Академія наук України
Інститут кібернетики імені В. М. Глушкова

На правах рукопису

ГОСТІЛОВА Світлана Валентинівна

УДК 681.324

ПРОГРАМНІ МЕХАНІЗМИ ПІДВИЩЕННЯ ВІДМОВОСТІЙКОСТІ РОЗПОДІЛЕНИХ СИСТЕМ

05.13.11 — математичне і програмне забезпечення обчислювальних машин, комплексів, систем і мереж

Автореферат
дисертації на здобуття ученого ступеня
кандидата технічних наук

Київ 1992

Робота виконана в Інституті кібернетики імені В. М. Глушкова АН України.

Науковий керівник: доктор технічних наук, професор
НІКІТІН А. І.

Офіційні опоненти: доктор технічних наук ПАРАСЮК І. М.,
кандидат технічних наук
ФУРСІН Г. І.

Провідне підприємство: Київський політехнічний інститут.

Захист відбудеться «22» січня 1993 р. о 14 год. на засіданні спеціалізованої ученої ради Д 016.45.01 при Інституті кібернетики імені В. М. Глушкова АН України за адресою:

252207 Київ 207, просп. Академіка Глушкова, 40.

З дисертацією можна ознайомитися у науково-технічному архіві інституту.

Автореферат розісланий «21» грудня 1992 р.

Учений секретар
спеціалізованої ученої ради СИНЯВСЬКИЙ В. Ф.

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ.

АКТУАЛЬНІСТЬ ТЕМИ. Розподілені системи (РС) обробки даних, що функціонують у мережному середовищі надають можливість використовувати широкий спектр інформаційних і обчислювальних послуг, мають більш високу надійність і продуктивність (можливий паралелізм обробки), а також наближують до користувача різні типі комп'ютерів. Нещодавно народився і розвивається науковий напрямок, пов'язаний з розробкою технологій розподіленої обробки даних. Існує ряд проблем, що виникли при спробі досягти указаних потенційних переваг РС і пов'язаних з ускладненням механізмів керування розподіленими децентралізованими інформаційними процесами. При цьому особливу роль відіграє створення механізмів, що забезпечуватимуть відмовостійкість РС.

В даній роботі відмовостійкість означає здатність розподіленої системи зберігати свою працездатність, а також цілістність інформації, що знаходиться в розподілених базах даних (РБД), при відмові деякої підмножини компонентів. Відмовостійкість - це та необхідна властивість, без якої практично неможливо гарантувати допустимі експлуатаційні характеристики РС.

Пов'язані з даним напрямком теоретичні проблеми були розглянуті у публікаціях вітчизняних та зарубіжних авторів, таких, як Лемпорт Л., Гарсіа-Моліна Г., Ротні Дж., Вальтер Б., Морген С., Зінов'єв Е.В., Калініченко Л.А., Нікітін А.І. та інші. Не зважаючи на досить велику кількість публікацій та отримані істотні результати в даній галузі, необхідні подальші теоретичні дослідження і практичні розв'язання багатьох питань. Вирішенню деяких з них присвячена дана робота.

МЕТА ДИСЕРТАЦІЙНОЇ РОБОТИ - дослідження і розвиток моделей механізмів забезпечення відмовостійкості РС в умовах паралельного виконання транзакцій, а також розробка на основі отриманих результатів алгоритмів, що у сукупності реалізують збереження цілісності і працездатності РБД.

Поставлені такі окремі задачі:

- узагальнити механізми забезпечення відмовостійкості РС в умовах паралельного виконання транзакцій;

– побудувати та обґрунтувати деякі нові алгоритми збереження і відтворення цілісності РБД у випадку відмов вузлів мережі для різних практично важливих припущень відносно умов роботи РС, які б відповідали сучасному рівневі надійності технічних компонентів РС;

– розробити програмний пакет моделювання систем управління транзакціями (СУТ), який би забезпечував знаходження найбільш важливих критеріїв якості такого управління.

НАУКОВА НОВИЗНА. У дисертаційній роботі розвинені та узагальнені механізми забезпечення відмовостійкості РС і на цій основі розроблений цілий ряд алгоритмів. Новими є:

– модифікований WT – протокол фіксації транзакції для N вузлів у рамках моделі служби відмовостійкості, що зібрана в єдину ієрархічну систему;

– неблокуючий WT – протокол трьохфазної фіксації, стійкий до довільних множинних відмов вузлів;

– модифікований механізм управління зарезервованими даними в рамках ієрархічної моделі служби відмовостійкості;

– узагальнення алгоритму "моментального знімку" глобального стану мережі, за Лемпортом, на основі "перефарбування" вузлів;

– модифіковані протоколи формування контрольних точок в РБД, що скорочують час простою РС;

– алгоритм верифікації глобального плану виконання субтранзакцій відносно його серіалізуємості при методі блокування.

ПРАКТИЧНА ЦІННІСТЬ РОБОТИ полягає у розробці програмної оболонки для моделювання СУТ, для дослідження динаміки практично важливих процесів і ситуацій при управлінні паралелізмом транзакцій і виявленню важливих критеріїв якості такого керування.

АПРОБАЦІЯ. Основні результати дисертаційної роботи доповідались і обговорювались на 3-й Всесоюзній конференції "Живучість і реконфігурація інформаційно-обчислювальних і керуючих систем" (м.Севастополь, 1991р.); семінарах наукової ради АН України з проблеми " Кібернетика"; Конференції молодих учених і спеціалістів Ін-ту Кібернетики ім.

В.М.Глушкова АН УРСР (м. Київ, 1988р.); 6-й Всесоюзній конференції "Обчислювальні мережі комутації пакетів" (м. Рига, 1988р.)

ПУБЛІКАЦІЇ. За темою дисертації опубліковано 4 наукові роботи.

СТРУКТУРА ТА ОБСЯГ РОБОТИ.

Дисертаційна робота складається із вступу, чотирьох розділів, висновку, списку літератури і додатку. Робота викладена на 123 сторінках машинописного тексту, містить 27 малюнків. Бібліографія складається з - 98 найменувань.

ЗМІСТ РОБОТИ.

У вступі обгрунтовується актуальність теми і формулюються конкретні задачі дослідження.

У першому розділі описані сучасні аспекти розвитку та застосування обчислювальних мереж і в зв'язку з цим нові задачі управління в РС, дані визначення основних понять, короткий огляд підходів до побудови механізмів забезпечення відмовостійкості РС, що дозволяють зберегти можливість користування нею при наявності відмов технічних або програмних засобів. Необхідно відзначити, що вказані механізми розподілені на всіх рівнях служб і протоколів моделі ВОС-МОС, причому на прикладному рівні вони, в свою чергу, можуть бути подані окремою ієрархічною моделлю. Саме на цій моделі і її компонентах зосереджені дослідження, що відображені в роботі. Включення того чи іншого механізму забезпечення відмовостійкості РС залежить як від характеру відмови технічних компонентів мережі, так і від фази обробки окремих паралельно працюючих транзакцій, яких істотно торкнулася відмова. В сукупності механізми повинні забезпечити досить швидке виявлення відмови, забезпечити успішне завершення тих транзакцій, які можуть бути "врятовані", відкіт тих транзакцій, які "врятовані" бути не можуть, і забезпечення подальшої роботи в умовах відмови. Основні проблеми при цьому - побудова системи контрольних точок РБД, забезпечення серіалізуємості глобального плану виконання субтранзакцій, узгоджена робота з зарезервованими даними. Нетривіальність розв'язання даних проблем зумовлена

припущенням, що управління мережами і РБД, яке на сьогоднішні є загальноприйнятим, децентралізоване. В кінці розділу формулюються конкретні задачі.

У другому розділі систематизовано ряд механізмів забезпечення відмовостійкості, що були запропоновані в різних публікаціях, і запропонована їх єдина ієрархічна модель. При цьому виділяються ті фази функціонування механізмів на всіх рівнях ієрархії, які пов'язані з необхідністю досягнення узгодження дій в децентралізованих РС. Таке узгодження (англ. INTERACTIVE CONSISTENCY), зведене Лемпортом в ранг однієї з парадигм РС, і є основою для розробки різних конкретних алгоритмів у рамках зазначених механізмів.

Будемо називати помилкою прояв деякої несправності (відмови) компоненту системи, який фіксується іншими її компонентами. Синонімом терміну "помилка" можна вважати термін "нерегулярна ситуація". Відмова одного і того ж технічного компоненту ЕОМ може бути причиною різних нерегулярних ситуацій. Так, наприклад, несправність процесора може проявитися у тому, що дана ЕОМ перестане реагувати на повідомлення інших ЕОМ мережі (мовчазний вузол), і в тому, що ЕОМ відповідає на повідомлення, але видає хибну інформацію. Ми припускаємо, що при відмові ЕОМ у деякому вузлі мережі даний вузол не видає іншим ЕОМ ніякої інформації до моменту свого відновлення. В літературі ця властивість відмови характеризується, звичайно, як властивість "FAIL STOP".

Припускається, що мережа, на основі якої побудована РС, забезпечує такі характеристики, як безпомилкова передача повідомлень між будь-якими двома справними вузлами та гарантія їх отримання по черзі відправлення. Службу відмовостійкості зображуємо як таку, що має чотири рівня.

Нижній рівень - рівень віртуального кільца. Віртуальне кільце призначене для швидкісного знаходження змін стану вузла (тобто відмови або відновлення) при невеликій кількості відправлених службових повідомлень, а також для необхідної простої реакції на ці події.

Віртуальне кільце утворюється таким чином. Вузли мережі

нумеруються послідовними натуральними числами (1,N). Ця нумерація і визначає кільцеву впорядкованість вузлів мережі згідно з відношенням слідування: к <- (к+1)mod N. При виконанні даного відношення вузол (к+1) будемо називати старшим, а вузол к- молодшим в цій парі (тут і нижче іменуватимемо вузол його номером у віртуальному кільці). Відзначимо, що номери присвоюються вузлам довільно, можливо, з урахуванням прагматичних міркувань, наприклад, виходячи з правила: два вузли з послідовними номерами не повинні бути занадто далеко розташовані один від одного за кількістю проміжних транзитних вузлів. Отриману згідно з введеною кільцевою впорядкованістю множину вузлів будемо називати віртуальним кільцем.

Базовий протокол рівня віртуального кільця складається з двох субпротоколів CONTROL і CHANGE.

Субпротокол CONTROL. Вузол к відправляє по віртуальному кільцю "UP"-повідомлення вузлу (к+1) і вмикає інтервальний таймер на період TSENDER, що відомий усім вузлам мережі. Кожне "UP"-повідомлення має тимчасову мітку, яка відповідає версії таблиці про стан вузлів, що дозволяє вузлам корегувати свої таблиці до отримання єдиної узгодженої версії. Коли вузол (к+1) отримує "UP"-повідомлення, він вмикає інтервальний таймер на період TCONTROL- максимально допустимий період між надходженнями двох "UP"-повідомлень: TCONTROL = TSENDER + LMAX - LMIN, де LMAX - максимальний час передачі повідомлення між двома будь-якими вузлами, LMIN - мінімальний час передачі.

Якщо нове "UP"-повідомлення не надійшло від вузла к протягом цього інтервалу часу, то таймер TCONTROL спрацьовує. Вузол к вважається недоступним, і ініціюється субпротокол CHANGE, який коректує таблицю станів (STATE-TABLE), що містить інформацію про інші вузли.

Другий рівень назвемо рівнем WATCH-служби. WATCH-служба організує слідкування за станом довільних вузлів, заданих прикладним процесом, і "доповідає" про будь-які зміни стану даному прикладному процесу. При цьому WATCH-служба, природно, спирається на інформацію, отриману від віртуального кільця. У розділі запропоновані і обґрунтовані

вирішальні правила вибору нової версії STATE-TABLE кожним вузлом. Запропонований алгоритм вибору нових версій STATE-TABLE заснований на такому припущенні: у всіх вузлах знаходяться годинники, показання яких не можуть відрізнятися один від одного більше ніж на ε одиниць часу.

$$\forall\, i,k\ (\ t_i - t_k\ ) \le \varepsilon,$$

де t_i, t_k -показання годинників вузла і і вузла к.

Провівши першу фазу субпротокола CHANGE - фазу збору недостатньої інформації, координатор (вузол к) помічає версію поточним часом свого вузла, тобто: PTS = t_k.

Нехай в деякий вузол к послідовно прийшли дві істотно різні версії STATE-TABLE з фізичними тимчасовими мітками PTS і PTS´ і нехай PTS´>PTS, тоді:

I) якщо різниця між цими тимчасовими мітками задовольняє нерівність

$$\delta = PTS' - PTS \ge 2(\ Lmax - Lmin),$$

то приймається версія STATE-TABLE, позначена PTS´;

2) якщо нерівність не виконується, то необхідне повторне виконання субпротоколу CHANGE;

3) якщо прийшли дві однакові версії, але з різними поточними мітками, то версія приймається і позначається міткою PTS´.

Третій рівень являє собою механізм управління фіксацією транзакцій, побудований за класичним протоколом двохфазної фіксації (2PC - протокол). Він використовує сервіс WATCH-служби як з боку вузла-координатора (TM), так і з боку вузла-виконавця (DM). Кожен з них звертається до WATCH-служби для слідкування за зміною стану вузлів, що беруть участь у виконанні транзакції. Завдяки WATCH-службі як вузол-координатор, так і вузли-виконавці можуть у процесі обробки транзакції реагувати на відмови. Тобто в 2PC-протоколі передбачена поведінка TM і DM на випадок знаходження відмов WATCH-службою.

У роботі запропонований WT-протокол для N-вузлів, який дозволяє скоротити кількість виданих повідомлень і станів вузлів порівняно з протоколом Вальтера, який був узятий за основу дослідження . Так, на першій фазі роботи протокола вузол-виконавець при неможливості виконання субтранзакції сам анулює її, переходить в стан BACKOUT і видає повідомлення "в". У другій фазі протокола вводиться додатковий стан чекання вузлів-виконавців WAIT, при якому відбувається узгодження їх подальших дій. Вводиться додатковий оператор з метою розмежування неактивних станів DM після фіксації або анулювання субтранзакцій. Цей протокол поряд з указаною перевагою нестійкий при множинних відмовах DM або при одночасній відмові TM і DM. У зв'язку з цим пропонується неблокуючий WT´-протокол, отриманий на базі WT-протокола способом введення додаткового повідомлення "WACK", яке надсилається координатором TM у стані готовності до фіксації COMMIT після отримання повідомлення "ACK" від усіх вузлів-виконавців, і скороченням деяких близьких за змістом станів вузлів. У роботі описані всі можливі переходи глобального стану при виконанні WT´-протокола і обґрунтована коректність останнього.

Четвертий рівень - рівень резервованих даних, призначенням якого є управління резервними даними. Зауважимо, що в процесі керування вузли, які містять ідентичну інформацію, слідкують за змінами станів один одного також за допомогою WATCH-служби. У розділі запропонований модифікований алгоритм, суть якого полягає у такому. Для кожного зарезервованого інформаційного об'єкту одна з копій оголошується основною (PRIMARY COPY). Після зміни основної копії список проведених змін (IL) надсилається до вузлів із вторинними копіями (при цьому підтранзакція знаходиться у стані DO). Останні повинні відправити підтвердження про виконання дій, вказаних в IL. В протилежному випадку (якщо зафіксована відмова вузла із вторинною копією) вузол з основною копією запам'ятовує IL у стабільній пам'яті з метою більш пізньої його доставки недосяжному вузлу із вторинною копією, викликає слідкування за відновленням вузла ("WU") і вписує відповідний запис в

таблицю UPDATE-TABLE. Цей запис містить в собі номер вузла, що вийшов з ладу, вказівник на IL, номер версії, що повинен бути збереженим. Після цих маніпуляцій стан підтранзакцій у вузлі з первинною копією змінюється на READY. Як тільки вузол із вторинною копією переходить у стан UP, він повідомляє номер своєї версії основній копії і всі втрачені IL будуть йому передані. Про відмову вузла з основною копією всі вузли досить швидко дізнаються від WATCH-служби. Підмножина вузлів, які зберігають вторинні копії, маркировані як ACTUEL, здійснює вибір нової основної копії, не чекаючи відновлення зазначеного вузла. Після відновлення колишній вузол основної копії відправляє широкомовний (серед елементів зазначеної підмножини) запит про те, де тепер зберігається основна копія, просить надіслати йому IL і починає функціонувати уже як вузол, що зберігає вторинну копію. Для обгрунтування коректності цього механізму належить зробити два зауваження. По-перше, всі вузли із вторинною копією мають одну і ту саму версію інформаційного об'єкту (це гарантує маркіровка ACTUEL). По-друге, тут не розглядається можливість фрагментації мережі. Це обмеження не дуже сильне, тому що у сучасних мережах передбачаються зарезервовані маршрути.

Третій розділ присвячений проблемі формування контрольних точок (CP) в РБД для координації відкатів локальних баз даних. Основою для розв'язання цієї проблеми є ідея визначення глобального стану РС шляхом її "моментального знімку" (SSS- SNAPSHOT STATE). Коротко описаний Лемпортом SS-алгоритм полягає у тому, що на часових вісях, що фіксують події, які виникають у процесах, проводиться переріз. При цьому, якщо позначати обмін повідомленнями між процесами стрілками, то стрілки повинні перетинати цей переказ тільки зліва направо.

Конкретно SS-алгоритм передбачає, що спочатку усі вузли "пофарбовані" в один колір (наприклад, білий). Потім ініціюючий вузол і виконує процедуру зміни кольору (PROCEDURE COLOR-CHANGE), що складається з чотирьох ступенів:

перефарбувати вузол і в червоний колір (COLOR(I)=RED);

записати стан вузла S_i в стабільну пам'ять;

по усіх вихідних каналах відправити попереджувальне повідомлення W_{ij};

записати усі надіслані у вузол білі повідомлення по усіх вхідних каналах з моменту пофарбування вузла у червоний колір до моменту надходження попереджувальних повідомлень (W_{ij}) по всіх вхідних каналах, які засвідчують про те, що всі відповідні вузли вже перефарбувалися у червоний колір і записали свій стан у стабільну пам'ять.

Решта j-х вузлів мережі, що отримали попереджувальне повідомлення W_{ij} від ініціатора i, порівнюють свій колір і на випадок його розбіжності з кольором попереджувального повідомлення виконують вищеописану процедуру COLOR.CHANGE.

Після того, як кожен вузол виконав процедуру COLOR.CHANGE, він продовжує роботу в червоному періоді. При ініціалізації наступного моментального знімку знову відбувається перефарбування вузлів і повідомлень від них у білий колір. Зазначимо, що колір вузла і його повідомлення не несе змістовного навантаження. Колір використовується в даному випадку тільки для моментального знімку стану вузлів як своєрідної мітки, яка може розмежовувати періоди (етапи) запису стану РС у процесі її безперервної роботи.

В розділі проведено дослідження коректності SS-алгоритму. Обгрунтовуються такі твердження:

чергове перефарбування закінчується за скінченний час;

якщо в процесі перефарбування один або декілька вузлів по своїй ініціативі викликають однойменний процес, то це не змінює результату SS-алгоритму;

якщо в черговому процесі перефарбування один або декілька вузлів по своїй ініціативі викличуть протилежний процес перефарбування (наступний, на їх думку, за черговим), то це не перешкодить закінченню чергового процесу.

Далі розглядається дана процедура на прикладі мережі з чотирьох вузлів і проводиться порівняльний аналіз SS-алгоритму з алгоритмом фіксації глобального стану в СУТ. SSS можна розглядати як глобальну контрольну точку (GCP) РС. Однак використання SSS як GCP порушує можливість відкату

тільки у відмовленій частині РС, якщо при цьому зберігається цілісний стан РБД. Крім того, створення GCP потребує жорсткої синхронізації процесів в РС аж до зупинки всієї системи (що може бути недопустимим в системах реального часу).

Згідно з розповсюдженим підходом, процеси формують СР незалежно один від одного і запам'ятовують їх у стабільній пам'яті. У випадку відмови процеси повинні знайти несуперечну множину контрольних точок серед сформованих. Система відмовляє і рестартує від цієї множини контрольних точок.

Цей підхід як головну вади містить в собі небезпеку "ефекту доміно". В розділі запропоновані два алгоритми простановки СР, які не підлягають впливу "ефекту доміно" і не передбачають простою РС, а також такі, що зменшують (порівняно з відомими) кількість транзакцій, які відкочуються.

Перший алгоритм полягає у простановці незалежних автономних тимчасових і постійних СР. Алгоритм допускає, що в пам'яті кожного вузла зберігається контрольна точка CP1. Сукупність СР відображає вищезазначену несуперечливу множину СР. Алгоритм включає такі кроки:

I. Кожен процес o самостійно ставить додаткову контрольну точку ACP2 тільки за умови, що від моменту створення останньої контрольної точки CP1 до створення нової контрольної точки CP2 було відіслано повідомлення m іншому процесу p. При цьому перед кожним відправленням процес o перевіряє повідомлення m на вірогідність (наприклад, спеціальним тестуванням).

2. До отримання підтвердження "ACK" про надходження повідомлення m процес o ніяких операцій не веде (простій).

3. При отриманні підтвердження "ACK" процес o записує додаткову контрольну точку ACP2 в стабільну пам'ять як постійну контрольну точку CP2 і видучає попередню контрольну точку CP1.

4. Процес p, який отримав повідомлення m, зразу ж формує контрольну точку CP2, відправляє підтвердження "ACK" і видучає CP1.

Припустимо, що вузол Q відмовив до отримання підтвердження "АСК", тоді впевнений у правильності повідомлення M (див.п.I алгоритму) він повертається до додаткової контрольної точки ACP2, відновлюється і чекає підтвердження "АСК". Припустимо, що вузол P відмовив до відправлення підтвердження "АСК". Тоді він повертається до контрольної точки CP2, відновлюється і відправляє підтвердження "АСК". Якщо ж P відмовив після відправлення підтвердження "АСК", то він повертається до CP2 і відновлюється без повторного відправлення "АСК".

Для випадку одночасної відмови процесів Q і P при відправленні і прийомі повідомлення M можливе порушення цілісності системи. Це очевидно на прикладі, коли процес Q відмовив після відправлення повідомлення M, але до створення ACP2, а процес P відмовив до отримання повідомлення M. Процес Q після відновлення від контрольної точки CP1 вдруге відправить повідомлення M, а процес P після відновлення від CP1 прийме старе і нове повідомлення з мережі. Тому введемо обмеження: припустимо, що одночасна відмова вузла, який відіслав повідомлення, і вузла, який отримав повідомлення, не може відбутися. Правомочність введеного обмеження обгрунтовується аналізом достить простої марківської моделі.

Таким чином, алгоритм, що дає несуперечний стан РС, забезпечує, по-перше, відсутність "ефекту доміно", по-друге, малу загальну довжину відкоту.

Пропонується таке координоване формування СР в РС з древовидною топологією. Така топологія мережі зв'язку часто зустрічається в ієрархічних регіональних і/або галузевих системах. Вважаємо, що ініціатором формування СР є центральний (вищий за ієрархією вузол) - корінь віртуального дерева (VT). Запит на формування СР він відправляє тільки множині безпосередньо підпорядкованих йому за ієрархією вузлів; ті, в свою чергу, - множинам підпорядкованих безпосередньо їм і т.д. Отже, елементарним актом протоколу являється протокол (1 <-> N) між одним старшим вузлом і N підпорядкованими йому вузлами більш низького рівня.

Запропоновано модифікований двоступеневий алгоритм формування контрольної точки (2SGCP), розроблені вирішальні

правила, що забезпечують термінацію і малий час роботи алгоритму:

1. Ініціатор a пропонує всім вузлам p_i сформувати глобальну контрольну точку ("chp(N)"). При цьому указується номер останнього повідомлення, отриманого a від p_i. Вузли p_i можуть висловити ініціатору свою згоду або незгоду в залежності від того, співпадає чи ні номер останнього відправленого ними повідомлення з номером, указаним в запиті ініціатору, а також залежності від відповіді нижче підпорядкованих вузлів. Для цього вузли p_i, перед тим як дати свою відповідь ("yes" або "no"), повинні запросити підпорядковані їм вузли-виконавці. І тільки якщо останні відповідають їм "yes", то до ініціатора a надходить та сама відповідь "yes". У протилежному випадку до a надходить "no".

2. Ініціатор a розсилає своє рішення про формування контрольної точки ("take-chp"), якщо всі вузли надіслали "yes", і відмову від формування контрольної точки ("undo-chp"), якщо хоча б один вузол надіслав "no". Якщо вузол – виконавець мав ненульовий номер у запиті ініціатора, тобто обмінювався повідомленнями, то він виконує намір ініціатора. Якщо вузол-виконавець мав нульовий номер у запиті ініціатора, тобто не обмінювався повідомленнями в жодним з вузлів, то він ігнорує намір ініціатора. Зауважимо, що з моменту надходження запиту від ініціатора a до моменту надходження рішення ініціатора a процеси p не обмінюються повідомленнями.

Кожного разу, коли вузол відмовив, VT повідомляє про це всі вузли системи. Тоді: якщо відмовить ініціатор a на будь-якому кроці алгоритму, то всі вузли p_i будуть чекати його відновлення. Якщо відмовив вузол p_i перед тим, як відповісти "yes" або "no", то ініціатор a буде про це повідомлений і прийме рішення "undo-chp". Тоді вузол p_i після свого відновлення повинен дізнатися про прийняте рішення від ініціатора або інших вузлів p_j. Якщо ж вузол p_i відмовив після відповіді "yes" або "no", то після свого відновлення він може самостійно прийняти рішення про "undo-chp", тільки якщо його відповідь була "no". У

протилежному випадку необхідно дізнатися про прийняте рішення від інших вузлів.

Відзначимо, якщо СР ініціюють одночасно два вузли мережі то, щоб уникнути тупикової ситуації, другий запит повинен отримати відмову. Таким чином, згідно з запропонованим алгоритмом, кожний вузол РС будує одну постійну контрольну точку за умови обміну повідомленнями, мінімізуючи при цьому витрати на пам'ять і забезпечуючи несуперечний стан даних і малу сумарну довжину відкоту.

В четвертому розділі поставлена задача створення набору засобів (функціональна оболонка) до системи СИМПАС для підтримання опису та моделювання в динаміці більшості практично важливих процесів і ситуацій в системах управління транзакціями і знаходження значень критеріїв якості такого управління (Додаток І до дисертації).

В основу розробки були покладені такі принципи:

І. Моделюванню підлягають тільки процеси, які управляються протоколами верхніх рівнів. Характеристики ж протоколів рівнів 2-4 (і конфігурації технічних засобів) задаються як початкові параметри на основі експертних оцінок.

2. Оболонка повинна підтримувати всі найбільш розповсюджені методи управління паралелізмом транзакцій і дозволяти знаходити і оцінювати міру дії різноманітних факторів і параметрів системи, а також тупикових ситуацій на ефективність роботи системи.

3. Оболонка повинна підтримувати проектування відмовостійкої системи і несуперечність даних систем на базі алгоритмів фіксації транзакцій і алгоритмів побудови контрольних точок для відновлення системи після відмов.

Зазначені функції і їх властивості реалізуються такими модулями: модуль вибору поточної події, модуль генерації транзакцій, модуль визначення необхідних DM-вузлів, модуль визначення необхідної гранули, модуль обліку часу розповсюдження повідомлень від TM до DM, модуль верифікації транзакцій перед постановкою в чергу до гранул, модуль забезпечення відмовостійкості, модуль фіксації транзакцій.

В процесі розробки оболонки були проведені додаткові

дослідження для реалізації алгоритмів перевірки глобального плану виконанню субтранзакцій щодо його серіалізуємості (така перевірка здійснюється методами, що відносяться до класу "оптимістичних"). Суть дослідження полягає в тому, що потрапивши у деякий DM, заявка на виконання субтранзакцій не одразу потрапляє в чергу до необхідної гранули, а попередньо проходить верифікацію згідно з алгоритмом " зростаючого лісу".

Процес перевірки використовує побудову в кожному вузлі TM дерева залежностей, яке являє собою підграф D-графа. Право говорити про те, що підграфи у всіх вузлах є деревами, дає нам врахування одного з правил, згідно якому, як тільки деяка транзакція стає причиною утворення циклу на графі-вова відкочується.

Алгоритм будує транзитивні замикання для дерева в кожному вузлі TM, в якому відбувається ратифікація нової субтранзакції по всім гілкам, які виникають в процесі росту дерева.

Завершення нової гілки і дерева в цілому означає побудову всіх транзитивних замикань, що гарантує відсутність циклів і можливість ратифікації нової транзакції.

Алгоритм вважається завершеним в двох випадках:
- коли в одному із вузлів TM знайдений хоч один цикл, і тоді вигляд дерев в інших TM не відіграє значення;
- коли всі гілки будуть завершені, цикл не знайдено, і шлях (проміжні стани) до цього стану байдужий.

Як експериментальний приклад імітаційного моделювання РС була досліджена модель РСБД, розподілена по п'яти DM-вузлам. Кожний вузол має чітко визначену частину даних РСБД з однаковим поділом на гранули (області блокування). Номери необхідних гранул задаються при генерації транзакції з TM. В експерименті використовувалась PP-стратегія (PRECLAIM POLICY) попередньої об'яви блокіровок. Прийнято, що транзакція не знімає жодної блокіровки до закінчення своєї роботи. Для знаходження тупиків використовувався механізм обмеження розмірів черг до кожної гранули. Для регулювання тривалості виконання транзакції використовувався механізм тайм-аута. Таким чином, враховано вплив таких

факторів:

I) закон розподілу потоку транзакцій;

2) закон розподілу часу обробки транзакцій;

3) максимальна довжина черги до гранул;

4) кількість гранул в DM;

5) тривалість тайм-аута чекання готовності DM;

6) системні параметри РСБД (зокрема, затримки при передачі).

Показники, за якими оцінена ефективність роботи системи:

I) продуктивність системи, тобто кількість транзакцій, виконаних за час моделювання;

2) процент втрат (невиконаних транзакцій), спричинених тупиковими ситуаціями або тайм-аутом.

Результати для кожного показника відображені у вигляді графіків, як функції від числа гранул.

Основні висновки.

У дисертаційній роботі були отримані такі основні результати.

У теоретичному плані:

I) модифіковано WT-протокол фіксації транзакції для N вузлів в рамках моделі служби відмовостійкості, зібраної в єдину ієрархічну систему;

2) запропоновано неблокуючий WT´-протокол трьохфазної фіксації, стійкий до довільних множинних відмов вузлів;

3) модифіковано механізм управління зарезервованими даними в рамках ієрархічної моделі системи відмовостійкості;

4) узагальнено і обґрунтовано алгоритм "моментального знімку" глобального стану мережі, за Лемпортом, на основі перефарбування вузлів;

5) розроблені протоколи формування контрольних точок в РС, в тому числі 2 модифікованих алгоритми, які скорочують час простою РС і кількість відкотів транзакцій;

6) запропоновано алгоритм верифікації глобального плану виконання субтранзакцій щодо його серіалізуємості при методі блокування.

У практичному плані:

Розроблена програмна оболонка для моделювання СУТ на базі системи СИМПАС, що підтримує аналіз динаміки практично важливих процесів і ситуацій в системах управління транзакціями і забезпечує знаходження значень найбільш важливих критеріїв якості такого управління.

Основні положення дисертації опубліковані в таких роботах:

I. Никитин А.И., Алиев А.А., Гостилова С.В. Глобальное непротиворечивое состояние распределенных баз данных. - Киев, I988. - 2Ic. - (Препр./ АН УССР. Ин-т кибернетики им. В.М.Глушкова; 88-48).

2. Никитин А.И., Гостилова С.В. Иерархия механизмов отказоустойчивости в распределенных системах// УСиМ. - I990.- №6. -С.84-93.

3. Никитин А.И., Гостилова С.В. Глобальное состояние распределенной базы данных и глобальная контрольная точка// УСиМ. - I99I. -№8. -С.68-76.

4. Никитин А.И., Гостилова С.В. Контрольная точка в РБД как метод восстановления процессов, происходящих в сетях// Тез. докл. 3-й Всесоюз. конф. "Живучесть и реконфигурация информационно-вычислительных и управляющих систем".- Севастополь, I99I.-С.I5.

Підп. до друку 19.11.92. Формат 60×84/16. Папір кн. журн. Офс. друк. Ум. друк. арк. 0,93. Ум. фарбо-відб. 1,05. Обл. вид. арк. 1,0. Тираж 100. Зам. 1815.

Редакційно-видавничий відділ з поліграфічною дільницею
Інституту кібернетики імені В. М. Глушкова АН України
252207 Київ 207, проспект Академіка Глушкова, 40