ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1995

Seria: INFORMATYKA z. 28 Nr kol. 1270

Henryk KRAWCZYK
Piotr W. UMIŃSKI

DIAGNOZOWANIE I ODTWARZANIE POPRAWNEGO STANU W ROZPROSZONYCH SYSTEMACH SIECIOWYCH

Streszczenie. Przedstawiono klasyfikacje różnych strategii diagnostyki uszkodzeń oraz różnych strategii odtwarzania poprawnego stanu wykorzystywane w komputerowych systemach rozproszonych. Omówiono ponadto dokładniej dwie strategie. Zaproponowano zintegrowany algorytm umożliwiający zarówno diagnostykę błędów, jak i odtwarzanie poprawnego stanu systemu. Bazuje on na porównywaniu zachowania się węzłów wykonujących to samo zadanie użytkowe.

DIAGNOSIS AND RECOVERY OF CORRECT STATE IN DISTRIBUTED NETWORK SYSTEMS

Summary. The paper presents classification criteria of various diagnostic strategies and various recovery strategies used in distributed computer systems. Two representative examples of such strategies are given. The integrated algorithm including diagnostic strategy as well as recovery strategy is proposed. It is based on behaviour comparison of nodes performing the same user task.

DIAGNOSE UND WIEDERHERSTELLUNG DES RICHTIGEN STANDES IN VERTEILTE SYSTEMEN

Zusammenfassung. In der Arbeit wird eine Klassifikation der verschiedenen Fehlerdiagnosestrategie und Wiederherstellung des richtigen Standes in verteilte

Praca zrealizowana w ramach grantu KBN nr 8S50301707.

Systemen gezeigt. Zwei represäntative Beispiele solcher Strategie wurden hinzufügt. Ausserdem schlug man einen integrierter Algorithmus vor, der die Diagnose und Wiederherstellung des richtigen Standes ermöglicht. Dieser Algorithmus ist gründet auf Vergleichung von Knoten, die gleiche Aufgaben realisieren.

1. Wstęp

Obserwujemy obecnie ciągły wzrost znaczenia systemów rozproszonych. Sieciowe systemy rozproszone pozwalają zarówno na zwiększenie efektywności przetwarzania za cenę niższą niż istniejące superkomputery, jak i na znacznie lepsze wykorzystanie posiadanych zasobów. Co więcej, zastosowanie systemów rozproszonych staje się niezbędne do zapewnienia dostępu do różnego rodzaju informacji z wielu punktów jednocześnie, na przykład w bankowości czy transporcie lotniczym i morskim. W tego rodzaju systemach część podejmowanych operacji ma charakter wybitnie lokalny (jak np. obsługa rachunku klienta w macierzystym oddziale banku) i nie ma potrzeby, aby był w nie angażowany cały system. Z drugiej strony pewne operacje wymagają zaangażowania wielu węzłów (jak np. rozliczenia międzybankowe). Wraz ze wzrostem znaczenia systemów komputerowych rosną stawiane im wymagania niezawodnościowe. I tu technologia systemów rozproszonych wychodzi naprzeciw tym wymaganiom. Poprzez rozproszenie zasobów zyskujemy możliwość zastąpienia wadliwych elementów systemu przez inne, sprawne. Wymiana ta może odbywać się w sposób płynny i nie zakłócać działania reszty systemu. Cały system uzyskuje przez to większą odporność na błędy.

Na ogół problemy diagnostyki i odtwarzania analizowane są niezależnie. Oznacza to, że nie istnieje wspólny model umożliwiający analizę i syntezę systemów posiadających zarówno własności diagnozowalności, jak i odtwarzalności obliczeń. Niniejsza praca stanowi próbę łącznego ujęcia tych dwóch problemów. Przedyskutowane zostaną najpierw różne metody lokalizowania uszkodzeń w sieciowych systemach rozproszonych. Następnie omówione zostaną podstawowe strategie odtwarzania stanu systemu w zależności od tego, czy poszczególne węzły systemu posiadają wewnętrzne mechanizmy odtwarzania, czy też ich nie posiadają. Na podstawie tego zostanie zaproponowana zintegrowana strategia diagnozowania i odtwarzania stanu w systemie rozproszonym, uwzględniająca sytuacje zarówno po wystąpieniu błędów sprzętowych, jak i programowych.

2. Diagnozowanie systemów rozproszonych

Diagnozowanie systemu rozproszonego jest zadaniem trudniejszym niż diagnozowanie pojedynczego komputera. Na poprawność działania całego systemu składają się: poprawność działania poszczególnych węzłów oraz poprawność działania połączeń między nimi. Rozróżnienie sytuacji, w której uszkodzony jest węzeł bądź uszkodzone jest połączenie z tym

węzłem, może być trudne. Wiele systemów nawet w czasie diagnozowania musi jednocześnie wykonywać zadania użytkowe. W wypadku wykrycia awarii w systemie powinna zostać podjęta akcja usuwania jej skutków, polegająca na rekonfiguracji systemu i przywróceniu poprawnego stanu. Opracowano wiele różnych strategii diagnostycznych. Ich ogólną klasyfikację przedstawia tabela 1.

Tabela 1

Klasyfikacja strategii diagnozowania systemów rozproszonych

Kryterium	Typ diagnostyki
Sposób testowania i podejmowania decyzji diagnostycznych	• scentralizowany testy i decyzje podejmowane przez jednostkę centralną • częściowo rozproszony testowanie przez poszczególne węzły, decyzje diagnostyczne przez jednostkę centralną • rozproszony testowanie i podejmowanie decyzji przez każdy z węzłów sieci
Zakres diagnozy	• diagnoza lokalna obejmuje podsieć wchodzącą w skład całego systemu • diagnoza globalna obejmuje cały system
Model uszkodzeń i testu	• modele statyczne zadane z góry typy uszkodzeń i moce testów • modele dynamiczne parametry zmieniające się w czasie • modele adaptacyjne dobór parametrów w zależności od aktualnego stanu sieci
Sposób identyfikacji uszkodzeń	• diagnoza jednokrokowa lokalizacja na podstawie odebranego syndromu • diagnoza wielokrokowa lokalizacja wraz ze stopniowym usuwaniem bądź wymianą uszkodzonych jednostek
Sposób realizacji testów	• off-line realizacja przy zawieszonym wykonywaniu zadań użytkowych • on-line testy wykonywane równolegle z zadaniami użytkowymi • współbieżnie część systemu realizuje w danym momencie diagnostykę, pozostała część zadania użytkowe
Wiarygodność diagnozy	• diagnoza poprawna jednostka sprawna nigdy nie jest uznana za uszkodzoną • diagnoza kompletna wszystkie jednostki uszkodzone są zidentyfikowane

Bezpośrednim celem diagnozowania systemu rozproszonego jest uzyskanie **syndromu**, czyli informacji o wynikach dokonanych sprawdzeń wszystkich elementów systemu. Syndrom powinien być dostępny dla wszystkich zainteresowanych jednostek i umożliwiać identyfikację

uszkodzonych jednostek w prosty sposób. Istotne jest również, żeby wszystkie jednostki podejmowały swoje decyzje na podstawie takiego samego syndromu (wymagana jest jego spójność) [6].

W pracy [4] przedstawiono strategię diagnozowania systemu rozproszonego składającego się z wielu węzłów. Bazuje ona na porównywaniu wyników zadań użytkowych wykonywanych na dwu węzłach jednocześnie. W praktyce, ze względu na trudności z porównywaniem skomplikowanych nieraz zbiorów wyników zadań porównywane są ich sygnatury.

Załóżmy, że mamy system rozproszony $W = \{W_1, W_2, ..., W_n\}$ składający się z n węzłów. Syndrom systemu przedstawmy jako macierz dwuwymiarową n×n. Oznaczmy ją przez $A=\{a_{ij}=0,1: i=1..n, j=1..n\}$. Przyjmujemy, że $a_{ij}=1$, gdy istnieją podejrzenia, że jeden z węzłów W_i lub W_j może być uszkodzony, w pozostałych przypadkach $a_{ij}=0$. Inicjalnie a_{ij} są równe 0 dla wszystkich i, j.

Diagnozowanie takiego systemu odbywa się w dwu fazach:

1. **Faza wykonywania zadań użytkowych**

Każde zadanie wykonywane jest na parze jednostek (W_i, W_j), a następnie wyniki wykonania zadania na obu tych węzłach są porównywane. Gdy jednostka W_i otrzyma wyniki różne od dostarczonych przez W_j, to ustawia $a_{ij}= 1$.

2. **Faza diagnozowania**

Po arbitralnie określonym czasie pracy systemu T wykonywany jest opisany poniżej algorytm lokalizacji uszkodzonych jednostek.

Algorytm 1 (algorytm diagnozy)

Oznaczmy przez S zbiór jednostek sprawnych, a przez U' zbiór jednostek, które podczas równoczesnego wykonania zadania dały wyniki niezgodne, czyli podejrzanych o to, że są uszkodzone. Tworzony jest graf nieskierowany G, którego wierzchołkami są wszystkie elementy zbioru S. Inicjalnie S zawiera wszystkie elementy W (S = W).

Dwa wierzchołki W_i i W_j grafu G są połączone krawędzią, jeżeli $a_{ij} <> 0$ lub $a_{ji} <> 0$. Oznaczmy stopień wierzchołka W_i grafu G jako $p(W_i)$, a stopień zredukowany wierzchołka W_i grafu G (czyli stopień W_i po usunięciu z G niektórych węzłów i związanych z nim krawędzi) jako $p'(W_i)$. Oznaczmy dalej przez U zbiór jednostek uznanych za uszkodzone. Inicjalnie jest on pusty, więc U = ∅. Inicjalnie mamy S = W.

W celu wyznaczenia zbioru U wykonujemy:

```
REPEAT
  U' = ∅
  FOR EACH Wi ∈ S
    IF p'(Wi) ≠ ∅
     U'=U' ∪ {Wi}
  FOR EACH Wi ∈ S
     BEGIN
       znajdź węzeł Wm dla którego p(Wm) jest najwiekszy
       S= S - {Wm}  (* usuń Wm ze zbioru węzłów sprawnych S *)
       U = U ∪ {Wm} (* włącz Wm do zbioru węzłów uszkodzonych *)
```

```
    END
UNTIL U' ≠ ∅
```

Algorytm kończy się w momencie, kiedy wszystkie węzły ze zbioru S jednostek sprawnych mają stopień wierzchołka równy 0. W rzeczywistym systemie fazy 1 i 2 występują naprzemiennie, przy czym moment dokonywania diagnozy może być wybierany okresowo, co z góry zadany przedział czasu, albo też może zależeć od stopnia obciążenia systemu.

Prezentowany algorytm identyfikacji uszkodzeń nie ma własności poprawności, tj. jednostka sprawna może zostać pod pewnymi warunkami zidentyfikowana jako uszkodzona. Jednak prawdopodobieństwo takiej sytuacji jest proporcjonalne do kwadratu prawdopodobieństwa uszkodzenia węzła a więc jest pomijalnie małe. Za zastosowaniem tego algorytmu przemawia natomiast jego złożoność obliczeniowa, wynosząca w najgorszym przypadku tylko $O(n^2)$.

3. Odtwarzanie poprawnego stanu systemu rozproszonego

W celu zapewnienia odporności systemu rozproszonego na błędy, po zlokalizowaniu uszkodzeń powinien on posiadać również możliwość odtworzenia swojego poprawnego stanu. Odtwarzanie stanu pojedynczego węzła może być realizowane w różny sposób. Generalnie, do odtwarzania stosuje się te same mechanizmy co do odtwarzania w środowisku jednoprocesorowym. Podstawowymi strategiami odtwarzania są [1]:

- strategia z użyciem punktów kontrolnych,
- strategia z zastosowaniem bloków odtworzeniowych.

W zależności od rodzaju wykrytych błędów oraz od wbudowanych w poszczególne elementy systemu mechanizmów odtwarzania, możemy wykorzystać różne strategie odtwarzania stanu systemu, które przedstawia tabela 2.

Strategia odtwarzania wstecznego polega na tym, że stan węzła jest okresowo zapamiętywany w czasie obsługi punktu kontrolnego. W przypadku awarii węzła po usunięciu jej przyczyny, praca węzła jest kontynuowana na podstawie stanu zapamiętanego w jednym z poprzednich punktów kontrolnych. Rozwinięciem omawianej strategii jest strategia wykorzystująca bloki odtworzeniowe [7]. Każde zadanie, które ma być wykonane przez dany węzeł, realizowane jest w kilku wariantach. Każdy wariant wykonuje te same funkcje, ale w różny sposób. Warianty powinny być wykonywane przez niezależne zespoły programistów, z wykorzystaniem w miarę możliwości różnych algorytmów, aby uniknąć tych samych błędów programowych. Wykonywanie zadania rozpoczyna się od wykonania pierwszego wariantu. Następnie otrzymane wyniki są sprawdzane pod względem ich poprawności przez tzw. test akceptacji, specyficzny dla danego zadania. Jeśli wyniki są zaakceptowane przez test akceptacji, są kwalifikowane jako poprawne i przekazywane do zleceniodawcy. W przeciwnym wypadku wykonywany jest kolejny wariant zadania dla tych samych danych.

Dopiero gdy żaden z wariantów nie dał poprawnych wyników, zadanie zwraca specjalny komunikat o błędach.

Tabela 2

Klasyfikacja strategii odtwarzania stanu w systemach rozproszonych

Klasyfikacja	Typ odtwarzania
Sposób odzyskiwania utraconej informacji	• wsteczne powrót do uprzednio zapamiętanego poprawnego stanu • bieżące narzucenie poprawnego stanu • mieszane łączące obie strategie
Sposób realizacji mechanizmów odtworzeniowych	• jednopoziomowe • wielopoziomowe
Koordynacja odtwarzania	• scentralizowana istnieje centralny węzeł-koordynator, przechowujący wszystkie informacje potrzebne do odtwarzania • rozproszona brak centralnego koordynatora, informacja na podstawie której odbywa się odtwarzanie, jest rozproszona
Sposób synchronizacji procesów odtwarzania	• zgodnie z zachodzącą komunikacją międzyprocesorową • zgodnie ze wspólną podstawą czasu
Skutki odtwarzania	• odtworzenie pełne informacja niezbędna do kontynuacji wykonywania zadania (zadań) zostaje odtworzona • odtworzenie poprawne żadna informacja nie zostaje odtworzona błędnie

W celu zapewnienia odporności całego systemu na błędy, oprócz mechanizmów wykorzystywanych przez poszczególne węzły potrzebne są mechanizmy diagnostyki i odtwarzania na poziomie systemowym. W sieci komputerowej z wbudowanymi mechanizmami odtwarzania poszczególne węzły przechowują syndrom systemu w postaci listy węzłów sprawnych oraz węzłów chwilowo niedostępnych. Listy te są tworzone w czasie inicjalizacji systemu w ten sposób, że wszystkie węzły aktywne w danym momencie umieszczane są na liście węzłów sprawnych, a lista węzłów niedostępnych pozostaje pusta. Włączenie się nowego węzła do systemu jest poprzedzone procedurą rejestracyjną, w wyniku której wszystkie pozostałe węzły umieszczają go na swoich listach węzłów sprawnych.

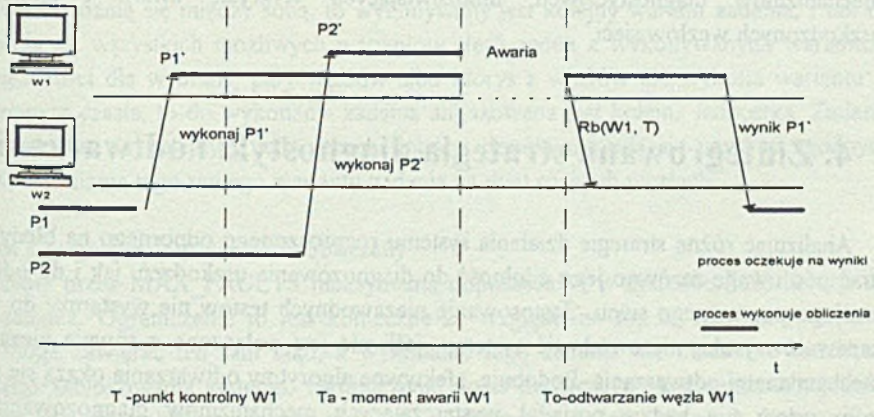
Węzeł zostaje uznany za węzeł niedostępny, gdy jest on uszkodzony bądź też przerwane zostanie z nim połączenie. Niesprawność węzła może być orzeczona na podstawie jego samodiagnozy albo jako rezultat działania diagnostyki rozproszonej. W obu tych wypadkach syndrom systemu zostaje uaktualniony. Węzeł niedostępny zostaje tymczasowo usunięty z listy

aktywnych węzłów i umieszczony na liście węzłów czasowo niedostępnych. Wskutek tej operacji nie są mu przydzielane nowe zadania (i nie postępuje dalsza dezorganizacja pracy sieci). Procesy innych węzłów oczekujące na efekty działania czasowo nieaktywnego węzła nie są jeszcze terminowane w nadziei na przywrócenie poprawnego stanu tego węzła.

Każdy węzeł ma przyporządkowany określony maksymalny czas nieaktywności, przed upływem którego powinien odtworzyć poprawny stan po awarii i rozpocząć normalną pracę. Jeśli uszkodzony węzeł nie powstanie w przewidzianym dla niego czasie, jest traktowany jako trwale uszkodzony. Każdy z pozostałych sprawnych węzłów usuwa węzeł uszkodzony z listy chwilowo niedostępnych, a następnie sygnalizuje procesom oczekującym na uszkodzony węzeł jego całkowity upadek. Procesy te bądź przekazują zadanie do wykonania innym sprawnym węzłom, bądź kończą swoją działalność. Jeśli uszkodzony węzeł rozpocznie poprawną pracę po czasie przewidzianym dla niego na odtwarzanie, musi włączyć się do pracy systemu jako nowy węzeł, wykonując przewidzianą protokołem procedurę zgłoszeniową.

Bardziej skomplikowane jest włączanie się węzła po dokonaniu udanego odtwarzania poprawnego stanu w przewidzianym czasie. Przez $R_{Bi}(W_j, T)$ oznaczamy **odtwarzanie wsteczne węzła W_i względem węzła W_j i chwili T**. Operacja ta polega na odtworzeniu stanu wszystkich procesów węzła W_i, które komunikowały się z węzłem W_j po chwili T, do stanu, jaki miały w momencie T.

Wszystkie węzły, które nie komunikowały się z węzłem W_j po chwili T, nie muszą w związku z tym podejmować żadnej akcji odtwarzania i mogą kontynuować swoją pracę. Z kolei, jeśli dany węzeł po chwili T komunikował się tylko z węzłem W_j, to musi jedynie odtworzyć swój stan z czasu T. Jeśli węzeł W_i komunikował się po chwili T nie tylko z węzłem W_j, ale i innymi węzłami, to po wykonaniu odtwarzania swojego stanu względem chwili T musi również rozesłać do tych węzłów żądanie przeprowadzenia przez nie operacji odtwarzania względem węzła W_j i chwili T: $R_{Bi}(W_i, T)$.



Rys. 1. Odtwarzanie stanu po awarii węzła
Fig. 1. Recovery after the node fail

Rys. 1 przedstawia przebieg odtwarzania poprawnego stanu dwu współpracujących węzłów po awarii jednego z nich. Węzeł W_2 zleca węzłowi W_1 wykonanie zadań P_1' i P_2', na których wyniki oczekują odpowiednio procesy P_1 i P_2. Węzeł W_1 w chwili T realizuje procedurę obsługi punktu kontrolnego, w której zapamiętuje swój bieżący stan. Oznaczmy przez T_a moment awarii węzła W_1. W tym czasie procesy P_1 i P_2 węzła W_2 oczekują na rezultat odtwarzania węzła W_1. W momencie T_0 węzeł dokonuje odtworzenia swojego poprawnego stanu z chwili T. W tej sytuacji węzeł W_2 zabija proces P_2 (gdyż proces P_2', rozpoczęty po chwili T nie będzie kontynuowany w W_1), a zachowuje P_1. W efekcie proces P_1 węzła W_2 otrzymuje poprawne wyniki wykonania procesu P_1'.

W zastosowanej strategii przypadek, gdy węzeł sieci nie ma wbudowanych mechanizmów odtwarzania ani samodiagnozowania, może być sprowadzony do przypadku węzła o nieskończenie krótkim dopuszczalnym czasie odtwarzania. Awaria takiego węzła może być stwierdzona tylko przez diagnostykę rozproszoną. Odtwarzanie poprawnego stanu systemu rozpoczyna się wówczas od wyizolowania uszkodzonego węzła, tak że poszczególne węzły zaprzestają komunikować się z węzłem uszkodzonym i zlecać mu nowe zadania do wykonania. Kolejnym etapem przywracania poprawnego stanu systemu jest wyodrębnienie na poszczególnych węzłach zadań oczekujących na wyniki z uszkodzonego węzła. Zadania te muszą być bądź bezwarunkowo zakończone, bądź też przejść do oczekiwania na wyniki zadań działających na innym węźle. Skrajnym rozwiązaniem jest przeinicjalizowanie węzła oczekującego na komunikacje z węzłem uszkodzonym. W ten sposób usunięte zostają z systemu wszelkie ślady awarii i system może kontynuować pracę.

Przedstawiona metoda odtwarzania kooperujących węzłów niesie za sobą poważne niebezpieczeństwo, znane jako *efekt domina* [1]. Powoduje ono, że system na skutek istnienia komunikacji międzyprocesorowej w okresie przed awarią zmuszony jest wykonać dużo więcej akcji odtwarzania, niż wynikałoby to z odtwarzania pojedynczych, nie komunikujących się ze sobą procesów. Poza tym strategia ta zakłada istnienie w systemie wyodrębnionych mechanizmów diagnostycznych, umożliwiających wykrycie awarii i zlokalizowanie uszkodzonych węzłów sieci.

4. Zintegrowana strategia diagnostyki i odtwarzania

Analizując różne strategie działania systemu rozproszonego odpornego na błędy, musimy brać pod uwagę zarówno jego zdolność do diagnozowania uszkodzeń, jak i do odtwarzania swojego poprawnego stanu. Zastosowanie niezawodnych testów nie wystarczy do tego, aby zapewnić wysoką niezawodność systemu, jeśli nie jest połączone z równie niezawodnymi mechanizmami odtwarzania. Podobnie, efektywne algorytmy odtwarzania okażą się zawodne, jeśli system nie będzie posiadał wystarczających mechanizmów diagnozowania swoich uszkodzeń. W szczególności, musimy traktować łącznie koszty diagnozy systemu oraz koszty poprawnego odtwarzania stanu. W związku z tym wśród różnych strategii diagnozowania systemów rozproszonych oraz różnych metod przywracania poprawnego stanu systemu należy zwrócić uwagę na te, w których diagnoza systemu daje od razu informację niezbędną do

odtwarzania poprawnego stanu. Taką metodą jest równoległe wykonywanie zadań na wielu węzłach i porównywanie wyników. Metoda ta znana jest w literaturze jako programowanie N-wariantowe [1]. Dzięki wykonywaniu tych samych zadań na kilku węzłach możliwe jest porównanie wyników, a w wypadku ich rozbieżności określenie poprawnego wyniku na drodze głosowania. Węzły, które nie dostarczą wyników w określonym czasie albo dostarczą wyniki inne niż pozostałe, są podejrzane o błąd. Z drugiej strony porównywanie wyników dwu różnych wariantów tego samego zadania może być czasem mylące: dwa różne warianty mogą dawać zbiory wyników różniące się przykładowo ilością spacji w tekście, a co za tym idzie - o różnych syndromach.

Uwzględniając powyższe ograniczenia, można zaproponować zintegrowaną strategię diagnozowania/odtwarzania stanu systemu. Proponowana przez autorów strategia stanowi rozwinięcie metod przedstawionych w pracach: [2] i [5], i polega na bieżącym odtwarzaniu stanu systemu z jednoczesnym uaktualnianiem syndromu uszkodzeń..

Załóżmy, że mamy do czynienia z systemem składającym się z n węzłów połączonych siecią komputerową. Każde zadanie Z wykonywane przez węzeł składa się z m wariantów Z_1, Z_2, Z_m. Każdy wariant zadania wykonuje tę samą operację, ale w inny sposób. Warianty zadania napisane są w taki sposób, aby uniknąć tych samych błędów programowych. Osiąga się to, podobnie jak w technikach z użyciem bloków odtworzeniowych, przez zaangażowanie niezależnych zespołów programistów i zastosowanie różnych algorytmów. Dla uproszczenia przyjmijmy, że każdy węzeł sieci może wykonać każde zadanie niezależne, to jest takie, które nie współpracuje z innymi zadaniami na innych węzłach. W systemie istnieje mechanizm szeregowania zadań, którego szczegóły możemy w tym momencie pominąć (patrz np. [3]).

Wykonywanie każdego kolejnego zadania użytkowego zlecane jest przez system dwu wybranym węzłom. Każdy z tych węzłów wykonuje wariant pierwszy zadania. Gdy oba węzły wykonają ten wariant, następuje porównanie wyników. W wypadku gdy jednostki osiągnęły zgodność, wynik jednej z nich jest przekazywany jako wynik zadania do zleceniodawcy. Jeśli obie jednostki w przewidzianym czasie ukończyły wykonywanie pierwszego wariantu zadania, ale ich wyniki różnią się między sobą, to wykonywany jest kolejny wariant zadania, i tak do wyczerpania się wszystkich możliwych wariantów. Jeśli żaden z wykonywanych wariantów nie da zgodności dla wybranej pary węzłów albo któryś z węzłów nie wykona wariantu w przewidzianym czasie, to do wykonania zadania angażowana jest kolejna jednostka. Zadanie wykonywane jest na kolejnych jednostkach dopóty, dopóki nie uda się uzyskać zgodnych wyników wykonania tego samego wariantu zadania na dwu różnych węzłach.

Algorytm 2 (algorytm odtwarzania obliczeń)

Oznaczmy przez MAX_FAULTS maksymalną dopuszczalną w systemie ilość iteracji dla danego zadania. Ograniczenie to jest konieczne ze względu na to, że wszystkie warianty zadania mogą zawierać ten sam błąd, a wykonanie tego zadania nie może prowadzić do całkowitego zablokowania systemu. Ograniczenie ilości iterakcji (czyli dopuszczalnej ilości uszkodzonych jednostek) jest również związane z minimalną ilością jednostek, które muszą pozostawać sprawne w systemie dla jego efektywnego funkcjonowania.

Oznaczmy dalej przez $R(Z_i, W_j)$ zbiór wyników (rezultatów) wykonania wariantu Z_i przez węzeł W_j.

```
FOR i=1 TO m
  BEGIN
   zleć  Zi węzłom W1 i W2
   IF zadania wykonane w przewidzianym czasie
     BEGIN
       IF  R(Zi, W1) = R (Zi, W2)
         EXIT R(Zi, W2)
     END
   ELSE
     BREAK; (*zakończ wykonywanie Zi przez parę W1, W2 *)
  END

a12=a21=1

FOR k = 3 TO MAX_FAULTS
  BEGIN
   FOR i=1 TO m
     BEGIN
      zleć  Zi węzłowi Wk
      IF zadanie wykonane w przewidzianym czasie
        BEGIN
          FOR j = 1 TO k - 1
            IF  R(Zi, Wj) = R (Zi, Wk)
              EXIT R(Zi, Wk)
        END
      ELSE
        BREAK; (*zakończ wykonywanie Zi przez  węzeł  Wk *)
     END
   FOR j = 1 TO k - 1
     ajk=akj=1 (* uaktualnij syndrom *)
  END
EXIT ERROR
```

Zaproponowany algorytm łączy w sobie dwie postulowane właściwości: pozwala na uzyskanie prawidłowego wyniku zadania, jeśli przynajmniej jeden z wariantów zadania nie zawiera błędu i przynajmniej jedna para węzłów jest sprawna, a także umożliwia uzyskanie syndromu systemu w postaci omawianej w rozdziale 2. Dzięki temu możliwa jest prawidłowa diagnoza systemu rozproszonego, przy zapewnieniu łatwego i szybkiego odtwarzania. Należy przy tym zwrócić uwagę, że inicjalna nadmiarowość węzłów dla wykonania jednego zadania wynosi dwa, podczas gdy klasyczne metody programowania N-wariantowego wymagają w każdym przypadku co najmniej 3 węzłów.

5. Podsumowanie

Przedstawiona w artykule kompleksowa metoda zapewnienia odporności systemu rozproszonego na błędy stanowi próbę łącznego traktowania zagadnień diagnozowania i

odtwarzania stanu systemu. Zapewnia ona odporność systemu na błędy sprzętowe oraz błędy oprogramowania. Przy zastosowaniu niezbyt dużej nadmiarowości sprzętowej (w porównaniu do klasycznych metod programowania N-wariantowego) pozwala na zachowanie dużej odporności na błędy sprzętowe. Jednocześnie diagnostyka i odtwarzanie w systemie przebiegają poprawnie w przypadku wystąpienia błędów programowych w realizowanych zadaniach.

Zaproponowano zintegrowaną metodę zwiększenia wiarygodności funkcjonowania systemu rozproszonego. Metoda ta łączy w sobie znane dotąd podejście dotyczące testowania porównawczego i techniki nawrotów do poprzednich stanów w systemie. W celu wyeliminowania błędów programowych zakłada się wykorzystanie różnych wariantów oprogramowania. W przedstawionym Algorytmie 2 poszczególne warianty są wykonywane zawsze w tej samej kolejności. Możliwe jest takie zorganizowanie pracy systemu, gdzie kolejność wykonywania wariantów może zmieniać się w poszczególnych węzłach.

W pracy rozpatrzono tylko przypadki zadań niezależnych. W przypadku zadań wymagających wzajemnej komunikacji i wymiany wyników pośrednich algorytm diagnostyki/odtwarzania musi być bardziej złożony. Dotyczy to zwłaszcza odtwarzania stanu. Należy wówczas wprowadzić pewne ograniczenia na komunikację między poszczególnymi zadaniami [8].

Proponowane rozwiązanie, choć jest niejako rozwiązaniem bardziej naturalnym i efektywniejszym od zwykłej techniki głosowania (programowanie N-wariantowe), wydaje się jednak być dalej bardzo kosztowne i jako takie może być stosowane jedynie w przypadku zadań o dużym znaczeniu. Pojawia się więc problem doboru ilości węzłów do realizacji pojedynczego zadania (Max) i ilości wariantów tego zadania (M). W najprostszym przypadku Max=1, M=1. Dla Max=2, M=1 mamy strategię zapewniającą odporność tylko na błędy sprzętowe (awarię węzła). Podobnie dla Max=1 i M=2 mamy system odporny tylko na błędy programowe. Dopiero Max=2 i M=2 zapewnia nam odporność na oba rodzaje błędów.

LITERATURA

[1] Anderson T., Lee P.: Fault Tolerance: Principles and Practice, Prentice-Hall 1981.

[2] Agrawal P.: Fault Tolerance in Multiprocessor Systems without Dedicated Redundancy, IEEE Transactions on Computers, vol. c-37, no. 3. March 1988.

[3] Błażewicz J., Drabowski M., Węglarz J.: Scheduling Multiprocessor Tasks to Minimize Schedule Length, IEEE Transactions on Computers, vol. c-35, no. 5. May 1986.

[4] Dahbura A. T., Sabnani K. K., King L. L.: The Comparasion Approach to Multiprocessor Fault Diagnosis, IEEE Fault-Tolerant Computing Symposium, 1985.

[5] Kozłowski W. E., Krawczyk H.: Distributed Diagnosis Strategy in Hybrid Fault Situations, Joint Symposium on Fault-Tolerant Computing, Chongqing , China 1989.

[6] Krawczyk H.: Rozproszona diagnoza Sieci Komputerowych, Materiały Krajowego Sympozjum Komunikacji, Bydgoszcz 1989.

[7] Lee P.: Consideration of The Recovery Block Scheme, The Computer Journal, Jun. 1978.
[8] Randell B.: System Structure for Software Fault Tolerance, IEEE Transactions on Software Engineering, June 1975.

Recenzent: Prof. dr hab. inż Andrzej Grzywak

Wpłynęło do Redakcji 18 listopada 1994 r.

Abstract

The technique of distributed networks allow to built more reliable computer systems. The diagnosis of distributed systems and the recovery of their proper state are usually consider as separate problems. This paper presents an uniform approach to both problems.

There are many strategies of diagnosis for distributed systems. The briefly classification of them is presented in Table 1 in Section 2. As an example of such strategy an algorithm for identification of faulty nodes in distributed system is presented later in this section. However this algorithm does not ensure, that no one of correct nodes will be identify as faulty, but probability of such situation is very small. This algorithm is also very fast ($O(n^2)$) comparing to other algorithm.

The classification of various recovery strategies is given in Table 2 in Section 3. Then, Section 3 contains the detailed presentation of the backward error recovery strategy.

Finally, in Section 4, the complex strategy for diagnosis and recovery of distributed systems is presented. It is based on a concept of N-version programming. Each task in the system has several versions. In the begining of the task execution, first variant of this task is scheduled on two nodes. The results of execution are compared. When results are identical, one of them is establish as a proper result of task. Otherwise, the next version of the task is executed on both nodes. When no one of version produce identical results on given nodes, the task is scheduled in the same way on another node. It is repeated until two identical results are obtained. The formal algorithm of scheduling versions of the task on nodes of the system is presented. The identification of the faulty nodes based on the results of compression is performed by algorithm presented in Section 2.

In summary (Section 5) the cost of several versions of proposed strategy is discussed.