NFSv4
i konkurencja

9 grudnia 2004

Plan prezentacji

- Projekt NFS i krótko o NFSv3
- NFSv4 – plany i implementacja
- Testy wydajnościowe NFSv4
- Aspiracje NFSv4
- NFS i inni – porównanie

Historia projektu

- Produkt Sun Microsystems
- Projekt prosty i przenośny
- Wersje:
  - 1: nie ujrzała światła dziennego
  - 2: wydana w 1985
  - 3: wydana w 1994
  - 4:
    - prace trwają od połowy ’90
    - RFC(3010) z 12.2000
    - RFC(3530) z 04.2003

Historia projektu (cd)

- NFS <= 3:
  - Specyfikacja - Sun
  - Implementacja – chętni
- NFS 4:
  - Specyfikacja – IETF *(Internet Engineering Task Force)*
  - Implementacja - chętni

NFSv3

- Przezroczystość dostępu
- Przezroczystość położenia
- Przezroczystość awarii

NFSv3 (cd)

- RPC i XDR
- Bezstanowy serwer
  - Klient identyfikuje się przy każdym odwołaniu do serwera
  - Operacje idempotentne
  - Bufor klienta aktualizowany przez klienta
- Bufor serwera
  - Buforowany odczyt
  - Zapis natychmiastowy

NFSv3: co przeszkadza?

- NFS = Network File System
- NFS ≈ Not For Speed
  - ls -l
  - Niedoskonały bufor klienta
- NFS ≈ Not For Security
  - Łatwo podsłuchać
  - Łatwo się podszyć
- Negocjowany port montowania


NFSv4

NFSv4: forma projektu

- Opracowywanie specyfikacji: IETF
- Implementacja:
  - Kto chce, w szczególności:
    - *Center for Information Technology Integration* na *University of Michigan*, wspierane przez
      - Sun Microsystems
      - Network Appliance
- RFC => publiczne debatowanie
  - m.in. *NAS Industry Conference*
  - Bake-A-Thon *(@CITI)*


Center for Information Technology Integration
NFSv4
2002

NFSv4: priorytety

- Szybkość
- Bezpieczeństwo
- Szybkość
- Bezpieczeństwo
- Szybkość
- Bezpieczeństwo

NFSv4: główne założenia

- Globalna przestrzeń nazw zasobów
- Przezroczystość:
  - Lokalizacji
  - Migracji i replikacji
- Skalowalność:
  - Unikanie scentralizowanych usług
  - Przerzucanie zadań na klientów
- Modularność i przenośność

NFSv4: główne założenia (cd)

- Czytanie wyłączne: bardzo częste
- Czytanie jednoczesne: często
- Pisanie wyłączne: nieczęsto
- Pisanie jednoczesne: bardzo rzadkie

NFSv4: umiędzynarodowienie

- Zastąpiono 7-bitowe ASCII/8-bitowe Latin I standardem UTF-8

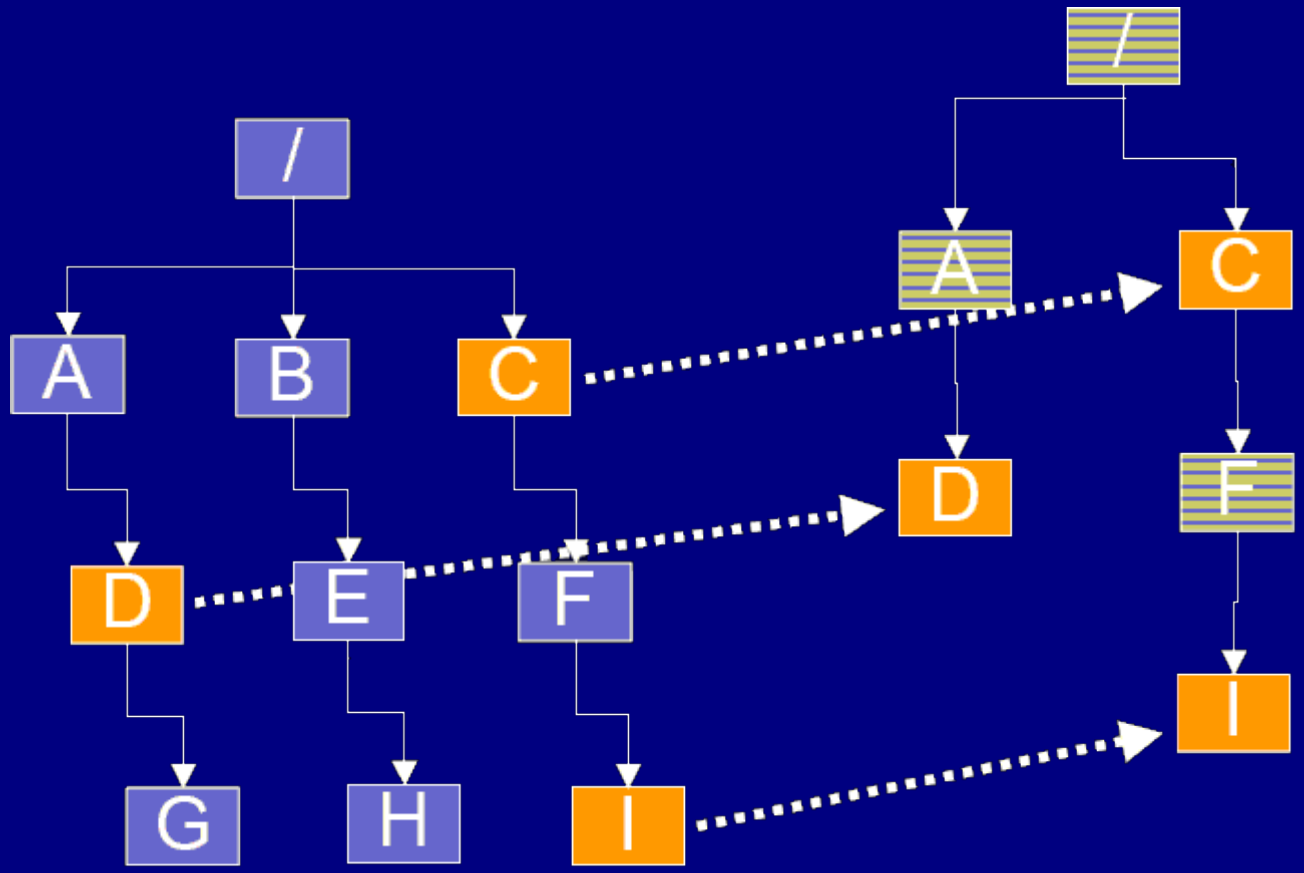
NFSv4: serwer – udostępnianie

- Administrator określa udostępniane katalogi
- Serwer buduje wirtualne drzewo z udostępnianymi katalogami
- Serwer udostępnia uchwyt do korzenia drzewa eksportowanych katalogów

NFSv4: pseudofs

FS serwera

pseudofs serwera



NFSv4: przestrzeń nazw - klient

- Klient odnajduje za pomocą DNS lokalizację serwera z zasobem
- Klient wykonuję operację PUTROOTFH
- Klient może przeglądać zawartość serwera

NFSv4: /nfs

- /nfs początkowo pusty
- cd /nfs/mimuw.edu.pl/home/jsem/me201258
- Serwer jest lokalizowany, klient podłącza się i wchodzi do odpowiedniego katalogu
- Po sukcesie w /nfs mieć będziemy podkatalog mimuw.edu.pl

NFSv4: AutoFS i Automount

- AutoFS *(FS w jądrze)* monitoruje i przekazuje
- Automount *(user-level deamon)* inicjuje podłączanie
- AutoFS sprząta nie używane wpisy

NFSv4: alternatywne lokalizacje

- Katalogi na serwerach NFSv4 mogą mieć dołączone informacje:
  - jak odnaleźć informacje o położeniu replik
    - ldap://ldapserver/lookup-key
    - dns://lookup-name
    - file://pathname/lookup-name
  - SERVER REDIRECT
    - server://hostname:/path [mount-options]
- Wybór serwera pozostawiony jest klientowi (Automount)

NFSv4: replikacje

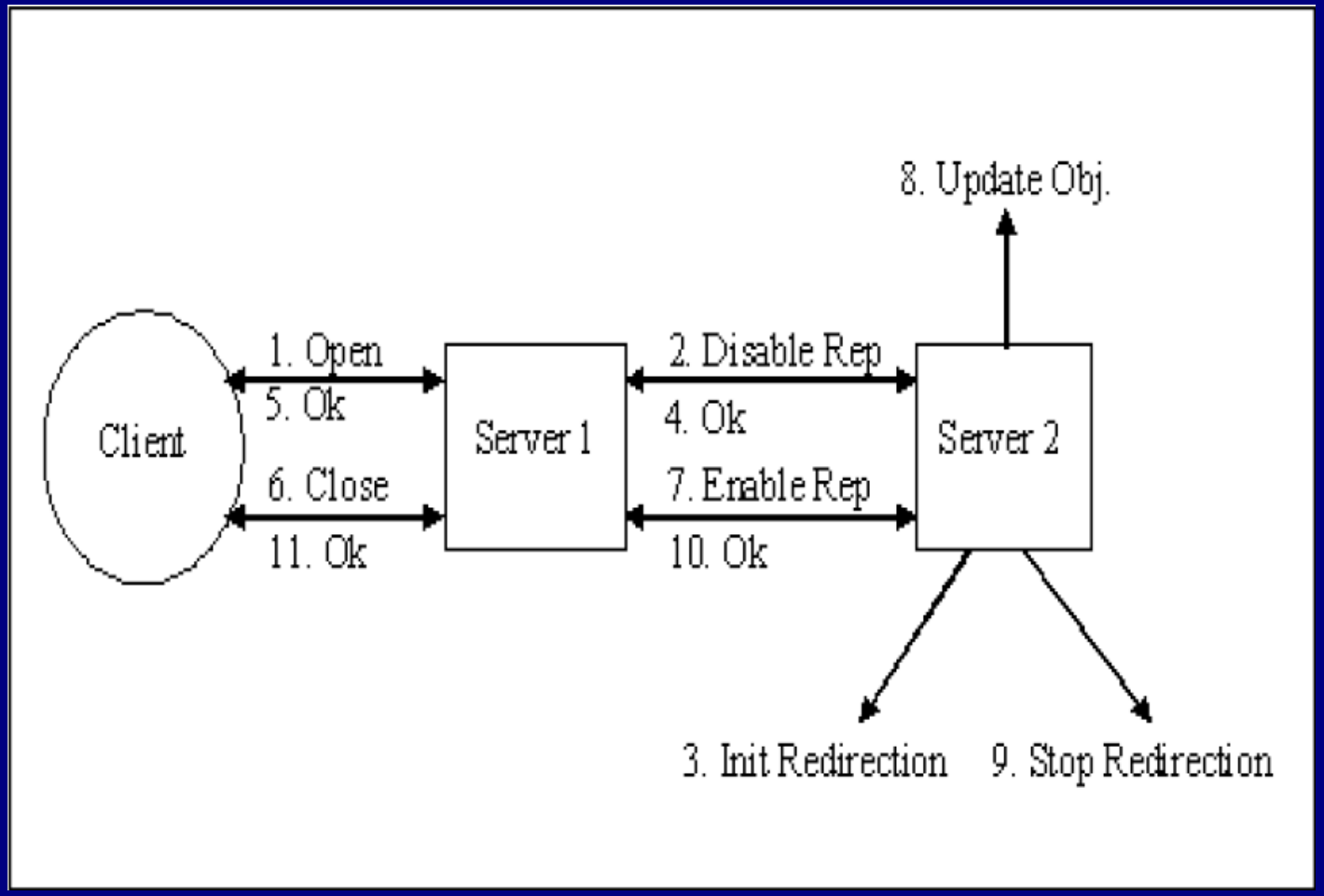
- RFC mówi: tylko zasoby tylko-do-odczytu
- CITI mówi: replikacja dla każdego

- RSYNC do uzgadniania kopii

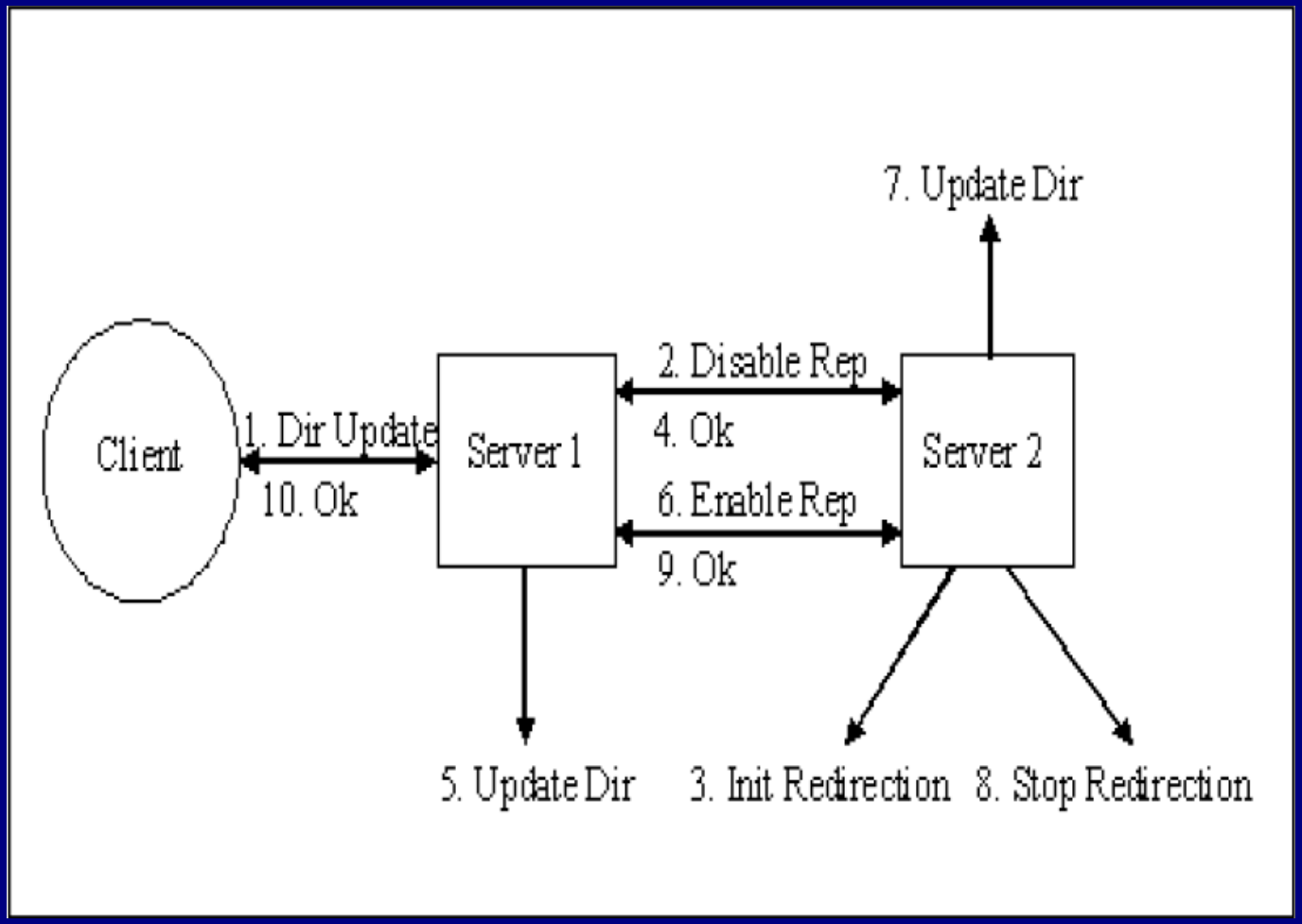
NFSv4: replikacje a aktualizacje

Serwer, który otrzymał żądanie zmiany replikowanego pliku/katalogu, zakazuje pozostałym serwerom oferowania zasobu

NFSv4: replikacja vs. zmiany plików



NFSv4: replikacja vs. zmiany katalogów



NFSv4: replikacja - konfilkty

Konflikt – gdy dwa serwery w tym samym czasie zakazują udostępniania

- Wygrywa ten, który zastopuje więcej serwerów
- Temat będzie jeszcze eksplorowany przez CITI

NFSv4: replikacja – awarie

Awaria – gdy serwer przestanie odpowiadać

- Klient, który chciał pisać/pisał wykrywa awarię serwera i zgłasza to do FDR *(Failure Detection and Resolution Program)*
  - Jeśli replikacja była wstrzymana wszędzie – wznawia ją i informuje klienta o awarii
  - Jeśli replikacja była wstrzymana gdzieniegdzie – wybiera dla klienta nowy serwer podstawowy i reaktywuje proces

NFSv4: replikacja – awarie (cd)

Dodatkowy problem: rozdzielenie sieci.

Jeśli którykolwiek z serwerów (nie podstawowych) stanie się niedostępny

=> wszystkie pozostałe przechodzą w tryb tylko-do-odczytu

NFSv4: procedury

- Podstawa: RPC + XDR
- Nowe operacje (OPEN, CLOSE)
- Tylko dwie procedury:
  - NULL – kontrolna
  - COMPOUND – wywołanie złożone
    - Grupuje potencjalnie wiele zapytań
    - Przetwarzane przez serwer do pierwszego błędu
    - Odpowiedź składa się z grupy rezultatów

NFSv4: tłumaczenie ścieżek

- NFSv3: przesłać całą scieżkę czy po kawałeczku?
- NFSv4: możemy przesłać całą i analizować po kawałeczku!

NFSv4: aktywny uchwyt

- Większość operacji wymaga uchwytu

- Serwer pamięta *aktywny uchwyt*
- Operacje nie zwracają uchwytu tylko aktualizują *aktywny*

- GETFH, SAVEFH

NFSv4: wygasły uchwyt

- NFSv3: stałe uchwyty
  - Łatwe zarządzanie
  - Nieprzenośne rozwiązanie
- NFSv4: nietrwałe uchwyty
  - Serwer NFS sam zarządza ważnością uchwytów
  - Gwarancja niezawodności

NFSv4: blokady

- Blokady zakresowe
- *Mandatory locks* (zgodne z Windows)
- Rezerwacja zasobu

NFSv4: identyfikatory

- clientid – jednoznacznie identyfikuje klienta i jego wcielenie
- stateid – identyfikuje stan blokad pliku

NFSv4: atrybuty

- Podział na:
  - obowiązkowe (8: typ obiektu, rozmiar)
  - zalecane (25)
- *Named attributes* – ukryty katalog z atrybutami

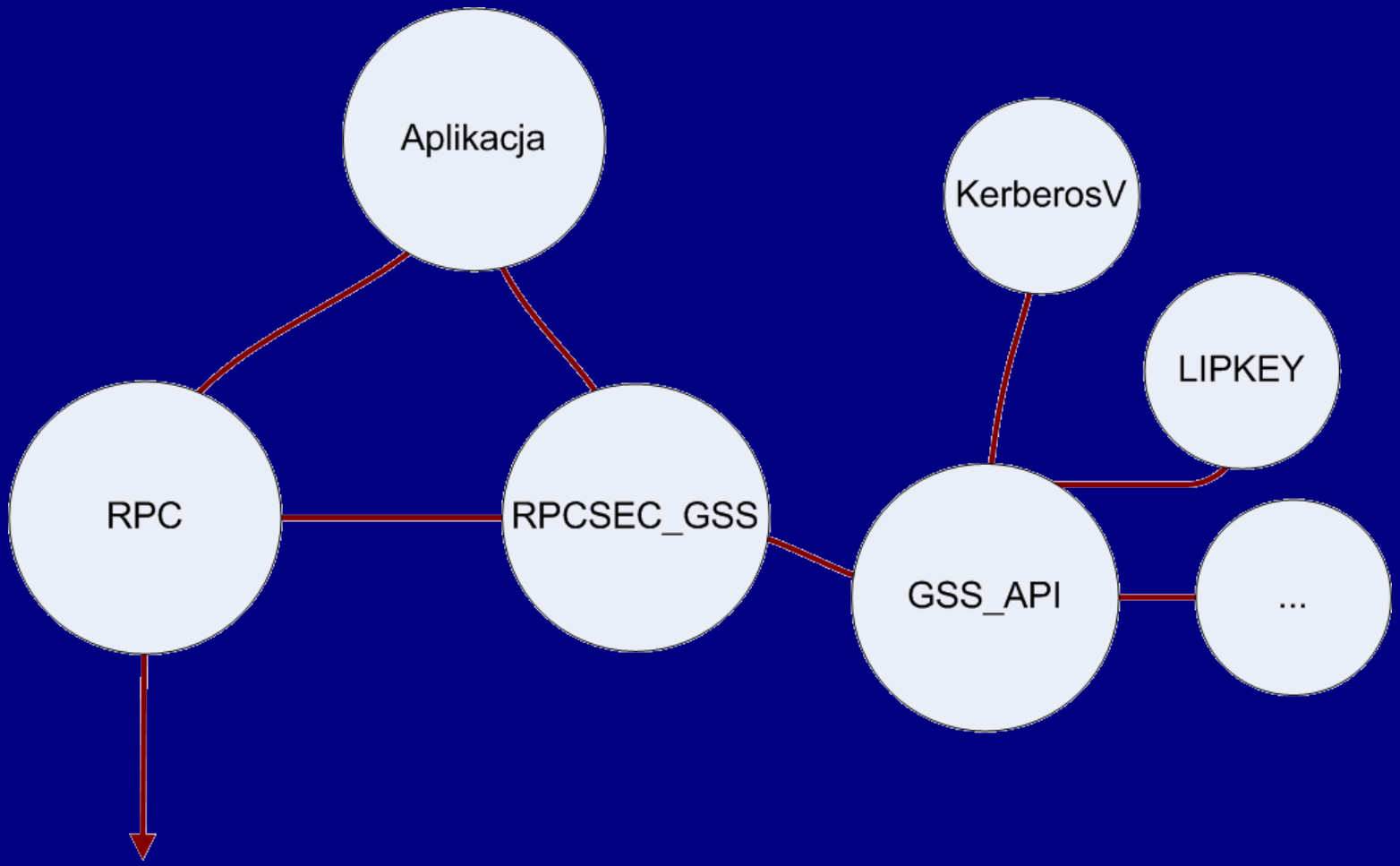
NFSv4: delegacje

The fastest packet is the one never sent

Brian Pawlowski

- Przy operacji OPEN serwer może przekazać klientowi delegację, o ile:
  - nikt inny nie pisze
  - klient przyjmuje komunikaty zwrotne *(callback)*
- Gdy delegacje nie są dostępne – cache po staremu

NFSv4: bezpieczeństwo



NFSv4: RPCSEC_GSS + GSS_API

- RPCSEC_GSS+GSS_API umożliwia:
  - autentykację
  - sprawdzanie integralności danych
  - zapewnienie prywatności
- Aplikacja przy tworzeniu wywołania RPC tworzy *kontekst* połączenia:
  - mechanizm bezpieczeństwa
  - QOP – *Quality of Protection*
  - typ usługi

NFSv4: GSS_API

- Niezależny od protokołu komunikacji interfejs zapewniający bezpieczeństwo
- GSS_API pośredniczy między implementacjami *mechanizmów* a aplikacjami
- Warstwa komunikacyjna przekazuje żetony, generowane i odczytywane przez GSS_API po obu stronach

NFSv4: Kerberos V

- Kerberos V wymaga sieci znających się serwerów
- Klient dostaje od swojego centrum dystrybucji kluczy (KDC) bilet
- Klient posługuje się biletem w komunikacji z docelową aplikacją

NFSv4: Kerberos V (cd)

- Efekty wymiany danych pomiędzy KDC a Klientem oraz Klientem a Serwerem:
  - Tylko KDC, Klient i Serwer znają klucz sesji
  - Serwer ma gwarancję, że biletem posługuje się jego właściciel
  - Serwer aplikacji nie musi autoryzować klientów – wystarczy, że zna się z KDC klienta

NFSv4: LIPKEY

- LIPKEY = *A Low Infrastructure Public Key Mechanism*
- Zapewnia bezpieczne połączenie klient-serwer a nie weryfikację tożsamości
- Od serwera wymaga:
  - wydania certyfikatu
- Od klienta wymaga:
  - posiadania klucza publicznego serwera

NFSv4: bezpieczeństwo

- Serwer i klient mogą negocjować poziom zabezpieczeń
- Kerberos V i LIPKEY – wymagane w każdej implementacji serwera/klienta
- SECINFO – klient może uzyskać listę mechanizmów obsługiwanych przez serwer

NFSv4-Linux: status implementacji

Funkcjonalność	**>= wersja jądra**
Podstawowe operacje	2.6.4
Blokady	2.6.4
Kerberos V	2.6.7
Delegowanie	2.6.9 (klient) 2.6.10 (serwer)

NFSv4: nad czym się pracuje (kernel 2.6.10)

- Bezpieczeństwo:
  - RPCSEC_GSS – prywatność
  - SECINFO
- Delegacje – serwer
- ACL
- *Named attributes*
- Nietrwałe uchwyty

NFSv4: testy

- Implementacja NFS ciągle nie jest stabilna
- Nie można w pełni testować (niezaimplementowane delegowanie!)
- Można testować fragmentami: jak nowa funkcjonalność wpływa na wydajność

NFSv4: opis testów

- Testowano prototypową implementację na *University of Michigan* jesienią 2003
- Testowano głównie pod kątem narzutów przy replikacji
- Warunki:
  - 100 Mbitowy LAN
  - Linux 2.5.68

Testy NFSv4: odnajdywanie położenia replik

Sposób uzyskiwania informacji	Czas
DNS TXT RR	1.49 ms
LDAP	12.3 ms
FILE	13.1 ms
SERVER REDIRECT	0.007 ms

Wnioski: zmierzony czas jest do zaakceptowania, tym bardziej, że operacja jest wykonywana przy pierwszym dostępie do obiektu

Testy NFSv4: otwieranie do zapisu

Repliki	Wstrzymanie udostępniania replik	Odpowiedź serwera
1	0 s	0.00349 s
1 + 1	0.000421 s	0.00392 s
1 + 2	0.000503 s	0.00398 s

Wnioski: replikacja nie powoduje opóźnień przy operacji OPEN

Testy NFSv4: zamykanie po zapisie

Repliki	Dystrybucja zmian	Włączenie replikacji	Odpowiedź serwera
1	0 s	0 s	0.00413 s
1 + 1	0.644 s	0.000386 s	0.649 s
1 + 2	0.687 s	0.000461 s	0.692 s

Wnioski: być może RSYNC nie był najlepszym pomysłem...

NFSv4: aspiracje WAN

- Globalna przestrzeń nazw użytkowników
- Wymagany minimalny poziom bezpieczeństwa
- Ogniomurkoprzyjazność

NFSv4: aspiracje
A common Internet file system

- Ściąganie danych
  - Lepszy niż FTP (bezpieczeństwo)
  - Lepszy niż HTTP (wznawianie)
- Komercyjne serwery kopii zapasowych
- Dyski Internetowe

- Ewolucja celów

OpenAFS

- 1984 – na *Carnegie-Mellon University* powstaje *AFS*
- 1998 – projekt zostaje skomercjalizowany i trafia do IBM
- 2000 – IBM decyduje się udostępnić AFS w ramach OpenSource

- Projekt rozwija się

OpenAFS – założenia

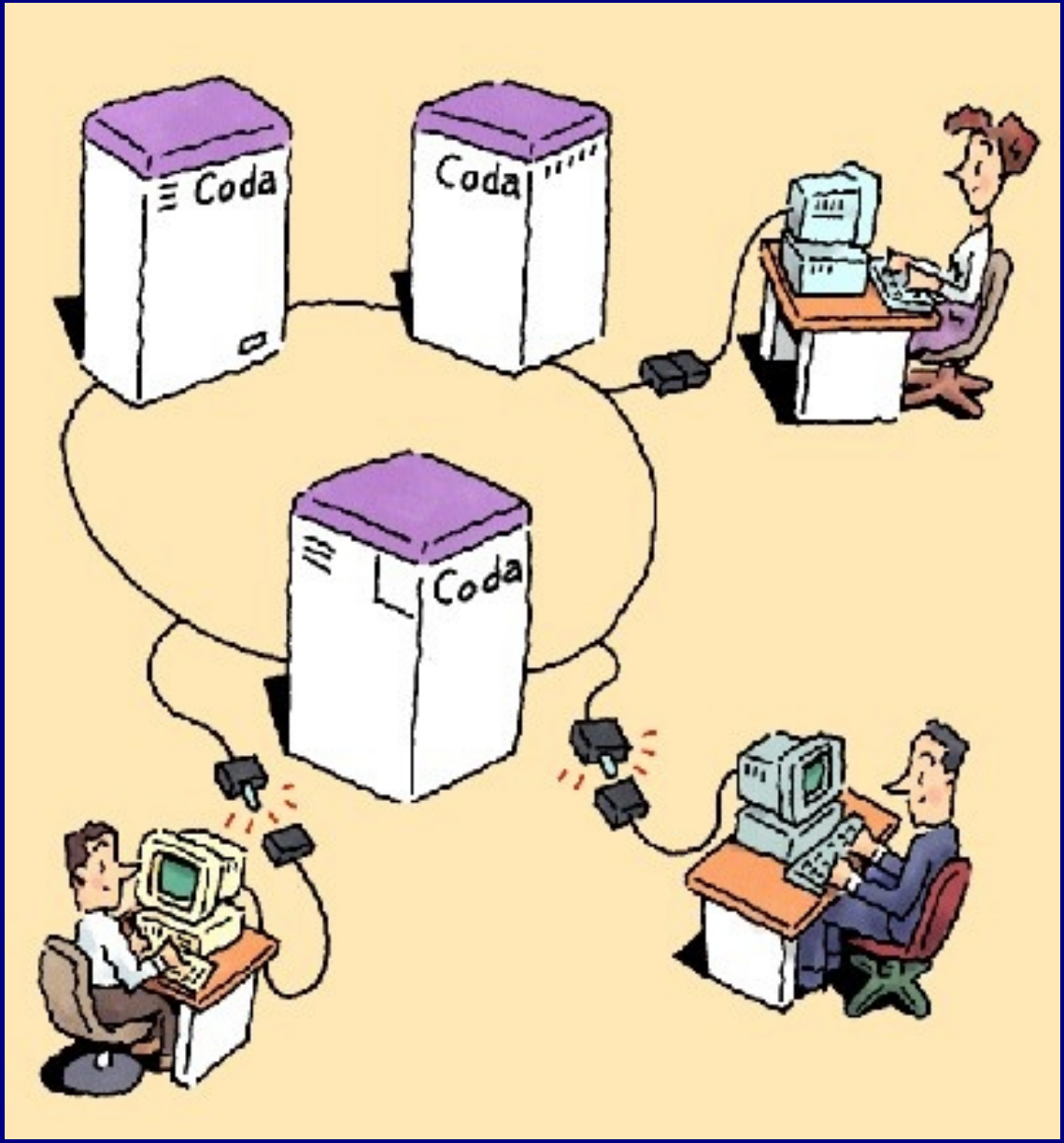
- Mnóstwo, mnóstwo ruchliwych klientów
- Pliki raczej małe
- Przeważnie tylko czytamy
- Jeśli piszemy to tylko my

OpenAFS - realizacja

- Sieć zorganizowana w komórki (domeny), klastry a dane w wolumeny
- Klient pracuje na lokalnych kopiach plików
- Serwer – *obietnice powiadomienia*
- Swobodna migracja wolumenów
- Możliwość replikacji
- Globalna przestrzeń nazw (/afs)
- Każdy serwer wie wszystko o lokalizacjach

OpenAFS vs NFSv4

OpenAFS	**NFSv4**
skalowalność	
migracje i replikacje	migracje i replikacje
przenośność (klient)	przenośność
	nietrudna instalacji


Coda
Coda
Coda

Słówko o systemie Coda

- Potomek AFS
- Żywy projekt prowadzony przez *Carnegie Mellon University*
- Założenia:
  - Małe pliki
  - Serwery bardzo awaryjne
  - Raczej bez współzawodnictwa w pisaniu
- Dołączony do jądra 2.4.0
- Ogromne problemy wydajnościowe


oceanstore

Oceanstore

- Zupełnie inne podejście: zbudowanie globalnej przestrzeni do przechowywania danych (stąd nazwa)
- Biliony użytkowników (10^{10} x 10,000 plików)
- Ogromna sieć (komercyjnych) serwerów
  - Wykupujemy dostęp
  - Publikujemy pliki
  - Zaszyfrowane dane rozprowadzane są po sieci
  - Mamy bezpieczeństwo danych i szybki dostęp

Oceanstore (cd)

- *Warstwa introspekcyjna*
  - Śledzi zapotrzebowanie na dane, steruje migracją i replikacją
- Lokalizacja obiektów (TAPESTRY)
  - Próbujemy probabilistycznie
  - Ew. deterministycznie
- Wersjonowanie plików i rozgłaszanie aktualizacji (sesji pracy z plikiem)
  - *deep archival storage*

Oceanstore (cd)

- Java
- Pond – instalacja prototypowej implementacji
  - 230,000 linii kodu
  - Świetne wyniki przy czytaniu
  - Gorsze przy pisaniu

Oceanstore vs NFSv4

- Różne cele
- Różne zastosowania
- Oba potrzebne

Moje miejsca sieciowe

CIFS/SMB

- Microsoft: *Common Internet File System* aka *Server Message Block*
- Korzenie sięgają lat 80-tych

  ...i dają o sobie znać

  7 równolegle funkcjonujących wersji

  ponad 100 poleceń, nierzadko pokrewnych
- Ma swoją Linuxową/Unixową implementację

  Opening Windows to a Wider World


samba

CIFS/SMB (cd)

- Siłą rzeczy niezwykle popularny

  ...ale ciekawy

  Command batching (grupowanie poleceń)

  Opportunistic locking

  File change notification

Przyszłość CIFS

- Microsoft zapewne uśmierci CIFS
  - Microsoft stracił kontrolę nad serwerami
  - Protokół zbyt złożony i drogi w rozwijaniu
- Microsoft ma swoje sposoby
  - Wprowadzenie WinFS
  - Zalecenie przepięcia się na nowy system
  - Licencjonowanie specyfikacji WinFS

WinFS

- *"WinFS" is the code name for the next generation storage system*
- Miał być nowy, bazować na SQLServer

  A może będzie bazować na NTFS?
- Miał być w Longhorn

  Ale nie będzie

WinFS – co wiadomo

- WinFS = wolumeny, foldery, przedmioty
- Folder = kolekcja (wirtualna lub fizyczna) przedmiotów
- Przedmiot = dokument wraz z metadanymi
- Wyszukiwanie oparte na *OPath*

- WinFS będzie dominować

Wielkie porównanie

	NFSv4	CIFS	OpenAFS	Oceanstore
spójność	+	+	-	-
bezpieczeństwo	+	-	+	+
LAN	+	+	-	-
WAN	+	-	+	+
sieć globalna	-	-	+	+
aut. migracje i replikacje	-	-	+	+

Podsumowanie

- Skala mikro (uczelnia, firma, korporacja):
  NFSv4
- Skala makro (e-dyski, ftp, nowe www):
  Oceanstore
  OpenAFS