ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: INFORMATYKA z. 30 Nr kol. 1315

Henryk KRAWCZYK
Maciej PIECHÓWKA

ZWIĘKSZENIE WIARYGODNOŚCI PRZETWARZANIA ZŁOŻONYCH ZADAŃ W SYSTEMACH ROZPROSZONYCH

Streszczenie. Przedstawiono możliwości zwiększenia niezawodności wykonywania złożonych zadań w systemach rozproszonych. W oparciu o algorytmiczną metodę tolerowania błędów zaproponowano dobór odpowiednich procedur kontrolnych. Dokonano również oceny wiarygodności przetwarzania tak zmodyfikowanych zadań.

DEPENDABILITY IMPROVEMENT OF WORKFLOW PROCESSING IN DISTRIBUTED SYSTEMS

Summary. The paper presents the way for increasing reliability of workflow processing in distributed systems. To choose the special checking procedures the algorithm based fault-tolerance approach is used. Dependability analysis of such modified workflow is also given.

ERHOHUNG DER ZUVERLASSIGKEIT VON VERARBEITUNG KOMPLIZIERTEN AUFGABEN IN VERTEILTEN SYSTEMEN

Zusammenfassung. Im Aufsatz werden Moglichkeiten von Erhohung der Zuverlassigkeit bei der Verwirklichung komplizierten Aufgaben dargestellt. Nach der algoritmischen Methode von Fehlertoleranz schlug man eine Auswahl der entsprechenden Kontollprozedur vor.

Praca sponsorowana przez grant KBN nr: 8 S503 017 07.

1. Wprowadzenie

Rozwój systemów rozproszonych na bazie sieci komputerowych umożliwia realizację różnorodnych zadań, które wykorzystują systemy baz danych rozlokowane w odległych rejonach geograficznych, a także wymagają skomplikowanych operacji rozłożonych w czasie, często wspomaganych decyzjami ludzkimi. Tego typu zadania nazywać będziemy złożonymi zadaniami rozproszonymi (ZZR) i opisywać za pomocą schematów operacyjnych, tzw. workflow [7].

Przykładami tego typu zadań są procesy wytwórcze, zarządzania przedsiębiorstwami czy obsługa finansowa, medyczna lub dokonywanie kontroli. Można stwierdzić, że takie złożone zadanie składa się z wielu podzadań, które mogą być wykonywane sekwencyjnie, współbieżnie bądź równolegle i których otrzymane w czasie wyniki są wzajemnie wykorzystywane. Co więcej, w wielu sytuacjach konieczne jest wskazanie alternatywnej drogi obliczeń, redystrybucji wcześniej założonych prac, czy obsługi zaistniałych sytuacji wyjątkowych.

Specyfikaja rozproszonych zadań złożonych (ZZR) nie jest sprawą łatwą. Na ogół opisuje się je na różnych poziomach abstrakcji poprzez wykorzystanie zagnieżdżania specyfikacji podzadań. Model ZZR składa się na ogół [3,7] z następujących elementów:

- **podzadań** zawierających opis akcji i funkcji oraz częściowy porządek wykonywanych operacji,
- **przetwarzanych obiektów**, takich jak: dokumenty, rekordy danych czy obrazy i dźwięki, na których wykonywane są określone operacje,
- **wymaganych usług** - określonych poprzez wykorzystywane systemy informatyczne bądź zdolności ludzkie niezbędne do wykonywania danego podzadania,
- **wykonawców** złożonych z ludzi i systemów informatycznych oferujących odpowiednie usługi, wykonujących podzadania i kooperujących przy realizacji ZZR.

Z kolei pełny opis ZZR zawiera:

- strukturę zadania określoną przez przepływ sterowania i przepływ danych, sposób wykonywania podzadań oraz warunków realizacji jego poszczególnych kroków,
- działania zaradcze dotyczące sytuacji wyjątkowych, które mogą się zdarzyć przy realizacji ZZR,
- parametry wykonywania podzadania, takie jak: czas jego rozpoczęcia i zakończenia, priorytety wykonania, czy istotność podzadania w całym problemie, działania pomocnicze (np. relokacji podzadań).

Zapewnienie wysokiej wiarygodności wykonywania ZZR jest sprawą istotną, decydującą w dużej mierze o powszechnym wykorzystywaniu sieciowych systemów rozproszonych. W literaturze brak jest jednak systematycznych rozwiązań zapewniających skuteczny i efektywny dobór działań zaradczych. Niniejsza praca proponuje adaptację algorytmicznej metody tolerowania błędów [1,2] do obsługi różnorodnych sytuacji wyjątkowych. Takie podejście umożliwia nie tylko określenie odpowiednich procedur zaradczych, ale także odpowiednie ich

rozmieszczenie w systemie w celu zapewnienia zadanego poziomu wiarygodności. W niniejszej pracy ograniczymy się przed wszystkim do opisu tej nowej idei oraz zilustrowania jej odpowiednim przykładem. Pełna ocena skuteczności proponowanego rozwiązania wymagać będzie oddzielnego opracowania.

2. Przykład złożonego zadania rozproszonego

Rozpatrzmy ZZR przedstawione na rys.1, dotyczące obsługi kredytu. Na rysunku prostokątami zaznaczono zbiory podzadań związanych z użytkownikami i bankiem, do którego zwraca się użytkownik o kredyt, zbiór systemów informatycznych przechowywujących listy wszelkiego rodzaju dłużników oraz pewne elementy zewnętrznej kontroli czasowej i merytorycznej kontroli bankowej. Do realizacji tak złożonego zadania wymagane są pewne operacje obliczeniowe (przeszukiwanie list), pewne transakcje (dokonywanie wpłaty, czy modyfikacje oprocentowania), a także typowy przesył informacji (np. list upominający). Oczywiste jest, że poszczególne zadania można szczegółowo wyspecyfikować. Dla ilustracji naszych rozważań pozostaniemy jednak na takim wysokim poziomie opisu.

Rys. 1. Schemat poglądowy obsługi kredytu
Fig. 1. General loan service schema

Założymy, że rozpatrujemy wykonanie się zadania złożonego w systemie rozproszonym. Wówczas można go opisać w następujący sposób:

$$ZZR = \langle T, P, C, W, A \rangle , \qquad (1)$$

gdzie:

$T = \{T_i\}$ jest zbiorem transakcji realizowanych w systemie rozproszonym, $i=1,2,...,t$;

$P = \{P_i\}$ jest zbiorem procesów realizowanych w systemie rozproszonym, $i=1,2,...,p$;

C = {C_i} jest zbiorem kontroli wykonywanych w systemie rozproszonym dotyczących zarówno transakcji T, jak i procesów P, i=1,2,...,c;

W = {W_i} jest zbiorem obserwowanych wyników analizowanego ZZR, np. sekwencje zmian stanów, warunki inicjalizacji, itp., i=1,2,...,w;

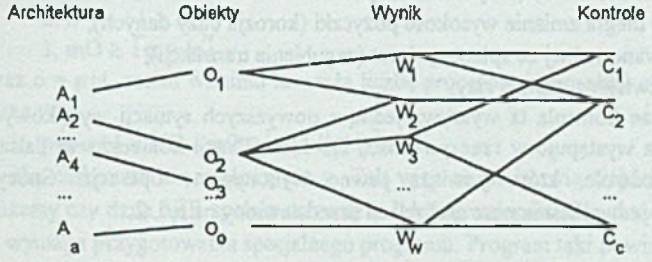
A = {A_i} architektura sieci komputerowej niezbędna dla realizacji ZZR, i=1,2,...,a.

Model zakłada, że ZZR stanowi zestaw transakcji i procesów, które są wykonywane w porządku określonym przez warunki W, zawierające również pewne decyzje ludzkie. Wykonanie zadania złożonego w warunkach rozproszenia, współbieżności i interwencji człowieka wymaga odpowiednich procedur kontrolnych. Zadanie to analizuje na bieżąco stan wykonywango zadania ZZR i w zależności od zaistniałej sytuacji otoczenia sugeruje podjęcie odpowiednich decyzji korekcyjnych. Zapewnienie odpowiedniej wiarygodności wykonania zadań wymaga odtworzenia spójności w sytuacjach nienormalnego zakończenia podzadania, korektę błędnych obliczeń czy powtórzenie pewnych usług lub wykonanie usług alternatywnych. Pełna analiza wszystkich działań kontrolnych nie jest możliwa. Oznaczmy przez W = {W_i} zbiór obserwowanych wyników pośrednich lub końcowych realizacji danego ZZR, i=1,2,..w. Każdy z wyników może być akceptowalny bądź nieakceptowalny. Pojawienie się pewnej liczby wyników nieakceptowalnych oznacza sytuację wyjątkową w systemie. Sytuacja taka powinna zostać wykryta przez odpowiedni zbiór kontroli C. Zbiór taki budowany jest na podstawie intuicji wykorzystując przy tym pewne zalecenia i doświadczenie. Pojawia się więc problem oceny jego skuteczności. Przedyskutowane to zostanie w następnym rozdziale.

3. Ocena wiarygodności wykonania złożonego zadania rozproszonego

Do oceny przyjętej strategii kontroli wykonywania się ZZR wykorzystamy algorytmiczną metodę tolerowania błędów. Metoda ta bazuje na przedstawieniu relacji między węzłami obliczeniowymi a otrzymywanymi wynikami oraz wynikami i dokonywaną kontrolą w systemie. W naszym przypadku otrzymujemy o jedną relację więcej (patrz rys. 2). Jak wynika z rys. 2, poszczególne węzły sieci komputerowej realizują podzadania ZZR. Opisujemy to relacją obliczeniową między zbiorem A oraz obliczeniem O, gdzie $O = T \cup P$, $o = p+t$.

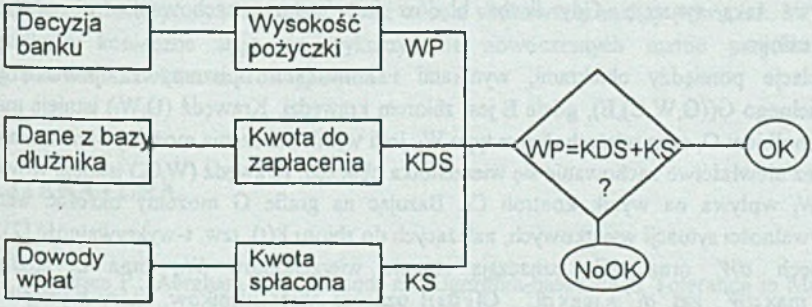
Można stwierdzić, że relacja $A \rightarrow O$ opisuje przyporządkowanie poszczególnych podzdań do węzłów sieci. Na ogół jest to odwzorowanie wzajemnie jednoznaczne, dlatego w dalszych rozważaniach pominiemy architekturę systemu, a operować będziemy na zbiorze O, reprezentującym konkretną implementację ZZR.



Rys. 2. Przykład relacji podstawowych obiektów ZZR
Fig. 2. An example of the object associations in ZZR

Relacja między obliczeniami a wynikami tych obliczeń (O→W) przedstawia zatem, w jaki sposób obserwowane wyniki zależą od konkretnych obliczeń. Na ogół realizowane podzadania są silnie powiązane i jeden wynik powstaje poprzez kooperację wielu procesów i transakcji. Analogicznie, planowane kontrole nie dotyczą pojedynczych wyników, a wzajemnego ich powiązania. To opisuje relacja W→C. W rzeczywistości mamy więc relacje złożone O→W→C (bądź A→O→W→C). Wynik obliczeń zależy więc od poprawności wykonania się odpowiednich procesów i transakcji. Zatem celem kontroli jest nie tylko wykrycie wyjątkowej sytuacji w systemie, ale także ustalenie jego pierwotnej przyczyny. Przyjmujemy, że wynik kontroli C_i wynosi 0, jeśli procedura C_i nie wykrywa sytuacji wyjątkowej w obserwowanych wynikach. Inaczej wynik kontroli C_i wynosi 1.

Rozpatrzmy obecnie przykład konkretnej procedury kontrolnej dla ZZR dotyczącego obsługi kredytu. Przedstawiona jest ona na rys. 3.



Rys. 3. Przykład procedury kontrolnej dla obsługi kredytu
Fig. 3. Example of checking procedure for loan service

Typowa kontrola spłaty kredytu polega na sprawdzeniu czy suma kwoty do zapłacenia pamiętanej w bazie danych dłużnika i rat spłaconych przez użytkownika stanowi wysokość przyznanej pożyczki. Jeśli tak, oznacza to, że spłacanie rat przebiega normalnie, inaczej należy określić przyczynę występujących rozbieżności.

Zauważmy, że możliwe są trzy kombinacje:

1. Przypadkowo uległa zmianie wysokość pożyczki (korozja bazy danych),
2. Nie zarejstrowano jednej ze spłaconych rat (zagubienie transakcji),
3. Użytkownik zwleka z zapłatą raty.

Zauważmy, że kontrola ta wykrywa jedną z powyższych sytuacji wyjątkowych, ale nie rozstrzyga, która występuje w rzeczywistości. Do identyfikacji konieczne są dalsze, na ogół szczegółowe kontrole, które powiążą pewne wykonywane operacje. Specyfikując je rozbudowujemy jednocześnie nasz graf relacji przedstawiony na rys. 2.

4. Warunki testowalności

Przyjmujemy,że procedura sprawdzająca C_i działa na zbiorze wyników $W^i = \{w_1^i, \ldots, w_r^i\}$ oraz że liczba błędnych wyników w_j^i nie przekracza wartości h. Wówczas procedurę C_i oznaczamy jako $C_i(g,h)$. Niech wynik sprawdzeń otrzymany przez procedurę $C_i(g,h)$ wynosi:

$$\omega_i = \begin{cases} 0 & \text{jeśli wszystkie } w_j^i \in W^i \text{ są akceptowalne,} \\ 1 & \text{jeśli r elementów } w_j^i \in W^i,\ r \le k \text{ jest nieakceptowalnych,} \\ \chi & \text{jeśli więcej niż r elementów jest nieakceptowalnych.} \end{cases}$$

Błędy w wynikach powstają wskutek niepoprawnie wykonujących się operacji ze zbioru O. Defekty w obliczeniach oznaczamy przez $F = \{F_i\}$, $i = 1,2,..f$, przy czym $f \le 2^{p+t}$. Zbiór F(t) oznacza zbiór wszystkich defektów ze zbioru F, dotyczących co najwyżej t obiektów biorących udział w obliczeniach. Zbiór F(t) może spowodować pewną liczbę błędnych wyników, przy czym w przypadku wystąpienia nie więcej niż h błędów procedura kontrolna wykrywa taką sytuację. Gdy liczba błędów jest wyższa, zachowanie procedury jest nieokreślone.

Relacje pomiędzy obiektami, wynikami i kontrolami opiszemy za pomocą grafu trójdzielnego G((O,W,C),E), gdzie E jest zbiorem krawędzi. Krawędź (O_iW_j) istnieje między wierzchołkiem O_i oraz wierzchołkiem typu W_j, jeśli wynik obliczenia może być zniekształcony poprzez niewłaściwe zachowanie się wierzchołka obliczeń. Krawędź (W_jC_k) istnieje wówczas, jeśli W_j wpływa na wynik kontroli C_k. Bazując na grafie G możemy określić warunki wykrywalności sytuacji wyjątkowych, należących do zbioru F(t), tzw. t-wykrywalność [2].

Niech dW_i oraz dC_j oznaczają stopnie wierzchołków W_i, oraz C_j grafu G, $dW = \max_i dW_i$, zaś $dC = \max_j dC_j$. O[F(α)] oznacza zbiór wyników, który wystąpił przy założeniu, że F(α) obliczeń zostało zrealizowanych nieprawidłowo, zaś max |O[F(α)]| = mO. Wówczas dla procedur kontrolnych typu C(g,h) ZZR jest w stanie zlokalizować każdą sytuację wyjątkową należącą do zbioru F(α) lub inaczej - ZZR jest α-lokalizowalny, jeśli zajdą następujące warunki:

1. $dC \geq g$
2. $o - \alpha + 1 \geq dO$ (2)
3. $mO \geq \lceil \alpha\, w/o \rceil$

Ponieważ o = p+t, zatem warunki te wiążą liczbę procesów i transakcji z parametrami α, *dO*, mO oraz w:

$$p+t-\alpha+1 > dO \text{ i } mO \geq \lceil \alpha\, w/(p+t) \rceil$$

$\lceil x \rceil$ jest liczbą całkowitą nie większą niż x. Tak więc na podstawie trójdzielnego grafu G określić możemy czy dane ZZR spełnia warunki α-lokalizowalności. Dla złożonych grafów G ocena taka wymaga przygotowania specjalnego programu. Program taki powinien bazować na przygotowanych specyfikacjach zbioru procesów, transakcji oraz procedur kontrolnych. Generacja grafu G ze zbioru specyfikacji jest sprawą znacznie trudniejszą niż sama ocena α-lokalizowalności.

5. Wnioski końcowe

W pracy zaproponowano systematyczną metodę doboru procedur kontrolnych dla zachowania α-lokalizowalności błędnie wykonujących się procesów i transakcji, które składają się na konkretne zadanie rozproszone. Z uwagi na rozmiar problemu podejście takie wymaga odpowiedniego narzędzia wspomagającego. W rozważaniach przyjęto prosty model relacji O→W→C, sprowadzający się do grafu trójdzielnego. W praktyce relacje te są bardziej złożone i wymagają znacznie szczegółowszej analizy poszczególnych procesów, transakcji i odpowiadających im procedur kontroli. Najlepszym rozwiązaniem wydaje się być podejście hierarchiczne, gdzie najwyższym poziomem rozważań jest proponowany graf G, zaś w razie potrzeby zmieniamy poziom szczegółowości w celu wnikliwszej analizy powiązań. W wielu przypadkach konieczne staje się wykorzystanie nowoczesnych metod projektowania oprogramowania, zwłaszcza prototypowanie.

LITERATURA

[1] Banerjee P., Abraham J.,Z.: Bounds on Algorithm-based Fault Tolerance in Multiple Processor Systems. IEEE Transactions on Computers, vol.35, No.4, April 1986, 296-306.

[2] Banerjee P., Abraham J.,Z.: Concurrent Fault Diagnosis in Multiple Processor Systems, FTCS 1986.

[3] Georgakopoulos D., Hornick M., Sheth A.: An overview of workflow management: From process modeling to workflow automation infrastructure. Distributed and Parallel Databases , 3(2):119-154 , April 1995.

[4] Gray J., Reuter A.: Transaction Processing: Concept and Techniques. The Morgan Kaufman Series in Data Management System, 1993.
[5] Elmagarmid A. (Ed.), Advanced Transaction Models for New Applications. The Morgan-Kaufmann Series in Data Management System, 1992.
[6] Krawczyk H.: Dependable Processing, skrypt TEMPUS , 1995, Gdańsk.
[7] Workflow Management Coalition Glossary; A Workflow Management Coalition Specification , no v. 1994.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 18 grudnia 1995 r.

Abstract

The paper presents dependability problems of workflow processing in distributed systems. We consider a typical example of loan workflow (see Fig.1) consisted of a bank control, multidatabase server and several users workstations. All these nodes connected by typical communication links create a distribued system processing. We describe formally such a system as quadruple: ZZR = <T, P, C, W, A} where T, P are a set of transactions and processes respectivley, C is a set of checks, W represents set of workflow results, A corresponds to the net architecture. Using this model an example of association of objects in workflow execution in the distributed system is shown in Fig.2. Then the dependability problem of workflow is considered. Such models are described in terms of dependencies between their components, states and conditions, and check procedures.

The algorithm-based fault tolerant protocol for processing workflow is defined and main bounds on data and error set cardinalities and on number of checks to achieve acceptable level of dependability is given. Creating tripartite graph we can evaluate different checking strategies and in interactive way choose some extra testing procedure to be closer the required dependability. Moreover, there is a need for various extensions to the model to selection of the check procedure given some level of system processing dependability.