NIEZAWODNOSC - 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 naprawlony.

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. I.
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).

Wady bizantyjskie (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.
Przykłady:

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.

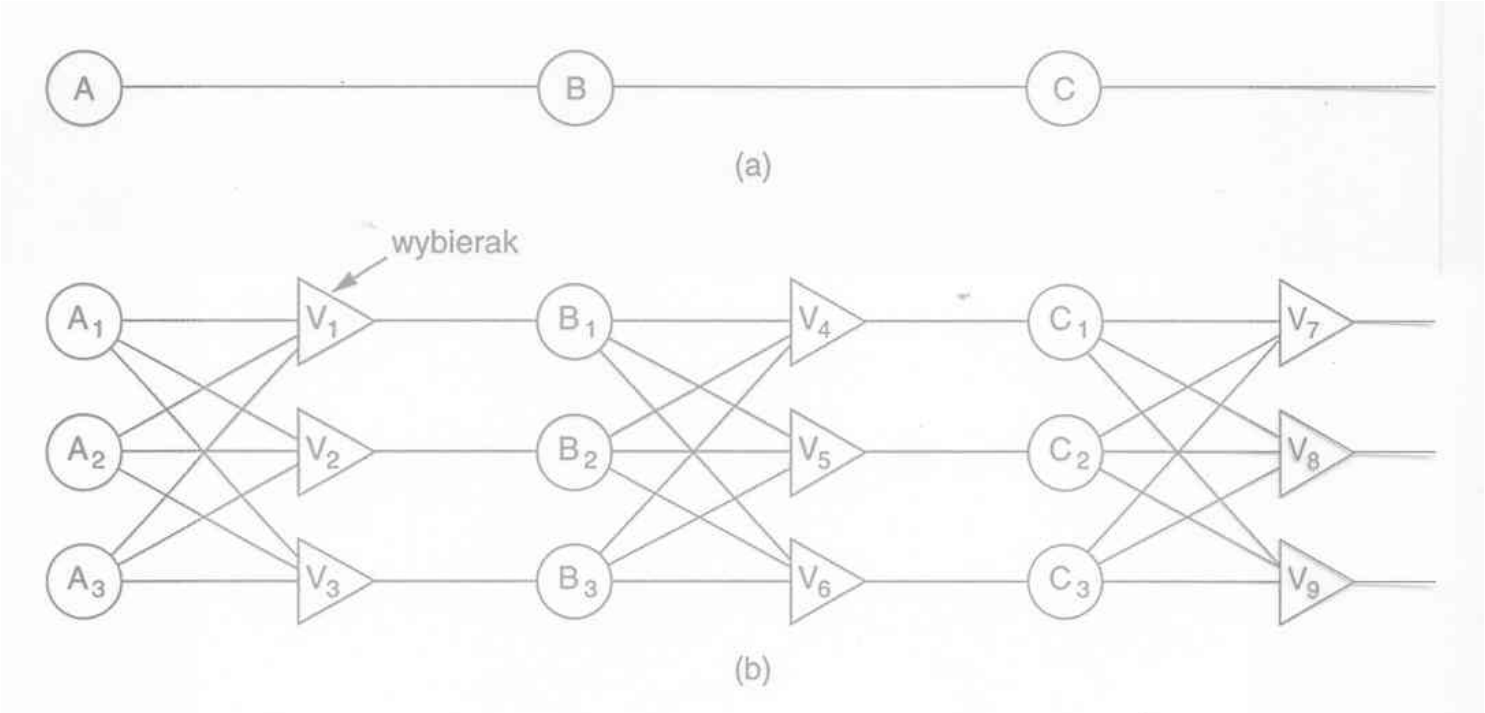
Zagadnienia analizy projektowej:

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 potrójnej redundancji modularnej (ang. TMR - Triple Modular Redundancy).



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.
Realizacji 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 czasu.

Zasoby rezerwowe

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

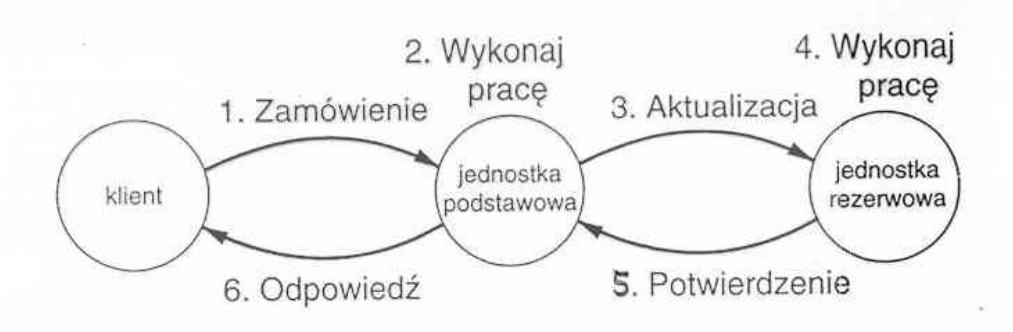
Zalety

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

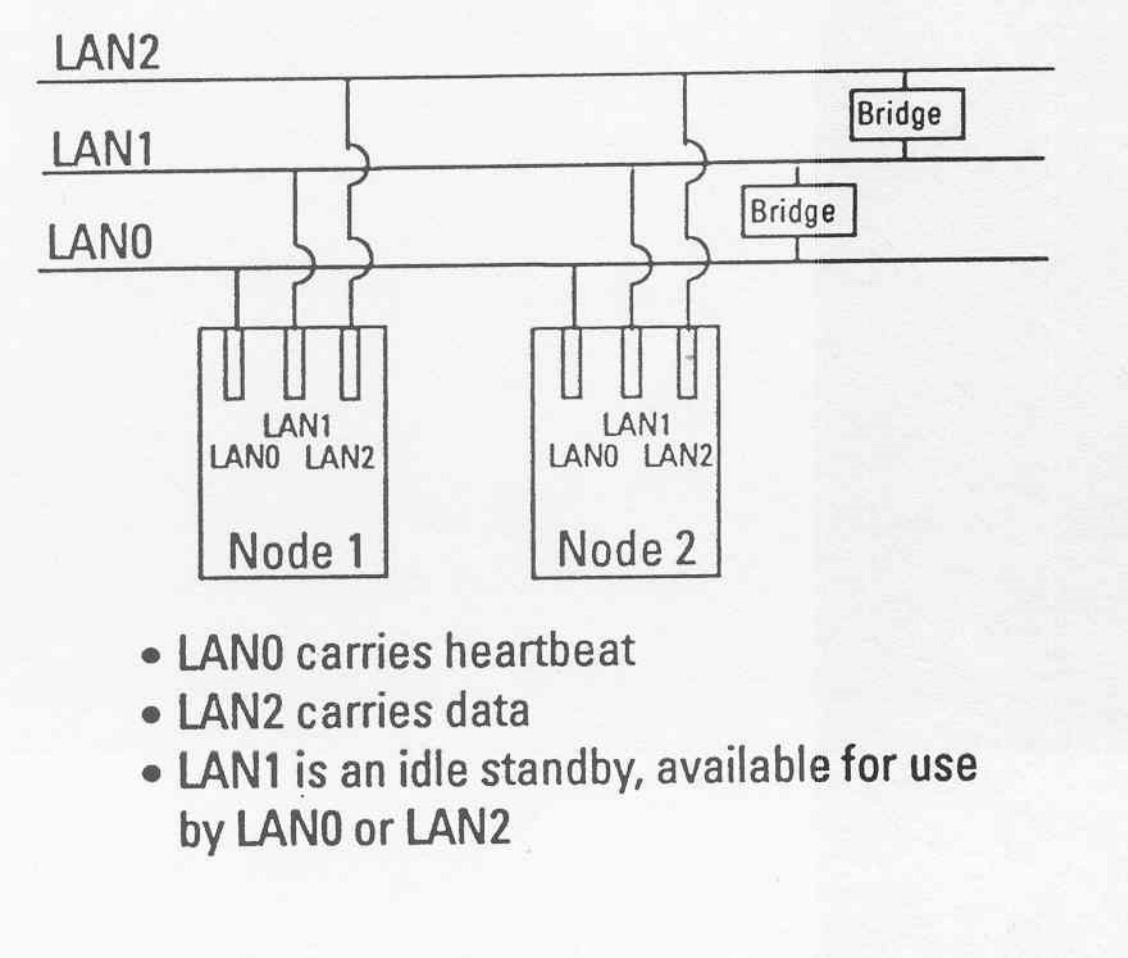
System odporny na (tolerujący) awarie sprzętu i oprogramowania, przeznaczony dla aplikacji wymagających wysokiej niezawodności (mission critical applications).

System rozproszony, składający się z kilku węzłów zorganizowanych jako klaster (cIaster). Węzłami mogą być systemy jedno lub wieloprocesorowe.
Węzły w klastrze mają wspólny dostęp do dysków z wykorzystaniem szyny (bus).
Połączone są również przez sieć ŁAN wykorzystywaną do:
- przesyłania informacji związanych z wykonywaniem aplikacji (dostęp klientów),
- przesyłania sygnałów monitorujących pracę węzłów (heartbeat).

MC Service Guard monitoruje prawidłowość działania (stanu) różnych elementów składowych systemu. W przypadku wykrycia wad podejmuje działanie – automatycznie eliminuje skutki wad, ewentualnie pozwala zminimalizować czas przerwy.
Wykrywa i reaguje na wady związane z pracą : jednostek centralnych, pamięci systemowych, sieci LAN, interfejsów sieciowych, procesów aplikacyjnych i systemowych.

Zasoby klastra (wszystkie zasoby niezbędne do wykonania określonych usług aplikacyjnych – pamięć dyskowa, zasoby sieciowe, procesy aplikacyjne i systemowe) organizowane są jako tzw. pakiety aplikacyjne (application packages). Pakiety te stanowią jednostki zarządzane w ramach klastra.

Przykład Redundantnej konfiguracji sieci LAN



Przykład redundantnej konfiguracji root/swap

W MC/Service Guard stosuje się redundancję w zakresie:

systemów komputerowych tworzących węzły,
linii sieci,
interfejsów sieciowych,
dysków: root, swap, danych,
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 następujących rozwiązań:
Mirror Disk/UX
RAID
Power Trust Uninterruptible Power Supplies (UPS),
HP Process Resource Manager
HP Open View Admin Center
HP Open View Operation Center