УДК 681.3.012 (045)
З 973.23-018.2

Гуменюк В.О. канд. техн. наук

ТЕХНОЛОГІЇ КЛАСТЕРНИХ АРХІТЕКТУР

Інститут інформатики Національного авіаційного університету

Широке використання кластерних структур взагалі обґрунтовано можливістю використання їх у вирішенні таких основних задач, як потужні обчислення и підтримка розподілених баз даних, особливо з надвисокими вимогами до надійності. Розглянуто паралельні бази даних, що реалізують високий рівень готовності. За допомогою моніторингу транзакцій обґрунтовано варіант із стійким навантаженням.

Постановка проблеми

Відомі численні засоби побудови надійних систем. В тому числі, дискові масиви *RAID*, наприклад, дозволяють не переривати обробку запитів до інформації, що зберігається на дисках, при виході з ладу одного або декількох елементів масиву. Використання надмірності інших підсистем сервера: резервні блоки живлення та джерела безперебійного живлення підтримують працездатність системи у випадку збоїв у мережі енергопостачання та відмови окремих її компонентів. Функціонування сервера у випадку відмови процесора можуть забезпечити багатопроцесорні материнські плати.

Але в тому випадку, коли з ладу виходить вся обчислювальна система, ефективною є кластеризація.

Як відомо, кластери умовно ділять на два класи:

- перший, у якому машини будуються цілком з стандартних складових;
- другий, у якому система має ексклюзивні або не дуже поширені складові.
- Найбільш визнаними типами кластерів є:
- системи високої надійності;
- системи для високопродуктивних обчислень;
- багатопоточні системи.

Зазначимо, що відмінності між цими типами досить умовні й часто існуючий кластер може мати ознаки, які виходять за рамки вказаних типів. Більш того, при конфігуруванні великого кластера, що використовується як система загального призначення, доводиться виділяти блоки, що виконують усі вказані функції.

У даній статті аналізуються й порівнюються архітектури кластерів, які останнім часом прийнято відносити до провідних з точки зору розв'язання проблеми забезпечення високої готовності, керованості й масштабованості.

Загальні риси кластерних структур

Структурно кластер поєднує кілька серверів, з'єднаних між собою спеціальним комунікаційним каналом, часто називаним також «системною мережею». Невід'ємною частиною кластера є спеціальне програмне забезпечення, що, власно, і вирішує проблему відновлення вузла у випадку збою, а також вирішує інші завдання, наприклад, змінює *IP*-адресу сервера додатків, якщо він вийшов з ладу й виконання перекладене на інший вузол. Кластерне ПЗ звичайно має трохи заздалегідь заданих сценаріїв відновлення працездатності системи, а також може надавати адміністраторові можливості настроювання таких сценаріїв. Відновлення після збоїв може підтримуватися як для вузла в цілому, так і для окремих його компонентів додатків, дискових томів і т. д. Ця функція автоматично ініціюється у випадку системного збою, а також може бути запущена адміністратором, якщо йому, наприклад, необхідно відключити один з вузлів для реконфігурації.

Кластери можуть мати поділювану пам'ять на зовнішніх дисках, як правило, на дисковому масиві *RAID*. Загальна файлова система дозволяє перезапускати додаток на вузлі, що вийшов з ладу, по загальній шині, вимагає певної координації доступу, для того щоб не виникало кілька

спроб одночасно змінити той самий файл на диску. Ця координація реалізується за допомогою спеціального механізму – диспетчера розподілених блокувань (*Distributed Lock Manager*, *DLM*), що закриває доступ до пристрою для всіх вузлів, крім одного, додаток якого в цей момент модифікує дані цього пристрою. Подібний механізм присутній, зокрема, на кластерах *OpenVMS*, але не реалізований в існуючих на сьогоднішній день кластерних рішеннях для *NT*. У цих системах додатки не можуть паралельно працювати з тими самими даними й загальною дисковою пам'яттю, якщо така є, вона призначається одному з вузлів у цей момент часу.

Таким чином, у кластерах, які не підтримують одночасного доступу до зовнішньої пам'яті, всі вузли є повністю автономними серверами. У випадку двох вузлів доступ до загальної пам'яті на дисках здійснюється за допомогою розділеної шини уведення/виводу – для кожного вузла шина закінчується в дисковому масиві. У кожен момент часу тільки один вузол володіє загальною файловою системою. Якщо один із серверів вийде з ладу, контроль над шиною й поділюваними дисками переходить до іншого вузла.

Замість поділюваної зовнішньої пам'яті, у кластерах може реалізовуватися принцип зеркалювання інформації, коли дані одного вузла тиражуються на дискові накопичувачі іншого.

Вузли кластера контролюють працездатність один одного й обмінюються специфічною «кластерною» інформацією, наприклад, про конфігурації кластера, а також передають дані між поділюваними накопичувачами й координують їхнє використання. Контроль працездатності здійснюється за допомогою спеціального сигналу (у літературі по кластерах він називається «*heartbeat*»), що вузли кластера передають один одному, для того щоб підтвердити своє нормальне функціонування. Зникнення такого сигналу в одному з вузлів сигналізує кластерному програмному забезпеченню про збій, що відбувся, і необхідності перерозподілити навантаження на вузли, що залишилися.

Як комунікаційний канал можуть використовуватися звичайні мережні технології (*Ethernet*, *Token Ring*, *FDDI*, *ATM*), поділювані шини уведення/виводу (*SCSI* або *PCI*), високошвидкісний інтерфейс *Fibre Channel* або спеціалізовані технології (*DSSI*, *CI*, *Memory Channel*). Вимоги до швидкодії комунікаційного каналу, залежать від ступеня інтеграції вузлів кластера й характеру роботи додатків. Скажемо, якщо додатки в різних вузлах не взаємодіють один з одним і не здійснюють одночасний доступ до дискових накопичувачів, то вузли обмінюються між собою тільки контрольними повідомленнями, що підтверджують їхню працездатність, а також інформацією про зміну конфігурації кластера, тобто додавання нових вузлів, перерозподіл дискових томів і т. д. Такий тип обміну не потребує значних ресурсів міжз'єднання й цілком може задовольнитися простим 10-мегабітним *Ethernet*.

Високошвидкісний комунікаційний канал потрібний, якщо додатка в різних вузлах кластера будуть працювати з тими самими даними. Реалізація механізму *DLM* припускає інтенсивний обмін повідомленнями між вузлами. А виходить, зажадає високої продуктивності міжз'єднання вузлів кластера.

Найпростіший варіант кластеризації – віддзеркалення, коли один з мережних серверів виступає в ролі «дзеркала» для іншого. На «дзеркальному» сервері встановлене те ж саме програмне забезпечення, що й на основному, причому актуальність цього ПО підтримується шляхом передачі даних по комунікаційному каналу, що з'єднує сервери в кластери. Єдина відмінність між двома системами полягає в тому, що дзеркальний сервер не реагує на мережні запити доти, доки не вийде з ладу основний.

Основні задачі, які вирішуються кластерами: потужні обчислення й підтримка розподілених баз даних, особливо таких, для яких потрібна підвищена надійність. Що стосується швидких обчислень, то вибір кластера, чі векторного або масово-паралельного суперобчислювача потребує додоткового аналізу особливостей рішення даного класу задач. Для підтримки паралельної бази даних з високим

рівнем готовності переваги кластера визначається насамперед можливістю побудувати унікальну архітектуру, що володіє достатньою продуктивністю, стійкістю до відмов апаратури і програмного забезпечення й при цьому легко нарощується й модернізується, але універсальними засобами, зі стандартних компонентів і за помірну ціну (незрівнянно меншу, ніж ціна унікального стійкого до відмов комп'ютера або системи з масовим паралелізмом).

Особливості архітектури UNIX-кластерів

Останнім часом передбачається значне підвищення інтересу до *UNIX*-кластерів, які у якості платформи для кластерів баз даних почали використовуватися в 1993 році. До цього часу спроби реалізації подібних систем, здійснені такими компаніями, як *IBM*, *Sequent*, *Pyramid* й ін., поширення не мали. Прогнозується, що функціональність *UNIX*-кластерів у самому найближчому майбутньому наздожене й пережене *VMS*-кластери.

На відміну від відмовостійких систем з надлишковими компонентами, а також різних варіантів багатопроцесорності, кластери поєднують відносно незалежні друг від друга машини, кожну з яких можна зупинити для профілактики або реконфігурування, не порушуючи при цьому працездатності кластера в цілому. Висока продуктивність кластера й мінімізація часу простоїв додатків досягається завдяки тому, що:

- у випадку збою ПО на одному з вузлів додаток продовжує функціонувати або автоматично перезапускається на інших вузлах кластера;
- вихід з ладу одного з вузлів (або декількох) не приведе до краху всієї кластерной системи;
- профілактичні й ремонтні роботи, реконфігурацію або зміну версій програмного забезпечення, як правило, можна здійснювати у вузлах кластера по черзі, не перериваючи роботи інших вузлів.

Parallel Server Option є спеціальним розширенням СУБД *Oracle*, орієнтованим на підтримку незначнозв'язаних комп'ютерних систем. Відзначимо, що є й спеціальні реалізації для систем з масовим паралелізмом й *SMP*. Логічна структура паралельного сервера обрана таким чином, щоб на кожному процесорі (незалежно, ототожнюється цей процесор із самостійним комп'ютером або є компонентом комп'ютера багатопроцесорного) виконувався свій екземпляр *Oracle*.

Основною вимогою, пропонованою до системи такого роду, є використання поділюваних накопичувачів. Вимоги поділюваної пам'яті не висуваються, що й уможливлює використання незначнозв'язаних і кластерних архітектур. Реалізація паралельного сервера базується на концепції розподіленого кэша, коли кожен процесор й, відповідно, кожен екземпляр працює зі своєю локальною пам'яттю (кэшем у термінах *Oracle*), але при цьому взаємодіє з іншими для обміну даними. Провідну роль у керуванні кластером грає розподілений менеджер блокувань (*Distributed Lock Manager - DLM*).

Для того щоб ефективність нарощування кластера не падала зі збільшенням кількості вузлів, необхідно такий засіб внутрикластерного взаємодії, що забезпечувало б адекватну швидкість обміну даними між вузлами кластера. Найпростіше рішення такого завдання – просто мережа *Ethernet*, *TokenRing* або *FDDI*. Однак при побудові мережі в такий спосіб неминуче наступає момент, коли продуктивності середовища стає недостатньо. Найбільш ефективні при побудові кластерів комунікаційні технології «канального» типу (коли встановлюється твердий канал між двома вузлами), але це спричиняє до квадратичного росту числа каналів при збільшенні числа вузлів. Тому найпривабливішими (за умови достатнього фінансування) виглядають системи з комутацією каналів високої продуктивності, *Fibre Channel*, наприклад. Незважаючи на те, що в консорціумі по розробці стандарту *Fibre Channel* брали участь *HP*, *Sun* й *IBM*, поки рішення на основі цієї технології пропонує тільки *Sun*.

Крім значної масштабованості, *OPS* забезпечує сервіс, що одержав назву «висока готовність» (пропонований словниками науково-технічних термінів переклад англійського *high avaliability*), власне кажучи означаючу можливість виявлення, локалізації й скасування одиноч-

них збоїв і відмов системи. При відмові однієї з обчислювальних систем, що входять у кластер (одного із прессорных модулів), черговий *SQL*-запит переадресується інший. Для підвищення надійності кластера й функціонуючої на ньому бази даних рекомендується використання поділюващих дискових масивів *RAID*, які можуть дублюватися в рамках кластера.

Для забезпечення рівномірного завантаження використаються монітори транзакцій. У даний момент доступно кілька різних продуктів, однак, найбільш популярний – *System Laboratories* монітор *Tuxedo*.

Особливості архітектури класте-ра з відбиттям пам'яті

Даний кластер (*Reflective Memory Cluster-RMC*) – це система з тиражуванням пам'яті або механізмом копіювання даних між вузлами, взаємодія вузлів у якій реалізується методом блокування (рис. 1). Копіювання пам'яті здійснюється за допомогою програмно реалізованої техніки когерентності. *RMC* - системи забезпечують більш швидку передачу повідомлень додаткам і звільняють вузли від необхідності звертання до диска для одержання тих самих сторінок пам'яті. У системі *RMC* одержання даних з пам'яті інших вузлів у сотні разів швидше, ніж обіг за ними до диска. Реальне підвищення продуктивності буде мати місце тільки в тому випадку, коли вузли дійсно розділяють дані, і додаток може цим скористатися.

RMC-системи швидше традиційних систем з передачею повідомлень по мережах, тому що, якщо зв'язок один раз

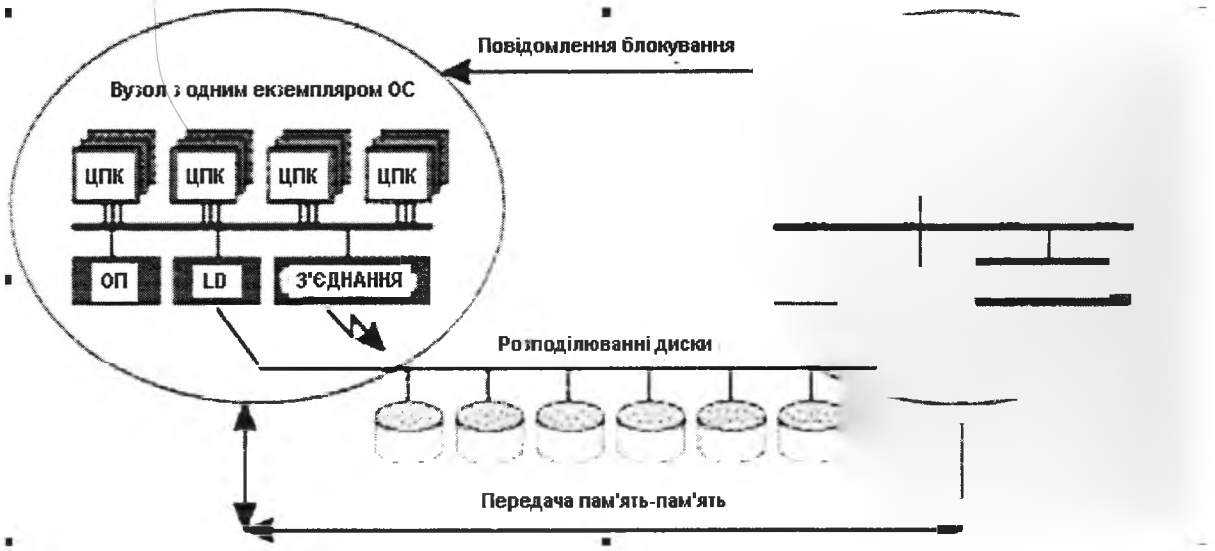

Рис. 1. Кластер з відбиттям пам'яті (*RMC*) із двома *SMP*-вузлами.

було встановлено, повідомлення може бути передане без втручання операційної системи. Копіювальна пам'ять-пам'ять тут аналогічно з'єднанням між *Mpp*-вузлами й виконує ті ж функції. Прикладами такого типу технології є *Scalable Data Interconnect* (*SDI*) компанії *Sequent*, *Memory Channel on the TruCluster* (*DEC*) і *ServerNet* (*Tandem*). Системна архітектура, побудована на базі *ServerNet*, поєднує властивості систем *NonStop* й *Integrity*, вирішуючи завдання масштабованих відмовостійких систем шляхом реалізації гнучких методів з'єднання стандартних функціональних блоків (модулів ЦП/пам'яті, підсистем зовнішньої пам'яті й комунікаційних адаптерів).

У типовій конфігурації більшість вузлів мають двопортові інтерфейси, що забезпечують приєднання кожного вузла до цих незалежних підмереж. Однієї з додаткових можливостей нової архітектури є наявність спеціальної шини когерентності, що допускає підключення до чотирьох ЦП. Ця шина забезпечує погоджений стан загальної для декількох процесорних вузлів пам'яті і їх кешів при виконанні програм, розрахованих на мультипроцесорну обробку в системі з поділюваною загальною пам'яттю.

При використанні операційних систем, у яких відсутні спеціальні засоби підтримки відмовостійкості, ця властивість може бути реалізоване за допомогою апаратних засобів шляхом створення конфігурацій ЦП у вигляді дуплексних пар. У цьому випадку пари вузлів ЦП виконують ідентичні потоки команд. Якщо один ЦП із пари відмовляє, інший продовжує працювати. Таким процесорам у мережі *ServerNet* привласнюється загальний ідентифікатор вузла, і всі пакети, адресуємі за допомогою цього ідентифікатора, дублюються й доставляються одночасно двом ЦП. При відсутності несправностей обоє ЦП у парі створюють ідентичні вихідні пакети, тому у випадку нормальної роботи логіка маршрутизації *ServerNet* може вибрати для пересилання пакети будь-якого вузла. При цьому для виявлення несправностей використаються можливості самої мережі *ServerNet*.

Основою для об'єднання архітектур є розробка головного транспортного засоба – системної мережі *ServerNet,* багатоступінчатої пакетної мережі, яку використовують для організації між процесорних зв'язків та для реалізації зв'язків з пристроями вводу-виводу. *ServerNet* забезпечує ефективні засоби для виявлення й ізоляції несправностей, а також реалізує пряму підтримку альтернативних каналів передачі даних для забезпечення неперервної роботи при наявності відмов мережі. Ця розробка надає нові можливості подальшого розвитку таких структур, в тому числі велику масштабованість з відкритими стандартами шин й покращену підтримку мультимедійних додатків. У реалізаціях *NonStop Cyclone* и *Himalaya K*10000/20000 для збільшення пропускної спроможності системи міжз'єднань застосована сегментація між процесорної шини на основі чотири процесорних секцій. Секції можуть об'єднуватися за допомогою оптоволоконних ліній зв'язку у вузлі – до чотирьох секцій у вузлі. Системи *NonStop* II, *TXP*, *VLX* та *Cyclone* підтримують також можливість побудови оптоволоконного кільця, яке дозволяє об'єднати між собою до 14 вузлів й забезпечує швидкий обмін даними у домені, який включає 224 процесори. У *Cyclone* до процесора можуть підключитися кілька каналів вводу-виводу, причому кожні чотири канали управляються своєю парою контролерів прямого доступу до пам'яті.

Усі апаратні компоненти систем *NonStop* побудовані на принципі «швидкого виявлення несправності» (*fail fast design*), у відповідності з яким кожний компонент повинен функціонувати вірно, або швидко зупинятися. У системах *Tandem* реалізація цього принципу спирається на використання методів перевірки парності, збитковості кодування або перевірки припустимості стану при виконанні кожної логічної функції. Сучасні конструкції виявлення помилок у складній логіці покладаються головним чином на методи дублювання та порівняння. Всі системи на основі мікропроцесорів для гарантії цільності даних й швидкого виявлення не справностей виконують порівняння виходів дубльованих й взаємно синхронізованих мікропроцесорів.

Одним з найбільш ефективних кластерів на *Intel*-платформі є програмне рішення *UnixWare NonStop Cluster*, плід спільних зусиль компаній *SCO* й *Tandem*. *Tandem* має великий досвід розробки відмовостійких систем. *UnixWare NonStop Cluster* поєднує надійність й можливості кластерного рішення від *Tandem* і засобу підтримки масштабованих систем *ОС UnixWare*.

Архітектура *SSI* організує роботу вузлів у кластері і їхнє подання додатку, користувачеві й адміністраторові системи, тому що якби це був один високо продуктивний *SMP*-сервер з високим рівнем доступності. *SSI* реалізує єдиний простір імен ОС *Unix*, завдяки чому додаток, фактично, не вимагає модифікації для роботи в кластері. Операційна система *UnixWare* тиражується на всі вузли кластера, і на кожному вузлі інтегрована з технологією кластеризації *NonStop Clusters*. Кластерні служби підтримують стандартний інтерфейс виклику, тому в

операційну систему не потрібно вносити зміни. Додаток здійснює доступ до кластерних засобів через стандартні системні бібліотеки *Unix*, які, у свою чергу використають інтерфейс сервісних викликів. Таким чином, клас-терна конфігурація виявляється зовсім прозорою для додатків, які працюють як із потужним *n*-процесорним *SMP*-серве-ром.

Висока доступність забезпечується двома способами. По-перше, *UnixWare NonStop Clusters* підтримує використання *SMP*-серверів як вузлів. По-друге, це можливість відновлення після збоїв технології *NonStop Clusters*, а також деякі елементи забезпечення відмовостійкості, такі як дублювання в режимі *L*-гарячого резерву: ряду апаратних компонентів, наприклад, дискових накопичувачів й елементів живлення, а також високо-надійних компонентів ядра ОС *UnixWare*. Архітектура *NonStop Clusters* підтримує відновлення додатків після збоїв за принципом «*n*+1», відповідно до якого резервна копія додатка може бути перезапущена на декількох вузлах кластера. Тобто один вузол кластера може виконувати резервні функції для всіх інших вузлів. У багатьох кластерних рішеннях, що підтримують більше двох вузлів, наприклад, в *HP ServiceGuard*, реалізований підхід «1+1», коли кожен активний вузол у кластері повинен мати свій власний резервний сервер.

UnixWare NonStop Clusters допускає динамічний розподіл навантаження як для додатків, так і для мережного трафіку. Це дозволяє домагатися оптимальної загальної продуктивності системи у випадку збою й високий рівень масштабованості. Цьому ж сприяє й принцип «активної міграції процесів», що означає можливість перенести додаток на будь-який інший вузол між виконанням команд. Таким чином, підтримується видалення й додавання вузлів у кластер у динамічному режимі.

Висновки порівняльного аналізу архітектур провідних видів кластерів

Найефективніше рішення проблеми забезпечення високої готовності, керованості й масштабованості, що засновано насамперед на реалізації архітектури *Single System Image* і розробленої *Tandem* високопродуктивної системної мережі *ServerNet*, яке поєднує вузли кластера на основі технології комутації. У даний момент можливості *ServerNet* забезпечують кластеризацію до шести *SMP*-серверів на платформі *IA*-32, але передбачено збільшити число вузлів до 32 і гарантувати кластеризацію серверів на платформі *IA*-64, як тільки почнеться їхній масовий випуск.

Список літератури

1. *Воеводин В. В., Воеводин Вл. В.* Параллельные вычисления. – С.Пб.: БХВ-Санкт-Петербург, 2002. – 357 с.

2. *Корнеев В. В.* Параллельные вычислительные системы. –М.: Нолидж, 1999. – 320 с.

3. *Корнеев В. В.* Современные микропроцессоры. – С.Пб.: БХВ-Санкт-Петербург, 2003. – 440 с.