RSO część II (1. kwiś)

PROCESY I WĄTKI W SYSTEMACH ROZPROSZONYCH

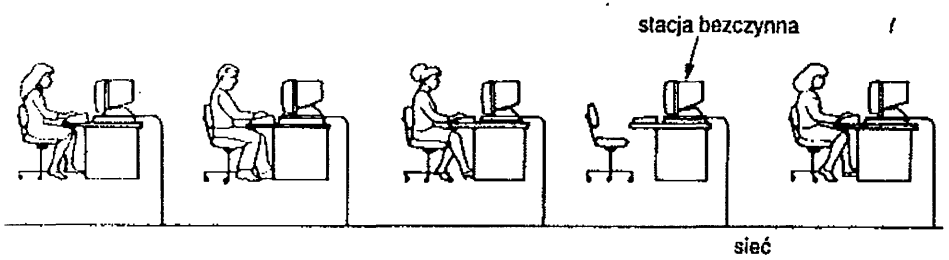
MODELE SYSTEMÓW

Różne sposoby organizacji systemów

2,50

Model stacji roboczych

Wiele stacji połączonych siecią LAN



Rys. 4.10. Sieć osobistych stacji roboczych wyposażonych w lokalne systemy plików

Stacje z prywatnymi dyskami, a bezdyskowe.
Bezdyskowe - systemy plików realizowane na zdalnych serwerach.

Zalety stacji bezdyskowych
niskie koszty,
łatwość eksploatacji,
symetria wykorzystania,
niski hałas, ...

Jak zostawić system rozproszony mając do dyspozycji wiele maszyn

Sposoby wykorzystania prywatnych dysków stacji roboczych

Stronicowanie i przechowywanie plików tymczasowych
pliki tymczasowe, tworzone w czasie sesji np. w trakcie kompilacji, nie muszą być przesyłane do serwera plików

Stronicowanie i przechowywanie plików tymczasowych, oraz systemowych plików binarnych
na dyskach lokalnych przechowuje się dodatkowo najczęściej wykorzystywane binaria - kompilatorów, edytorów tekstu, programy obsługi

Stronicowanie i przechowywanie plików tymczasowych, systemowych plików binarnych, oraz podręczna pamięć plików
w czasie sesji użytkownik ściąga potrzebne pliki z serwera na dysk lokalny, pracuje wykorzystując dysk lokalny, odsyła ostateczne wersje plików do serwera przed zakończeniem sesji.
Zalety:
redukcja obciążenia sieci,
utrzymuje się zcentralizowaną pamięć długoterminową.
Wady:
Problem utrzymania spójności pamięci podręcznych stacji roboczych

Kompletny lokalny system plików
Każda maszyna ma własny system plików z możliwością montowania systemów plików innych maszyn.
Zalety:
gwarantowany czas odpowiedzi,
małe obciążenie sieci
Wady:
utrudnione dzielenie informacji,
realizuje idee operacyjne system sieciowy, a nie przezroczystego systemu rozproszonego.

Pakiety wsparcia

Pakiet wsparcia - zbiór elementarnych działań - wywołań bibliotecznych dostępnych dla programistów

Przykłady działań elementarnych

tworzenie wątku
likwidacja wątku
oczekiwanie na zakończenie innego wątku

Wątek zamyka zamek aby wejść do sekcji krytycznej w której sprawdza strukturę danych, jeśli zasób jest wolny, to oznacza zasób jako zajęty i proces, który chciał na ten zasób jest budzony i może z niego korzystać. Są to działania niepodzielne. Zmienna warunkowa otwiera zasoby aby inne procesy z nich nie korzystały

System Mach

- Założenia i cele projektu Mach
- Charakterystyka planowania w systemie Mach

Dokonywanie uzgodnień w systemach rozproszonych

Przykłady uzgodnień:
wybór koordynatora, zatwierdzenie transakcji, synchronizacja, ...

Problem
- doprowadzić do uzgodnienia mimo, że pewne procesy są wadliwe,
- wykonać procedurę uzgodnienia w skończonej liczbie kroków.

Rozwiązanie zależy od
- niezawodności dostarczania komunikatów,
- czy awarie procesorów mają charakter wad wyciszających czy wad bizantyjskich,
- czy system jest synchroniczny - czy zapewnia odpowiedź w założonym, skończonym czasie, czy nie.

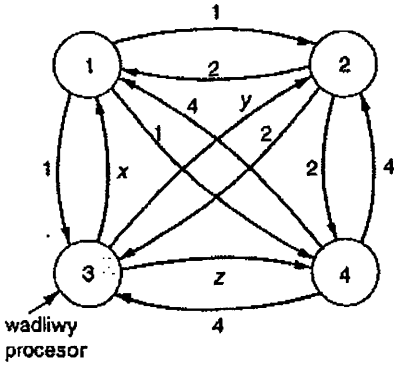
Niepewna komunikacja

Przykład uzgadniania wspólnego ataku dwóch oddalonych oddziałów armii. Wykorzystanie posłańców do przenoszenia komunikatów.

Niezawodna komunikacja, wadliwe niektóre procesy

tzw. problem bizantyjskich generałów
n - dowódców, wymieniając między sobą komunikaty, próbuje uzyskać informacje o łącznej sile całej armii,
m - spośród nich jest zdrajcami - udzielają nieprawidłowych informacji,
znaleźć algorytm zapewniający, że zdrajcy nie zakłócą przesyłania informacji między lojalnymi dowódcami.

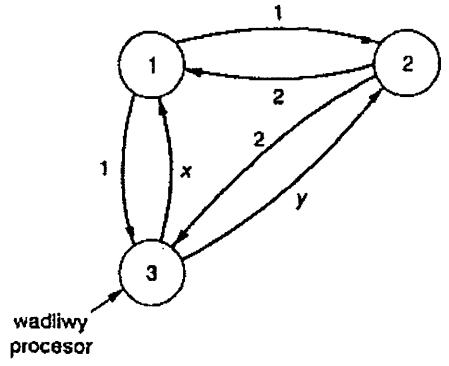
Algorytm Lamporta



1 otrzymał (1,2,x,4)	1 otrzymał	2 otrzymał	4 otrzym
2 otrzymał (1,2,y,4)	(1, 2, y, 4)	(1, 2, x, 4)	(1, 2, x,
3 otrzymał (1,2,3,4)	(a, b, c, d)	(e, f, g, h)	(1, 2, y,
4 otrzymał (1,2,z,4)	(1, 2, z, 4)	(1, 2, z, 4)	(i, j, k,

(a) (b) (c)

Rys. 4.23. Problem bizantyjskich generałów dla trzech lojalnych generałów i jedne zdrajcy. (a) Generałowie meldują o sile swoich oddziałów (w jednostkach 1KB). (b) We tory zebrane przez każdego z generałów na podstawie danych z (a). (c) Wektory otrzyma przez generałów w kroku 3.



1 otrzymał (1,2,x)	1 otrzymał	2 otrzymał
2 otrzymał (1,2,y)	(1, 2, y)	(1, 2, x)
3 otrzymał (1,2,3)	(a, b, c)	(d, e, f)

(a) (b) (c)

Rys. 4.24. To samo, co na rys. 4.23, z tym że teraz lojalnych jest dwóch generałów

Wykorzystanie bezczynnych stacji

Ogólny problem zdalnego wykonywania procesów w sposób przezroczysty.

Pierwsza próba - UNIX BSD
`rsh maszyna polecenie`
wady: trzeba określić maszynę, środowisko zdalne na ogół inne niż lokalne.

Problemy
znalezienie bezczynnej maszyny
zapewnienie przezroczystości wykonania
czynności po powrocie właściciela

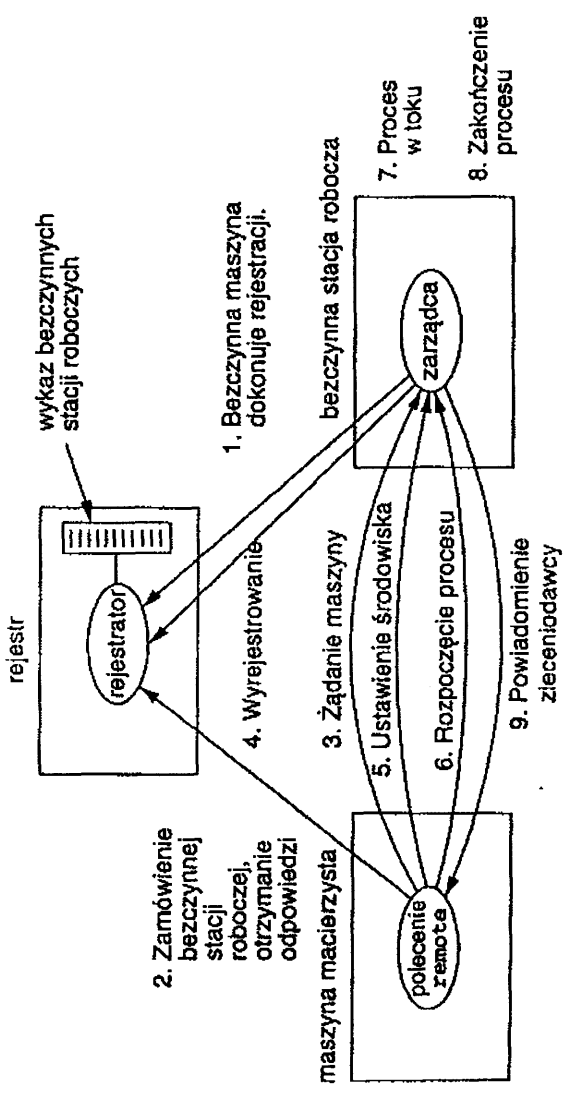
Znalezienie bezczynnej stacji

Definicja bezczynności stacji.

Algorytm lokalizacji bezczynnej stacji sterowany za pomocą serwera

Stacja robocza
- stwierdza swoją bezczynność
- ogłasza swoją dostępność - niezbędne informacje (dane stacji) są wpisywane do pliku rejestracyjnego

Użytkownik
- wykonuje: `remote polecenie`
- program `remote` sprawdza rejestr



Rys. 4.12. Algorytm znajdywania i zatrudniania bezczynnej stacji roboczej, oparty na rejestrowaniu

Algorytm sterowany przez klienta

Program remote rozgłasza zamówienie, podaje jako informacje:
- program, który potrzebuje stację,
- wielkość potrzebnej pamięci,
- zapotrzebowanie na obliczenia, ...

Po nadejściu odpowiedzi program remote wybiera stację.

Wykonanie zdalne procesu:
- przemieszczenie kodu,
- zapewnienie tego samego środowiska
  potrzebny ten sam obraz plików
  ten sam katalog roboczy
  te same zmienne środowiska.

Problemy działania jądra systemu

Odwołania do systemu plików np. operacja read:
system bezdyskowy - zamówienie do serwera plików
dyski lokalne z kompletnymi systemami plików - do stacji macierzystej.

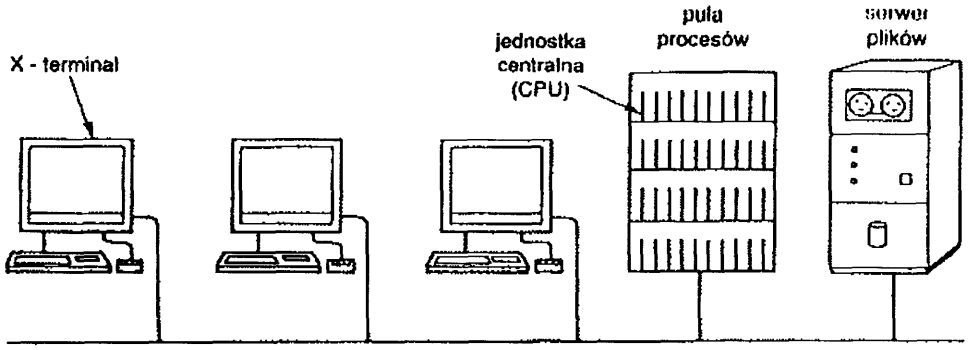
Odwołania dot. klawiatury i monitora:
przesyłane do stacji macierzystej.

Inne odwołania, np. dot. priorytetu, segmentu danych, nazwy maszyny, adresu sieciowego itp.:
wykonywane zdalnie.

Problemy synchronizacji czasu.

Model pu[illegible] procesów

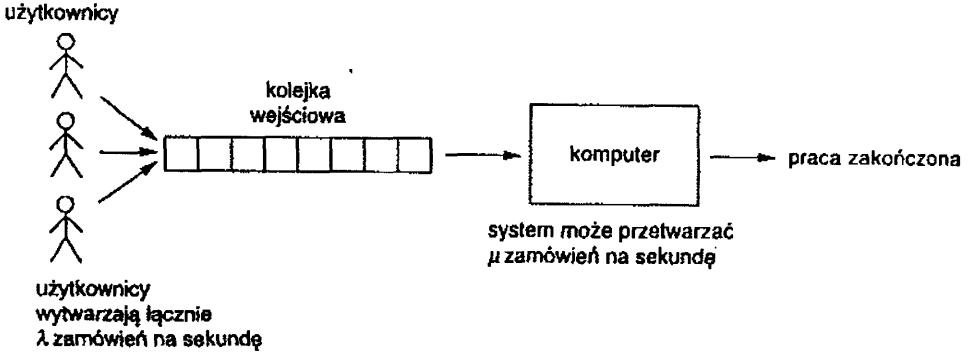
Wiele jednostek centralnych w jednej szafie
Użytkownicy mają szybkie terminale graficzne.



Rys. 4.13. System oparty na modelu puli procesorów

Zalety:
redukcja kosztów - wspólny system zasilania, obudowa, ...
łatwość powiększania mocy obliczeniowej,
możliwość udostępnienia użytkownikowi tylu procesorów, ile potrzebuje.

System masowej obsługi
użytkownicy generują losowo zamówienia,
zamówienia ustawiane są w kolejce do obsługi.



Rys. 4.14. Elementarny system masowej obsługi

Powrót właściciela do stacji

Jakie podjąć działania

- Żadne - stacja przestaje być osobista.
- Zlikwidować proces zewnętrzny
  - nagle - utrata całej pracy, system w chaosie
  - ostrzec proces, aby mógł sam się zamknąć
  - przenieść proces na inną maszynę
    tzn. kod i dane użytkownika, jądrowe struktury danych
  - oczyścić maszynę źródłową
    kończący proces musi zostawić maszynę w stanie, w jakim ją zastał

NIEZAWODNOŚĆ - TOLEROWANIE AWARII W SYSTEMACH ROZPROSZONYCH

Systemy komputerowe zawodzą z powodu wad elementów składowych

wada (fault) - niewłaściwe działanie elementu,
które może wynikać z różnych powodów: błędu projektanta, błędu w produkcji, błędu w programie, . . .

Klasyfikacja wad

Wady przejściowe (transient faults)
Pojawiają się i znikają. Przy powtórzeniu operacji wada zwykle się już nie pojawia.

Wady nieciągłe (intermitted faults)
Wielokrotnie pojawiają się i znikają w sposób przypadkowy.

Wady trwałe (permanent faults)
Po pojawieniu się nie ustępują, aż uszkodzony element zostanie naprawiony.

Cel projektowania i budowy systemu tolerującego awarie: uzyskanie pewności, że system będzie działał nawet w przypadku obecności wad.

Tradycyjne badania tolerowania uszkodzeń - analiza statystyczna wad elementów elektronicznych.

Awarie w systemie rozproszonym

W systemie rozproszonym jest wiele elementów składowych.

Niewłaściwe działanie procesora może być spowodowane zarówno fizyczną wadą produkcyjną, uszkodzeniem, błędem programu.

Tolerowanie awarii przez system rozproszony polega bardziej na takiej jego budowie, aby **mógł przetrwać uszkodzenia elementów** składowych (zwłaszcza procesorów), niż na całkowitym wyeliminowaniu prawdopodobieństwa wystąpienia wad.

Formy uszkodzeń

Uszkodzenie wyciszające (fail- silent fault)
Procesor się zatrzymuje i nie odpowiada. Następuje **wadliwe zatrzymanie** (fail-stop fault).

Wada bizantyjska (Byzantine fault)
Procesor po wystąpieniu takiej wady dalej działa, ale błędnie odpowiada na pytania i niewłaściwie współpracuje z innymi. Stwarza wrażenie poprawnej pracy.

Redundancja

Rozproszone systemy tolerujące awarie buduje się wykorzystując redundancję.

Redundancja informacji.
Przesyłanie dodatkowych bitów informacji, umożliwiających odtworzenie zniekształconych bitów.
Kod Hamminga stosowany w transmisji.

Redundancja czasu.
Wykonanie operacji, a jeśli wykonana błędnie, powtórzenie jej wykonania.
Przykład - użycie transakcji niepodzielnych.

Redundancja fizyczna.
Specjalna budowa, dodatkowe wyposażenie, zwielokrotnienie elementów składowych, aby system działał mimo awarii niektórych elementów.
Sposoby realizacji:

- aktywne zwielokrotnienie,
- zasoby rezerwowe.

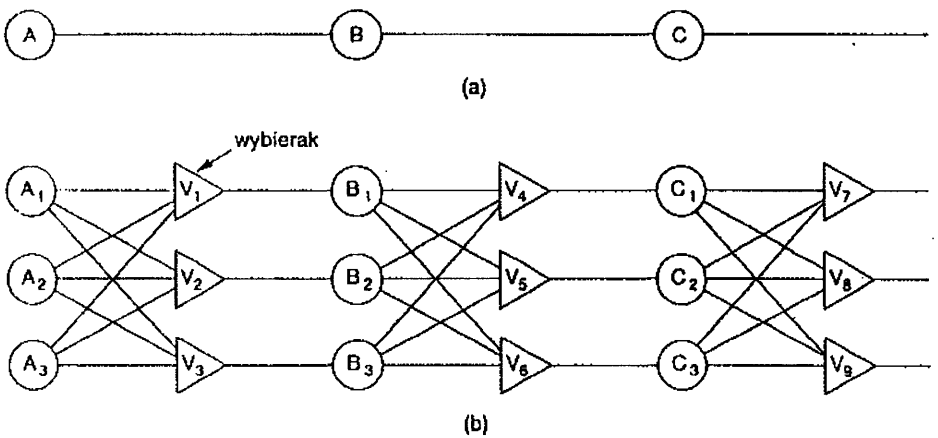
Zagadnienia analizy projektowej:

- wymagany stopień zwielokrotnienia,
- działanie systemu, gdy nie ma uszkodzeń - średnie i najgorsze,
- działanie systemu, gdy uszkodzenia występują - średnie i najgorsze.

Aktywne zwielokrotnienie (active replication)

Zwielokrotnienie elementów działających równolegle.
Podejście autonomiczne (state machine approach).

Przykład zwielokrotnienia urządzenia
Technika TMR (Triple Modular Redundancy).



Rys. 4.21. Potrójna redundancja modularna

Zagadnienia zwielokrotnienia serwerów w systemach rozproszonych

Serwer - maszyna skończenie stanowa: przyjmuje zamówienia i generuje odpowiedzi.
Zamówienia od klienta wysyłane do wielu serwerów. Jeśli zostaną odebrane i przetworzone w tym samym porządku, to po przetworzeniu wszystkie sprawne serwery będą w tym samym stanie i wygenerują te same odpowiedzi. Wyniki można połączyć, aby wyeliminować uszkodzenia.

Jakie zwielokrotnienie

Odpowiedź zależy od założenia projektowego stopnia odporności systemu na uszkodzenia.

Def.
System tolerujący k uszkodzeń (k-fault tolerance)
jest to system, który przetrwa uszkodzenia k elementów i będzie działał właściwie.

Problem **niepodzielnego rozgłaszania** -
wymaganie, aby wszystkie zamówienia dochodziły do serwerów w tej samej kolejności.

Zagadnienie **przetwarzana zamówień w tej samej kolejności** na wszystkich serwerach

- globalne ponumerowanie - zastosowanie globalnego serwera numerów,

- logiczne zegary Lamporta - każdy komunikat ma znacznik czasu, przetwarzanie w serwerach zgodnie ze znacznikami

Zasoby rezerwowe

Aktywnie wykorzystywane są zasoby podstawowe (serwer podstawowy). W przypadku awarii, funkcje uszkodzonego zasobu (serwera) przejmuje zasób (serwer) rezerwowy.

Zalety

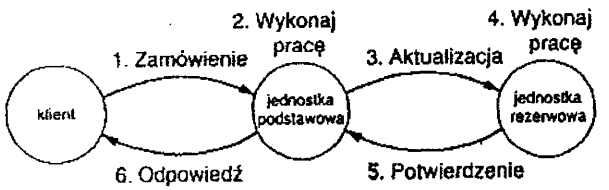
- prostsza realizacja - komunikaty są przesyłane tylko do jednego serwera, nie trzeba ich porządkować,
- potrzeba mniej maszyn niż w przypadku aktywnego zwielokrotnienia

Wady

- mała odporność na wady bizantyjskie
- czasochłonne, złożone przywracanie serwera podstawowego do pracy

Przykład realizacji

Protokół operacji zapisu



Rozwiązanie bardziej zaawansowane

Wspólny dysk dla jednostki podstawowej i rezerwowej z oddzielnymi partycjami. Zamówienia i wyniki zapisywane są na dysku.

Przykład zastosowania redundancji

Multi Computer Service Guard firmy Hewlett Packard

System odporny na (tolerujący) awarie sprzętu i oprogramowania, przeznaczony dla aplikacji wymagających wysokiej niezawodności (mission critical applications).

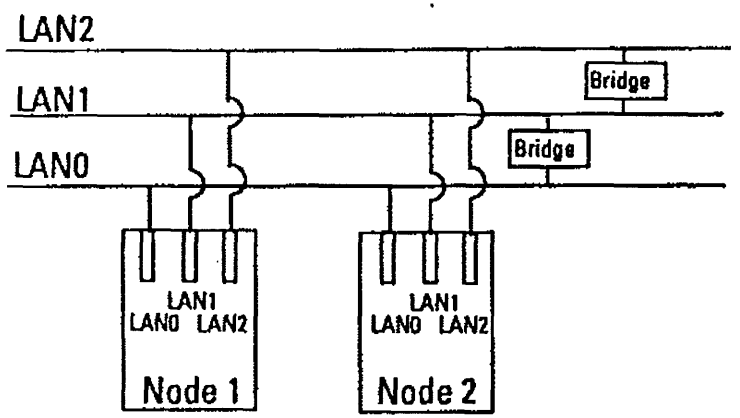
System rozproszony, składający się z kilku węzłów zorganizowanych jako klaster (claster).
Węzłami mogą być systemy jedno lub wieloprocesorowe.
Węzły w klasterze mają wspólny dostęp do dysków z wykorzystaniem szyny (bus).
Połączone są również przez sieć LAN wykorzystywaną do

- przesyłania informacji związanych z wykonywaniem aplikacji (dostęp klientów),
- przesyłania sygnałów monitorujących pracę węzłów (heardbeat)

MC Service Guard monitoruje prawidłowość działania (stanu) różnych elementów składowych systemu, w przypadku wykrycia wad podejmuje działanie - eliminuje skutki wad, ewentualnie pozwala zminimalizować czas przerwy.
Wykrywa i reaguje na wady: jednostki centralne, pamięci systemowe, LAN, interfejsy sieciowe, procesy aplikacyjne i systemowe.

Zasoby klastra (wszystkie zasoby niezbędne dowykonania określonych usług aplikacyjnych - dyski, zasoby sieciowe, procesy aplikacyjne i systemowe) organizowane są jako tzw.
pakiety aplikacyjne (application packages).
Pakiety te stanowią jednostki zarządzane w ramach klastra.

Figure 6.4 Fully Redundant, Data Intense LAN Configuration



- LAN0 carries heartbeat
- LAN2 carries data
- LAN1 is an idle standby, available for use by LAN0 or LAN2

Figure 6.6 Root/Swap Disk Configuration Examples

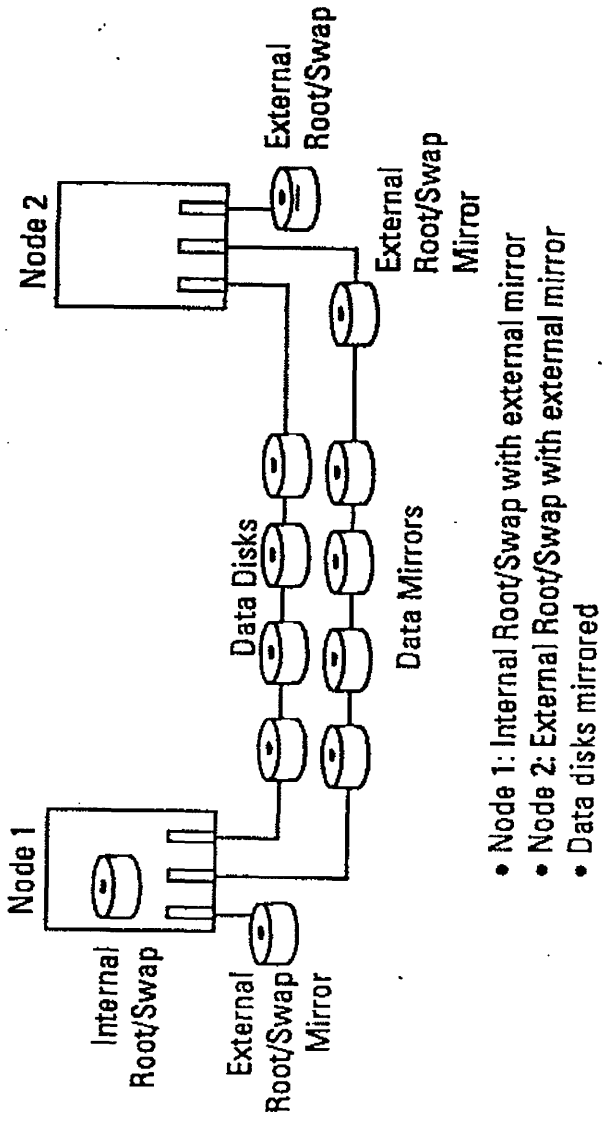
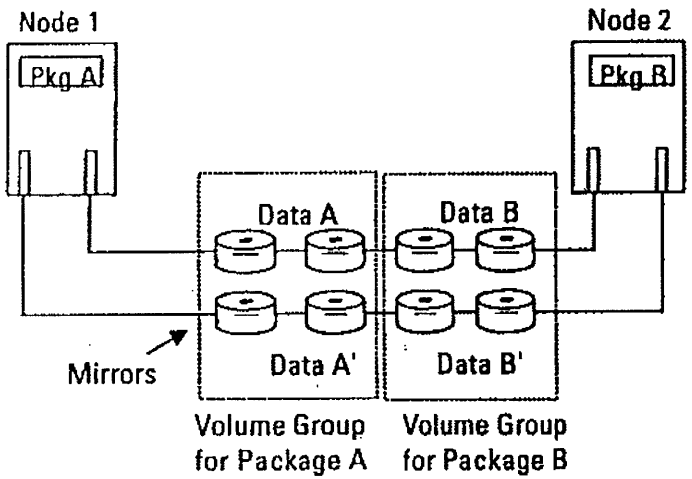
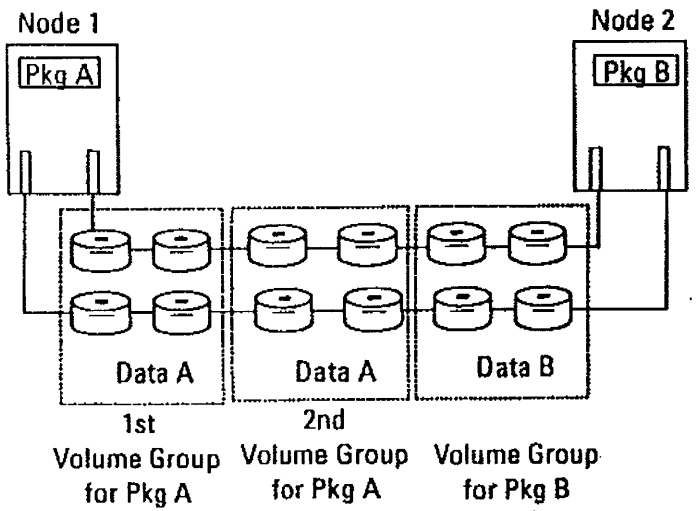


Figure 6.7 Basic Disk Configuration



- Data disks attached to shared F/W SCSI bus
- Volume groups have 4 physical drives
- Data disks mirrored

Figure 6.8 Package with More than One Volume Group



W MC/Service Guard stosuje się redundancję w zakresie:
systemów komputerowych tworzących węzły,
linii sieci,
interfejsów sieciowych,
dysków: root, swap, data
szyn (bus).

Każdy system (węzeł klastra) wykonuje określone aplikacje, ale w przypadku awarii jednego z nich - inny przejmuje wykonanie - kontynuacje zadania.

Korzystając z MC/Service Guard można tworzyć pełne środowisko wykonywania aplikacji odporne na uszkodzenia.

Zaleca się stosowanie, razem z MC/Service Guard:
Mirr Disk /UX
RAID
Power Trust Uninterraptible Power Suplies (UPS),
HP Process Resource Manager
HP Open View Admin Center
HP Open View Operation Center