

УДК 621.391.8

Коновалов О. Ю., к.т.н. (Київський коледж зв'язку)

ПЛАНУВАННЯ АРХІТЕКТУРИ РОЗПОДІЛЕНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

Коновалов О. Ю. Планування архітектури розподілених обчислювальних систем. У роботі розглянуті питання планування архітектури розподілених обчислювальних систем і реалізація механізмів розподілу навантаження в розподілених обчислювальних системах. Виконаний аналіз різних підходів до мінімізації міжпроцесорних взаємодій при організації обчислювального процесу, розподілу ресурсів, мінімізації простоїв обчислювальних вузлів і забезпечення їх постійним звантаженням.

Ключові слова: розподілена обчислювальна система, Grid-системи, планування архітектури, міжпроцесорні взаємодії, розподіл завдань

Коновалов А. Ю. Планирование архитектуры распределенных вычислительных систем. В работе рассмотрены вопросы планирования архитектуры распределенных вычислительных систем и реализация механизмов распределения нагрузки в распределенных вычислительных системах. Выполнен анализ разных подходов к минимизации межпроцессорных взаимодействий при организации вычислительного процесса, распределения ресурсов, минимизации простоев вычислительных узлов и обеспечение их постоянной нагрузкой.

Ключевые слова: распределенная вычислительная система, Grid-системы, планирование архитектуры, межпроцессорные взаимодействия, распределение заданий

Konovалov O. Iu. Planning the architecture of distributed computing systems. The paper discusses the issues of planning the architecture of distributed computing systems and the implementation of mechanisms for distributing the load in distributed computing systems. The analysis of different approaches to minimize interprocessor communication in the organization of the computational process, the allocation of resources, minimizing downtime computing nodes and providing them with constant load.

Keywords: distributed computing system, Grid-systems, architecture planning, interprocessor communication, distribution of tasks

Вступ. Розробка розподілених обчислювальних систем (РОС) базувалась на різних підходах, що залежали від існуючих програмно-апаратних рішень, в основі яких лежала концепція створення розподілених середовищ для розробки і виконання локальних застосувань, що в свою чергу, забезпечувало доступ до мережевих ресурсів. Швидкий розвиток і збільшення швидкодії комп'ютерних мереж зумовив перехід від локальних до територіально-розподілених мереж обробки даних. Розподілені обчислення (РО) на базі глобальних комп'ютерних мереж є програмно-апаратною інфраструктурою, яка забезпечує надійний, сумісний, повсюдний і недорогий доступ до обчислювальних ресурсів великої потужності; виконує координований розподіл ресурсів на кожному етапі обчислень і роботу в інтерактивному режимі для всіх користувачів.

В процесі проектування РОС багатоагентна технологія дозволяє поєднати в єдиній системі як протокольне, так і будь-яке прикладне програмне середовище обробки і взаємодії з різними типами даних. Така система має гнучкість, масштабованість і ефективність розподілу навантаження між серверами.

При організації систем доступу до інформаційних ресурсів [1], особливо систем, орієнтованих на використання в онлайн-режимі в мережі Internet, одними з ключових технічних характеристик системи, що визначають зручність її використання, являється час відгуку на запит і надійність системи, що зумовлене в багатьох випадках об'єднанням

великої кількості територіально розподілених комп'ютерів в одну територіально розподілену обчислювальну систему.

Метою даної статті є огляд методів планування архітектури РОС та взаємодії обчислювальних середовищ із клієнтськими додатками.

При плануванні архітектури РОС виділяють наступні технології забезпечення паралельної роботи :

- симетричні багатопроесорні системи (SMP – symmetrical multiprocessing);
- кластерні конфігурації;
- розподілені обчислювальні системи (Grid).

SMP вимагає підтримки як з боку апаратури, так і з боку операційної системи (ОС), а кластери і Grid-середовища більше залежать від способів організації мережевої взаємодії.

Якщо розглядати функції ядра ОС, то такі процеси, як підтримка кластерів і РОС, залежать значною мірою від ефективності роботи в мережі. Взагалі, будь-яку сучасну високопродуктивну обчислювальну систему можна представити як набір багатопроесорних обчислювальних вузлів, що пов'язані між собою комунікаційними мережами.

Важлива загальна характеристика таких систем – логічна організація оперативної пам'яті, з якою працюють обчислювальні вузли.

Оперативна пам'ять може бути:

- такою, що розділяється для усіх вузлів;
- розподіленою – доступною тільки для процесорів свого вузла;
- розподіленою що розділяється – доступною для процесорів свого вузла і з інших вузлів, але із застосуванням спеціальних програмно-апаратних засобів.

При розробці апаратних засобів і програмного забезпечення (ПЗ) подібних систем недостатньо лише додаткового резервування складових, оскільки окрім високої надійності при роботі в штатному режимі увесь програмно-апаратний комплекс повинен забезпечувати передбачувану і безпечну поведінку у разі виходу з ладу окремих компонентів.

Технічні обмеження, умови управління такими системами, що накладаються конкретними об'єктами, а також вимоги, що регламентуються державними, галузевими і внутрішніми стандартами підприємств, не дозволяють знайти повністю готове універсальне рішення, проте дозволяють запропонувати базову архітектуру апаратного комплексу і принципи організації ПЗ, достатні для побудови надійних і безпечних систем.

Ще на етапі проектування, ускладнення логіки роботи може значно утруднити оцінку характеристик створюваної системи, основними з яких є надійність, безпека і живучість [2].

І якщо у випадку апаратної частини, складеної із стандартних компонентів, можна скористатися даними про середній час напрацювання на відмову, терміні служби і т. п., що надаються виробниками, то для програмної частини використання статистичного підходу, принаймні на першому етапі, непридатне, зважаючи на унікальність системи управління і запланованого штучного використання. Таким чином існує необхідність розробки принципів організації ПЗ, що дозволяють складати складні і логічно несуперечливі програмні комплекси з тих відносно простих модулів, що піддаються аналізу.

При проектуванні архітектури РОС також можна вирішувати завдання, у яких результати підзадач взаємозв'язані між собою. Проте при реалізації таких завдань в комп'ютерних мережах існує проблема ефективного розподілу завдань між процесорами.

Суть проблеми полягає в тому, що при розподілі завдань між комп'ютерами на продуктивність системи впливають два конфліктуючі чинники – рівномірне завантаження [3] і міжкомп'ютерні взаємодії [4].

Існує також підхід, при якому система вважається однорідною [5], а також пропонується модель ефективного розподілу завдань в розподіленій системі з урахуванням завантаженості комп'ютерів і міжкомп'ютерних взаємодій в неоднорідному середовищі [6].

Такі підходи орієнтовані на використання існуючих комп'ютерних систем і мереж і покликані в першу чергу підвищити ефективність роботи існуючої системи шляхом розширення її обчислювальних можливостей [7].

Проаналізувавши роботи в цій області, можна констатувати, що реальна робота із створення і використання РОС сьогодні ведеться по трьох напрямках [8].

Перший напрям – це створення універсальних розподілених обчислювальних середовищ, над якими працюють основні виробники ПЗ.

Другий напрям полягає в розробці інструментарію для організації РО.

Третій напрям виходить з першого, якщо універсальність середовища замінити чіткою орієнтацією на конкретні завдання. Наприклад, створення спеціалізованих РОС для вирішення невеликого набору багаторазово використовуваних складних обчислювальних завдань. Така постановка набагато реалістичніша, оскільки специфіка завдання відома заздалегідь, що допомагає спроектувати ефективний інструмент для її вирішення.

Універсальні середовища є перспективним напрямом, але їх створення – дуже складний і трудомісткий процес. Тому потрібний простий інструментарій, який допоміг би швидко створювати розподілені застосування і використовувати доступні обчислювальні ресурси. При такому підході, вирішення задачі інтеграції обчислювальних ресурсів комп'ютерів різних організацій, найбільш серйозними є дві проблеми [9]:

- складність адміністрування комп'ютерів різних установ, організацій та ін;
- раціональне використання так званої низькоякісної обчислювальної потужності, зосередженої в мережі.

Складність використання комп'ютерів, що належать різним організаціям, полягає в значних труднощах при забезпеченні адміністративного доступу для сторонніх користувачів до ОС і файлів.

З іншого боку, ОС робочої станції, що використовується як обчислювальний вузол розподіленої системи, конфігурується відповідно до вимог досягнення максимальної продуктивності. При роботі на такому комп'ютері для досягнення максимального контролю над його ресурсами дотримується найсуворіша дисципліна доступу. Так досягаються керованість і цілісність системи.

Аналіз існуючих на сьогодні технологій і програмних засобів, що дозволяють вирішувати обчислювальні завдання в РОС, показує, що процес розробки додатків для вирішення завдань з використанням мережі як обчислювального ресурсу є складним, оскільки містить безліч етапів, починаючи від розробки паралельного алгоритму і закінчуючи моніторингом ресурсів і розподілу навантаження.

Гетерогенність складу обчислювальних вузлів і непередбачувані зміни обчислювального середовища під час рішення задачі призводять до проблеми раціонального використання обчислювальної потужності, зосередженої в мережі [10].

Проблема розподілу навантаження в паралельних обчисленнях є однією з найважливіших, особливо в динамічно змінюваному середовищі. І саме від вирішення цієї

проблеми в основному залежить ефективність паралельного вирішення задачі, тобто виграш в часі, який можна отримати в порівнянні з послідовною моделлю [11].

Деякі з даних засобів реалізують методи розподілу навантаження, що знижує трудомісткість розробки додатків [12]. Але ці методи не забезпечують раціонального використання ресурсів мережі, оскільки не виключають простою комп'ютерів.

Різні системи використовують різні алгоритми розподілу навантаження в системі, і на сьогодні немає оптимального рішення задачі використання обчислювальних ресурсів мережі з урахуванням мінімізації часу рішення задачі і накладних витрат.

Раціональне використання таких ресурсів визначається нижче перерахованими критеріями [13]:

- необхідність використання усіх доступних обчислювальних ресурсів;
- мінімізація простоїв обчислювальних вузлів і забезпечення їх постійним завантаженням;
- мінімізація накладних витрат;
- забезпечення безпеки;
- забезпечення надійності.

На сьогодні найширше застосовуються статистичні методи розподілу навантаження [14]. У статистичному методі усе завдання ділиться на підзадачі однакового розміру. При цьому вирівнювання навантаження відбувається за рахунок кількості підзадач, що переглядаються кожним комп'ютером. Кожен комп'ютер після обробки чергової підзадачі просить наступну.

Чим більша потужність комп'ютера, тим більше підзадач він устигає обробити. Таким чином, швидкий комп'ютер обробляє більше число, а повільний менше число підзадач. Це досить відомий підхід до розподілу навантаження. Але ця методика не дозволяє збалансувати величини накладних витрат (вартість обчислювальних ресурсів, що орендуються, пам'ять, канал зв'язку та ін.) і час рішення підзадачі. Накладні витрати при організації обчислень однієї підзадачі для різних комп'ютерів будуть приблизно однакові.

Отже, величина накладних витрат залежить від кількості підзадач, а кількість підзадач залежить від розміру однієї підзадачі. Таким чином, збільшуючи розмір підзадачі, ми одночасно зменшуємо їх кількість, а отже, і накладні витрати.

Але, з іншого боку, збільшуючи розмір підзадачі, ми збільшуємо величину часу рішення задачі, яка визначається часом, що пройшов з моменту закінчення роботи першим комп'ютером до моменту закінчення роботи останнім комп'ютером. При цьому час рішення задачі дорівнюватиме часу рішення однієї підзадачі на найповільнішому комп'ютері.

Для мінімізації часу рішення задачі і накладних витрат використовується динамічний (адаптивний) метод розподілу навантаження з урахуванням коефіцієнта завантаженості комп'ютерів [15].

При створенні розподіленого середовища необхідно враховувати ряд важливих вимог до обчислювального середовища, до яких належать надійність і безпека. Аналіз різних підходів до організації РО на базі локальних і глобальних комп'ютерних систем показує, що разом з перевагами ці підходи мають ряд недоліків, пов'язаних з питаннями забезпечення надійності [16]. Поза сумнівом, що об'єднання великої кількості географічно розподілених комп'ютерів в єдину систему пред'являє жорсткі вимоги до надійності її функціонування. Тому при організації РО необхідно враховувати цей важливий чинник.

Також при організації РО на базі глобальних комп'ютерних систем потрібно враховувати і питання безпеки [17]. Об'єднання великої кількості географічно розподілених комп'ютерів

в єдину розподілену систему обчислень пред'являє жорсткі вимоги до забезпечення безпеки рішення задачі (завдання і дані не повинні втрачатися і мають бути захищені від несанкціонованого доступу до них).

Для того, щоб забезпечити надійний захист ресурсів розподіленої системи, треба здійснити низку організаційно-технічних заходів. У числі цих заходів особливу роль грають міжмережеві екрани, які є основним засобом контролю зовнішнього доступу злоумисників до ресурсів комп'ютерної мережі. Застосування міжмережевих екранів для безпеки інформації в РОС вимагає здійснення оперативного налаштування параметрів на основі існуючих аутентифікаційних правил між комп'ютерами мережі, а також координування їх функціонування з урахуванням вимог політики безпеки.

Нині стандартною схемою резервування в інформаційних системах є дублювання серверів. Якщо основний сервер виходить з ладу або обмін даними з ним припиняється, резервний сервер бере на себе його функції і починає обробляти запити клієнтів [18].

Ця архітектура, незважаючи на пропоноване значне спрощення розробки ПЗ і підвищену швидкодію, не позбавлена ряду істотних недоліків. Можливість хоча і рідкісних колізій в каналі зв'язку і затримок ПЗ при отриманні запиту і формуванні відповіді, а також іноді виникаюча необхідність декількох повторних передач примушує збільшувати час для прийняття рішення [19].

Переважає більшість РОС призначена для використання з ОС Windows, а ті що залишилися, або не складають їм конкуренції в плані функціональності, або розроблені для якої-небудь конкретної ОС і не є багатоплатформеними.

Це є серйозною перешкодою при розробці розподілених систем, в яких до деяких вузлів пред'являються підвищені вимоги по відмовостійкості (що вимагає використання рішень, що добре зарекомендували себе, на базі ОС QNX або GNU/Linux), а другорядні вузли, що реалізують тільки функції спостереження, можуть бути стандартними персональними комп'ютерами із звичнішою для користувачів і обслуговуючого персоналу ОС Windows.

При проектуванні РОС, яке має багато спільного з проектуванням ПЗ загалом, все ж слід враховувати деякі специфічні особливості і наступні характеристики РОС [20]:

- спільне використання ресурсів – РОС допускають спільне використання як апаратних (жорстких дисків, принтерів), так і програмних (файлів, компіляторів) ресурсів;
- відкритість – це можливість розширення системи шляхом додавання нових ресурсів;
- паралельність – у РОС декілька процесів можуть одночасно виконуватися на різних комп'ютерах в мережі. Ці процеси можуть взаємодіяти під час їх виконання;
- масштабованість – можливість додавання нових властивостей і методів;
- відмовостійкість – декілька комп'ютерів дозволяють дублювати інформацію, що забезпечує стійкість до деяких апаратних і програмних помилок. Розподілені системи у разі помилки можуть підтримувати часткову функціональність. Повний збій в роботі системи відбувається тільки при мережевих помилках;
- прозорість – користувачі мають повний доступ до ресурсів в системі, в той же час від них прихована інформація про розподіл ресурсів у системі.

Розподілені системи мають і ряд недоліків:

- складність – набагато важче зрозуміти і оцінити властивості розподілених систем в цілому, їх складніше проектувати, тестувати і обслуговувати. Також

продуктивність системи залежить від швидкості роботи мережі, а не окремих процесорів. Перерозподіл ресурсів може істотно змінити швидкість роботи системи;

- безпека – доступ до системи можна отримати з декількох різних машин, повідомлення в мережі можуть бути видимими і перехоплюватися. Тому в розподіленій системі набагато важче підтримувати безпеку;
- керованість – система може складатися з різнотипних комп'ютерів, на яких можуть бути встановлені різні версії ОС. Помилки на одній машині можуть поширитися непередбачуваним чином на інші машини;
- непередбачуваність – реакція розподілених систем на деякі події непередбачувана і залежить від повного завантаження системи, її організації і мережевого навантаження;

Аналіз цих недоліків дає можливість прогнозувати, що при проектуванні розподілених систем виникає ряд проблем, які потрібно враховувати розробникам:

- ідентифікація ресурсів – ресурси в розподілених системах розташовуються на різних комп'ютерах, тому систему імен ресурсів слід продумати так, щоб користувачі могли без зусиль відкривати необхідні ресурси і посилатися на них. Прикладом може служити система URL (уніфікований покажчик ресурсів), яка визначає імена Web – сторінок;
- комунікація – універсальна працездатність Internet і ефективна реалізація протоколів TCP/IP в Internet для більшості розподілених систем служать прикладом найбільш ефективного способу організації взаємодії між комп'ютерами. Проте в деяких випадках, коли потрібна особлива продуктивність або надійність, можливе використання спеціалізованих засобів;
- якість системного сервісу – цей параметр відбиває продуктивність, працездатність і надійність. На якість сервісу впливає ряд чинників: розподіл процесів, ресурсів, апаратні засоби і можливості адаптації системи;
- архітектура ПЗ – описує розподіл системних функцій по компонентах системи, а також розподіл цих компонентів по процесорах. Якщо необхідно підтримувати високу якість системного сервісу, вибір правильної архітектури є вирішальним чинником.

Завдання розробників РОС – спроектувати програмне і апаратне забезпечення так, щоб врахувати і забезпечити усі необхідні характеристики системи. А для цього вимагається знати переваги і недоліки різної архітектури РОС. Виділяється три типи архітектур:

- архітектура клієнт/сервер – у цій моделі систему можна представити як набір сервісів, що надаються серверами клієнтам. У таких системах сервери і клієнти значно відрізняються один від одного;
- триланкова архітектура – у цій моделі сервер надає клієнтам сервіси не безпосередньо, а за допомогою сервера бізнес-логіки;
- архітектура розподілених об'єктів – в цьому випадку між серверами і клієнтами немає відмінностей і систему можна представити як набір взаємодіючих об'єктів, місце розташування яких не має особливого значення. Між постачальником

сервісів і їх користувачами не існує відмінностей. Ця архітектура широко застосовується нині і носить також назву архітектури веб-сервісів.

Веб-сервіс – це застосування, доступне через Internet і надає деякі послуги, форма яких не залежить від постачальника (оскільки використовується універсальний формат даних – XML) і платформи функціонування. Зараз існує три різні технології, що підтримують концепцію розподілених об'єктних систем. Це технології EJB, CORBA і DCOM.

Висновки. ROC використовуються у багатьох корпоративних інформаційних системах, об'єднуються телекомунікаційними каналами і використовують розподілені бази даних.

Існують відпрацьовані методи розподілу даних і управління розподіленими даними, архітектурні підходи, що забезпечують масштабованість систем, і реалізують принципи багатоланкової архітектури “клієнт-сервер”, а також архітектури проміжного шару.

Починає застосовуватися на практиці мобільна архітектура. Це відноситься як до систем баз даних, так і до додатків Web. Відроджується підхід до побудови розподілених систем, заснований на одноранговій архітектурі (Peer to Peer), при якому, на відміну від домінуючої сьогодні в розподілених системах архітектури “клієнт-сервер”, ролі взаємодіючих сторін в мережі не фіксуються. Вони призначаються залежно від ситуації в мережі, від завантаженості її вузлів.

У роботі розглянуті питання планування архітектури розподілених обчислювальних систем і реалізація механізмів розподілу навантаження в розподілених обчислювальних системах. Виконаний аналіз різних підходів до мінімізації міжпроцесорних взаємодій при організації обчислювального процесу, розподілу ресурсів, мінімізації простоїв обчислювальних вузлів і забезпечення їх постійним завантаженням.

Література

1. Федотов В. Б. Построение распределенной системы доступа к информационным ресурсам на основе многоагентной архитектуры / В. Б. Федотов [Электронный ресурс] // – Режим доступа: http://www.benran.ru/Magazin/cgi-bin/Sb_03/pr03.exe?!10
2. Надежность в технике. Основные понятия. Термины и определения. ГОСТ 27.002–89. – М.: Государственный Комитет СССР по управлению качеством продукции и стандартам. Издательство стандартов, 1990.
3. Лобков С. Н. К вопросу оценки одного метода оптимального распределения заданий в многопроцессорных вычислительных системах / С. Н. Лобков, С. Б. Ревин // Изв. вузов Северо-Кавказского региона. Естественные науки. – 1998. – №2. – С. 34-42.
4. Лобков С. Н. Подход к минимизации межпроцессорных взаимодействий при организации вычислительного процесса в многопроцессорных системах / С. Н. Лобков, В. А. Фатхи // Изв. Российской академии наук. ТиСУ. – 2003. – №6. – С.106-112.
5. Zhang Weiqi, Hvang Yaping, Xia Hongshan, etc. WWW-based distributed computing // Aeron. and Astronaut. – 2000. – 32, №6. – P.686-690.
6. Алгулиев Р. М. Подход к оптимальному назначению заданий в распределенной системе / Р. М. Алгулиев, Р. М. Алыгулиев, Р. К. Алекперов // Проблемы управления и информатики. – 2004. – №5. – С.140-145.

7. Боголепов Д. К. Использование ресурсов GPU для выполнения распределенных вычислений / Д. К. Боголепов // Научный сервис в сети Интернет: решение больших задач: Труды Всероссийской научной конференции (22–27 сентября 2008 г., г.Новороссийск). – М.: Изд-во МГУ, 2008. – С. 322-325.
8. Филамофитский М. П. Система поддержки метакомпьютерных расчетов X-COM. Архитектура и технология работы / М. П. Филамофитский // Вычислительные методы и программирование (РФ). – 2004. – Т.5. – С.1-9.
9. Крюков Ю. А. Вычислительная инфраструктура для прикладных задач, будущее и настоящее / Ю. А. Крюков // Геоинформатика (РФ). –2004. – №9. – С.57-61.
10. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Ван Стеен. – СПб.: Питер, 2003. – 877 с.
11. Степанченко И. В. Алгоритмы самоорганизации распределенной вычислительной системы / И. В. Степанченко // Труды второй международной конференции «Параллельные системы вычисления и задачи управления». Москва, 4-6 октября 2004. – С.164-173.
12. Lawton George. Distributed net application create virtual Supercomputers // Computer. – 2000. – 33, №6. – P.16-20.
13. Топорков В. В. Оптимизация распределения ресурсов в системах жесткого реального времени / В. В. Топорков // Известия Академии наук. Теория и системы управления (РФ). – 2004. – №3. – С. 61-71.
14. Степанченко И. В. Адаптивный алгоритм назначения подзадач в гетерогенной распределенной вычислительной системе / И. В. Степанченко // Математические методы в технике и технологиях: Сб. трудов XVI международной научной конференции, Ростов-на-Дону.: под общ. ред. В. С. Балакирева / – 2003. – Т.6. – С.119-122.
15. Фомичев В. С. Методы динамической загрузки компьютеров при параллельном решении задач в сети / В. С. Фомичев, А. Н. Смирнов // Сб. докл. V международной конференции по мягким вычислениям (SMS 2002). – СПб.: «Гидрометеиздат». – 2002. –Т.1. – С. 274-277.
16. Алгулиев Р. М. Об одном методе организации надежной распределенной вычислительной среды с применением мультиагентной технологии / Р. М. Алгулиев, Р. К. Алекперов, И. М. Алиев // Труды Института системного анализа Российской академии наук (ИСА РАН). Сетевые и алгоритмические задачи распределенных вычислений. – М.: 2004. С.166-172.
17. Управление метакомпьютерными системами [А. Киселев, В. Корнеев, Р. Семенов и др.] // Открытые системы (РФ). – 2005. – №2. – С.11-16.
18. Димаки А. В. Интегрированные системы проектирования и управления / А. В. Димаки. – Томск: Кафедра информационно-измерительной техники, 2005. – 183 с.
19. Конотоп Д. Г. Механизм отложенной передачи в системах диспетчерского контроля и управления / Д. Г. Конотоп, А. Р. Стефанюк А.Р., В. П. Яхно // Промышленные контроллеры. АСУ (РФ). – 2008. – Т.7. – С. 4-10.
20. Kate Matsudaira. Scalable Web Architecture and Distributed Systems [Электронный ресурс] // – Режим доступа: <http://www.aosabook.org/en/distsys.html>.