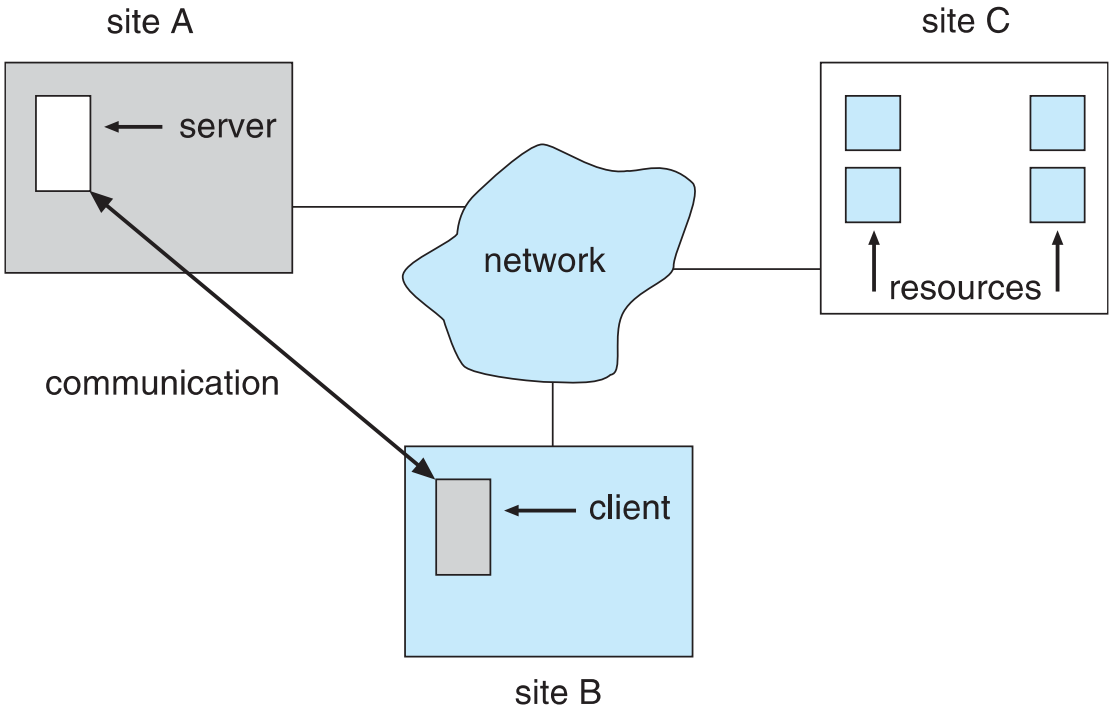
# Rozproszone systemy plików

Ostatnia modyfikacja: 22.06.2017

# Wprowadzenie

- **System rozproszony (*distributed system*)** to zbiór luźno połączonych maszyn, połączonych za pomocą sieci komunikacyjnej.



# Rozproszony system plików (DFS)

- **Rozproszony system plików (*distributed file system*, *DFS*)** – rozproszona implementacja klasycznego modelu lokalnego współdzielenia systemu plików, gdzie użytkownicy współdzielą pliki oraz urządzenia pamięci pomocniczej
- DFS zarządza zbiorem różnorodnych urządzeń pamięci pomocniczej (dysków) rozmieszczonych w różnych lokalizacjach.
- Zazwyczaj urządzenia w jednej lokalizacji przechowują pewien podzbiór plików, którym zarządza DFS. **Jednostka składowa** (*component unit*), to najmniejszy zbiór plików, który można przechowywać w jednej maszynie niezależnie od innych jednostek.
- Wyzwania:
  - Nazewnictwo i przezroczystość
  - Zdalny dostęp do plików

# Podstawowe terminy DFS

- **Usługa (*service*)** – jednostka oprogramowania działająca na jednej lub wielu maszynach, wypełniająca dla nieznanego uprzednio klienta funkcję konkretnego typu.
- **Serwer (*server*)** – oprogramowanie usługowe wykonywane na jednej maszynie.
- **Klient (*client*)** – proces, który zamawia usługę za pomocą zbioru operacji tworzących **interfejs klienta** (create, delete, read, write)
- System plików dostarcza więc klientom **usług plikowych**.
- Interfejs DFS klienta powinien być **przezroczysty**, tzn. nie powinno dać się odróżnić plików lokalnych od zdalnych.

# Nazewnictwo i przezroczystość

- **Nazewnictwo** – odwzorowanie między obiektami logicznymi i fizycznymi. Odwzorowanie to ma charakter wielopoziomowy. Użytkownik na ogół odwołuje się do pliku za pomocą nazwy tekstowej, która jest odwzorowywana na identyfikator liczbowy niższego poziomu, a ten z kolei – na bloki dyskowe

- **Przezroczysty** DFS ukrywa miejsce przechowywania pliku.

- **Zwielokrotnianie pliku** (*file replication*) oznacza przechowywanie pliku w wielu lokalizacjach. Dla danej nazwy pliku wartością odwzorowania jest zbiór umiejscowień kopii tego pliku. W przezroczystym DFS ukrywa się zarówno istnienie wielu kopii pliku, jak i ich położenie.

# Struktury nazewnicze

Należy rozróżnić dwa powiązane ze sobą pojęcia dotyczące odwzorowywania nazw w DFS:

- **Przezroczystość położenia (*location transparency*)** – nazwa pliku nie daje jakiejkolwiek wskazówki co do fizycznego miejsca przechowywania pliku.

- **Niezależność położenia (*location independence*)** – nazwy pliku nie trzeba zmieniać, gdy plik zmienia swoje fizyczne umiejscowienie.

# Podejścia do schematów nazewniczych

1. Pliki są nazywane za pomocą pewnych kombinacji nazw przechowujących je komputerów i nazw lokalnych, co gwarantuje jednoznaczność nazw w obrębie całego systemu.
2. Katalogi zdalne są montowane do katalogów lokalnych, co daje wrażenie spójnego drzewa katalogów.
3. Pełna integracja składowych systemu plików
   - Jedna globalna struktura nazw obejmująca wszystkie pliki w systemie
   - Jeżeli serwer przestaje działać, to pewien (przypadkowy z punktu widzenia użytkownika) zbiór katalogów na różnych maszynach staje się niedostępny. Wskutek tego w komputerze jednego klienta użytkownik może mieć do zdalnego drzewa katalogów, podczas gdy w innym komputerze temu samemu użytkownikowi dostępu tego się zabrania.

# Zdalny dostęp do pliku

- Po znalezieniu lokalizacji za pomocą odpowiedniego schematu nazewniczego dostęp do pliku jest możliwy np. za pomocą **mechanizmu obsługi zdalnej** (***remote-service mechanism***). Zamówienia na dostępy są kierowane do serwera; maszyna serwera wykonuje odpowiednie operacje, a ich wyniki są przekazywane do użytkownika.
- Najczęściej obsługa zdalna jest realizowana w postaci zdalnego wywołania procedury (RPC).
- Dla poprawy wydajności obsługi zdalnej stosuje się przechowywanie podręczne (*caching*), zmniejszające liczby operacji dyskowych i zmniejszające ruch w sieci.
  - Jeśli potrzebnego pliku (czy jego części) nie ma w pamięci podręcznej dane są kopiowane z serwera
  - Dostęp do pliku jest realizowany na kopii w pamięci podręcznej
  - **Problem spójności pamięci podręcznej (cache-consistency problem)** – zachowywanie zgodności kopii lokalnych z kopią główną (*master file*)

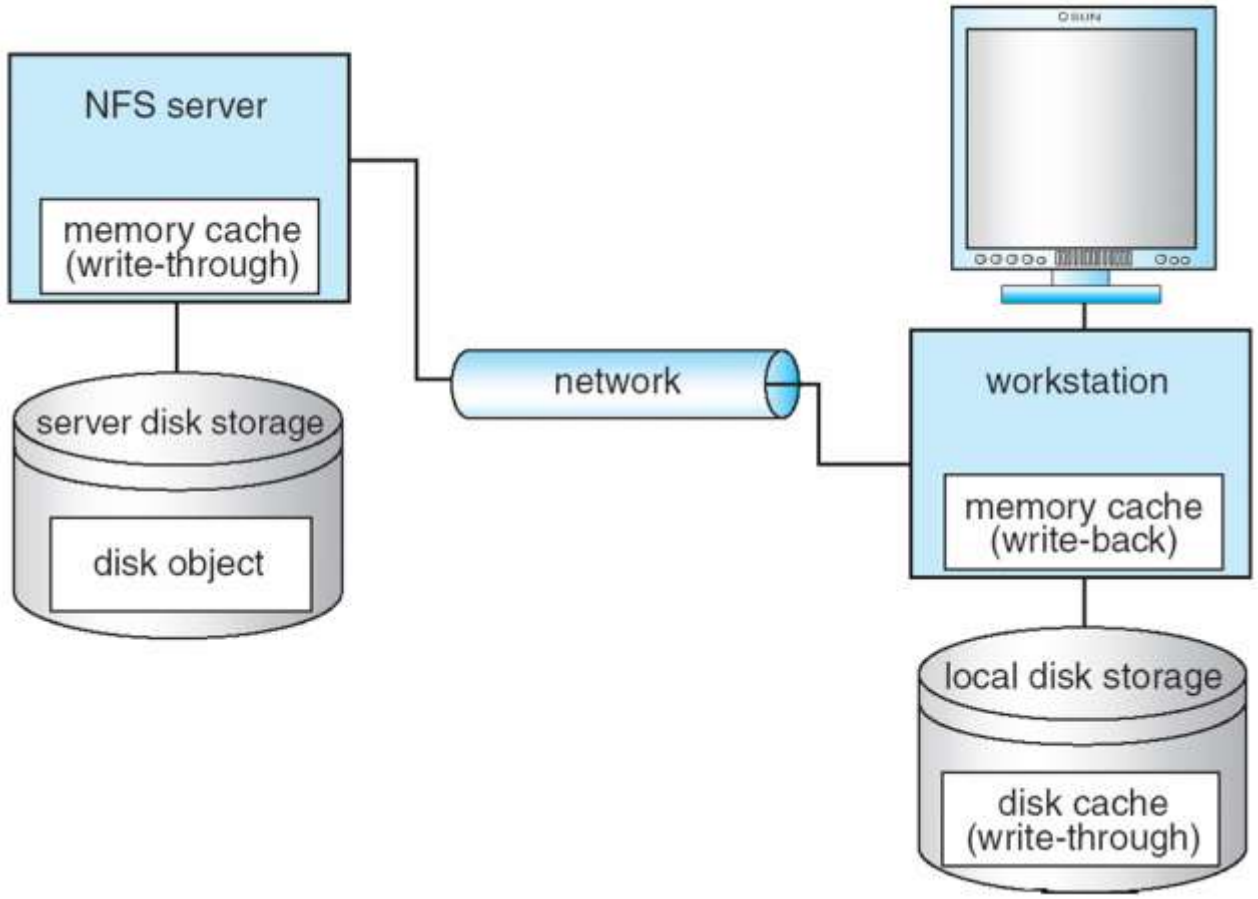
# Umiejscowienie pamięci podręcznej

- Zalety lokalizacji pamięci podręcznej na dysku
  - Pamięć nieulotna -> bardziej niezawodna
  - Dane przechowywane na dysku mogą być użyte po restarcie systemu bez potrzeby ponownego pobrania z serwera.
- Zalety lokalizacji w pamięci głównej:
  - Można zrealizować bezdyskową stację roboczą
  - Szybki dostęp
  - Coraz lepszy relacje pojemność-szybkość-cena.
  - Pamięci podręczne serwerów, wykorzystywane do przyspieszania dyskowych operacji wejścia/wyjścia, są w pamięci głównej. Lokalizacja pamięci podręcznej DFS w pamięci głównej klienta pozwala na użycie jednolitego sposobu przechowywania kopii podręcznych po stronie serwera i klienta.

# Polityka uaktualniania pamięci podręcznej

- **Zasada przepisywalności (*write-through*)** – przepisywanie danych na dysk zaraz po tym, gdy znajdą się w dowolnej pamięci podręcznej
  - Niezawodność kosztem wydajności (oczekiwanie na przesłanie danych do serwera sprawia, że pamięć podręczna przyspiesza dostęp jedynie przy odczycie)
- **Opóźnianie zapisywania (*delayed-write, write-back*)** – zmiany wprowadzone do pamięci podręcznej są przepisywane do serwera w późniejszym czasie.
  - Operacje zapisu wykonują się szybko. Dane nadpisane nie muszą być przesyłane wielokrotnie do serwera. Dobre rozwiązanie dla plików otwartych przez dłuższy okres czasu i często modyfikowanych
  - Marna niezawodność; niezapisane dane są tracone przy awarii komputera.
  - Warianty
    - Okresowe przeglądy pamięci podręcznej + przesyłanie do serwera bloków zmodyfikowanych od ostatniego przeglądu
    - **Pisanie przy zamykaniu (*write-on-close*)**, modyfikacje pliku są zapisywane na serwerze przy zamykaniu dostępu do pliku (n.p. AFS)

# System plików cachefs i pamięci podręczne



# Spójność

- Maszyna klienta musi rozstrzygać, czy lokalna kopia danych przechowywana w pamięci podręcznej jest spójna z kopią główną (więc może być używana), czy też nie.
- Metody weryfikacji aktualności danych:
  - **Podejście polegające na inicjatywie klienta**
    - Klient inicjuje sprawdzenie ważności danych.
    - Najistotniejsza jest w tym podejściu częstość sprawdzania ważności; to ona przesądza o wynikowej semantyce spójności.
  - **Podejście polegające na inicjatywie serwera**
    - Serwer zapamiętuje dla każdego klienta które pliki czy fragmenty zostały skopiowane oraz jaki jest zamierzony tryb użycia.
    - Po wykryciu, że plik otwarto równocześnie w konfliktowych trybach, wtedy zakaże przechowywania podręcznego w odniesieniu do tego pliku. Zakaz powoduje de facto przejście do trybu obsługi zdalnej.

# Porównanie przechowywania podręcznego i obsługi zdalnej

Wybór pomiędzy przechowywaniem podręcznym a obsługą zdalną oznacza znalezienie kompromisu pomiędzy potencjalnie lepszą wydajnością a zmniejszoną prostotą.

- Znaczna liczba zdalnych dostępów może być skutecznie obsłużona za pomocą lokalnej pamięci podręcznej. Obsługa zdalna takich dostępów oznacza duży ruch sieciowy i obciążenie serwera (zła skalowalność).
- Całkowity koszt przesyłania siecią dużych porcji danych jest mniejszy przy stosowaniu pamięci podręcznych niż przy przesyłaniu serii odpowiedzi na poszczególne zamówienia.
- Problem spójności pamięci podręcznej jest główną wadą przechowywania podręcznego. Używanie pamięci podręcznej ma przewagę, gdy dostępy do pisania zdarzają się rzadko.
- Aby przechowywanie podręczne przynosiło korzyść – maszyny klienckie powinny mieć szybkie dyski lokalne i/lub dużą pamięć główną.

# Doglądana obsługa pliku

- Mechanizm doglądanej obsługi pliku (*stateful file service*)
  - Klient otwiera dostęp do pliku
  - Serwer pobiera do pamięci informację o pliku dyskowym; zwraca klientowi identyfikator, unikalny dla pary klient-otwarty plik.
  - Identyfikator jest wykorzystywany przez klienta do kolejnych dostępów do otwartego pliku; identyfikator traci ważność przy kończeniu sesji (plikowej).
  - Serwer musi odzyskiwać obszary pamięci głównej wykorzystywane na obsługę zakończonych sesji plikowych.

- Zalety:
  - Mała liczba dostępów do dysku
  - Serwer może wiedzieć, że plik został otwarty do dostępu sekwencyjnego i czytać bloki dyskowe z wyprzedzeniem.

# Niedoglądana obsługa plików

Niedoglądana obsługa plików (*stateless file service*)

- Unika przechowywania stanu, traktując każde zamówienie jako samowystarczalne (każde zamówienie w pełni identyfikuje plik i miejsce w pliku).
- Nie ma potrzeby otwierania i zamykania sesji plikowej.
- Operacje obsługi plików są bezstanowe.

# Skutki awarii serwera

- Serwer usługi doglądanej traci informacje o stanie przechowywane w pamięci ulotnej.
  - Stan może być odtwarzany przez protokół rekonstrukcji oparty na dialogu z klientami. Mniej gładka rekonstrukcja polega na zaniechaniu operacji, które były wykonywane w chwili awarii.
- Dla serwera usługi niedoglądanej skutki awarii są prawie niezauważalne.
  - Proces klienta nie odróżnia powolnego serwera od serwera zrekonstruowanego. Klient powtarza zamówienie, jeśli nie otrzymał odpowiedzi.
- Koszty serwera usługi niedoglądanej:
  - Dłuższe komunikaty zamawiania operacji
  - Dłuższe przetwarzanie zamówień
  - Dodatkowe ograniczenia DFS (brak adresowania przyrostowego, by operacje były idempotentne)
- W pewnych środowiskach obsługa doglądana jest koniecznością
  - Serwer sprawdzający spójność nie może realizować obsługi niedoglądanej
  - UNIX owe sesje plikowe sugerują obsługę doglądaną

# Zwielokrotnianie pliku

- Repliki (kopie) tego samego pliku są umieszczone w maszynach, które są niezależne od siebie w wypadku awarii.
- Poprawa dostępności i czasu obsługi.
- Istnienie kopii nie powinno być widoczne dla użytkownika
- Uaktualnienie dowolnej kopii musi znaleźć odzwierciedlenie we wszystkich pozostałych kopiach – chyba że spójność może być poświęcona na rzecz dostępności i wydajności.

# Deduplikacja plików

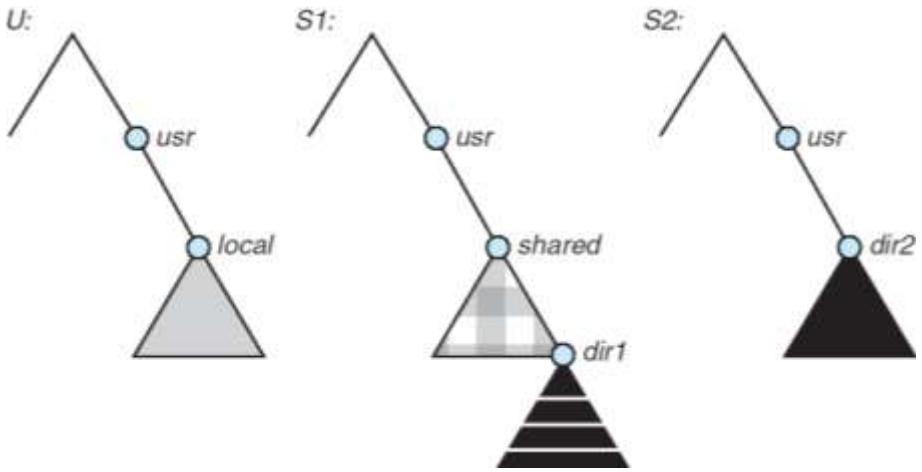
- Funkcja deduplikacji danych polega na wyszukiwaniu i usuwaniu zduplikowanych danych bez ich zmiany, zachowując ich poprawność i integralność.
- Głównym celem tej funkcji jest przechowywanie większej ilości danych na tych samych, a nawet mniejszych powierzchniach dyskowych.
- Jest to możliwe dzięki: segmentacji plików na mniejsze kawałki (wielkości 32-128 KB), identyfikacji zduplikowanych kawałków oraz zachowania jednej kopii każdego elementu. Kopie fragmentów zostaną zastąpione przez odniesienie do ich pojedynczych kopii . Większe kawałki są kompresowane i organizowane w specjalnych kontenerach plikowych.
- W przypadku rozproszonych systemów plików deduplikacja może być realizowana po stronie
  - Klienta. Zduplikowane dane nie są przesyłane do serwera.
  - Serwera – jako funkcjonalność lokalnych systemów plików.

# Sieciowy system plików NFS

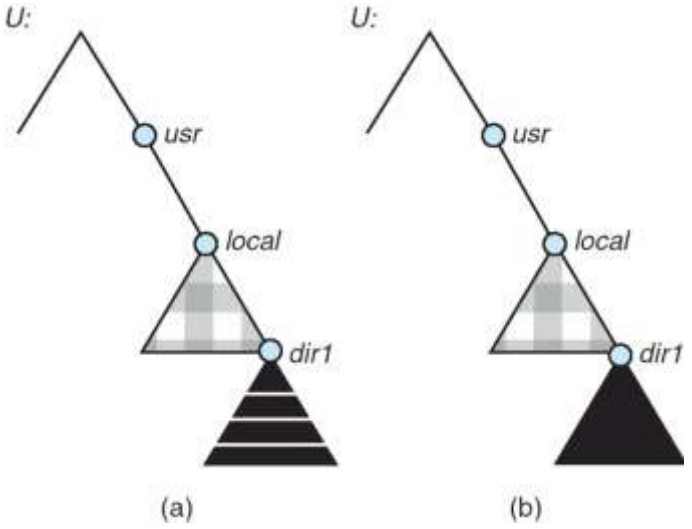
- NFS jest standardem dostępu sieciowego do plików w sieciach LAN.
- Stacje robocze połączone siecią mają niezależne systemy plików. NFS umożliwia udostępnianie wybranych plików jednego komputera innym w sposób niewidoczny (*transparent*) dla użytkownika.
  - Zdalny katalog jest montowany w lokalnym systemie plików. W ten sposób zamontowany katalog wygląda jakby znajdował się na lokalnym komputerze.
  - Określanie który zdalny katalog którego serwera ma być domontowany lokalnie zależy od administratora systemu lokalnego
- NFS zaprojektowano do pracy w środowisku heterogenicznym (różne architektury komputerów, systemy operacyjne, architektury sieci). Cel ten osiągnięto przez wykorzystaniu techniki zdalnego wykonywania procedur (RPC) w połączeniu ze standardem reprezentacji danych: External Data Representation (XDR).
- Specyfikacji zdalnego dostępu do pliku towarzyszy specyfikacja montowania zdalnego katalogu. Komputer udostępniający pliki przechowuje ich położenie na liście eksportowej – utworzonej przez administratora tego komputera.

# Przykład

Trzy niezależne systemy plików na serwerach: U, S1, S2

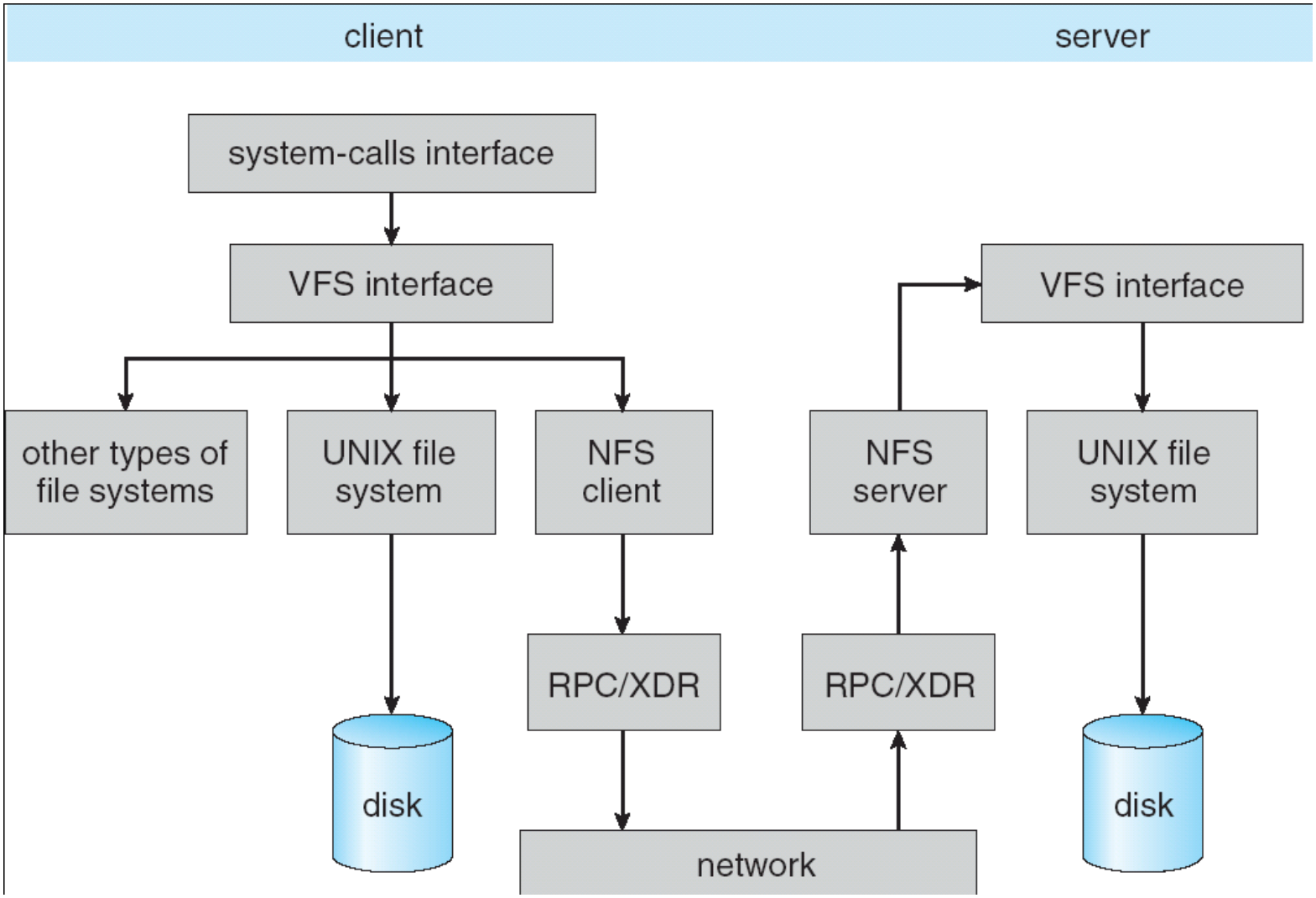



a) b)




Przykłady: a) montażu katalogu S1:/usr/shared/dir1 pod U:/usr/local
b) montażu kaskadowego katalogu S2:/usr/shared/dir2

# Architektura NFS



# NFS – zdalne operacje plikowe

- Tłumaczenie nazw ścieżkowych:
  - Podział ścieżki na segmenty i oddzielne poszukiwania (NFS lookup) dla każdej pary: węzeł (vnode) katalogu - nazwa.
  - Dla przyspieszenia tłumaczenia klient buforuje węzły (vnode) katalogów zdalnych To make lookup faster, a directory name lookup cache on the client's side holds the vnodes for remote directory names
- Występuje odpowiedniość podstawowych funkcji operacji zdalnych NFS (RPC calls) oraz funkcji systemowych UNIXa (za wyjątkiem zamykania i otwierania plików – w NFS v.3).
- NFS wykorzystuje paradygmat zdalnych operacji plikowych, ale wykorzystuje buforowanie (bloków dyskowych i atrybutów plików) ze względu na wydajność:
- Strona klienta nie zwalnia bloków zapisywanych z opóźnieniem – aż serwer potwierdzi zapis na dysk.