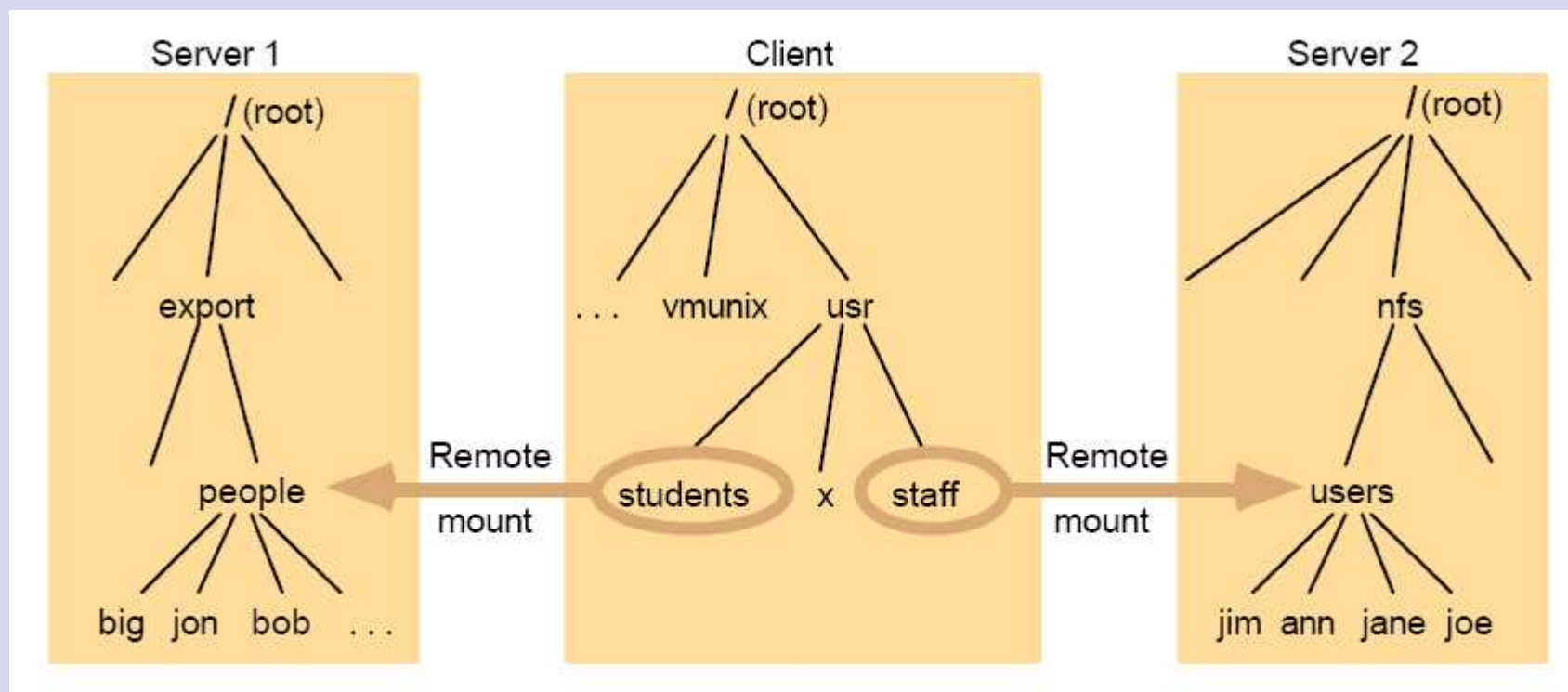


# Transparentnost přístupu

- Dovoluje aplikacím přistupovat k vzdáleným souborům jako k lokálním
- Jmenný prostor vzdáleného souborového systému musí být syntakticky konzistentní s lokálním jmenným prostorem
  - Jména souborů vzdáleného souborového systému musí odpovídat lokálním jménům
    - Soubory, adresáře, linky
  - Musí být vyřešen transparentní mechanismus mapování vzdálených souborů na lokální
    - Vzdálený souborový systém překrývá lokální prostor jmen
  - Problém s atributy souborů (přístupová práva)

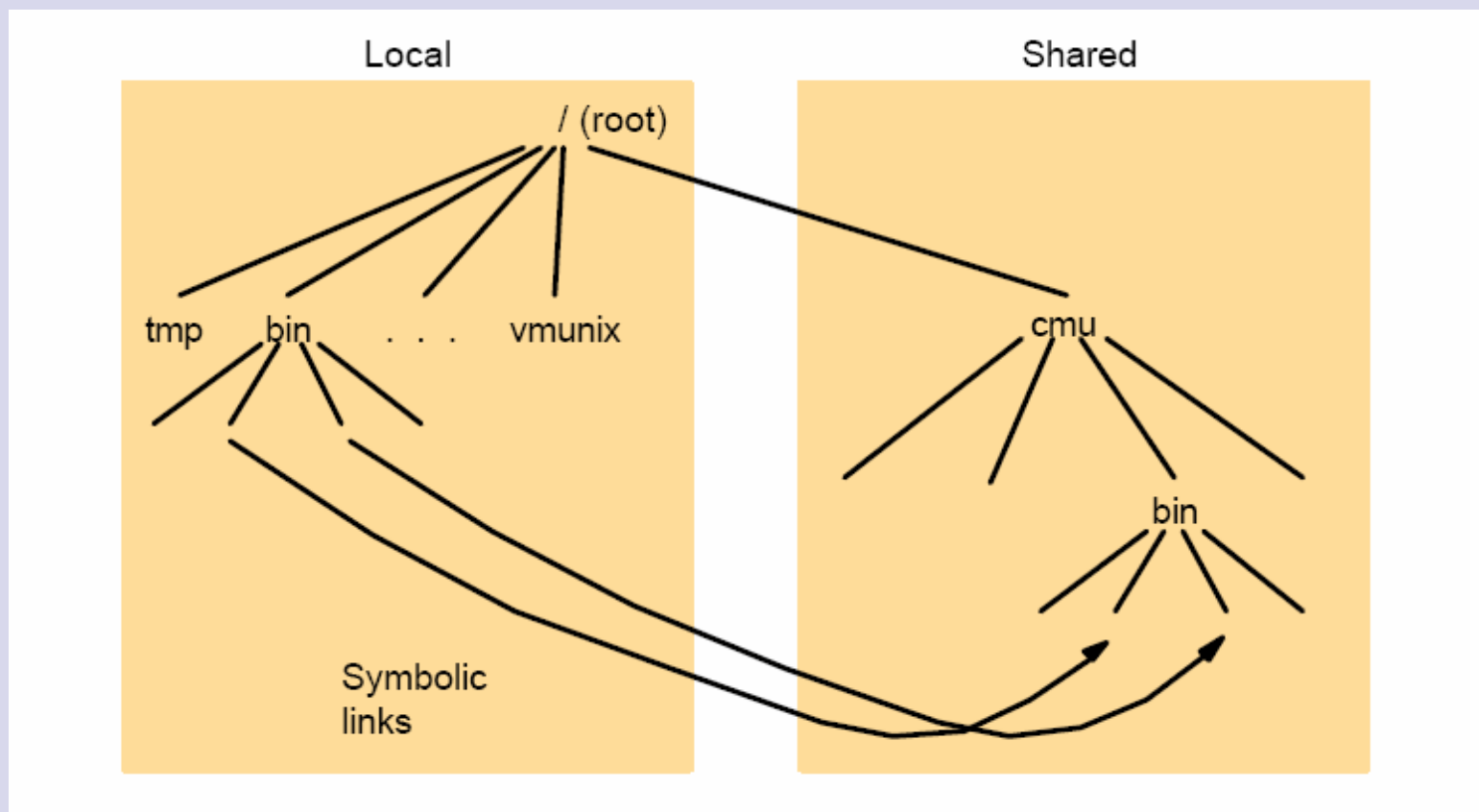


# Transparentnost jmen v NFS





# Transarentnost jmen v AFS





# Sémantika sdílení souborů

- Sekvenční sémantika
  - Sémantika používaná v Unixu
  - Výsledkem je poslední zapsaná hodnota
- Relační sémantika
  - Změny v souboru jsou pro ostatní viditelné po jeho uzavření
- Immutable files (neměnitelné soubory)
  - Sémantika používaná v AFS
  - Výsledkem je nová kopie souboru
- Transakční sémantika
  - Provedení operací čtení a zápisu jako by to byla transakce
  - Nejnáročnější velká režie





# Absolutní uspořádání podle času

- Sekvenční sémantika
  - Operace čtení vždy vrací hodnotu posledního zápisu (striktně konzistentní)
  - Jednoduše dosažitelné pro
    - Systém s jedním souborovým serverem
    - Na straně klientů se nepoužívají vyrovnávací paměti
  - Nevýhoda
    - Problémy s výkonností pokud nepoužívám cache
    - Použití cache (pro čtení) a operace write-through
    - Musí existovat mechanismus pro oznámení modifikace všem klientům, majícím kopii dat ve vyrovnávací paměti
    - Vyžaduje stavy navíc a přenosy navíc



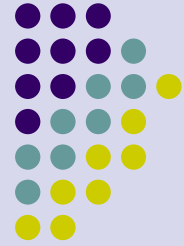
# Relační sémantika

- Session semantics (relační)
- Změny v otevřeném souboru jsou viditelné pouze procesu (nebo počítači), který jej modifikuje
- Uvolnění pravidel oproti sekvenční sémantice



# Neměnitelné soubory

- Vytváření neměnitelných souborů
  - Výhoda při replikacích
  - Nad souborem jsou povoleny pouze operace vytvoření souboru, a čtení souboru
  - Původní soubor nelze modifikovat
  - Oprava souboru znamená
    - Vytvoření nové verze souboru
    - Čtení původního souboru, vytvoření nového souboru kopírováním nebo přidáváním/ubíráním dat.
  - Neřeší problém detekce změn
    - Problém zatřídění dvou modifikovaných souborů



# Transakční sémantika

- Ke každému souboru se přistupuje přes transakční mechanismus
  - Uzamčení přístupu (čtení/zápis)
  - Provedení transakce
  - Ukončení transakce
- Výsledkem je sekvenční přístup (sekvenční modifikace)
- Příklad jak lze řešení jednoho problému převést na řešení na jiné úrovni



## Přehled charakteristik souborů

- Řešíme sdílení souborů v síťovém nebo distribuovaném souborovém systému bez znalosti četností výskytu jednotlivých typů souborů
- V r.1981 byla provedena studie (Satyanarayanan)
  - Rozdělil soubory podle délky
  - Podle požadavků na sdílení
  - Podle kritéria trvanlivosti (dočasné/trvalé)
- Na základě tohoto rozdělení navrhnout optimální systém pro manipulaci se soubory

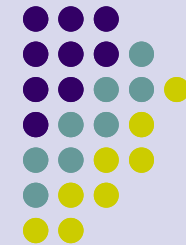


# Analýza použití souborů

- Mnoho souborů je < 10KB
  - Malé soubory vhodné pro přenos jako celek
  - Musí se však podporovat i přenos velkých souborů
- Mnoho souborů má krátkou dobu života
  - Možnost udržovat je pouze lokálně (ne na serveru)
- Minimum souborů je sdíleno
  - Přetrvávající problém
  - Relační sémantika by nemusela v mnoha případech způsobovat problém

# Problémy s návrhem systému

---





# Práce se jmény

- Soubor je identifikován
  - Jménem (cestou k souboru)
  - Identifikátorem (id)
- Používá se id souboru
  - Pamatuje se mapování jméno - id





# Stavový server

- Server udržuje informaci o stavu klienta
  - Seznam otevřených souborů
  - Pozice čtení/zápisu v otevřeném souboru
- Kratší dotazy
- Lepší výkonnost při zpracování dotazů
- Jednodušší udržení konzistentního obsahu vyrovnávací paměti
  - Server může registrovat kdo přistupuje ke kterému souboru
  - Sdílení souborů
- Není problém s uzamykáním souborů



# Bezestavový server

- Server neudrží o klientovi žádnou informaci
- Každý požadavek musí identifikovat soubor, klienta, offset
- Není problém s výpadkem serveru
  - Nedojde ke ztrátě informace o stavu klientů
- Není problém s výpadkem klienta a jeho obnovou
- Operace open a close pouze ovlivňují práva přístupu k souboru
- Server si nemusí pamatovat stav – není problém se škálovatelností
- Není možné uzamykání souboru



# Vyrovnávací paměti

- Vyrovnávací paměti výrazně zlepšují výkonnost systému
- Mohou být umístěny
  - Na disku serveru
  - Ve vyrovnávací paměti serveru
  - Ve vyrovnávací paměti klienta
  - Na disku klienta
- Použití vyrovnávacích pamětí s sebou nese problém s udržením konzistentnosti vyrovnávací paměti
  - Konzistentnost vyrovnávací paměť – diskové úložiště (zápis)
  - Konzistentnost mezi vyrovnávacími paměťmi
    - Na serveru
    - Na klientech



# Vyrovnávací paměti

- Problémy
  - Vyrovnávací paměti na serveru
    - Není problém se čtením, problém se zápisy
      - Společná vyrovnávací paměť
      - Vyrovnávací paměť pro každého klienta zvlášť
  - Vyrovnávací paměti na klientech
    - Není problém se čtením
    - Problém s udržení konzistentnosti při sdílení a zápisu



# Možnosti organizace zápisu

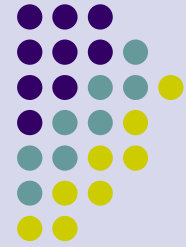
- Problém udržení konzistentnosti vyrovnávací paměti a diskového úložiště
- Souvisí se sémantikou manipulace se soubory
- Existují čtyři způsoby
  1. Přímý zápis (write-through)
    - Snížení průchodnosti systému
    - Příliš mnoho zápisů
  2. Zpožděný zápis
    - Omezení počtu zápisů
    - Shromažďování změn v lokální vyrovnávací paměti
    - Vyrovnávací paměť obsahuje nejnovější data – čtení z vyrovnávací paměti



## Možnosti organizace zápisu

3. Zápis při uzavření souboru (write on close)
  - Přípustné pokud budeme uvažovat relační sémantiku
4. Centralizované řízení
  - Stavový souborový systém s velkým množstvím synchronizačních zpráv
  - Degradace distribuovaného přístupu

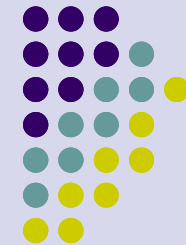
# Příklady distribuovaných a síťových souborových systémů



- NFS – Network File System (ver 1 – ver 4) (Sun Microsystems)
- AFS – Andrew File System (Carnegie-Mellon University)
- DFS - Distributed File System (Open Group)
- RFS – Remote File Sharing (AT&T Unix System V - 1986)
- Coda – Constant Data Availability (Carnegie-Mellon University)
- SMB – Server Message Blocks (Microsoft)

# Network File System

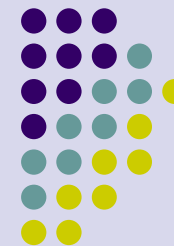
---



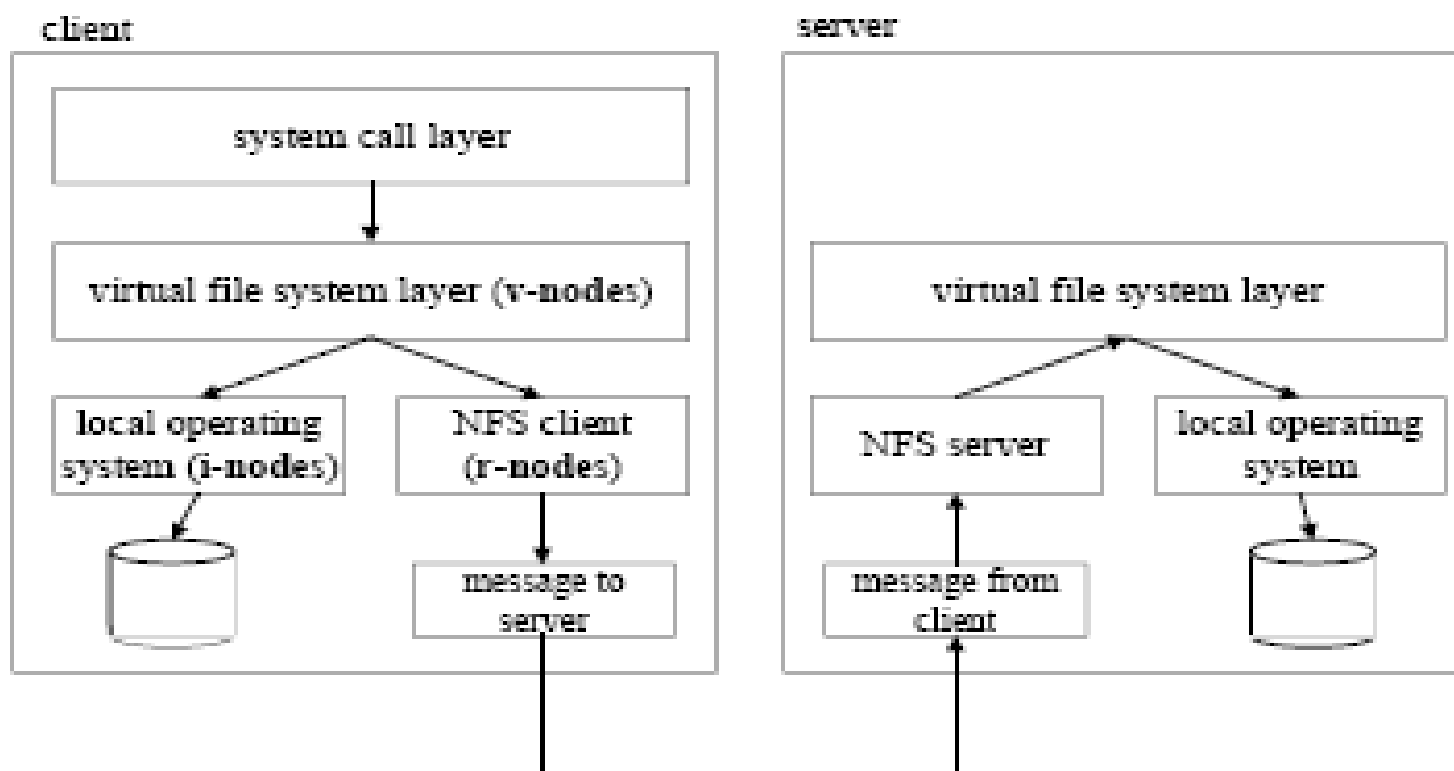
Sun Microsystems 1985



# Implementace NFS v operačním systému



## NFS Implementation





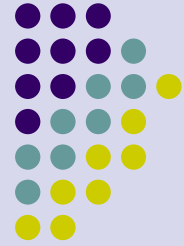
## Cíle návrhu NFS

- Jakýkoliv počítač může být klientem nebo serverem
- Musí podporovat bezdiskové stanice
- Musí být podporovány heterogenní systémy
  - Různý hardware, operační systém, souborový systém
- Transparentnost přístupu
  - Požadavek přistupovat k vzdáleným souborům stejně jako k lokálním prostřednictvím normálních systémových volání OS
- Obnova po chybě
  - Bezestavový server, použití UDP, obnova opakováním požadavku, odolný proti výpadku serveru i klienta



## Cíle návrhu NFS

- Vysoká propustnost
  - Použití vyrovnávacích pamětí a dopředného čtení
- Nezahrnuje transparentnost migrace
  - Jména serverů uvedena v konfiguračních souborech
- Nepodporuje UNIX sémantiku pro přístup k souborům
  - Bezstavové servery, problém s uzamykáním souborů (ver.3)
- Musí podporovat bezdiskové stanice
  - Všechny soubory jsou vzdálené
- Problém s ověřováním uživatelů
  - NIS, NYS, ...



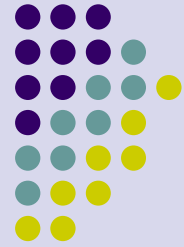
# Cíle návrhu NFS

- Transportní protokol
  - Používá UDP plus SUN RPC
- Snaha o minimalizaci režie přenosového protokolu
  - UDP je rychlejší než TCP
  - UDP je jednodušší
  - Nemusí udržovat spojení
  - Pro prostředí Ethernetu relativně spolehlivý
  - Zahrnuta detekce chyb
    - Nemá korekci
    - Korekce opakováním požadavků
- Výhodné z hlediska spolehlivosti



# NFS protokoly

- Mount protocol
  - Mapování exportovaného adresářového stromu na lokální disk
- NFS protocol
  - Přístup k souborům a adresářům
  - Zajištění souborových operací jako pro lokální souborový systém
  - Manipulace s právy
  - Manipulace s linky
- Daemoni
  - nfsd, biod
  - Daemon biod pro urychlení přístupu k souborům



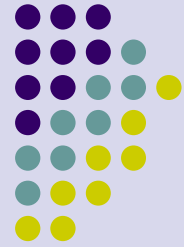
# Montovací protokol

- Klient posílá na server cestu k adresářovému stromu
- Server kontroluje práva přístupu
- Server vrací **file handle**
  - Číslo zařízení, inode, instance
- Klient vytváří bod přístupu



# Přístup k souborům a adresářům

- Vyhledání souboru
  - Vrací handle souboru a atributy
  - Handle se použije pro další přístup k souboru (read, write)
- NFS podporuje 16 (v2), 24(v3) funkcí
  - Lookup
  - Vytváření souborů
  - Vytváření adresářů
  - Práce s atributy
  - Vytváření symbolických linků
  - Zjištění stavu
  - uzamykání



# Výkonnost NFS

- Vzdálený přístup je pomalejší než lokální
- Využití vyrovnávacích pamětí na straně klienta
  - Snížení operací se vzdáleným serverem
  - Běžně operace čtení (soubor, atributy, adresáře)
  - Ukládání
    - Data, atributy souborů, mapování jmen
- Využití vyrovnávacích pamětí na straně serveru
  - Automatické použití vyrovnávacích pamětí při přístupu na disk
  - Okamžitý zápis na disk
  - Obrana proti ztrátě dat při výpadku serveru





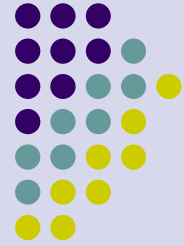
# Technika dopředného čtení

- Data jsou přenášena v 8KB blocích
- Předpokládá se sekvenční přístup k souboru
- Další čtení následuje za předchozím – z vyrovnávací paměti



# Problémy s NFS

- Konzistentnost souborů
- Předpokládá synchronizované hodiny
- Negarantuje přidávání za soubor
- Uzamykání přidáno jako separátní manager zámků (stavový)
- Nemá čítač otevřených souborů
  - Možnost smazání souboru pod rukami jiných klientů
- Předpokládá globální prostor UID
  - Přístupová práva k souborům podle UID a GID



# Problémy s NFS

- Mohou se měnit práva přístupu k souborům
  - Zneplatnění přístupu k souboru
- Nezajišťuje šifrování
  - Požadavky jsou přenášeny nešifrovaným RPC
  - Šifrování na úrovni uživatele
  - Možnost ověřování
    - Deffie-Hellman, Kerberos, UNIXové



# Automounter

- Problémy s montováním
  - Více montování trvá příliš dlouho
  - Každý počítač může mít vlastní prostor jmen
- Automounter
  - Dovoluje administrátorům vytvořit globální prostor jmen
  - Podporuje montování na přání
  - Řeší problém statického montování
    - Klient posílá požadavek na všechny servery
    - Nejrychlejší server vyhrává
    - Autokonfigurace založená na periodickém pokusu o odmontování svazku



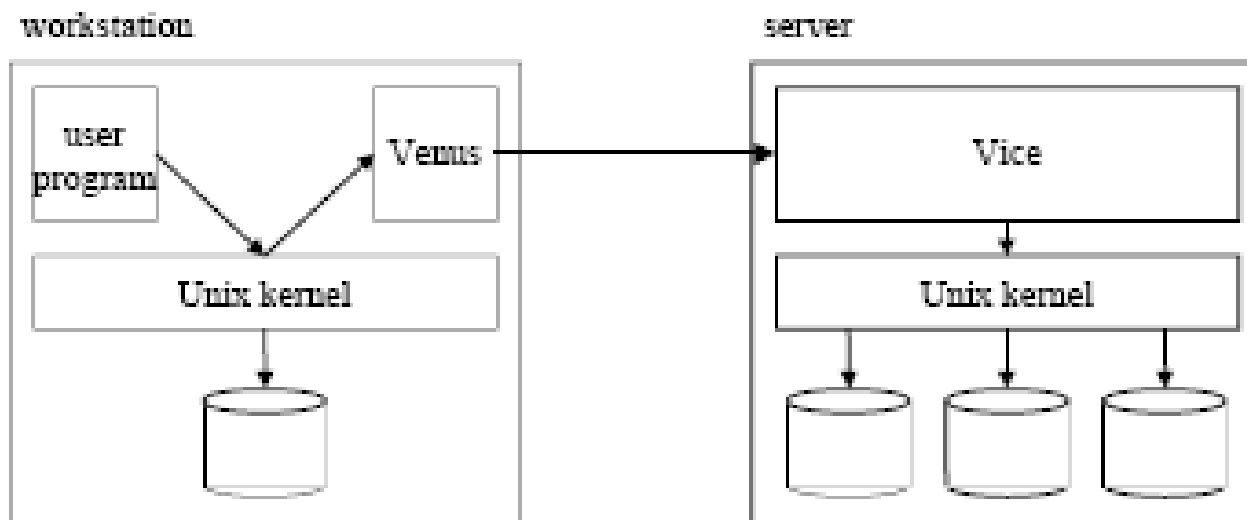
# Souborový systém AFS

- Vytvořen na CMU (Carnegie Mellon University)
- Později komerční produkt (IBM Transarc)
- Nyní volně k dispozici (Public Licence)
- Podpora velkého počtu klientů



# Implementace AFS

## AFS Implementation





# Předpoklady

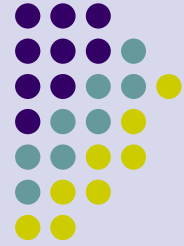
- Mnoho souborů je malých
- Počet čtení převyšuje počet zápisů
- K mnoha souborům přistupuje v dané chvíli jen jeden uživatel
- Soubory jsou využívány v časových shlucích
  - Soubor je chvíli využíván a pak dlouho ne



# Základní vlastnosti

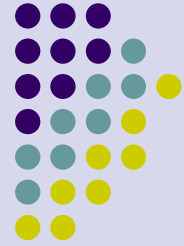
- Sdílení celých souborů
- Ukládání celých souborů do vyrovnávacích pamětí
  - Na disku klienta
  - Zápis zpět na souborový server v případě modifikace po uzavření
  - Zachování kopie souboru pro příští použití
- Klient komunikuje s množinou serverů
- Každý server předkládá identický prostor jmen
- Transparentní umístění souborů





# Svazky

- Fyzický diskový svazek rozdělen na volumes
- Volumes obsahují soubory i adresáře
- Administrativní organizační jednotka
- Každý volume je adresářovým stromem s jedním kořenem
- Volume je domácím adresářem uživatele
- Obsahuje prostředky pro omezení rozsahu volume (quota)



# Buňky

- Servery jsou seskupovány do administrativních entit – cell (buňka)
- Cell
  - Soubor serverů
  - Uživatelů
  - Klientským počítačů
- Cell je autonomní z hlediska administrace
- Cell mohou spolupracovat
- Mohou prezentovat uživatele v jednotném jmenném prostoru

# Jména



- Klienti získávají informaci prostřednictvím „cell directory server“
- Cíl:
  - Všichni vidí tentýž prostor jmen
  - /afs/cellname/path
  - /afs/zcu.cz/users/n/novak/...



# Ověřování a přístup k souborům

- **Ověřování pomocí Kerbera**
  - Důvěryhodné tickety
  - Vzájemné ověřování (server/klient)
- **Ověření uživatele vůči Kerberu (klog)**
- **Získání tokenu**
- **Předkládání při přístupu k souboru**
- **Neověření (anonymní) uživatelé**
  - `system:anyuser`

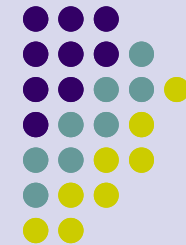


# Replikace a caching

- Replikace se týká svazků (volumes) R/O
- Mohou být replikovány na více serverech
- Pro příliš velké soubory není vhodné je přetahovat jako celek
  - AFS ukládá do vyrovnávací paměti implicitně části o velikosti 64KB
  - Adresáře jsou přenášeny do vyrovnávací paměti celé
- Podpora uzamykání

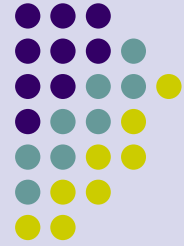
# **Coda** **(Constant Data Availability)**

---



Carnegie-Mellon University  
1990-1992

# Úvod



- Distribuovaný souborový systém
- AFS 1985
- Coda 1987 - návrh
- CMU M. Satyanarayanan
- Nyní implementace pro
  - Linux
  - BSD
  - Windows



# Architektura systému

- Coda vychází z AFS
- Základem je Coda cell
- Zahrnuje
  - Klienty
  - Souborové a datové servery
  - Řídicí systém (System Control machine)
  - Zálohovací počítače
- Vše může být jeden stroj





# Architektura systému

- Klienti
  - Obsahují cache paměť, disková paměť, trvalé uložení informace
  - Rozsah 20MB až 200MB
  - Uchování celého souboru (pro maximální velikost souboru)
  - Data ve vyrovnávací paměti organizuje cache manager (Venus)



# Cache manager Venus

- Volán při požadavku otevření souboru
  - Volání open() z programu
  - Vyvolá se jádro a VFS (Virtual File Systém)
  - Vyvolá se modul jádra Coda
  - Vyvolá se proces Venus
  - Kontrola existence souboru v cache a volání souborového serveru
  - Případný přesun souboru do cache (up-to-date)
  - Návrat do jádra s odkazem na soubor



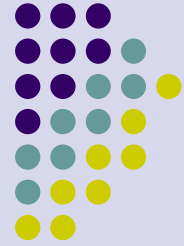
## Svazky - volumes

- Volume je virtuální svazek (partition)
- Pokud se z ní má číst nebo do ní zapisovat, musí být namontována v celé oblasti (cell)
- Zálohována může být bez namontování
- Rozsah svazku je dán velikostí povolené quoty
- Není-li quota určena, může dosáhnout velikosti fyzické partition na disku.



# Klonování

- Klonování se provádí podobně jako při vytváření paralelního procesu v Unixu
- Originál (zapisovatelný) i klon (pouze pro čtení) sdílí stejný prostor na disku (na počátku)
- Pokud se originál změní, je odkaz na klon zrušen a je vytvořen nový soubor
- Klon slouží jako záložní kopie – pomáhá při zálohování a při replikaci R/O svazků



# Typy svazků

- Jednoduchý (R/W)
- Read-only (jako backup) – klon
  - Jakoby kopie v cache klienta
  - Obnova po smazání souboru
- Replicated (R/O)
  - Klonování se provádí na více serverů – identické kopie
  - Určené pouze pro čtení
  - Celé svazky se replikují periodicky (za běhu)
- Replicated (R/W)



# Operace bez spojení se serverem

- Základní myšlenka Coda systému
  - Umožnit mobilní počítání
  - Ztráta spojení se serverem
    - při přesunu,
    - vlivem výpadku serveru
    - Vlivem chyby v síti
  - Off-line operace čtení i zápisu
  - Řešení konfliktů zápisu (porovnání souborů) po obnově spojení



# Uživatelé a skupiny

- Coda zavání vlastní uživatele a skupiny
  - Celá čísla (kladná – uživatelé, záporná – skupiny)
  - Nemusí být stejná jako v Unixu (problémy)
  - Skupina má vlastníka
  - Může přidávat do skupiny další uživatele
  - pdbtool – na řídicím systému, root
- Ověřování
  - Coda (kclog, Coda token)
  - Kerberos 4
  - Kerberos 5



# ACL – Access Control List

- ACL vztažen k adresářům
- Přístupové bity UNIXu ignorovány pro přístup k adresářům
- Přístupové bity skupiny a ostatních k souborům ignorovány, pro uživatele ne
- Přísnější kontrola přístupu k souborům (r,x)
- Oddělení prostoru jmen UNIX od Coda
  - problémy



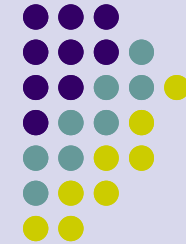


# Coda server

- codasrv – souborový server
- updateclnt, updatesrv
  - Řeší problém konzistentnosti dat mezi servery a serverem a klientem
- rpc2portmap – rpc daemon – požadavky na Coda služby
- kauth2 – generování Coda tokens podle uživatelských Kerberos ticketů

# Distributed File System

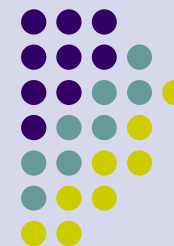
---





# DFS

- Součást DCE (Distributed Computing Enviroment)
- Následník AFS
- Předpokládá
  - Většina přístupů k souborům je sekvenční
  - Doba života většiny souborů je malá
  - Převážná část přístupů vyžaduje celý soubor
  - Mnoho přístupů je do malých souborů

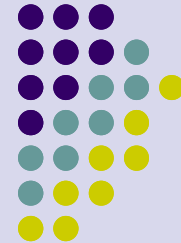


# Cíle

- Využívá ukládání celých souborů do vyrovnávacích pamětí na straně klienta jako AFS
- Nepoužívá relační sémantiku
- Zavádí striktní konzistentnost (UNIX sémantiku)

# Server Message Blocks

---





# Cíle

- Sdílení souborů v prostředí Windows
- Protokol pro sdílení
  - Sdílení souborů, zařízení, komunikačních prvků (sériový), poštovní schránky
- Servery umožňují přistupovat klientům ke zdrojům
- Klienti mají přístup k souborům, tiskárnám, ...
- Zajištění konzistentnosti přístupu, uzamykání, přístupová práva



# Vlastnosti SMB

- Protokol typu dotaz/odpověď
  - Posílá message blocks (dle DOS)
  - Spojově orientovaný protokol
  - Každá zpráva obsahuje
    - Záhlaví
    - Příkazový řetězec
    - Nebo řetězec odpovědi