МІНІСТЕРСТО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

К.М. Марченко, О.Г. Собінов, О.В. Оришака, В.В. Босько

КОМП'ЮТЕРНІ СИСТЕМИ

Навчальний посібник

Кропивницький
2022

УДК 004.7

Рекомендовано вченою радою Центральноукраїнського національного технічного університету,
протокол №8 від 26 квітня 2022 року

Комп'ютерні системи : навч. посіб. // Укладачі: К.М. Марченко, О.Г. Собінов, О.В. Оришака, В.В. Босько. – Кропивницький: ЦНТУ., 2022. – 178 с.

Навчальний посібник надає теоретичний та практичний матеріал про організацію комп'ютерних систем, засоби та методи обробки даних в комп'ютерних системах. Розглянуті питання архіиектури комп'ютерних систем, принципи побудови коунікаційних середовищ. види процесорів, що використовуються в комп'ютерних системах та особливості їх роботи. Також приділено увагу новітнім підходам тп технологіям побудови комп’ютерних систем, зокрема, квантовим обчислювальним системам, клітинним та ДНК-процесорам. Приділено увагу надійності та відмовостійкості комп’ютерних систем.

Призначений для студентів, що навчаються за спеціальністю «Комп’ютерна інженерія». Для полегшення засвоєння наведеного матеріалу надаються ілюстрації та приклади.

УДК 004.7

ЗМІСТ

1. ВСТУП ДО КОМП'ЮТЕРНИХ СИСТЕМ

1.1. Основні поняття системи

СИСТЕМА — це сукупність елементів довільного природного походження, що знаходяться в відношеннях та зв'язках один з одним, і які створюють визначену цілістність.

Рисунок 1 - Що є системою, а що нею не є

Енергія зв'язків між елементами системи перевищує енергію їх зв’язків з елементами інших систем, тим самим формуючи систему як цілістне утворення. Категорія системи встановлює **онтологічне ядро** системного підходу.

Форми об'єктивації цієї категорії в різних варіантах підходу - різні та визначаються теоретико-методологічними представленями, що визначаються, і засобами, які для цього використовуються.

Виключне різноманіття уяв про систему в людському пізнаванні породжує прагнення приведення характеристик системи до деякого мінімуму. При всьому різномаїтті тлумачень, розуміння системи в самому загальному плані традиційно включає в себе уявлення про єдність і цілістність взаємопов'язаних між собою її елементів, тобто

передбачає розгляд системи як об'єкта, перше за все, з точки зору цілого.

В **семантичне поле** такого розуміння входять термини:

«елемент»,

«ціле»,

«єдність»,

«зв'язок»,

«взаємодія»,

а також **«структура»** — схема св'язків між елементами системы.

Структура системи передбачає упорядкованність, організацію, пристрій, які обумовлені характером взаємовідношень між елементами та її (системи) взаємовідносини з зовнішнім середовищем, у яких проявляються дві протилежних властивості системи: обмеженність (зовнішня властивість системи) та цілістність (внутрішня властивість системи).

Поняття системи має надзвичайно широку область застосування (практично кажний об'єкт може бути розглянуто як систему), тому достатньо повне розуміння категорії системи передбачає побудову родинно відповідних визначень — як змістовних, так і формальних. Лише у рамках такої родинних визначень вдається відобразити **основні ознаки систем** та відповідні їм **системні принципи:**

Цілістність — визначена незалежність системи від зовнішнього середовища та від інших систем; визначена залежність кажного елемента, властивості та відношення системи від його місця, функцій і так далі всередені цілого.

Зв'язність — наявність зв'язків та відношень, які дозволяють за допомогою переходів по них від елемента до елементу з'єднати два будь яких елемента системи;

Структурність — можливість опису системи через встановлення її структури, по іншому схема св'язків та відношень; обумовленість поведінки системи не стільки поведінкою її окремих елементів, скількі властивостями її структури.

Ієрархічність — кожний компонент системи, у свою чергу, може розглядатися як система, а та що досліджується, у такому випадку, система є у собі одним з компонентів більш широкої системи.

Функція — наявність можливостей, при цьому які не є простою сумою можливостей элементів, які входять до системи; принципова незнанність (ступінь незнанності) властивостей системи до суми властивостей її елементів, що зветься емерджентністю.

Емерджéнтність (англ. emergence — виникнення, поява нового) в теорії систем — наявність у будь-якої системи особливих властивостей, не властивих її підсистемам і блокам, а також сумі елементів, не пов'язаних системоутвірними зв'язками; неможливість зведення властивостей системи до суми властивостей її компонентів.

У силу принципової складності кожної системи її адекватне пізнання вимагає побудови великої кількості різних моделей, кожна з яких описує лише визначений аспект системи, тому постає питання про множинність опису кожної системи

Відповідноно до вказаного підходу, загальну схему компонентів системи можна представити наступним чином:

Елемент системи - неподільна частина системи, що характеризується конкретними властивостями, які визначають її у

даній системі однозначно. Множина що складає єдність елементів, їх св'язків і взаємодій між собою та між ними, та зовнішнім середовищем, створюють притаманну системі цілістність, якісну визначенність та цілеспрямованність.

Кількість різних елементів та їх взаємозв'яків, які містить у собі система, визначають її складність.

Зв'язки системи. Сукупність залежностей властивостей одного елемента від властивостей інших елементів системи: односторонніх; двосторонніх, багатосторонніх. Свя'зки визначають важливий для системи порядок обміну між елементами речовиною, енергією, інформацією. Найпростішими св'язками є послідовні і паралельні з'єднані елементів та додатний і від'ємний зворотній св'язок. У складних системах особливе значення мають інформаційні зв'язки, але не менше важливі та энергетичні і матеріальни связки. Складна сукупність св'язків у подібних системах створює таку властивість як ієрархічність, яка притаманна не тільки побудові, морфології системи, но і її поведінці: окремі рівні системи обумовлюють визначенні аспекти її поведінки, а цілістне функціювання стає результатом взаємодії всіх її сторін та ровнів.

Структура системи. Упорядкованість відносин, які зв'язують елементи системи, визначає структуру системи як множину елементів, що функціонують між елементами системи у відповідності з зв'яками, які встановилися між елементами. Структуру можна уявити як схему — статичну модель системи, яка характеризує тільки побудову системи, не враховуючи великої кількості властивостей та станів її елементів. Як правило, при введені поняття структури систему відображують шляхом поділу на підсистеми, компоненти, елементи з

взаємозв'язками, які можуть мати різний характер. Одна й та ж система може бути надана різними структурами у залежності від стадії пізнання об'єктів або процесів, від аспекту, як їх разглядають, мети створення і так далі. При цьому, в міру розвитку досліджень або в ході проектурвання структура системи может змінюватися. Структури можуть бути представлені в матричній формі, у формі теоретико-множинних описів, за допомогою мови топології, алгебри та інших засобів моделювання систем. Найбільш розповсюджені наступні класи структур:

Мережева структура є декомпозицією системи в часі. Такі структури можуть відбивати порядок дії техничної системи (наприклад, телефонна мережа, електрична мережа і подібні їм), етапи діяльності людини (наприклад, при виробництві продукції — мережовий графік, при проектуванні — мережева модель, при плануванні — мережевий план і подобні їм).

Ієрархічна структура є декомпозицією системи у просторі. Всі компоненти та св'язки існують у ціх структурах одночасно (не рознесені у часі). Такі структури можуть мати велику кількість рівнів декомпозиції (структуризації). Структури, в яких кажний елемент рівня, що лежить нижче, є підлеглим одному вузлу, що стоїть вище (і це справедливо для всіх рівнів ієрархії), називають деревовидними структурами, або ієрархічними структурами с «сильними» св'язками. Структури, в яких елемент що лежить нижче рівня может бути пдлеглим двум і більше взлам , які стоять вище, называють ієрархічними структурами з «слабкими» св'язками.

Матрична структура є ієрархічною структурою зі «слабкими» св'язками, яка базується на принципі множинної ієрархії.

Відносини, які мають вигляд «слабких» св'язків між двох рівнів, побудовано за функціональним принципом та подібні відношенням у матриці, яку створено зі складових цих двох рівнів.

Багаторівнева ієрархічна структура є ієрархічною структурою з «сильними» та «слабкими» св'язками, яка базується на принципі множинної ієрархії. Так, в теории систем М. Месаровича запропоновано особливі класи ієрархічних структур, які відрізняються різними принципами взаємовідносин елементів у межах рівня та різним правом втручання рівня, який стоїть вище в організацію взаємовідносин між елементами, які лежать нижче, для іменування яких він запропанував наступні термини: «страти», «слої», «ешелони».

Змішана ієрархічна структура структурою з вертикальними та горизонтальними св'язками.

Структура з довільними зв'язками можуть мати будь яку форму, об'єднувати принципи разних видів структур та порурушиати їх.

Взаємодія системи. Процес взаємного впливу елементів, системы та зовнішнього середовища один на одного, а також сукупність взаємозв'язків та взаємоовідношень між їх властивостями, коли вони отримують характер взаємодії.

Зовнішнє середовище системи. Все, що не входить у систему, об'єднується поняттям «зовнішнє середовище». По суті, виявлення системи є поділ за визначеною основою деякої області матеріального або абстрактного світу на дві частини, одна з яких розглядається як система, а інша — як зовнішнє середовище. Під цим розуміється, що зовнішнє середовище являє по собі множину об'єктів, які існують у просторі і в часі та інших систем, які, як передбачається, діє на систему тим або іншим чином. При цьому, між системою та зовнішнім

середовищем існує визначений взаємозв'язок — система формує та проявляє свої властивості у процесі взаимодії з середовищем, будучи активним компонентом цієї взаємодії.

1.2. Властивості системи

Серед великої кількості властивостей, які притаманні системам, виділяються найбільш важливі, які характеризують їх функціювання:

Стан системы. Набір значень основних параметрів системи, які визначають характер її функціювання на визначеному часовому інтервалі. Стан системы можна уявити як сукупність станів її ***n*** элементів та св'язків меж ними (двосторонніх св'язків не може буть больш ніж ***n(n – 1)*** у системі з ***n*** елментами). Завдання конкретної системи зводиться до завдання її станів на протязі всього її життєвого циклу. Реальна система не може знаходитися в будь якому стані, тому що завжди є відомі обмеження — деякі внутрішні та зовнішні фактори. *Можливі стани реальної системи створюють у просторі її станів деяку множину допустимих станів системи.* Визначають стан системи (у випадку системи материальної природи) або через вхідні впливи та вихідні сигнали (результати), або через макропараметри, макровластивості системи.

Поведінка системи. Якщо система здібна переходити з одного стану в інший (наприклад, s1->s2->s3-> …), тоді мається на увазі, що вона володії поведінкою. Цим поняття користуються, коли невідомі закономірності або правила переходу системи з одного стану в інший. У таких випадках кажуть, що система має деяку поведінку та з`ясовують її характер, алгоритм і інші особливості.

Рівновага системи. Здібність системи у відсутності зовнішніх збурюючих впливів (або при постійних впливах) зберігати свій стан

скільки завгодно довго (або на протязі заданого часового інтервалу) называють станом рівноваги.

Стійкість системи. Під стійкістю розуміють здібністьь системи повертатися у стан рівноваги після того, як вона була з цього стану виведена під впливом (а у системах з активними елементами — внутрішніх) збурюючих впливів. Ця здібність відносна і за звичай, притаманна системам тільки тоді, коли відхилення не превищують деякої межі. Стан рівноваги, у якій система здібна повертатися, називають стійким станом рівноваги. Повернення у цей стан може супроводжуватися коливальним процесом. Відповідно, у складних системах можливі нестійкі стани рівноваги.

Розвититок системи. Кажна система у своєму развитку проходить ряд основних етапів: ***виникнення; становлення; перетворення.***

Винекнення системы — складний суперечливий процес, пов'язаний з поняттям «новий». Цей процес, у свою чергу, можна поділити на два этапи:

- **скритий етап** — поява нових елементів та нових зв'язків у рамках старого;
- **явний етап,** коли нові фактори, які накопичилися, приводять до стрибка — появи нової якості.

Процес становлення системи пов'язано з подальшим кількісним збільшенням якісно тотожних множин її елементів та з появою у системі нових якостей.

Протирічча між якісно тотожними елементами є одним з джерел розвитку системи. У наслідок цього протирічча — намагання элементов розійтися у просторі. З іншого боку, існують

системоутворюючі фактори, які не дають системі разпастися. До того ж існує межа системи, вихід за яку може бути згубним для елементів системи та для системи в цілому. Крім того, на кажну систему діють інші системи, які перешкоджають розширенню системних меж. Все це визначає цілістність як специфічні властивості зрілої системи.

Нові функціональні якості, які отримує система включають в себе специфічні властивості, що отримані системою у процесі її спілкування з зовнішнім середовищем. Найбільш перспективними виявляються ті елементи системи, функції яких відповідають потебам існування системи у конкретнму зовнішньому середовищі. Система в цілому стає спеціалізованою. Вона може успішно функціювати тільки у тому середовищі, у якому вона сформувалася. Будь який перехід системи в інше средовище неменуче викликає її перетворення.

Система в період зрілості внутрішнє суперечлива внаслідок подвійності свого існування як системи, яка завершує одну форму руху і така, що є носієм більш високої форми руху. Навіть при сприятливих зовнішніх умовах внутрішні протирічча приводять систему у стан перетворення — до неменучого этапу її розвитку.

Зовнішні причини перетворення системи:

- змінення зовнішнього середовища;
- проникнення у систему чужих елементів, які впливають на структуру системи.

Внутрішні причини перетворення системи:

- обмеженність простору розвитку та загострення протиріч між елементами системи;
- накопничення помилок при развитку системи (мутації у живих організмах);

- припинення відтворення елементів, які складають систему.

Перетворення системи може привести як до загибелі системи, так і до виникнення якісно, що дорівнює, або більш высокої, ніж ступінь організованості системи, яка перетворюється.

Таким чином, при визначених умовах можливий стрибкоподібний перехід системи на новий больш високий (або більш низький) рівень упорядкування. Причому перехід системи до різних властивих їй станів, а також руйнування системи можуть бути результатом як досить сильних зовнішних впливів, так і відносно слабкых флюктуацій, які довго існують або такі, що підсилюють за рахунок додатних зворотніх св'язків. Перехід системи на новий рівень організованості у визначених ситуаціях представляє собою випадковий процес вибору системою одного з можливих шляхів еволюції. У цьму випадку знову потрібно підкреслити слово «можливих», або розумно говорити про створення умов переходу системи в одне з можливих, властивостей її станів.

Можливі два крайніх варіанти змнення структури систем, які можна умовно позначити як революційнний та еволюціоний. При революційному перетворенні передбачається, що створення нової організації системи, нової її структури повинна передувати насильницька поламка старої структури. За звичай після такої насильницької поламки система переходит на більш низкий рівень упорядкування, при цьому формування нової структури затягується на довго, іноді, невизначений строк. При еволюційному перетворенні нові відношення формуються у рамках існуючої структури, виникають нові тенденції развитку систем. При накопиченні якісних змін можливий і стрибкоподібний, і в цьому сенсі революційный, перехід системи в

новий рівноважний стан — до нової структури, з якою система «внутрішнє» готова. В цьому випадку суть революційного перетворення зводиться до знищення елементів, які перешкоджають становленню нової структури (наприклад, в соціально-економічних системах такими елементами є органи управління).

Якщо припустити, що стан системи може бути представлено набором з ***n*** параметрів, то кожному стану системи буде відповідати точка в ***n***-мірному просторі станів системи, а функціювання системи проявится в переміщенні цієї точки за деякою траекторією в просторі станів. Мабуть, досягнення бажаного стану можливо в загальному випадку за декількома траекторіями. Перевага траекторії визначається оцінкою якості траекторії та залежить також від обмежень, які накладаються на систему, в тому числі зовнішнім середовищем. Ці обмеження визначають область допустимих траекторій. Для визначення кращої траекторії з числа допустимих вводиться критерій якості функціювання системи — у загальному випадку [формально] у вигляді деяких цільових функцій (функціоналів, відношень). На кращій [оптимальній] траекторії цільової функції досягають екстремальних значень. ***Цілеспрямоване втручання у поведінку системи, яке забезпечує вибір системою оптимальної траекторії развитку, називається керуванням.***

Рух системи. Процес послідовного змінення стану системи. Рух буває як вимушеним, так і власним. Вимушений рух системи — це змінення її стану під впливом зовнішнього середовища. Так, прикладом вимушенного руху системи «організація» може служити переміщення ресурсів за наказом, який прийшов в систему зовні. Власний рух системи — це змінення стану системи без впливу

зовнішнього середовища (тільки під дією внутрішніх причин). Так, власним рухом системи «людина» буде його життя як біологічного (а не суспільного) індивіду, по іншому це є живлення, сон, розмноження і тому подібне.

Обмеження системи. Набір факторів, які визначають умови функціювання системи (реалізацію процесу). Обмеження бувають як внутрішніми, так і зовнішніми. Одними з основних зовнішніх обмежень є мета функціювання системи. Прикладом внутрішніх обмежень можуть бути ресурси, які забезпечують реализацію того або іншого процесу.

Процеси системи. Сукупність последовних змін стану системи для досягнення мети. До процесів системи відносяться:

- ***вхідний процес*** — множина вхідних впливів, які змінюються з плином часу;
- ***вихідний процес*** — множина вихідних впливів на зовнішнє середовище, які змінюються з плином часу та визначаються вихідними значеннями (реакціями);
- ***перехідний процесс*** — безліч перетворень початкового стану і вхідних впливів системи в вихідні величини, які змінюються з плином часу за певними правилами.

1.3. Функції системи

Властивості системи, що призводять до досягнення мети. Функціонування системи проявляється в її переході з одного стану в інший або в збереженні будь-якого стану протягом певного періоду. У цьому сенсі поведінка системи - це її функціонування в часі. Цілеспрямоване (цілеспрямоване) поведінка орієнтоване на досягнення системою кращою для неї мети. В системі, що складається

з пов'язаних між собою, взаємодіючих підсистем, оптимум для всієї системи не є функцією (наприклад, сумою) оптимумів підсистем, що входять в систему. Це положення іноді називають теоремою оптимумів системного підходу.

Поширення усюди ідей системних досліджень і системного підходу є однією з характерних особливостей наукового та технічного знання XX століття. Розвиток інженерного підходу і технологій в XX столітті відкриває еру штучно-технічного освоєння систем. Тепер системи не тільки досліджуються, але проектовані і сконструйовані. Одночасно оформляється і організаційно-управлінська установка: об'єкти управління також починають розглядатися як системи. Це призводить до виділення все нових і нових класів систем: цілеспрямованих, що самоорганізуються, рефлексивних та інших. Сам термін «система» входить в лексикон практично всіх професійних сфер. Починаючи з середини XX століття широко розгортаються дослідження з загальної теорії систем і розробки в галузі системного підходу, складається міжпрофесійні і міждисциплінарний системний рух.

У всьому різноманітті трактувань систем продовжують зберігатися два підходи. З точки зору першого з них (його можна назвати онтологічним або, більш жорстко, натуралістичним), системність інтерпретується як фундаментальне властивість об'єктів пізнання. Тоді завданням системного дослідження стає вивчення специфічно системних властивостей об'єкта: виділення в ньому елементів, зв'язків і структур, залежностей між зв'язками і тому подібних категорій. Причому елементи, зв'язку, структури і залежності трактуються як «натуральні», властиві «природі» самих об'єктів і в

цьому сенсі об'єктивні. Система в такому підході покладається як об'єкт, що володіє власними законами життя. Інший підхід (його можна назвати епістемолого-методологічним) полягає в тому, що система розглядається як епістемологічний конструкт, який не має природної природи, і задає специфічний спосіб організації знань і мислення. Тоді системність визначається не властивостями самих об'єктів, але цілеспрямованістю діяльності та організацією мислення. Різниця в цілях, засобах і методах діяльності неминуче виробляє множинність описів одного і того ж об'єкта, що породжує в свою чергу установку на їх синтез і конфігурація.

1.4. Класифікація систем

Суттєвим аспектом розкриття змісту трактувань систем є виділення різних типів систем, при цьому різні типи і аспекти систем - закони їх будови, поведінки, функціонування, розвитку і так далі - описуються у відповідних спеціалізованих теоріях систем. Для виділення класів систем можуть використовуватися різні класифікаційні ознаки. Основними з них вважаються: природа елементів системи, походження, тривалість існування, мінливість властивостей, ступінь складності, ставлення до середовища, реакція на впливи, характер поведінки і ступінь участі людей в реалізації керуючих впливів. До теперішнього часу сформувався ряд класифікацій систем, що використовують зазначені підстави.

Проста класифікація систем. У найбільш загальному плані системи можна розділити за природою їх елементів на матеріальні (реальні) і ідеальні (абстрактні). Розподіл систем на матеріальні і абстрактні дозволяє розрізняти реальні системи (об'єкти, явища, процеси) і системи, які є певними відображеннями (моделями)

реальних об'єктів або чистими абстракціями.

Матеріальні системи являють собою цілісні сукупності об'єктів різних областей дійсності і, в свою чергу, діляться на системи, що складаються з елементів неорганічної природи (фізичні, геологічні, хімічні та інші) і живі системи, куди входять як найпростіші біологічні системи, так і дуже складні біологічні об'єкти типу організму, виду, екосистеми. Матеріальні системи бувають відносно простими і відносно складними. Простіші системи складаються з відносно однорідних безпосередньо взаємодіючих елементів. У більш складних системах елементи групуються в підсистеми, що вступають у взаємини як деякі цілісності. Особливий клас матеріальних живих систем утворюють соціальні системи, різноманітні за типами та формами (від найпростіших соціальних об'єднань до соціально-економічної структури суспільства).

Ідеальні (абстрактні) системи є продуктами людського мислення, елементи яких не мають прямих аналогів в реальному світі і являють собою ідеальні об'єкти - поняття або ідеї, пов'язані певними взаємовідносинами. Вони створюються шляхом уявного відволікання від тих чи інших сторін, властивостей і/або зв'язків предметів і утворюються в результаті творчої діяльності людини. Вони також можуть бути розділені на безліч різних типів (особливі системи являють собою наукові поняття, гіпотези, теорії, системи рівнянь і тому подібні). Абстрактної системою є, наприклад, система понять тієї чи іншої науки. До числа абстрактних систем відносяться і наукові знання про системи різного типу, як вони формулюються в загальній теорії систем, спеціальних теоріях систем та інших областях. У сучасній науці велика увага приділяється вивченню мови як

[семіотичної] системи; в результаті узагальнення цих досліджень виникла загальна теорія знаків - семіотика (див. Семіотика).

Завдання обгрунтування математики і логіки викликали інтенсивну розробку принципів побудови формалізованих логічних систем. Результати цих досліджень широко застосовуються у всіх областях науки і техніки. В цілому, формалізовані логічні системи підрозділяються на три основні класи:

1 статичні математичні системи або моделі, які описують об'єкт в будь-який момент часу;

2 динамічні математичні системи або моделі відображають поведінку об'єкта в часі;

3 знаходяться в нестійкому положенні між статикою і динамікою, які при одних впливах поводяться як статичні, а при інших впливах - як динамічні.

Залежно від походження систем, виділяють природні і штучні системи. Природні системи, будучи продуктом розвитку природи, виникли без втручання людини. Штучні системи являють собою результат творчої діяльності людини, причому з часом їх кількість постійно збільшується.

За тривалістю існування системи підрозділяються на постійні і тимчасові. До постійних зазвичай ставляться природні системи, хоча з точки зору діалектики всі існуючі системи - тимчасові. До постійних прийнято відносити і штучні системи, які в процесі заданого часу функціонування зберігають істотні властивості, які визначаються призначенням цих систем.

Залежно від ступеня мінливості властивостей систем, виділяються статичні і динамічні системи. Для статичної системи

характерно, що її стан з часом залишається постійним (наприклад, газ в обмеженому обсязі - в стані рівноваги). Динамічна система змінює свій стан у часі (наприклад, живий організм). Якщо знання значень змінних системи в даний момент часу дозволяє встановити стан системи в будь-який інший або будь-який попередній моменти часу, то така система є однозначно детермінованою. Для ймовірнісної (стохастичної) системи знання значень змінних в даний момент часу дозволяє передбачити ймовірність розподілу значень цих змінних в наступні моменти часу. Поведінка зазначених класів систем описується за допомогою диференціальних рівнянь, завдання побудови яких вирішується в математичної теорії систем.

За характером взаємовідносин систем із зовнішнім середовищем, виділяють закриті і відкриті системи.

Закриті (ізольовані) системи фізично ізольовані від зовнішнього середовища. Всі статичні системи є закритими, що, однак, не виключає присутності динамічних процесів в закритих системах. Згідно з другим законом термодинаміки, здатність ізольованих фізичних систем підтримувати постійний обмін речовин і енергії з часом слабшає, в результаті чого система витрачає запас енергії, внаслідок чого ентропія такої системи прагне до свого максимуму. У таких системах нівелюються відмінності, а процеси самоорганізації в них неможливі. Другий закон термодинаміки пророкує досить песимістичний прогноз однорідного майбутнього ізольованих систем. Ізольованих і закритих систем в природі практично не існує.

Якщо проаналізувати приклад будь-який з таких систем, то можна переконатися, що не існує абсолютних «ізолюючих екранів» відразу від усіх форм матерії або енергії, що будь-яка система швидше

або повільніше розвивається або деградує. У вічності поняття «швидко» і «повільно» сенсу не мають, тому, строго кажучи, існують тільки відкриті системи, близькі до рівноваги, умовно названі відкритими рівноважними системами. З цієї точки зору ізольовані і закриті системи - свідомо спрощені схеми відкритих систем, корисні при наближеному вирішенні приватних завдань.

Відкриті системи характеризуються постійним обміном речовини і енергії з зовнішнім середовищем. Так, в біологічних організмах домінує рухлива рівновага при постійному обміні речовини і енергії із середовищем. Такі відкриті системи уникають ентропії через метаболізм і постійне надходження інформації з зовнішнього середовища. Всі відкриті системи характеризуються самостабілізації і саморегуляцією. Ці системи виявляються здатними на підтримку готівкового стану в результаті включення процесів контролю. Негативні зворотні сигнали протидіють надходить із середовища, елімінує обурення і, таким чином, реставрують бажаний стан системи. У відкритих органічних системах здатність на динамічну самостабілізації бажаного стану називається гомеостазом. Такі системи характеризує плавне рівновагу, оскільки абсорбування збурень середовища призводить не до первісного стану, а до нового рівноважного стану. Самоорганізація та морфогенез представляють найбільш загальні процеси системних змін в еволюції відкритих систем. У той час як самостабілізації досягається за допомогою негативних зворотних зв'язків, самоорганізація досягається за допомогою позитивних зворотних зв'язків. Розвиток системи (морфогенез) передбачає адаптацію початкового рівноважного стану зовнішніх збурень і, відповідно, досягнення нового етапу розвитку.

Обурення середовища викликають посилення механізмів самостабілізації.

Нове трактування другого початку термодинаміки була запропоноване І. Р. Пригожиним. На його думку, ентропія - це не просто невпинне зісковзування системи до стану, позбавленому будь-якої було організації. Необоротні процеси є джерелом порядку. У сильно нерівноважних умовах може відбуватися перехід від безладу, хаосу до порядку. Можуть виникати нові динамічні стану матерії, що відображають взаємодію даної системи з навколишнім середовищем. Ці нові структури Пригожин називає диссипативними, оскільки їх стабільність покоїться на дисипації енергії і речовини. Теорії нерівноважної динаміки і синергетики задають нову парадигму еволюції систем, що долає термодинамічний принцип прогресивного зісковзування до ентропії. З точки зору цієї нової парадигми, порядок, рівновагу і стійкість систем досягаються постійними динамічними нерівновагими процесами.

Залежно від реакції на впливи виділяють активні і пасивні системи. Активні системи здатні протистояти впливам зовнішнього середовища та інших систем і самі можуть впливати на них. У пасивних систем це властивість відсутня.

За характером поведінки все системи підрозділяються на системи з керуванням і без управління. Клас систем з управлінням утворюють системи, в яких реалізується процес визначення мети й целеосуществления. Прикладом систем без управління може служити Сонячна система, в якій траєкторії руху планет визначаються чинними у Всесвіті законами гравітації.

У прикладних науках, а також в теорії і практиці управління

широко використовуються класифікації систем в залежності від ступеня їх складності та організованості. На цих підставах системи діляться на великі, прості, складні і організаційні. Як правило, коли мова йде про різні види систем управління, перш за все мається на увазі саме такий загальний їх розподіл.

До організаційних систем належать соціальні системи - групи, колективи, спільноти людей, суспільство в цілому (див. Суспільство).

Простими системами називають системи, що складаються з обмеженого і відносного малого числа елементів з однотипними однорівневими зв'язками. Такі системи з достатнім ступенем точності можуть бути описані відомими математичними співвідношеннями.

Великими системами називають багатокомпонентні системи, що включають значне число елементів з однотипними багаторівневими зв'язками. Великі системи - це просторово-розподілені системи високого ступеня складності, в яких підсистеми (їх складові частини) також відносяться до категорій складних. Додатковими ознаками, що характеризують велику систему, є:

- великі розміри;
- складна ієрархічна структура;
- циркуляція в системі великих інформаційних, енергетичних і матеріальних потоків;
- високий рівень невизначеності в описі системи.

Складними системами називають структурно і функціонально складні багатокомпонентні системи з великим числом взаємозалежних і взаємодіючих елементів різного типу і з численними і різнорідними зв'язками між ними. Складні системи відрізняються багатомірністю, різнорідністю структури, різноманіттям природи елементів і зв'язків,

організаційної разносопротівляемостью і разночувствітельностью до впливів, асиметричністю потенційних можливостей здійснення функціональних і дисфункціональних змін. При цьому кожен з елементів такої системи може бути також представлений у вигляді системи (підсистеми). До складної можна віднести систему, що володіє принаймні одним з таких ознак:

- система в цілому має властивості, якими не володіє ні одна зі складових її елементів;
- систему можна розділити на підсистеми і вивчати кожну з них окремо;
- система функціонує в умовах суттєвої невизначеності і впливу середовища на неї, що обумовлює випадковий характер зміни її показників;
- система здійснює цілеспрямований вибір своєї поведінки.

У кібернетиці міра складності пов'язується з поняттям різноманітності. Зокрема, з принципу різноманітності слід, що аналіз систем (процесів, ситуацій), що володіють певним різноманітністю, можливий лише з використанням керуючих систем, здатних породжувати, по крайней мере, не меншу різноманітність.

Важливою особливістю складних систем, особливо живих, технічних і соціальних, є передача в них інформації, що веде до суттєвих взаємозв'язку їх властивостей. Тому значну роль у функціонуванні таких систем грають процеси управління. До найбільш складних видів подібних систем відносяться цілеспрямовані системи, поведінка яких підпорядковане досягненню певних цілей, і самоорганізуються, здатні в процесі функціонування видозмінювати свою структуру. При цьому для багатьох складних систем характерна

наявність різних за рівнем, часто які не узгоджуються між собою цілей.

Системи, що містять активні елементи (підсистеми), тобто такі елементи, які мають можливість самостійно приймати рішення щодо свого стану, називаються організаційними системами (організаціями). В організаційних системах властивістю цілеспрямованості володіє як вся система, так і окремі її елементи. Цим організація відрізняється від системи, яку називають організмом. Між окремими елементами (органами) організму існує поділ системних функцій, але тільки організм в цілому може бути цілеспрямованим.

2. ПРИЗНАЧЕННЯ, ОБЛАСТЬ ЗАСТОСУВАННЯ І СПОСОБИ ОЦІНКИ ПРОДУКТИВНОСТІ БАГАТОПРОЦЕСОРНИХ КОМПЬЮТЕРНИХ СИСТЕМ

В даний час сфера застосування багатопроцесорних комп’ютерних систем безперервно розширюється, охоплюючи все нові області в різних галузях науки, бізнесу і виробництва. Стрімкий розвиток кластерних систем створює умови для використання багатопроцесорної компьютерної техніки в науці, на виробництві, космічних дослідженнях, військовій справі та у реальному секторі економіки.

Якщо традиційно багатопроцесорні комп’ютерні системи застосовувалися в основному в науковій сфері для вирішення обчислювальних завдань, що вимагають потужних обчислювальних ресурсів, то зараз через бурхливий розвиток бізнесу різко зросла кількість компаній, які надають використанню комп'ютерних технологій та електронного документообігу головну роль. У зв'язку з цим постійно зростає потреба в побудові централізованих комп’ютерних систем для критично важливих додатків, пов'язаних з обробкою транзакцій, управлінням базами даних і обслуговуванням телекомунікацій. Можна виділити дві основні сфери застосування описуваних систем: обробка транзакцій в режимі реального часу (OLTP, on-line transaction processing) і створення сховищ даних для організації систем підтримки прийняття рішень (Data Mining, Data Warehousing, Decision Support System). Система для глобальних корпоративних обчислень - це, перш за все, централізована система, з якою працюють практично всі користувачі в корпорації, і, відповідно, вона повинна весь час перебувати в робочому стані. Як правило,

рішення подібного рівня встановлюють в компаніях і корпораціях, де навіть короткочасні простої мережі можуть привести до величезних збитків. Тому для організації такої системи не підійде звичайний сервер зі стандартною архітектурою, цілком придатний там, де немає жорстких вимог до продуктивності і часу простою. Високопродуктивні системи для глобальних корпоративних обчислень повинні відрізнятися такими характеристиками як підвищена продуктивність, масштабованість, мінімально допустимий час простою.

Поряд з розширенням сфери застосування у міру вдосконалення багатопроцесорних комп'ютерних систем відбувається ускладнення і збільшення кількості завдань в областях, традиційно використовують високопродуктивну комп'ютерну техніку. В даний час виділено коло фундаментальних і додатних проблем, ефективне вирішення яких можливе лише з використанням надпотужних обчислювальних ресурсів. Це коло, що позначається поняттям "Grand challenges", включає наступні завдання:

- передбачення погоди, клімату і глобальних змін в атмосфері;
- науки про матеріали;
- побудова напівпровідникових приладів;
- надпровідність;
- структурна біологія;
- розробка фармацевтичних препаратів;
- генетика;
- квантова хромодинамика;
- астрономія;
- транспортні завдання;
- гідро- і газодинаміки;

- керований термоядерний синтез;

- ефективність систем згоряння палива;

- геоінформаційні системи;

- розвідка надр;

- наука про світовому океані;

- розпізнавання і синтез мови;

- розпізнавання зображень.

Багатопроцесорні комп'ютерні системи можуть існувати в різних конфігураціях. Найбільш поширеними типами цих систем є:

- системи високої надійності;

- системи для високопродуктивних обчислень;

- багатопотокові системи.

Відзначимо, що межі між цими типами багатопроцесорних комп'ютерних систем до деякої міри розмиті, і часто система може мати такі властивості або функції, які виходять за рамки перерахованих типів. Більш того, при конфігуруванні великої системи, яка використовується як система загального призначення, доводиться виділяти блоки, виконують всі перераховані функції.

Багатопроцесорна комп'ютерна система є ідеальною схемою для підвищення надійності інформаційно-обчислювальної системи. Завдяки єдиному уявленню, окремі вузли або компоненти багатопроцесорних комп'ютерних систем можуть непомітно для користувача замінювати несправні елементи, забезпечуючи безперервність і безвідмовну роботу навіть таких складних додатків як бази даних.

Катастрофостійкі рішення створюються на основі того, що вузли багатопроцесорної системи розносяться на сотні кілометрів і

забезпечується механізми глобальної синхронізації даних між такими вузлами.

Багатопроцесорні комп’ютерні системи для високопродуктивних обчислень призначені для паралельних розрахунків. Є багато прикладів наукових розрахунків, виконаних на основі паралельної роботи декількох недорогих процесорів, які підтримують одночасне проведення великого числа операцій.

Багатопроцесорні комп’ютерні системи для високопродуктивних обчислень зазвичай зібрані з багатьох комп'ютерів. Розробка таких систем - процес складний, що вимагає постійного узгодження таких питань як інсталяція, експлуатація і одночасне управління великим числом комп'ютерів, технічних вимог паралельного і високопродуктивного доступу до одного і того ж системного файлу (або файлів), межпроцессорной зв'язку між вузлами і координації роботи в паралельному режимі. Ці проблеми найпростіше вирішуються при забезпеченні єдиного образу операційної системи для всього кластера. Однак реалізувати подібну схему вдається далеко не завжди, і зазвичай вона застосовується лише для невеликих систем.

Багатопотокові системи використовуються для забезпечення єдиного інтерфейсу до ряду ресурсів, які можуть з часом довільно нарощуватися (або скорочуватися). Типовим прикладом може служити група web-серверів.

Головною відмінною рисою багатопроцесорних комп’ютерних систем є її продуктивність, тобто кількість операцій, вироблених системою за одиницю часу. Розрізняють пікову і реальну продуктивність. Під піковою розуміють величину, що дорівнює добутку пікової продуктивності одного процесора на число таких

процесорів в даній машині. При цьому передбачається, що всі пристрої комп'ютера працюють в максимально продуктивному режимі. Пікова продуктивність комп'ютера обчислюється однозначно, і ця характеристика є базовою, по якій виробляють порівняння високопродуктивних обчислювальних систем. Чим більше пікова продуктивність, тим (теоретично) швидше користувач зможе вирішити свою задачу. Пікова продуктивність є величина теоретична і, взагалі кажучи, недосяжна при запуску конкретного додатка. Реальна ж продуктивність, що досягається на даному додатку, залежить від взаємодії програмної моделі, в якій реалізовано додаток, з архітектурними особливостями машини, на якій під час запуску програми.

Існує два способи оцінки пікової продуктивності комп'ютера. Один з них спирається на число команд, виконуваних комп'ютером за одиницю часу. Одиницею виміру, як правило, є MIPS (Million Instructions Per Second). Продуктивність, виражена в MIPS, говорить про швидкість виконання комп'ютером своїх же інструкцій. Але, по-перше, заздалегідь не ясно, в скільки інструкцій відобразиться конкретна програма, а по-друге, кожна програма має свою специфіку, і число команд від програми до програми може змінюватися дуже сильно. У зв'язку з цим дана характеристика дає лише загальне уявлення про продуктивність комп'ютера.

Інший спосіб вимірювання продуктивності полягає у визначенні числа речових операцій, які виконуються комп'ютером за одиницю часу. Одиницею вимірювання є Flops (Floating point operations per second) - число операцій з плаваючою точкою, вироблених комп'ютером за одну секунду. Такий спосіб є більш

прийнятним для користувача, оскільки йому відома обчислювальна складність програми, і, користуючись цією характеристикою, користувач може отримати нижню оцінку часу її виконання.

Однак пікова продуктивність виходить тільки в ідеальних умовах, тобто при відсутності конфліктів при зверненні до пам'яті при рівномірному завантаженні всіх пристроїв. У реальних умовах на виконання конкретної програми впливають такі апаратно-програмні особливості даного комп'ютера як: особливості структури процесора, системи команд, склад функціональних пристроїв, реалізація введення / виведення, ефективність роботи компіляторів.

Одним з визначальних чинників є час взаємодії з пам'яттю, яке визначається її будовою, об'ємом і архітектурою підсистем доступу в пам'ять. У більшості сучасних комп'ютерів в якості організації найбільш ефективного доступу до пам'яті використовується так звана багаторівнева ієрархічна пам'ять. Як рівнів використовуються регістри і реєстрова пам'ять, основна оперативна пам'ять, кеш-пам'ять, віртуальні і жорсткі диски, стрічкові роботи. При цьому витримується наступний принцип формування ієрархії: при підвищенні рівня пам'яті швидкість обробки даних повинна збільшуватися, а обсяг рівня пам'яті - зменшуватися. Ефективність використання такого роду ієрархії досягається за рахунок зберігання часто використовуваних даних в пам'яті верхнього рівня, час доступу до якої мінімально. А оскільки така пам'ять обходиться досить дорого, її обсяг не може бути більшим. Ієрархія пам'яті відноситься до тих особливостей архітектури комп'ютерів, які мають величезне значення для підвищення їх продуктивності.

Для того щоб оцінити ефективність роботи обчислювальної

системи на реальних задачах, був розроблений фіксований набір тестів. Найбільш відомим з них є LINPACK - програма, призначена для вирішення системи лінійних алгебраїчних рівнянь з щільною матрицею з вибором головного елемента по рядку. LINPACK використовується для формування списку Top500 - п'ятисот найпотужніших комп'ютерів світу. Однак LINPACK має істотний недолік: програма распараллеливается, тому неможливо оцінити ефективність роботи комунікаційного компонента суперкомп'ютера.

В даний час великого поширення набули тестові програми, взяті з різних предметних областей і представляють собою або модельні, або реальні промислові додатки. Такі тести дозволяють оцінити продуктивність комп'ютера дійсно на реальних завданнях і отримати найбільш повне уявлення про ефективність роботи комп'ютера з конкретним додатком.

Найбільш поширеними тестами, побудованими за цим принципом, є: набір з 24 Ліверморської циклів (The Livermore Fortran Kernels, LFK) і пакет NAS Parallel Benchmarks (NPB), до складу якого входять дві групи тестів, що відображають різні сторони реальних програм обчислювальної гідродинаміки. NAS тести є альтернативою LINPACK, оскільки вони відносно прості і в той же час містять значно більше обчислень, ніж, наприклад, LINPACK або LFK.

Однак при всій різноманітності тестові програми не можуть дати повного уявлення про роботу комп'ютера в різних режимах. Тому завдання визначення реальної продуктивності багатопроцесорних обчислювальних систем залишається поки невирішеною.

3.. КЛАСИФІКАЦІЯ АРХІТЕКТУР ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ЗА ПАРАЛЕЛЬНОЮ ОБРОБКОЮ ДАНИХ

Щоб дати більш повне уявлення про багатопроцесорні комп’ютерні системи, крім високої продуктивності необхідно назвати і інші відмінні риси. Перш за все, це незвичайні архітектурні рішення, спрямовані на підвищення продуктивності (робота з векторними операціями, організація швидкого обміну повідомленнями між процесорами або організація глобальної пам'яті в багатопроцесорних системах та ін.).

Поняття архітектури високопродуктивної системи є досить широким, оскільки під архітектурою можна розуміти і спосіб паралельної обробки даних, який використовується в системі, і організацію пам'яті, і топологію зв'язку між процесорами, і спосіб виконання системою арифметичних операцій. Спроби систематизувати велику кількість існуючих архітектур вперше було вжито в кінці 60-х років і тривають донині.

У 1966 р. М.Флінном (Flynn) був запропонований надзвичайно зручний підхід до класифікації архітектур обчислювальних систем. В його основу було покладено поняття потоку, під яким розуміється послідовність елементів, команд або даних, обробляється процесором. Відповідна система класифікації заснована на розгляді числа потоків інструкцій і потоків даних і описує чотири архітектурні класу:

SISD - Single Instruction Single Data

MISD - Multiple Instruction Single Data

SIMD - Single Instruction Multiple Data

MIMD - Multiple Instruction Multiple Data

SISD (single instruction stream / single data stream) - одиночний потік команд і одиночний потік даних. До цього класу належать послідовні комп'ютерні системи, які мають один центральний процесор, здатний обробляти тільки один потік послідовно виконуваних інструкцій. В даний час практично всі високопродуктивні системи мають більше одного центрального процесора, однак кожен з них виконує незв'язані потоки інструкцій, що робить такі системи комплексами SISD-систем, що діють на різних просторах даних. Для збільшення швидкості обробки команд і швидкості виконання арифметичних операцій може застосовуватися конвеєрна обробка. У разі векторних систем векторний потік даних слід розглядати як потік з одиночних неподільних векторів. Прикладами комп'ютерів з архітектурою SISD можуть служити більшість робочих станцій Compaq, Hewlett-Packard і Sun Microsystems.

MISD (multiple instruction stream / single data stream) - множинний потік команд і одиночний потік даних. Теоретично в цьому типі машин безліч інструкцій повинно виконуватися над єдиним потоком даних. До сих пір жодної реальної машини, що потрапляє в даний клас, створено не було. Як аналог роботи такої системи, мабуть, можна розглядати роботу банку. З будь-якого терміналу можна подати команду і щось зробити з наявним банком даних. Оскільки база даних одна, а команд багато, ми маємо справу з множинним потоком команд і одиночним потоком даних.

SIMD (single instruction stream / multiple data stream) - одиночний потік команд і множинний потік даних. Ці системи зазвичай мають велику кількість процесорів, від 1024 до 16384, які можуть виконувати одну і ту ж інструкцію щодо різних даних в

жорсткій конфігурації. Єдина інструкція паралельно виконується над багатьма елементами даних. Прикладами SIMD-машин є системи CPP DAP, Gamma II і Quadrics Apemille. Іншим підкласом SIMD-систем є векторні комп'ютери. Векторні комп'ютери маніпулюють масивами подібних даних подібно до того, як скалярні машини обробляють окремі елементи таких масивів. Це робиться за рахунок використання спеціально сконструйованих векторних центральних процесорів. Коли дані обробляються за допомогою векторних модулів, результати можуть бути видані на один, два або три такти частотогенератора (такт частотогенератора є основним тимчасовим параметром системи). При роботі у векторному режимі векторні процесори обробляють дані практично паралельно, що робить їх в кілька разів швидшими, ніж при роботі в скалярному режимі. Прикладами систем подібного типу є, наприклад, комп'ютери Hitachi S3600.

MIMD (multiple instruction stream / multiple data stream) - множинний потік команд і множинний потік даних. Ці машини паралельно виконують кілька потоків інструкцій над різними потоками даних. На відміну від згаданих вище багатопроцесорних SISD-машин, команди і дані пов'язані, тому що вони представляють різні частини однієї і тієї ж задачі. Наприклад, MIMD-системи можуть паралельно виконувати безліч підзадач з метою скорочення часу виконання основного завдання. Велика розмаїтість потрапляють в даний клас систем робить класифікацію Флінна в повному обсязі адекватною. Дійсно, і чотирипроцесорний SX-5 компанії NEC, і тисячепроцесорний Cray T3E потрапляють в цей клас. Це змушує використовувати інший підхід до класифікації, інакше описує класи комп'ютерних систем. Основна ідея такого підходу може складатися, наприклад, в

наступному. Будемо вважати, що множинний потік команд може бути оброблений двома способами: або одним конвеєрним пристроєм обробки, що працює в режимі поділу часу для окремих потоків, або кожен потік обробляється своїм власним пристроєм. Перша можливість використовується в MIMD-комп'ютерах, які зазвичай називають конвеєрними або векторними, друга - в паралельних комп'ютерах. В основі векторних комп'ютерів лежить концепція конвейеризации, тобто явного сегментування арифметичного пристрою на окремі частини, кожна з яких виконує свою підзадачу для пари операндів. В основі паралельного комп'ютера лежить ідея використання для вирішення одного завдання декількох процесорів, що працюють спільно, причому процесори можуть бути як скалярними, так і векторними.

Класифікація архітектур обчислювальних систем потрібна для того, щоб зрозуміти особливості роботи тієї чи іншої архітектури, але вона не є достатньо детальної, щоб на неї можна було спиратися при створенні багатопроцесорних комп’ютерних систем, тому слід вводити більш детальну класифікацію, яка пов'язана з різними архітектурами ЕОМ і з використовуваним обладнанням.

4. АРХІТЕКТУРА ОБЧИСЛЮВАЛЬНИХ СИСТЕМ. SMP І MPP-АРХІТЕКТУРИ. ГІБРИДНА АРХІТЕКТУРА (NUMA). ОРГАНІЗАЦІЯ КОГЕРЕНТНОСТІ БАГАТОРІВНЕВОЇ ІЄРАРХІЧНОЇ ПАМ'ЯТІ

4.1. SMP-архітектура

SMP (symmetric multiprocessing) - симетрична багатопроцесорна архітектура. Головною особливістю систем з архітектурою SMP є наявність загальної фізичної пам'яті, що розділяється всіма процесорами.

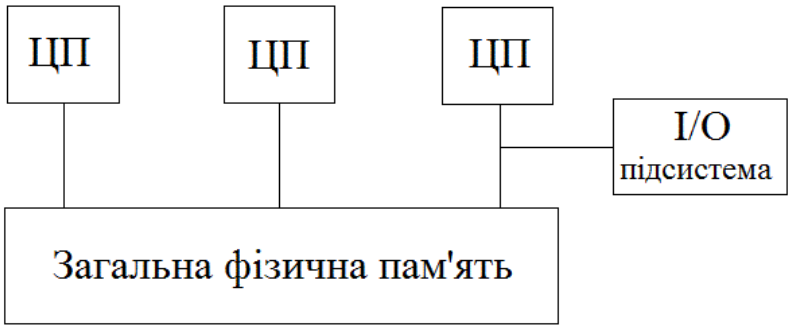

Рисунок 4.1 - Схематичний вигляд SMP-архітектури.

Пам'ять служить, зокрема, для передачі повідомлень між процесорами, при цьому всі обчислювальні пристрої при зверненні до неї мають рівні права і одну і ту ж адресацію для всіх елементів пам'яті. Тому SMP-архітектура називається симетричною. Остання обставина дозволяє дуже ефективно обмінюватися даними з іншими обчислювальними пристроями. SMP-система будується на основі високошвидкісної системної шини (SGI PowerPath, Sun Gigaplane, DEC TurboLaser), до слотів якої підключаються функціональні блоки типів:

процесори (ЦП), підсистема введення/ виведення (I/O) і т. П. Для під'єднання до модулів I/O використовуються вже повільніші шини (PCI, VME64). Найбільш відомими SMP-системами є SMP-сервер і робочі станції на базі процесорів Intel (IBM, HP, Compaq, Dell, ALR, Unisys, DG, Fujitsu і ін.) Вся система працює під управлінням єдиної ОС (зазвичай UNIX-подібної, але для Intel-платформ підтримується Windows NT). ОС автоматично (в процесі роботи) розподіляє процеси по процесорам, але іноді можлива і явна прив'язка.

Основні переваги SMP-систем:

- простота і універсальність для програмування. Архітектура SMP не накладаються обмежень на модель програмування, яка використовується під час створення програми: зазвичай використовується модель паралельних гілок, коли всі процесори працюють незалежно один від одного. Однак можна реалізувати і моделі, що використовують міжпроцесорний обмін. Використання загальної пам'яті збільшує швидкість такого обміну, користувач також має доступ відразу до всього об'єму пам'яті. Для SMP-систем існують досить ефективні засоби автоматичного розпаралелювання;

- простота експлуатації. Як правило, SMP-системи використовують систему кондиціонування, засновану на повітряному охолодженні, що полегшує їх технічне обслуговування;

- відносно невисока ціна.

Недоліки:

- системи із загальною пам'яттю погано масштабуються.

Цей істотний недолік SMP-систем не дозволяє вважати їх по-справжньому перспективними. Причиною поганої масштабованості є те, що в даний момент шина здатна обробляти тільки одну транзакцію,

внаслідок чого виникають проблеми вирішення конфліктів при одночасному зверненні декількох процесорів до одних і тих же областях загальної фізичної пам'яті. Обчислювальні елементи починають один одному заважати. Коли відбудеться такий конфлікт, залежить від швидкості зв'язку та від кількості обчислювальних елементів. В даний час конфлікти можуть відбуватися при наявності 8-24 процесорів. Крім того, системна шина має обмежену (хоч і високу) пропускну здатність (ПС) і обмежене число слотів. Все це очевидно перешкоджає збільшенню продуктивності при збільшенні числа процесорів і числа підключаються користувачів. У реальних системах можна задіяти не більше 32 процесорів. Для побудови масштабованих систем на базі SMP використовуються кластерні або NUMA-архітектури. При роботі з SMP-системами використовують так звану парадигму програмування з пам'яттю (shared memory paradigm).

MPP-архітектура. MPP (massive parallel processing) - масивно-паралельна архітектура. Головна особливість такої архітектури полягає в тому, що пам'ять фізично розділена. В цьому випадку система будується з окремих модулів, що містять процесор, локальний банк операційної пам'яті (ОП), комунікаційні процесори (Рутер) або мережеві адаптери, іноді - жорсткі диски і / або інші пристрої введення / виводу. По суті, такі модулі представляють собою повнофункціональні комп'ютери (див. Рис.3.2). Доступ до банку ОП з даного модуля мають тільки процесори (ЦП) з цього ж модуля. Модулі з'єднуються спеціальними комунікаційними каналами. Користувач може визначити логічний номер процесора, до якого він підключений, і організувати обмін повідомленнями з іншими процесорами. Використовуються два варіанти роботи операційної системи (ОС) на

машинах MPP-архітектури. В одному повноцінна операційна система (ОС) працює тільки на керуючу машину (front-end), на кожному окремому модулі функціонує сильно урізаний варіант ОС, що забезпечує роботу тільки розташованої в ньому гілки паралельного застосування. У другому варіанті на кожному модулі працює повноцінна UNIX-подібна ОС, що встановлюється окремо.

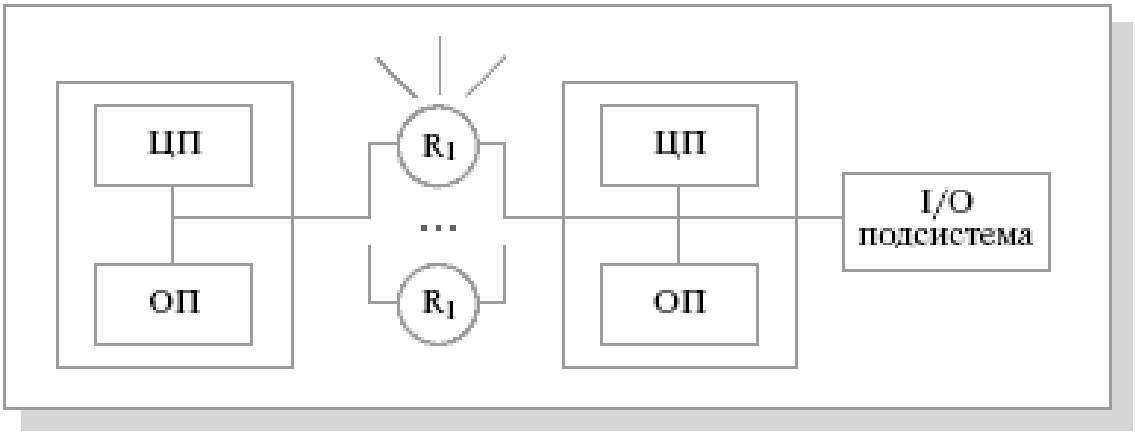

Рисунок 4.2 - Схематичний вид архітектури з роздільною пам'яттю

Головною перевагою систем з роздільною пам'яттю є хороша масштабованість: на відміну від SMP-систем, в машинах з роздільною пам'яттю кожен процесор має доступ тільки до своєї локальної пам'яті, у зв'язку з чим не виникає необхідності в потактовой синхронізації процесорів. Практично всі рекорди по продуктивності на сьогодні встановлюються на машинах саме такої архітектури, що складаються з декількох тисяч процесорів (ASCI Red, ASCI Blue Pacific).

Недоліки архітектури:

- відсутність спільної пам'яті помітно знижує швидкість межпроцессорного обміну, оскільки немає загального середовища для зберігання даних, призначених для обміну між процесорами. Потрібна

спеціальна техніка програмування для реалізації обміну повідомленнями між процесорами;

- кожен процесор може використовувати тільки обмежений обсяг локального банку пам'яті;

- внаслідок зазначених архітектурних недоліків потрібні значні зусилля для того, щоб максимально використовувати системні ресурси. Саме цим визначається висока ціна програмного забезпечення для масивно-паралельних систем з роздільною пам'яттю.

Системами з роздільною пам'яттю є суперкомп'ютери БКС-1000, IBM RS/ 6000 SP, SGI/CRAY T3E, системи ASCI, Hitachi SR8000, системи Parsytec.

Машини останньої серії CRAY T3E від SGI, засновані на базі процесорів Dec Alpha 21164 з піковою продуктивністю 1200 Мфлопс (CRAY T3E-1200), здатні масштабироваться до 2048 процесорів.

При роботі з MPP-системами використовують так звану Massive Passing Programming Paradigm - парадигму програмування з передачею даних (MPI, PVM, BSPlib).

4.2. Гібридна архітектура NUMA

Головна особливість гібридної архітектури NUMA (nonuniform memory access) - неоднорідний доступ до пам'яті.

Гібридна архітектура поєднує переваги систем зі спільною пам'яттю і відносну дешевизну систем з роздільною пам'яттю. Суть цієї архітектури - в особливій організації пам'яті, а саме: пам'ять фізично розподілена по різних частинах системи, але логічно вона є спільною, так що користувач бачить єдиний адресний простір. Система побудована з однорідних базових модулів (плат), що складаються з невеликої кількості процесорів і блоку пам'яті. Модулі

об'єднані за допомогою високошвидкісного комутатора. Підтримується єдиний адресний простір, апаратно підтримується доступ до віддаленої пам'яті, тобто до пам'яті інших модулів. При цьому доступ до локальної пам'яті здійснюється в кілька разів швидше, ніж до віддаленої. По суті, архітектура NUMA є MPP (масивно-паралельної) архітектурою, де в якості окремих обчислювальних елементів беруться SMP (сімметрічная многопроцессорная архітектура) вузли. Доступ до пам'яті та обмін даними всередині одного SMP-вузла здійснюється через локальну пам'ять вузла і відбувається дуже швидко, а до процесорів іншого SMP-вузла теж є доступ, але більш повільний і через більш складну систему адресації.

Структурна схема комп'ютера з гібридною мережею: чотири процесори зв'язуються між собою за допомогою кроссбара в рамках одного SMP-вузла. Вузли зв'язані мережею типу "метелик" (Butterfly):

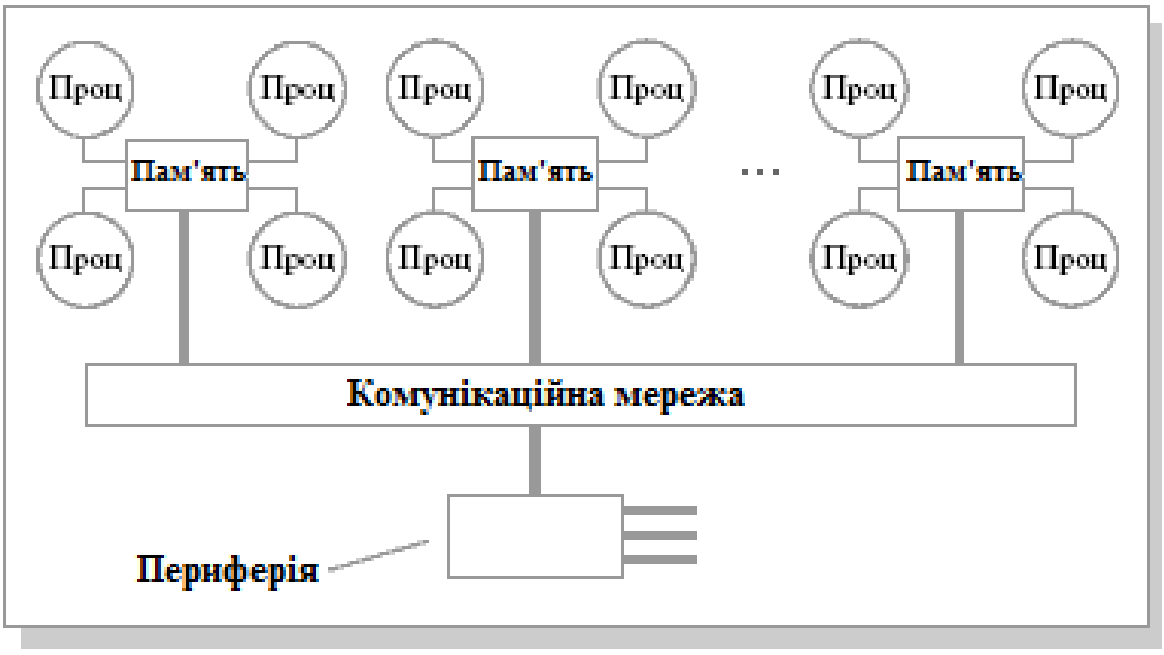

Рисунок 4.3 - Структурна схема комп'ютера з гібридною мережею.

Вперше ідею гібридної архітектури запропонував Стів Воллох, він втілив її в системах серії Exemplar. Варіант Воллоха - система, що складається з восьми SMP-вузлів. Фірма HP купила ідею і реалізувала на суперкомп'ютерах серії SPP. Ідею підхопив Сеймур Крей (Seymour R.Cray) і додав новий елемент - когерентний кеш, створивши так звану архітектуру cc-NUMA (Cache Coherent Non-Uniform Memory Access), яка розшифровується як "неоднорідний доступ до пам'яті з забезпеченням когерентності кешей". Він її реалізував на системах типу Origin.

4.3. Організація когерентності багаторівневої ієрархічної пам'яті

Поняття когерентності кешей описує той факт, що всі центральні процесори отримують однакові значення одних і тих же змінних в будь-який момент часу. Дійсно, оскільки кеш-пам'ять належить окремому комп'ютеру, а не всієї многопроцессорной системі в цілому, дані, що потрапляють в кеш одного комп'ютера, можуть бути недоступні іншому. Щоб цього уникнути, слід провести синхронізацію інформації, що зберігається в кеш-пам'яті процесорів.

Для забезпечення когерентності кешей існує кілька можливостей:

- використовувати механізм відстежування шинних запитів (snoopy bus protocol), в якому кеші відстежують змінні, що передаються до будь-якого з центральних процесорів і при необхідності модифікують власні копії таких змінних;

- виділяти спеціальну частину пам'яті, що відповідає за відстеження достовірності всіх використовуваних копій змінних.

Найбільш відомими системами архітектури cc-NUMA є: HP

9000 V-class в SCA-конфігураціях, SGI Origin3000, Sun HPC 15000, IBM / Sequent NUMA-Q 2000. На сьогодні максимальне число процесорів в cc-NUMA-системах може перевищувати 1000 (серія Origin3000). Зазвичай вся система працює під управлінням єдиної ОС, як в SMP. Можливі також варіанти динамічного "підрозділу" системи, коли окремі "розділи" системи працюють під управлінням різних ОС. При роботі з NUMA-системами, так же, як з SMP, використовують так звану парадигму програмування із загальною пам'яттю (shared memory paradigm).

5. АРХІТЕКТУРА ОБЧИСЛЮВАЛЬНИХ СИСТЕМ. PVP-АРХІТЕКТУРА. КЛАСТЕРНА АРХІТЕКТУРА

5.1. PVP-архітектура

PVP (Parallel Vector Process) - паралельна архітектура з векторними процесорами. Основною ознакою PVP-систем є наявність спеціальних векторно-конвеєрних процесорів, в яких передбачені команди однотипної обробки векторів незалежних даних, ефективно виконуються на конвеєрних функціональних пристроях. Як правило, кілька таких процесорів (1-16) працюють одночасно із загальною пам'яттю (аналогічно SMP) в рамках багатопроцесорних конфігурацій. Кілька вузлів можуть бути об'єднані за допомогою комутатора (аналогічно MPP). Оскільки передача даних у векторному форматі здійснюється набагато швидше, ніж в скалярному (максимальна швидкість може складати 64 Гбайт / с, що на 2 порядки швидше, ніж в скалярних машинах), то проблема взаємодії між потоками даних при розпаралелювання стає несуттєвою. І те, що погано распараллеливается на скалярних машинах, добре распараллеливается на векторних. Таким чином, системи PVP-архітектури можуть бути машинами загального призначення (general purpose systems). Однак, оскільки векторні процесори досить дорого коштують, ці машини не можуть бути загальнодоступними.

Найбільш популярні машини PVP-архітектури:

1. CRAY X1, SMP -Архітектура. Пікова продуктивність системи в стандартній конфігурації може становити десятки терафлопс.

2. NEC SX-6, NUMA-архітектура. Пікова продуктивність системи може досягати 8 Тфлопс, продуктивність одного процесора

становить 9,6 Гфлопс. Система масштабується з єдиним чином операційної системи до 512 процесорів.

Рисунок 5.1 - CRAY SV-2

3. Fujitsu-VPP5000 (vector parallel processing), MPP - Архітектура. Продуктивність одного процесора складає 9.6 Гфлопс, пікова продуктивність системи може досягати 1249 Гфлопс, максимальна ємність пам'яті - 8 Тбайт.

Парадигма програмування на PVP-системах передбачає векторизацію циклів (для досягнення розумної продуктивності одного процесора) і їх розпаралелювання (для одночасного завантаження декількох процесорів одним додатком).

На практиці рекомендується виконувати наступні процедури:

- виробляти векторизацію вручну, щоб перевести задачу в матричну форму. При цьому, відповідно до довжини вектора, розміри матриці повинні бути кратні 128 або 256;

- працювати з векторами у віртуальному просторі, розкладаючи шукану функцію в ряд і залишаючи число членів ряду, кратне 128 або 256.

Рисунок 5.2 - Fujitsu-VPP5000

За рахунок великого фізичного пам'яті (частки терабайта) навіть погано векторізуемие завдання на PVP-системах вирішуються швидше на машинах зі скалярними процесорами.

Кластер являє собою два або більше комп'ютерів (часто званих

вузлами), що об'єднуються за допомогою мережевих технологій на базі шинної архітектури або комутатора і постають перед користувачами в якості єдиного інформаційно-обчислювального ресурсу. Як вузлів кластера можуть бути обрані сервери, робочі станції і навіть звичайні персональні комп'ютери. Вузол характеризується тим, що на ньому працює єдина копія операційної системи. Перевага кластеризації для підвищення працездатності стає очевидним у разі збою будь-якого вузла: при цьому інший вузол кластера може взяти на себе навантаження несправного вузла, і користувачі не помітять переривання в доступі. Можливості масштабованості кластерів дозволяють багаторазово збільшувати продуктивність додатків для більшого числа користувачів технологій (Fast / Gigabit Ethernet, Myrinet) на базі шинної архітектури або комутатора. Такі суперкомп'ютерні системи є найдешевшими, оскільки збираються на базі стандартних комплектуючих елементів ("off the shelf"), процесорів, комутаторів, дисків і зовнішніх пристроїв.

Кластеризація може здійснюватися на різних рівнях комп'ютерної системи, включаючи апаратне забезпечення, операційні системи, програми-утиліти, системи управління і додатки. Чим більше рівнів системи об'єднані кластерної технологією, тим вище надійність, масштабованість і керованість кластеру.

5.2. Типи кластерів

Умовний розподіл на класи запропоновано Язеком радаевского і Дугласом Едлайном:

- Клас I. Клас машин будується цілком із стандартних деталей, які продають багато постачальників комп'ютерних компонентів (низькі ціни, просте обслуговування, апаратні компоненти доступні з різних

джерел).

- Клас II. Система має ексклюзивні або не дуже широко поширені деталі. Таким чином можна досягти дуже хорошою продуктивності, але при більш високій вартості.

Як уже зазначалося, кластери можуть існувати в різних конфігураціях. Найбільш поширеними типами кластерів є:

-системи високої надійності;

- системи для високопродуктивних обчислень;

- багатопотокові системи.

Відзначимо, що межі між цими типами кластерів до деякої міри розмиті, і кластер може мати такі властивості або функції, які виходять за рамки перерахованих типів. Більш того, при конфігуруванні великого кластера, використовуваного як система загального призначення, доводиться виділяти блоки, виконують всі перераховані функції.

Кластери для високопродуктивних обчислень призначені для паралельних розрахунків. Ці кластери зазвичай зібрані з великої кількості комп'ютерів. Розробка таких кластерів є складним процесом, що вимагає на кожному кроці узгодження таких питань як інсталяція, експлуатація і одночасне управління великим числом комп'ютерів, технічні вимоги паралельного і високопродуктивного доступу до одного і того ж системного файлу (або файлів) і межпроцессорная зв'язок між вузлами, і координація роботи в паралельному режимі. Ці проблеми найпростіше вирішуються при забезпеченні єдиного образу операційної системи для всього кластера. Однак реалізувати подібну схему вдається далеко не завжди і зазвичай вона застосовується лише для не дуже великих систем.

Багатопотокові системи використовуються для забезпечення єдиного інтерфейсу до ряду ресурсів, які можуть з часом довільно нарощуватися (або скорочуватися). Типовим прикладом може служити група web-серверів.

У 1994 році Томас Стерлінг (Sterling) і Дон Беккер (Becker) створили 16-вузловий кластер з процесорів Intel DX4, з'єднаних мережею 10 Мбіт / с Ethernet з дублюванням каналів. Вони назвали його "Beowulf" за назвою старовинної епічної поеми. Кластер виник в центрі NASA Goddard Space Flight Center для підтримки необхідними обчислювальними ресурсами проекту Earth and Space Sciences. Проектно-конструкторські роботи швидко перетворилися в те, що відомо зараз як проект Beowulf.

Проект став основою загального підходу до побудови паралельних кластерних комп'ютерів, він описує многопроцессорную архітектуру, яка може з успіхом використовуватися для паралельних обчислень. Beowulf- кластер, як правило, є системою, що складається з одного серверного вузла (який зазвичай називається головним), а також одного або декількох підлеглих (обчислювальних) вузлів, з'єднаних за допомогою стандартної комп'ютерної мережі. Система будується з використанням стандартних апаратних компонентів, таких як ПК, що запускаються під Linux, стандартні мережеві адаптери (наприклад, Ethernet) і комутатори. Немає особливого програмного пакету, званого "Beowulf". Замість цього є кілька шматків програмного забезпечення, які багато користувачів знайшли придатними для побудови кластерів Beowulf. Beowulf використовує такі програмні продукти як операційна система Linux, системи передачі повідомлень PVM, MPI, системи управління чергами завдань і інші стандартні

продукти. Серверний вузол контролює весь кластер і обслуговує файли, що направляються до клієнтських вузлів.

5.3. Проблеми виконання мережі зв'язку процесорів в кластерної системі

Архітектура кластерної системи (спосіб з'єднання процесорів один з одним) більшою мірою визначає її продуктивність, ніж тип використовуваних в ній процесорів. Критичним параметром, що впливає на величину продуктивності такої системи, є відстань між процесорами. Так, з'єднавши разом 10 персональних комп'ютерів, ми отримаємо систему для проведення високопродуктивних обчислень. Проблема, однак, буде складатися в пошуку найбільш ефективного способу з'єднання стандартних засобів один з одним, оскільки при збільшенні продуктивності кожного процесора в 10 разів продуктивність системи в цілому в 10 разів збільшено.

Розглянемо для прикладу завдання побудови симетричної 16-процесорної системи, в якій всі процесори були б рівноправні. Найбільш природним представляється з'єднання у вигляді плоскої решітки, де зовнішні кінці використовуються для під'єднання зовнішніх пристроїв.

При такому типі з'єднання максимальна відстань між процесорами виявиться рівним 6 (кількість зв'язків між процесорами, що відокремлюють найближчий процесор від самого далекого). Теорія ж показує, що якщо в системі максимальна відстань між процесорами більше 4, то така система не може працювати ефективно. Тому при з'єднанні 16 процесорів один з одним плоска схема є недоцільною. Для отримання більш компактною конфігурації необхідно вирішити задачу про знаходження фігури, що має максимальний обсяг при мінімальній

площі поверхні. У тривимірному просторі такою властивістю володіє куля. Але оскільки нам необхідно побудувати вузлову систему, замість кулі доводиться використовувати куб (якщо число процесорів дорівнює 8) або гіперкуб, якщо число процесорів більше 8. Розмірність гіперкуба визначатиметься в залежності від числа процесорів, які необхідно з'єднати. Так, для з'єднання 16 процесорів буде потрібно чотиривимірний гіперкуб. Для його побудови слід взяти звичайний тривимірний куб, зрушити в потрібному напрямку і, з'єднавши вершини, отримати гіперкуб розміром 4.

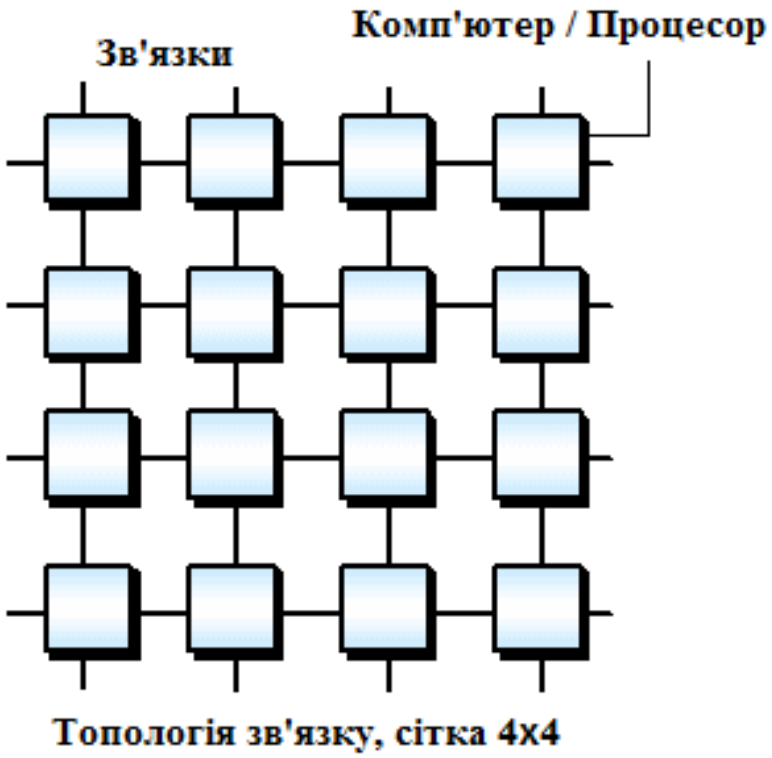

Рисунок 5.3 - Схема з'єднання процесорів у вигляді плоскої решітки

Архітектура гіперкуба є другою за ефективністю, але найбільш наочною. Використовуються й інші топології мереж зв'язку: тривимірний тор, "кільце", "зірка" і інші.

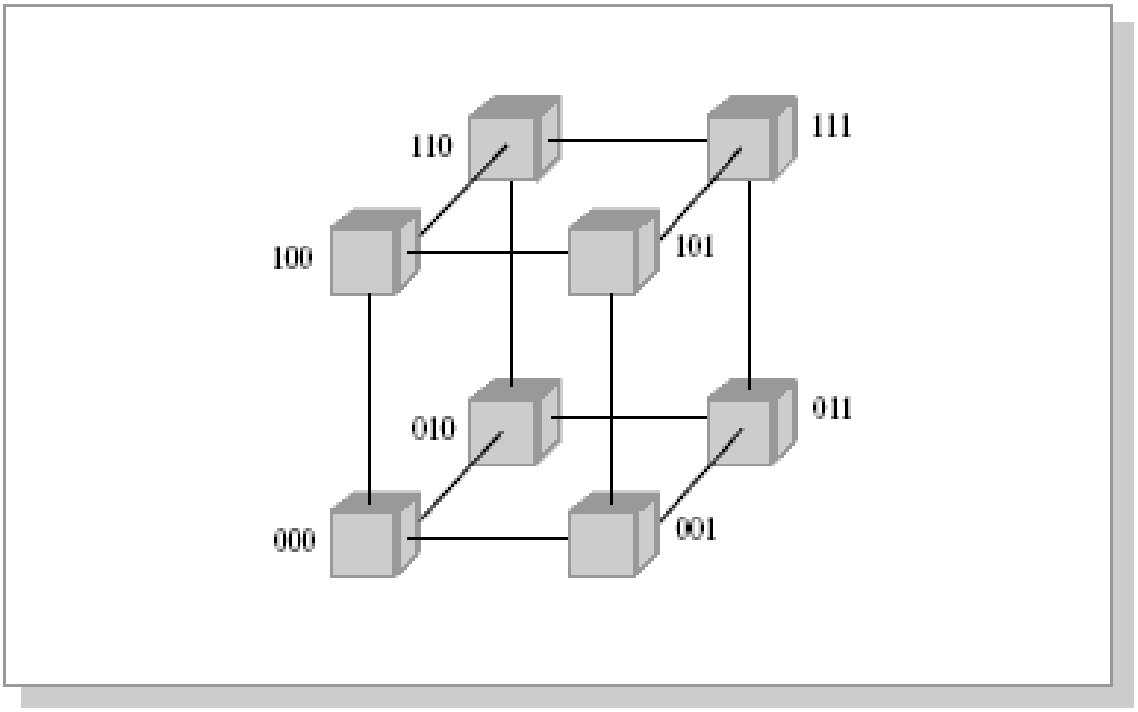

Рисунок 5.4 - Топологія зв'язку, 3-х мірний гіперкуб

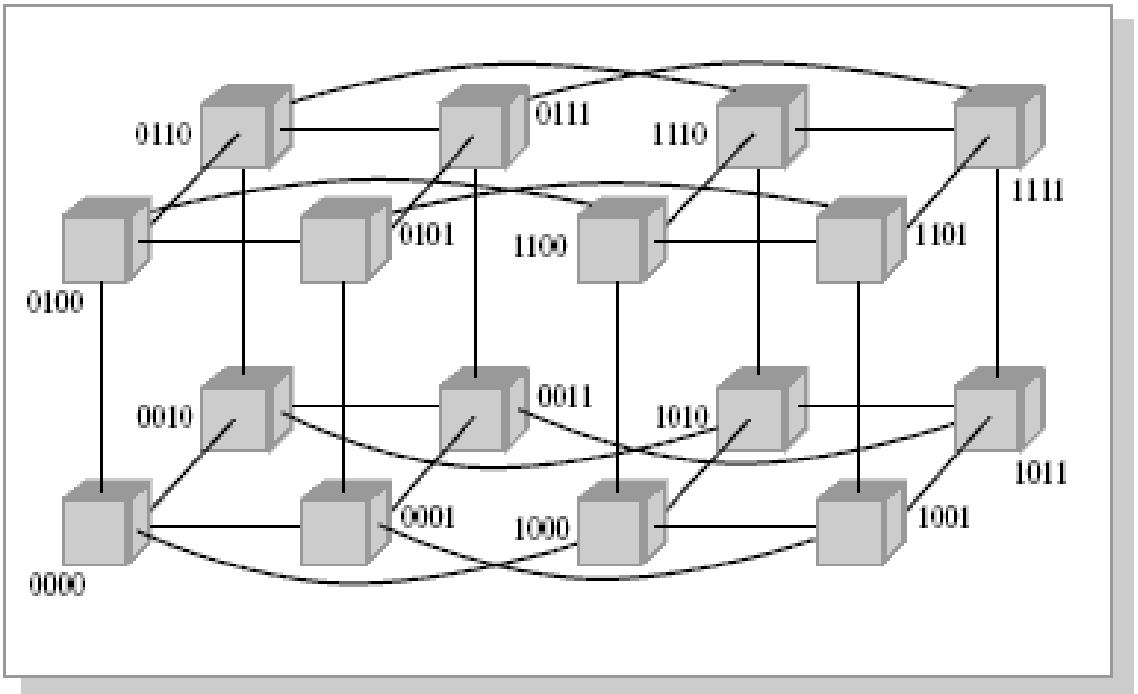

Рисунок 5.5 - Топологія зв'язку, 4-х мірний гіперкуб

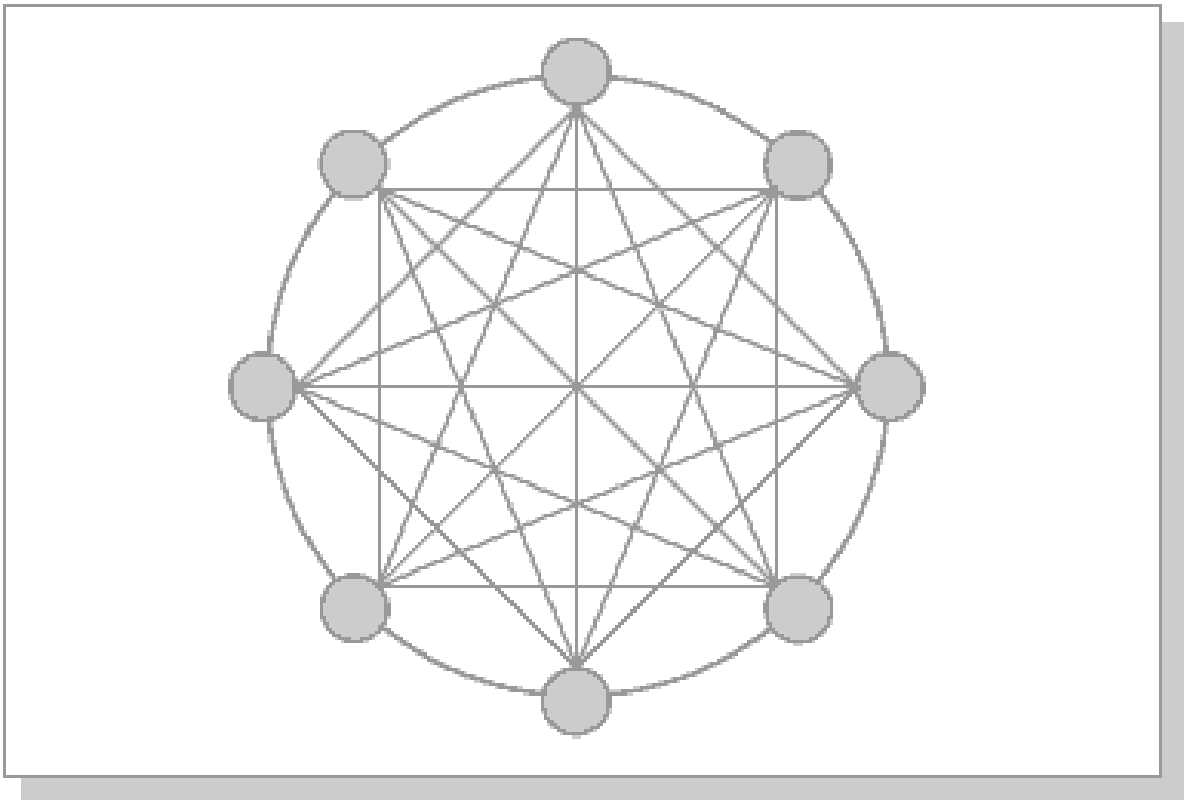

Рисунок 5.6 - Архітектура кільця з повним зв'язком по хордам (Chordal Ring)

Найбільш ефективною є архітектура з топологією "товстого дерева" (fat-tree). Архітектура "fat-tree" (hypertree) була запропонована Лейзерсон (Charles E. Leiserson) в 1985 році. Процесори локалізовані в листі дерева, в той час як внутрішні вузли дерева скомпоновані у внутрішню мережу. Піддерева можуть спілкуватися між собою, не зачіпаючи більш високих рівнів мережі.

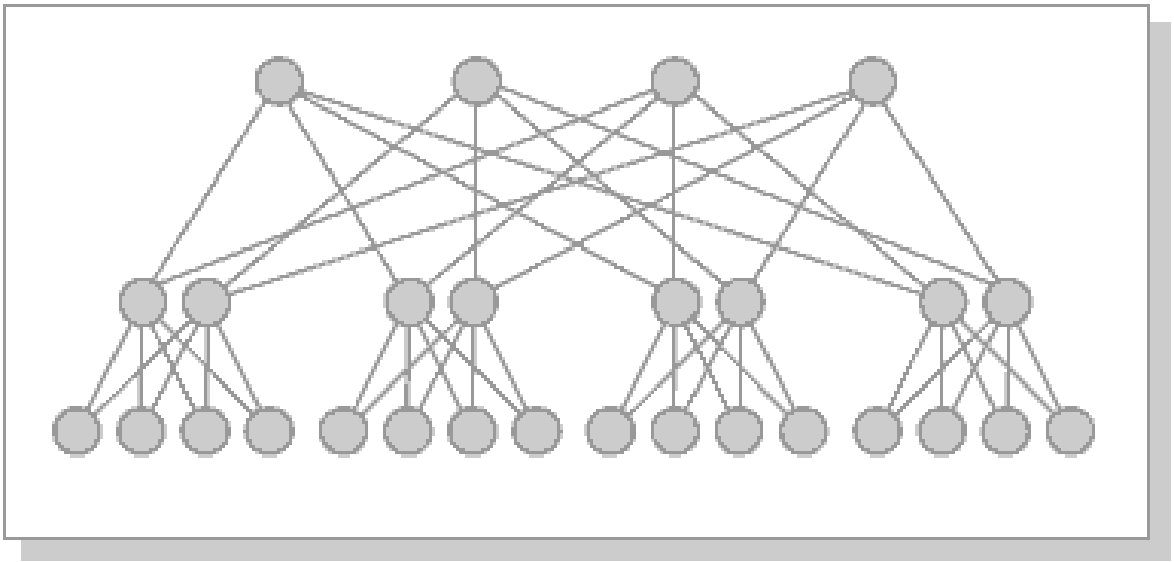

Рисунок 5.7 - Кластерна архітектура "Fat-tree"

Оскільки спосіб з'єднання процесорів один з одним більше впливає на продуктивність кластера, ніж тип використовуваних в ній процесорів, то може виявитися більш доцільним створити систему з більшого числа дешевих комп'ютерів, ніж з меншого числа дорогих. У кластерах, як правило, використовуються операційні системи, стандартні для робочих станцій, найчастіше вільно поширювані (Linux, FreeBSD), разом зі спеціальними засобами підтримки паралельного програмування і балансування навантаження. При роботі з кластерами, так само, як і з MPP-системами, використовують так звану Massive Passing Programming Paradigm - парадигму програмування з передачею даних (найчастіше - MPI). Помірна ціна подібних систем обертається великими накладними витратами на взаємодію паралельних процесів між собою, що сильно звужує потенційний клас розв'язуваних задач.

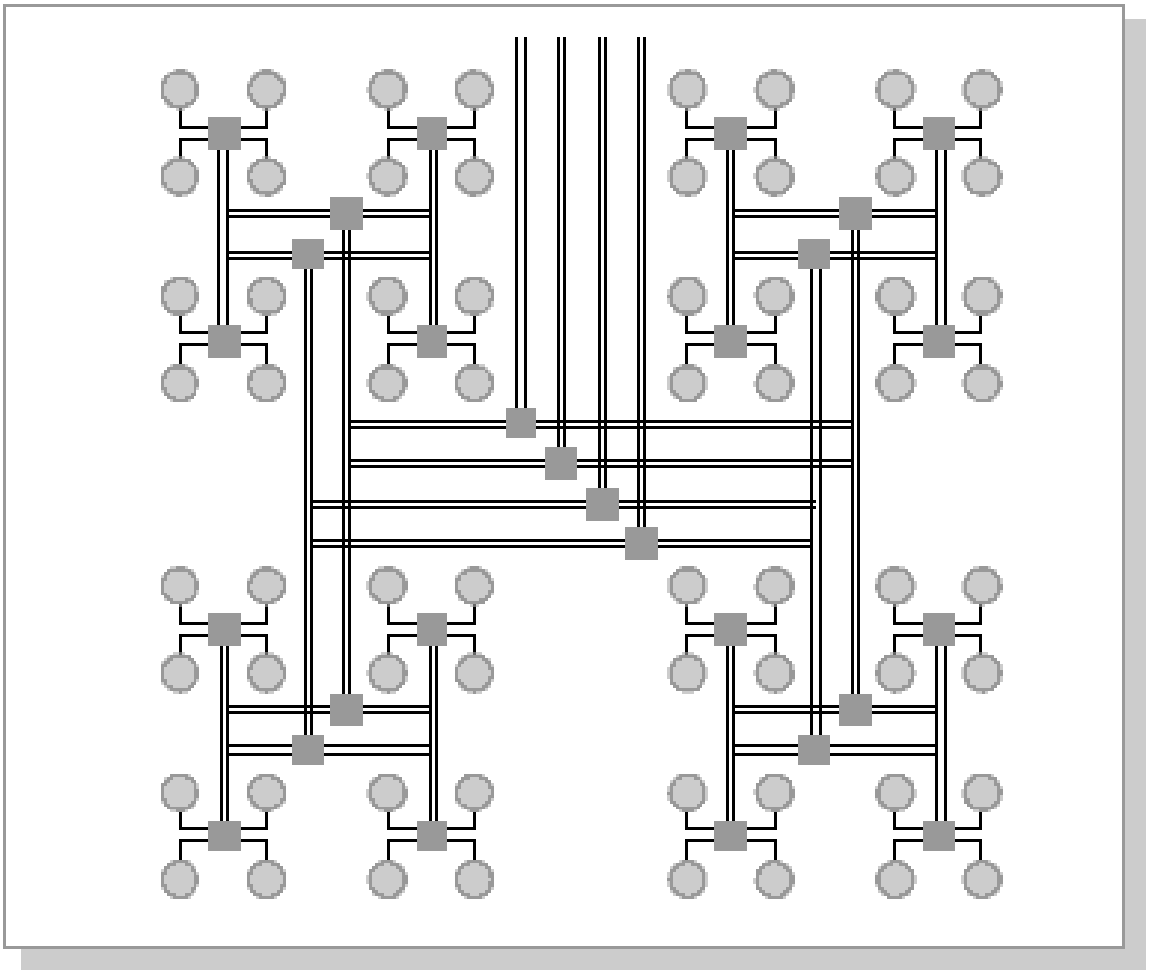

Рисунок 5.8 - Кластерна архітектура "Fat-tree"

(вид зверху на попередню схему)

6. ПРИНЦИПИ ПОБУДОВИ КОМУНІКАЦІЙНИХ СЕРЕДОВИЩ

6.1. Способи з'єжнання комп'ютерів між собою

У найзагальнішому сенсі архітектуру комп'ютера можна визначити як спосіб з'єднання комп'ютерів між собою, з пам'яттю і з зовнішніми пристроями. Реалізація цього з'єднання може йти різними шляхами. Конкретна реалізація такого роду сполук називається комунікаційним середовищем комп'ютера. Одна з найпростіших реалізацій - це використання загальної шини, до якої підключаються як процесори, так і пам'ять. Сама шина складається з певного числа ліній зв'язку, необхідних для передачі адрес, даних і керуючих сигналів між процесором і пам'яттю. Цей спосіб реалізований в SMP-системах. Основним недоліком таких систем, як було зазначено вище, є погана масштабованість. Збільшення, навіть незначне, числа пристроїв на шині викликає помітні затримки при обміні з пам'яттю і катастрофічне падіння продуктивності системи в цілому. Необхідні інші підходи для побудови комунікаційного середовища, і одним з них є поділ пам'яті на незалежні модулі та забезпечення можливості доступу різних процесорів до різних модулів одночасно за допомогою використання різного роду комутаторів .

При цьому можливі різні конфігурації утворюються систем зв'язку. Так, в комп'ютерах сімейства Cray T3D/T3E все процесори були об'єднані спеціальними високошвидкісними каналами в тривимірний тор, в якому кожен обчислювальний вузол мав безпосередні зв'язки з шістьма сусідами. У комп'ютерах IBM SP/2 взаємодія процесорів відбувається через ієрархічну систему комутаторів, також забезпечує можливість з'єднання кожного

процесора з будь-яким іншим. Ці оригінальні унікальні рішення значно збільшують ціну комп'ютерів.

Набагато більш простим і дешевим виявилося використання зв'язків на базі мереж Ethernet - методика, розроблена компанією Xerox. Спочатку використовувалася звичайна 10-мегабітних мережу, потім стали застосовувати Fast Ethernet, а останнім часом іноді і Gigabit Ethernet. Але для Fast Ethernet характерна велика латентність (затримка в передачі даних), що оцінюється в 160-180 мікросекунд, а Gigabit Ethernet відрізняється високою вартістю. Крім того, він ефективний тільки при з'єднанні точка-точка, при з'єднанні декількох вузлів його ефективність різко падає, а при з'єднанні більше 5-6 вузлів вона не перевищує по продуктивності навіть Fast Ethernet. Тому при створенні багатопроцесорних обчислювальних систем часто перевагу надають технологіям SCI, Myrinet або Raceway.

6.2. Приклади побудови комунікаційних середовищ на основі масштабується когерентного інтерфейсу SCI.

SCI (Scalable Coherent Interface) прийнятий як стандарт в 1992 р (ANSI / IEEE Std 1596-1992). Він призначений для досягнення високих швидкостей передачі з малим часом затримки і при цьому забезпечує масштабовану архітектуру, що дозволяє будувати системи, що складаються з безлічі блоків. SCI є комбінацією шини і локальної мережі, забезпечує реалізацію когерентності кеш-пам'яті, що розміщується в вузлі SCI, за допомогою механізму розподілених директорій, який покращує продуктивність, приховуючи витрати на доступ до віддалених даних в моделі з розподіленою пам'яттю. Продуктивність передачі даних зазвичай знаходиться в межах від 200 Мбайт/с до 1000 Мбайт/с на відстанях десятків метрів з

використанням електричних кабелів і кілометрів - з використанням оптоволокна. SCI зменшує час міжвузлових комунікацій в порівнянні з традиційними схемами передачі даних в мережах шляхом усунення звернень до програмних рівнями - операційній системі і бібліотекам часу виконання; комунікації представляються як частина простої операції завантаження даних процесором (командами load або store). Зазвичай звернення до даних, фізично розташованим в пам'яті іншого обчислювального вузла і не знаходяться в кеші, призводить до формування запиту до віддаленого вузла для отримання необхідних даних, які протягом декількох мікросекунд доставляються в локальний кеш, і виконання програми продовжується. Колишній підхід вимагав формування пакетів на програмному рівні з подальшою передачею їх апаратного забезпечення. Точно так же відбувався і прийом, в результаті чого затримки були в сотні разів більше, ніж у SCI. Однак для сумісності SCI має можливість переносити пакети інших протоколів.

Ще одна перевага SCI - використання простих протоколів типу RISC, які забезпечують більшу пропускну здатність. Вузли з адаптерами SCI можуть використовувати для з'єднання комутатори або ж з'єднуватися в кільце. Зазвичай кожен вузол виявляється включеним в два кільця (рис. 5.1).

На відміну від HIPPI, дана технологія оптимізована для роботи з динамічним трафіком, проте може бути менш ефективна при роботі з великими блоками даних. Протокол передачі даних забезпечує гарантовану доставку та відсутність дедлоків. Протокол SCI досить складний, він передбачає широкі можливості управління трафіком, але використання цих можливостей передбачає наявність розвиненого

програмного забезпечення. На комунікаційної технології SCI заснована система зв'язку гіперузлов CTI (Convex Torroidal Interconnect) в системах HP / Convex Exemplar X-class, крім того, на ній побудовані кластерні системи SCALI Computer, системи сімейства hpcLine компанії Siemens, а також cc-NUMA сервера Data General і Sequent.

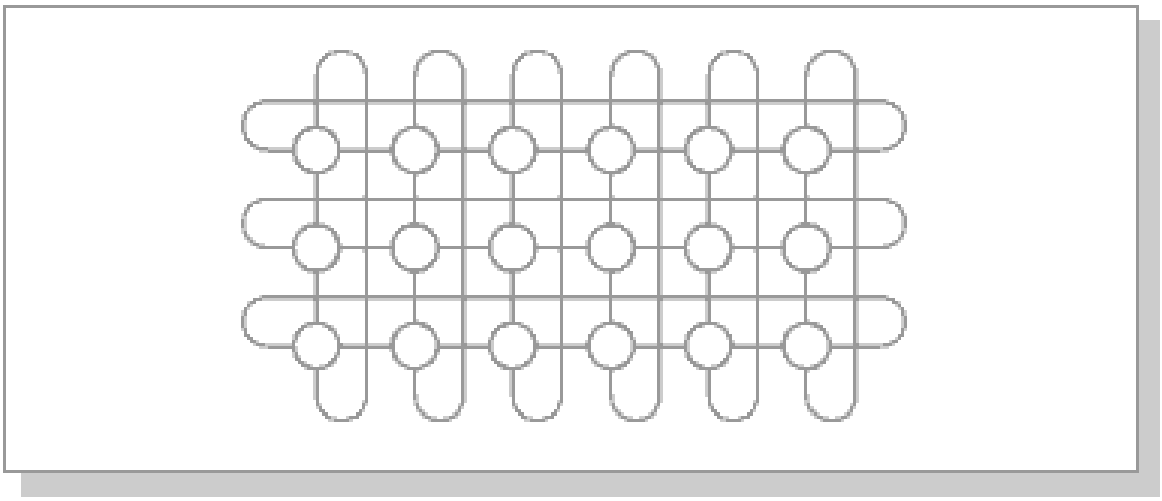

Рисунок 6.1 - Матриця вузлів кластера на основі мережі SCI

Традиційна область застосування SCI - це комунікаційні середовища багатопроцесорних систем. На основі цієї технології побудовані, зокрема, комп'ютери серії hpcLine від Siemens або модульні сервери NUMA-Q від IBM, раніше відомі як Sequent.

Модульні SCI-комутатори Dolphin дозволяють споживачам будувати масштабовані кластерні рішення класу підприємства на платформах Windows NT/2000/ XP, Linux, Solaris, VxWorks, LynuxWorks і NetWare з використанням стандартизованого обладнання та програмного забезпечення.

Таблиця 5.1. технологія SCI

Виробники обладнання	Dolphin Interconnect Solutions та ін.
Показники продуктивності	Для продуктів Dolphin: пікова пропускна здатність - 667 Мбіт/с. Апаратна латентність - 1.46 мксек
Програмна підтримка	Драйвери для Linux, Windows NT, Solaris. ScaMPI - реалізація MPI компанії Scali Computer для систем на базі SCI. SISCI API - інтерфейс програмування нижнього рівня
Коментарі	SCI (ANSI / IEEE l596-1992) – стандартизо-вана технологія. Крім стандартної мережевої середовища, SCI підтримує побудову систем з пам'яттю та з когерентністю кешей. На комунікаційної технології SCI засновані кластерні системи компанії SCALI Computer, системи сімейства hpcLine компанії Siemens, а також cc-NUMA-сервера Data General і Sequent. Технологія SCI використовувалася для зв'язку гиперузлов в системах HP / Convex Exemplar X-class.

6.3. Комунікаційне середовище MYRINET

Мережеву технологію Myrinet представляє компанія Myricom, яка вперше запропонувала свою комунікаційну технологію в 1994 році, а на сьогодні має вже понад 1000 інсталяцій по всьому світу.

Технологія Myrinet заснована на використанні багатопортових комутаторів при обмежених декількома метрами довжинах зв'язків вузлів з портами комутатора. Вузли в Myrinet з'єднуються один з одним через комутатор (до 128 портів). Максимальна довжина ліній зв'язку варіюється в залежності від конкретної реалізації.

Як коммутируемая мережу, аналогічна за структурою сегментам Ethernet, сполученим за допомогою комутаторів, Myrinet може одночасно передавати кілька пакетів, кожен з яких йде зі швидкістю, близькою до 2 Гбіт / с. На відміну від некомутованих Ethernet і FDDI мереж, які поділяють загальну середу передачі, сукупна пропускна здатність мережі Myrinet зростає зі збільшенням кількості машин. На сьогодні Myrinet найчастіше використовують як локальну мережу (LAN) порівняно невеликого розміру, пов'язуючи разом комп'ютери усередині кімнати або будівлі. Через свою високу швидкість, малого часу затримки, прямої комутації і помірної вартості Myrinet часто використовується для об'єднання комп'ютерів в кластери. Myrinet також використовується як системна мережа (System Area Network, SAN), яка може об'єднувати комп'ютери в кластер всередині стійки з тією ж продуктивністю, але з більш низькою вартістю, ніж Myrinet LAN. Пакети Myrinet можуть мати будь-яку довжину. Таким чином, вони можуть включати в себе інші типи пакетів, в тому числі IP-пакети. Об'єднання обчислювальних вузлів з адаптерами Myrinet в мережу відбувається за допомогою комутаторів, які мають зараз 4, 8, 12 або 16 портів. У комутаторах використовується передача пакетів шляхом встановлення з'єднання на час передачі, для маршрутизації повідомлень застосовується алгоритм прокладки шляху (wormhole, "червоточина"). Комутатори, як і мережеві адаптери, побудовані на

спеціалізованих мікропроцесорах LANai компанії Myricom.

Таблиця 5.2. технологія Myrinet

Виробники обладнання	Myricom
Показники продуктивності	Пікова пропускна здатність - 2 Гбіт / с, повний дуплекс. Латентність - близько 4 мксек.
Програмна підтримка	Драйвери для Linux (Alpha, x86, PowerPC, UltraSPARC), Windows NT (x86), Solaris (x86, UltraSPARC) і Tru64 UNIX. GM - інтерфейс програмування на нижньому рівні. Пакети HPVM (включає MPI-FM, реалізацію MPI для Myrinet), BIP-MPI і ін.
Коментарі	Myrinet є відкритим стандартом. Myricom пропонує широкий вибір мережевого обладнання за порівняно невисокими цінами. На фізичному рівні підтримуються мережеві середовища SAN (System Area Network), LAN (CL-2) і оптоволокно. Технологія Myrinet надає широкі можливості масштабування мережі і в даний час дуже часто використовується при побудові високопродуктивних кластерів

На фізичному рівні лінки Myrinet складаються з 9 провідників: 8 бітів призначені для передачі інформації, що інтерпретується в залежності від стану дев'ятого біта як байт даних або керуючий

символ; при цьому на кожному лінк забезпечується управління потоком і контроль помилок. Середа Myrinet вигідно відрізняється від багатьох інших середовищ передачі, зокрема, SCI, простотою концепції і апаратної реалізації протоколів. Вона містить обмежений набір засобів управління трафіком, що використовують припливів-буфер, керуючі символи і таймерні інтервали. Myrinet є відкритим стандартом, компанія Myricom пропонує широкий вибір мережевого обладнання за порівняно невисокими цінами. Технологія Myrinet надає широкі можливості масштабування мережі і часто використовується при побудові високопродуктивних обчислювальних кластерів.

7. СПОСОБИ ОРГАНІЗАЦІЇ ВИСОКОПРОДУКТИВНИХ ПРОЦЕСОРІВ. АСОЦІАТИВНІ ПРОЦЕСОРИ. КОНВЕЄРНІ ПРОЦЕСОРИ. МАТРИЧНІ ПРОЦЕСОРИ

Існуючі в даний час алгоритми прикладних задач, системне програмне забезпечення і апаратні засоби переважно орієнтовані на традиційну адресну обробку даних. Дані повинні бути представлені у вигляді обмеженої кількості форматів (наприклад, масиви, списки, записи), повинна бути явно створена структура зв'язків між елементами даних за допомогою покажчиків на адреси елементів пам'яті, при обробці цих даних повинна бути виконана сукупність операцій, які забезпечують доступ до даних по вказівниками. Такий підхід зумовлює громіздкість операційних систем і систем програмування, а також служить перешкодою до створення обчислювальних засобів з архітектурою, орієнтованої на більш ефективне використання паралелізму обробки даних.

Асоціативні процесори. Асоціативний спосіб обробки даних дозволяє подолати багато обмежень, властиві адресного доступу до пам'яті, за рахунок завдання деякого критерію відбору і проведення необхідних перетворень, тільки над тими даними, які задовольняють цьому критерію. Критерієм відбору може бути збіг з будь-яким елементом даних, достатньою для виділення шуканих даних з усіх наявних. Пошук даних може відбуватися за фрагментом, що має більшу або меншу кореляцію з заданим елементом даних.

Досліджено і в різному ступені застосовуються кілька підходів, що розрізняються повнотою реалізації моделі асоціативної обробки. Якщо реалізується тільки асоціативна вибірка даних з подальшим послідовним використанням знайдених даних, то говорять про

асоціативної пам'яті або пам'яті, що адресується за вмістом. При досить повної реалізації всіх властивостей асоціативної обробки використовується термін "асоціативний процесор".

Асоціативні системи відносяться до класу: один потік команд - безліч потоків даних (SIMD = Single Instruction Multiple Data). Ці системи включають велике число операційних пристроїв, здатних одночасно по командам керуючого пристрою вести обробку декількох потоків даних. В асоціативних обчислювальних системах інформація на обробку надходить від асоціативних запам'ятовуючих пристроїв (АЗП), що характеризуються тим, що інформація в них вибирається не по певній адресі, а по її змістом.

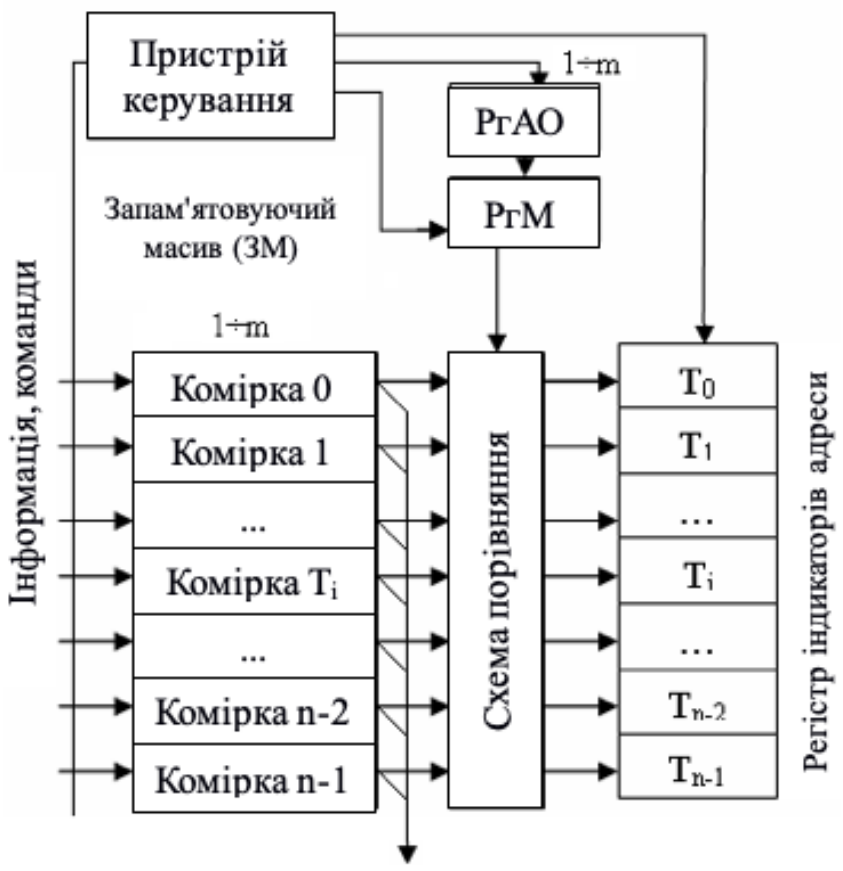

Рисунок 7.1 - Схема асоціативної системи

Конвеєрні процесори. Процесори сучасних комп'ютерів використовують особливу технологію - конвеєри, які дозволяють обробляти більше однієї команди одночасно.

Обробка команди може бути розділена на кілька основних етапів, назвемо їх мікрокомандами. Виділимо основні п'ять микрокоманд:

- вибірка команди;
- розшифровка команди;
- вибірка необхідних операндів;
- виконання команди;
- збереження результатів.

Всі етапи команди задіюються тільки один раз і завжди в одному і тому ж порядку: одна за одною. Це, зокрема, означає, що якщо перша мікрокоманда виконала свою роботу і передала результати другої, то для виконання поточної команди вона більше не знадобиться, і, отже, може приступити до виконання наступної команди. Виділимо кожну команду в окрему частину пристрою і розташуємо їх у порядку виконання. У перший момент часу виконується перша мікрокоманда. Вона завершує свою роботу і починає виконуватися друга мікрокоманда, в той час як перша готова для виконання наступної інструкції. Перша інструкція може вважатися виконаною, коли завершать роботу всі п'ять микрокоманд.

Така технологія обробки команд носить назву конвеєрної обробки. Кожна частина пристрою називається щаблем конвеєра, а загальне число ступенів - довжиною конвеєра.

У багатьох обчислювальних системах поряд з конвеєром команд використовуються і конвеєри даних.

Поєднання цих двох конвеєрів дозволяє досягти дуже високої продуктивності на певних класах завдань, особливо якщо використовується кілька різних конвеєрних процесорів, здатних працювати одночасно і незалежно один від одного.

Однією з найбільш високопродуктивних обчислювальних конвеєрних систем вважається CRAY. У цій системі конвеєрний принцип обробки використовується в максимальному ступені. Є і конвеєр команд, і конвеєр арифметичних і логічних операцій. В системі широко застосовується поєднана обробка інформації декількома пристроями. Максимальна пікова продуктивність процесора може становити 12 GFLOPS.

В даний час створені однокристальних векторно-конвеєрні процесори, основними компонентами яких є скалярний процесор і 8 ідентичних векторних пристроїв, сумарна продуктивність яких становить 64 GFLOPS. На їх основі побудована система SX-6 компанії NEC.

Матричні процесори. Найбільш поширеними з систем класу один потік команд - безліч потоків даних (SIMD) є матричні системи, які найкраще пристосовані для вирішення завдань, що характеризуються паралелізмом незалежних об'єктів або даних. Організація систем подібного типу, на перший погляд, досить проста. Вони мають загальне керуючий пристрій, що генерує потік команд і велике число процесорних елементів, що працюють паралельно і обробних кожна свій потік даних. Таким чином, продуктивність системи виявляється рівною сумі продуктивностей всіх процесорних елементів. Однак на практиці щоб забезпечити достатню ефективність системи при вирішенні широкого кола завдань, необхідно організувати

зв'язку між процесорними елементами з тим, щоб найбільш повно завантажити їх роботою. Саме характер зв'язків між процесорними елементами і визначає різні властивості системи.

Одним з перших матричних процесорів був SOLOMON (60-ті роки).

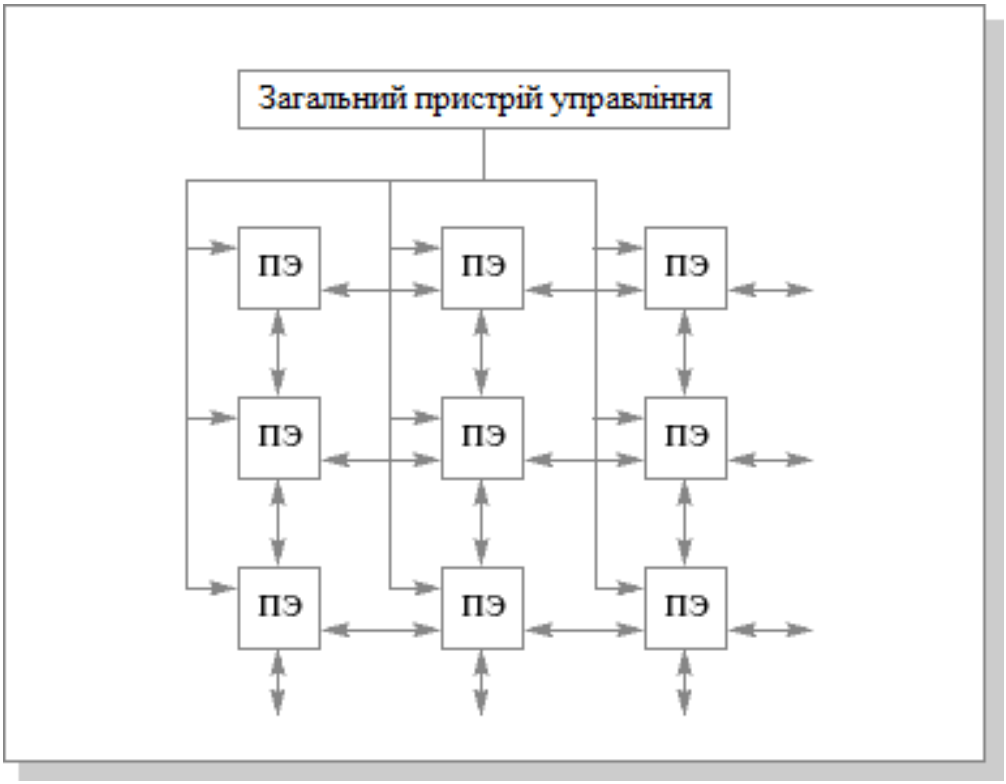

Рисунок 7.2 - Структура матричної обчислювальної системи SOLOMON

Система SOLOMON містить 1024 процесорних елемента, які з'єднані у вигляді матриці: 32x32. Кожен процесорний елемент матриці включає в себе процесор, що забезпечує виконання послідовних порозрядних арифметичних і логічних операцій, а також оперативне ЗУ ємністю 16 Кбайт. Довжина слова - мінлива від 1 до 128 розрядів. Розрядність слів встановлюється програмно. По каналах зв'язку від

пристрою управління передаються команди і загальні константи. В процесорному елементі використовується так звана многомодальним логіка, яка дозволяє кожному процесорному елементу виконувати або не виконувати загальну операцію в залежності від значень оброблюваних даних. У кожен момент всі активні процесорні елементи виконують одну і ту ж операцію над даними, що зберігаються у власній пам'яті і мають один і той же адресу.

Ідея многомодальним полягає в тому, що в кожному процесорному елементі є спеціальний регістр на 4 стани - регістр моди. Мода (модальність) заноситься в цей регістр від пристрою управління. При виконанні послідовності команд модальність передається в коді операції і порівнюється з вмістом регістра моди. Якщо є збіги, то операція виконується. В інших випадках процесорний елемент не виконує операцію, але може, в залежності від коду, пересилати свої операнди сусіднього процесорного елементу. Такий механізм дозволяє виділити рядок або стовпець процесорних елементів, що дуже корисно при операціях над матрицями. Взаємодіють процесорні елементи з периферійним обладнанням через зовнішній процесор.

Подальшим розвитком матричних процесорів стала система ILLIAC-4, розроблена фірмою BURROUGHS. Спочатку система повинна була включати в себе 256 процесорних елементів, розбитих на групи, кожна з яких має управлятися спеціальним процесором. Однак з різних причин була створена система, яка містить одну групу процесорних елементів і керуючий процесор. Якщо на початку передбачалося досягти швидкодії 1 млрд. Операцій в секунду, то реальна система працювала з швидкодією 200 млн. Операцій в

секунду. Ця система протягом ряду років вважалася однією з найбільш високопродуктивних в світі.

На початку 80-х років в СРСР була створена система ПС-2000, яка також є матричної. Основою цієї системи є мультипроцессор ПС-2000, що складається з вирішального поля і пристрої управління мультипроцесором. Вирішальне поле будується з одного, двох, чотирьох або восьми пристроїв обробки, в кожному з яких 8 процесорних елементів. Мультипроцессор з 64 процесорних елементів забезпечує швидкодію 200 млн. Операцій в секунду на коротких операціях.

8. СПОСОБИ ОРГАНІЗАЦІЇ ВИСОКОПРОДУКТИВНИХ ПРОЦЕСОРІВ. КЛІТИННІ ТА ДНК-ПРОЦЕСОРИ. КОМУНІКАЦІЙНІ ПРОЦЕСОРИ

8.1. Клітинні та ДНК-процесори

В даний час в пошуках реальної альтернативи напівпровідникових технологій створення нових обчислювальних систем вчені звертають дедалі більшу увагу на біотехнології, або біокомпьютінг, який представляє собою гібрид інформаційних, молекулярних технологій, а також біохімії. Біокомпьютінг дозволяє вирішувати складні обчислювальні задачі, використовуючи методи, прийняті в біохімії і молекулярної біології, організовуючи обчислення за допомогою живих тканин, клітин, вірусів і біомолекул. Найбільшого поширення набув підхід, де в якості основного елемента (процесора) використовуються молекули дезоксирибонуклеїнової кислоти. Центральне місце в цьому підході займає так званий ДНК-процесор. Крім ДНК, як Біопроцесор можуть використовуватися також білкові молекули і біологічні мембрани.

8.2. Особливості ДНК-процесорів

Так само, як і будь-який інший процесор, ДНК-процесор характеризується структурою і набором команд. У нашому випадку структура процесора - це структура молекули ДНК. А набір команд - це перелік біохімічних операцій з молекулами. Принцип пристрою комп'ютерної ДНК-пам'яті заснований на послідовному з'єднанні чотирьох нуклеотидів (основних цеглинок ДНК-ланцюги). Три нуклеотиду, з'єднуючись в будь-якій послідовності, утворюють елементарну комірку пам'яті - кодон, сукупність яких формує потім

ланцюг ДНК. Основні труднощі в розробці ДНК-комп'ютерів пов'язана з проведенням виборчих однокодонних реакцій (взаємодій) всередині ланцюга ДНК. Однак прогрес є вже і в цьому напрямку. Існує експериментальне обладнання, що дозволяє працювати з одним з 1020 кодонів або молекул ДНК. Іншою проблемою є самосборка ДНК, що призводить до втрати інформації. Її долають введенням в клітку спеціальних інгібіторів - речовин, що запобігають хімічну реакцію самосшівкі.

Використання молекул ДНК для організації обчислень - це не дуже нова ідея. Теоретичне обґрунтування такої можливості було зроблено ще в 50-х роках минулого століття (Р.П. Фейманом). В деталях ця теорія була опрацьована в 70-х роках Ч. Бенетті і в 80-х М. Конрадом.

Перший комп'ютер на базі ДНК був створений ще в 1994 р американським ученим Леонардом Адлеманом. Він змішав в пробірці молекулу ДНК, в якій були закодовані вихідні дані, і спеціальним чином підібрані ферменти. В результаті хімічної реакції структура ДНК змінилася таким чином, що в ній в закодованому вигляді був представлений відповідь завдання. Оскільки обчислення проводилися в ході хімічної реакції за участю ферментів, на них було витрачено дуже мало часу.

Слідом за роботою Адлемана пішли інші. Ллойд Сміт з Університету Вісконсін вирішив за допомогою ДНК завдання доставки чотирьох сортів піци за чотирма адресами, що мала на увазі 16 варіантів відповіді. Вчені з Прінстонського університету вирішили комбінаторних шахову задачу: за допомогою РНК знайшли правильний хід шахового коня на дошці з дев'яти клітин (всього їх 512 варіантів).

Річард Ліптон з Прінстона першим показав, як, використовуючи ДНК, кодувати двійкові числа і вирішувати проблему задоволення логічного виразу. Суть її в тому, що, маючи деякий логічне вираження, що включає n логічних змінних, потрібно знайти всі комбінації значень змінних, що роблять вираз істинним. Завдання можна вирішити тільки перебором 2n комбінацій. Всі ці комбінації легко закодувати за допомогою ДНК, а далі діяти за методикою Адлемана. Ліптон запропонував також спосіб злому шифру DES (американський криптографічний), трактують як своєрідне логічне вираз.

Першу модель біокомп'ютера, правда, у вигляді механізму з пластмаси, в 1999 р створив Іхуд Шапіро з Вейцмановского інституту природничих наук. Вона імітувала роботу "молекулярної машини" в живій клітині, яка щороку збирає білкові молекули за інформацією з ДНК, використовуючи РНК в якості посередника між ДНК і білком.

А в 2001 р Шапіро вдалося реалізувати обчислювальний пристрій на основі ДНК, яке може працювати майже без втручання людини. Система імітує машину Тьюринга - одну з фундаментальних концепцій обчислювальної техніки. Машина Тьюринга крок за кроком зчитує дані і в залежності від їх значень приймає рішення про подальші дії. Теоретично вона може вирішити будь-яку обчислювальну задачу. За своєю природою молекули ДНК працюють аналогічним чином, розпадаючись і рекомбініруясь відповідно до інформації, закодованої в ланцюжках хімічних сполук.

Розроблена в Вейцмановского інституті установка кодує вхідні дані і програми в складаються з двох ланцюгів молекулах ДНК і змішує їх з двома ферментами. Молекули ферменту виконували роль апаратного, а молекули ДНК - програмного забезпечення. Один

фермент розщеплює молекулу ДНК з вхідними даними на відрізки різної довжини в залежності від міститься в ній коду. А інший рекомбинирует ці відрізки відповідно до їх кодом і кодом молекули ДНК з програмою. Процес триває уздовж вхідного ланцюга, і, коли доходить до комунікаційні процесори

Комунікаційні процесори - це мікрочіпи, що представляють собою щось середнє між жорсткими спеціалізованими інтегральними мікросхемами і гнучкими процесорами загального призначення. Комунікаційні процесори програмуються, як і звичні для нас ПК-процесори, але побудовані з урахуванням мережевих завдань, оптимізовані для роботи в мережі і на їх основі виробники - як процесорів, так і обладнання - пишуть програмне забезпечення для специфічних додатків. Комунікаційний процесор має власну пам'ять і оснащений високошвидкісними зовнішніми каналами для з'єднання з іншими процесорними вузлами. Його присутність дозволяє в значній мірі звільнити обчислювальний процесор від навантаження, пов'язаної з передачею повідомлень між процесорними вузлами. Швидкісний комунікаційний процесор з RISC-ядром дозволяє управляти обміном даними з кількох незалежних каналах, підтримувати практично всі поширені протоколи обміну, гнучко і ефективно розподіляти і обробляти послідовні потоки даних з тимчасовим поділом каналів.

Сама ідея створення процесорів, призначених для оптимізації мережевої роботи і при цьому досить універсальних для програмної модифікації, народилася в зв'язку з необхідністю усунути відмінності в підходах до створення локальних мереж (різні підходи до архітектури мережі, класифікації потоків і т.д.). Безсумнівно, справжньою причиною буму мережевих процесорів стало прискорення темпів

розвитку ринку. Коли ринок рухається на "Internet-швидкості", постачальники обладнання вже не можуть витрачати по два роки на розробку спеціалізованих мікросхем для реалізації конкретних мережевих функцій. Ці два роки (і вкладені гроші) будуть витрачені даремно, якщо ринок за цей час піде в іншому напрямку. Вихід один - розробляти процесори, які постачальники обладнання можуть впровадити і випустити в новому продукті протягом декількох місяців. Бум мережевих процесорів, остаточно оформився в середині 1999 року, ні коротким, і в наступні роки промисловість розвивалася вкрай бурхливо.За прогнозами одних аналітиків, дуже скоро спеціальні мікросхеми будуть витіснені стандартними мережевими процесорами. Інші аналітики вважають, що у мережевих процесорів, без сумніву, є майбутнє, але вони зможуть переважати тільки на деяких сегментах ринку, де необхідні укорочені цикли розробки, швидкість і гнучкість.

Передбачається, що на цьому ринку не буде переважати якась одна компанія, як, наприклад, Intel на ринку ПК. Однак вважається, що Intel залишиться одним з ключових гравців, розділивши $ 2,9 млрд. З IBM, Motorola і дюжиною інших компаній.

Нова серія комунікаційних процесорів INTEL IXP4xx побудована на базі розподіленої архітектури XScale і включає потужні мультимедійні можливості, а також розвинені мережеві інтерфейси Ethernet. Поєднання високої продуктивності і низького енергоспоживання дозволяє ефективно застосовувати комунікаційні процесори INTEL не тільки в класичних мережевих додатках, але і для побудови Internet-орієнтованих вбудованих систем промислового призначення.

Ефективність роботи промислових підприємств сьогодні

напряму залежить від гнучкості застосовуються систем автоматизованого управління. Великі виробничі установки вимагають використання декількох децентралізованих систем управління, пов'язаних один з одним потужної інформаційної мережею, здатної працювати в складних промислових умовах. Найчастіше ці кошти промислової комунікації покликані забезпечити можливість гнучкого управління, програмування і контролю роботи розподілених систем управління з віддалених диспетчерських пунктів. Досягнення цих цілей можливо за допомогою комунікаційних процесорів, призначених для підключення персональних комп'ютерів до промислових інформаційних мереж. Додаткові можливості, що забезпечуються комунікаційними процесорами, повинні бути цікаві, перш за все, тим користувачам, яким необхідно здійснювати складні транзакції або налагодити прямий голосовий та відеозв'язок в рамках мережевої інфраструктури.

9. СПОСОБИ ОРГАНІЗАЦІЇ ВИСОКОПРОДУКТИВНИХ ПРОЦЕСОРІВ. ПРОЦЕСОРИ БАЗ ДАНИХ. ПОТОКОВІ ПРОЦЕСОРИ. НЕЙРОННІ ПРОЦЕСОРИ. ПРОЦЕСОРИ З БАГАТОЗНАЧНОЮ (НЕЧІТКОЇ) ЛОГІКОЮ

9.1. Процесори баз даних

Процесорами (машинами) баз даних в даний час прийнято називати програмно-апаратні комплекси, призначені для виконання всіх або деяких функцій систем управління базами даних (СКБД). Якщо свого часу системи управління базами даних призначалися в основному для зберігання текстової та числової інформації, то тепер вони розраховані на різні формати даних, в тому числі графічні, звукові і відео. Процесори баз даних виконують функції управління і поширення, забезпечують дистанційний доступ до інформації через шлюзи, а також реплікацію оновлених даних за допомогою різних механізмів тиражування. У великих інформаційних системах намітився перехід від тривіальної архітектури "клієнт - сервер" до трирівневої архітектурі з розподіленими базами даних (клієнт, сервер з СУБД і сервери власне з даними).

Сучасні процесори баз даних повинні забезпечувати природну зв'язок накопичуваної в базах даних інформації із засобами оперативної обробки транзакцій і Internet-додатками. Це повинні бути системи, які дають користувачам можливість в будь-який момент звернутися до корпоративних даних і проаналізувати їх, незалежно від того, де ці дані розміщуються.

Рішення таких завдань потребує суттєвого збільшення продуктивності систем управління базами даних. Однак традиційна

програмна реалізація численних функцій сучасних СУБД на ЕОМ загального призначення призводить до появи громіздких і непродуктивних систем з недостатньо високою надійністю. Необхідний пошук нових архітектурних і апаратних рішень. Інтенсивні дослідження, проведені в цій області в даний час, привели до розуміння необхідності використання в якості процесорів баз даних спеціалізованих паралельних обчислювальних систем. Створення такого роду систем зв'язується з реалізацією паралелізму при виконанні послідовності операцій і транзакцій, а також конвеєрної потокової обробки даних.

9.2. Потокові процесори

Потоковими називають процесори, в основі роботи яких лежить принцип обробки багатьох даних за допомогою однієї команди. Відповідно до класифікації Флінна, вони належать до SIMD (single instruction stream / multiple data stream) архітектурі. Технологія SIMD дозволяє виконувати один і той же дію, наприклад, віднімання і додавання, над декількома наборами чисел одночасно. SIMD-операції для чисел подвійної точності з плаваючою комою прискорюють роботу ресурсоємних додатків для створення контенту, тривимірного рендеринга, фінансових розрахунків і наукових завдань. Крім того, вдосконалені можливості 64-розрядної технології MMX (цілочисельних SIMD-команд); ця технологія поширена на 128-розрядні числа, що дозволяє прискорити обробку відео, мови, шифрування, обробку зображень і фотографій. Потоковий процесор підвищує загальну продуктивність, що особливо важливо при роботі з 3D-графікою.

Може бути окремий потоковий процесор (Single-streaming

processor - SSP) і багатопотокових процесор (Multi-Streaming Processor — MSP).

Яскравим представником потокових процесорів є сімейство процесорів Intel, починаючи з Pentium III, в основі роботи яких лежить технологія Streaming SIMD Extensions (SSE, потоковая обробка за принципом "одна команда - багато даних"). Ця технологія дозволяє виконувати такі складні і необхідні в століття Internet завдання як обробка мови, кодування і декодування відео-та аудіо, розробка тривимірної графіки і обробка зображень.

Представниками класу SIMD вважаються матриці процесорів: ILLIAC IV, ICL DAP, Goodyear Aerospace MPP, Connection Machine 1 і т.п. У таких системах єдиний керуючий пристрій контролює безліч процесорних елементів. Кожен процесорний елемент отримує від пристрою управління в кожен фіксований момент часу однакову команду і виконує її над своїми локальними даними.

Іншими представниками SIMD-класу є векторні процесори, в основі яких лежить векторна обробка даних. Векторна обробка збільшує продуктивність процесора за рахунок того, що обробка цілого набору даних (вектора) проводиться однією командою. Векторні комп'ютери маніпулюють масивами подібних даних подібно до того, як скалярні машини обробляють окремі елементи таких масивів. У цьому випадку кожен елемент вектора треба розглядати як окремий елемент потоку даних. При роботі у векторному режимі векторні процесори обробляють дані практично паралельно, що робить їх в кілька разів швидшими, ніж при роботі в скалярному режимі. Максимальна швидкість передачі даних у векторному форматі може становити 64 Гбайт / с, що на 2 порядки швидше, ніж в скалярних машинах.

Прикладами систем подібного типу є, наприклад, процесори фірм NEC і Hitachi.

9.3. Нейронні процесори

Одне з найбільш перспективних напрямків розробки принципово нових архітектур обчислювальних систем тісно пов'язане зі створенням комп'ютерів нового покоління на основі принципів обробки інформації, закладених в штучних нейронних мережах.

Перші практичні роботи по штучним нейромереж і нейрокомп'ютерів почалися ще в 40-50-і роки. Під нейронною мережею зазвичай розуміють сукупність елементарних перетворювачів інформації, званих "нейронами", які певним чином пов'язані один з одним каналами обміну інформацією - "синаптическими зв'язками".

Нейрон, по суті, являє собою елементарний процесор, який характеризується вхідним і вихідним станом, передавальної функцією (функція активації) і локальною пам'яттю.

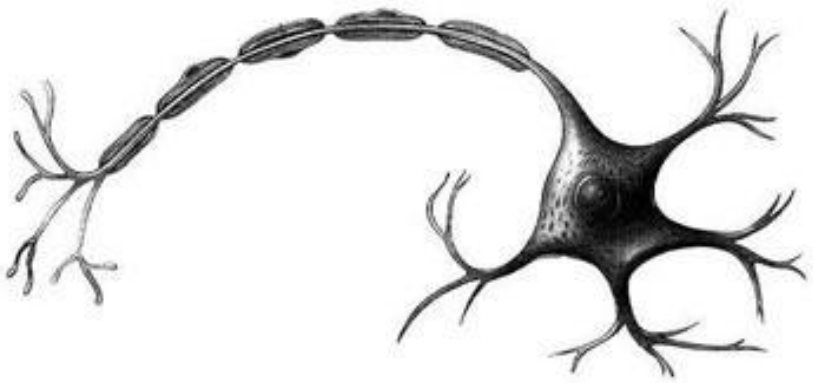

Рисунок 9.1 - Нейрон являє собою елементарний процесор

Стани нейронів змінюються в процесі функціонування і складають короткочасну пам'ять нейромережі. Кожен нейрон обчислює зважену суму прийшли до нього по синапсах сигналів і виробляє над

нею нелінійне перетворення. При пересиланні по синапсах сигнали множаться на деякий ваговий коефіцієнт. У розподілі вагових коефіцієнтів полягає інформація, що зберігається в асоціативної пам'яті нейронної мережі. Основним елементом проектування мережі є її навчання. При навчанні та перенавчанні нейронної мережі її вагові коефіцієнти змінюються. Однак вони залишаються постійними при функціонуванні нейромережі, формуючи довгострокову пам'ять.

Нейронна мережа може складатися з одного шару, з двох, з трьох і більшого числа шарів, однак, як правило, для вирішення практичних завдань більше трьох шарів в нейронній мережі не потрібно.

Число входів нейронної мережі визначає розмірність гіперпростору, в якому вхідні сигнали можуть бути представлені точками або гіперобластямі з близько розташованих точок. Кількість нейронів у шарі мережі визначає число гіперплощин в гіперпросторі. Обчислення зважених сум і виконання нелінійного перетворення дозволяють визначити, з якого боку від тієї чи іншої гиперплощини знаходиться точка вхідного сигналу в гіперпросторі.

Візьмемо класичну задачу розпізнавання образів: визначення належності точки одного з двох класів. Таке завдання природним чином вирішується за допомогою одного нейрона. Він дозволить розділити гіперпростір на дві непересічні і невкладені гіперобласті. Вхідні сигнали в задачах, що вирішуються за допомогою нейромереж, утворюють в гіперпросторі сильно вкладені або накладається, розділити які за допомогою одного нейрона неможливо. Це можна зробити, тільки провівши нелінійну гіперповерхню між областями. Її можна описати за допомогою полінома n-го порядку. Однак статична функція занадто повільно обчислюється і тому дуже незручна для

обчислювальної техніки.

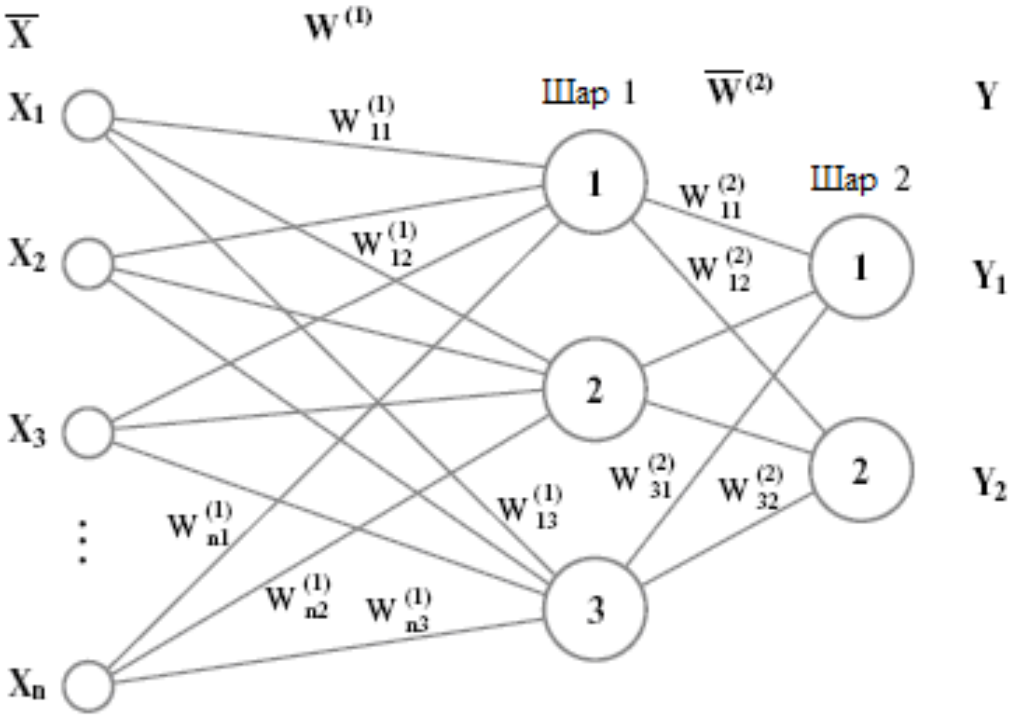

Рисунок 9.2 – Схема двошарової нейронної мережі

Альтернативним варіантом є апроксимація гіперповерхні лінійними гіперплощинами. Зрозуміло, що при цьому точність апроксимації залежить від числа використовуваних гіперплощин, яке, в свою чергу, залежить від числа нейронів в мережі. Звідси виникає потреба в апаратної реалізації якомога більшої кількості нейронів в мережі. Кількість нейронів в одному шарі мережі визначає її роздільну здатність. Одношарова нейронна мережа не може розділити лінійно залежні образи. Тому важливо вміти апаратно реалізовувати багатошарові нейронні мережі.

Штучні нейронні мережі відрізняються дивовижні властивості. Вони не вимагають детальної розробки програмного забезпечення і відкривають можливості вирішення завдань, для яких відсутні

теоретичні моделі або евристичні правила, що визначають алгоритм рішення. Такі мережі мають здатність адаптуватися до змін умов функціонування, в тому числі до виникнення заздалегідь непередбачених чинників. За своєю природою нейронні мережі є системами з дуже високим рівнем паралелізму.

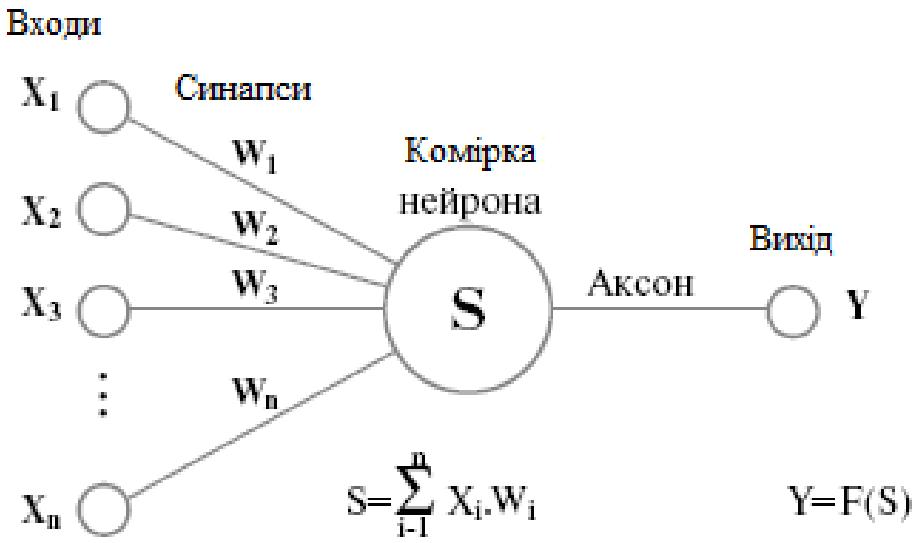

Рисунок 9.3 – Модель елемента нейронної мережі

У нейрокомп'ютер використовуються принципи обробки інформації, здійснювані в реальних нейронних мережах. Ці принципово нові обчислювальні засоби з нетрадиційною архітектурою дозволяють виконувати високопродуктивну обробку інформаційних масивів великої розмірності. На відміну від традиційних обчислювальних систем, нейромережеві обчислювачі, аналогічно нейронних мереж, дають можливість з більшою швидкістю обробляти інформаційні потоки дискретних і безперервних сигналів, містять прості обчислювальні елементи і з високим ступенем надійності дозволяють вирішувати інформаційні завдання обробки даних,

забезпечуючи при цьому режим самоперестройкой обчислювального середовища в залежності від отриманих рішень.

Взагалі кажучи, під терміном "нейрокомп'ютер" в даний час мається на увазі досить широкий клас обчислювачів. Це відбувається з тієї простої причини, що формально нейрокомп'ютером можна вважати будь-яку апаратну реалізацію нейросетевого алгоритму, від простої моделі біологічного нейрона до системи розпізнавання символів або рухомих цілей. Нейрокомп'ютери не є комп'ютерами в загальноприйнятому сенсі цього слова. В даний час технологія ще не досягла того рівня розвитку, при якому можна було б говорити про нейрокомп'ютер загального призначення (який був би одночасно штучним інтелектом). Системи з фіксованими значеннями вагових коефіцієнтів - взагалі самі вузькоспеціалізовані з нейросетевого сімейства. Ті, що навчаються мережі більш адаптовані до різноманітності вирішуваних завдань. Ті, що навчаються мережі більш гнучкі і здатні до вирішення різноманітних завдань. Таким чином, побудова нейрокомп'ютера - це кожен раз найширше поле для дослідницької діяльності в області апаратної реалізації практично всіх елементів нейронної мережі.

На початку 21 століття, на відміну від 40-50-х років минулого століття, існує об'єктивна практична потреба навчитися створювати нейрокомп'ютери, тобто необхідно апаратно реалізувати досить багато паралельно діючих нейронів, з мільйонами фіксованих або паралельно адаптивно модифікуються зв'язків-синапсів, з декількома повнозв'язну шарами нейронів.

У той же час фізичні можливості технології інтегральної електроніки не безмежні. Геометричні розміри транзисторів більше не

можна фізично зменшувати: при технологічно досяжних розмірах близько 1 мкм і менше проявляються фізичні явища, непомітні при великих розмірах активних елементів - починають сильно позначатися квантові розмірні ефекти. Транзистори перестають працювати як транзистори.

Для апаратної реалізації нейронної мережі необхідний новий носій інформації. Таким новим носієм інформації може бути світло, який дозволить різко, на кілька порядків, підвищити продуктивність обчислень.

Єдиною технологією апаратної реалізації нейронної мережі, здатної в майбутньому прийти на зміну оптики і оптоелектроніки, є нанотехнологія, здатна забезпечити не тільки фізично гранично можливу ступінь інтеграції субмолекулярних квантових елементів з фізично гранично можливою швидкодією, але і таку необхідну для апаратної реалізації нейронної мережі тривимірну архітектуру.

Тривалий час вважалося, що нейрокомп'ютери ефективні для вирішення так званих неформалізованих і погано формалізованих задач, пов'язаних з необхідністю включення в алгоритм вирішення задачі процесу навчання на реальному експериментальному матеріалі. В першу чергу до таких завдань ставилася задача апроксимації приватного виду функцій, що приймають дискретну безліч значень та інші.

9.4. Завдання розпізнавання образів

В даний час до цього класу задач додається клас задач, іноді не вимагає навчання на експериментальному матеріалі, але добре представимо в нейромережевому логічному базисі. До них відносяться завдання з яскраво вираженим природним паралелізмом обробки

сигналів, обробка зображень і ін. Підтвердженням точки зору, що в майбутньому нейрокомп'ютери будуть більш ефективними, ніж інші архітектури, може, зокрема, служити різке розширення в останні роки класу общематематических завдань, що вирішуються в нейромережевому логічному базисі. До них, крім перерахованих вище, можна віднести завдання вирішення лінійних і нелінійних алгебраїчних рівнянь і нерівностей великої розмірності; систем нелінійних диференціальних рівнянь; рівнянь в приватних похідних; задач оптимізації та інших завдань.

9.5. Процессори с багатозначною (нечіткою) логікою

Идея построения процессоров с нечеткой логикой (fuzzy logic) основывается на нечеткой математике. Математическая теория нечетких множеств, предложенная проф. Л.А. Заде, являясь предметом интенсивных исследований, открывает все более широкие возможности перед системными аналитиками. Основанные на этой теории различные компьютерные системы, в свою очередь, существенно расширяют область применения нечеткой логики.

Подходы нечеткой математики позволяют оперировать входными данными, непрерывно меняющимися во времени, и значениями, которые невозможно задать однозначно, такими, например, как результаты статистических опросов. В отличие от традиционной формальной логики, известной со времен Аристотеля и оперирующей точными и четкими понятиями типа истина и ложь, да и нет, ноль и единица, нечеткая логика имеет дело со значениями, лежащими в некотором (непрерывном или дискретном) диапазоне.

Функция принадлежности элементов к заданному множеству также представляет собой не жесткий порог "принадлежит – не

принадлежит", а плавную сигмоиду, проходящую все значения от нуля до единицы. Теория нечеткой логики позволяет выполнять над такими величинами весь спектр логических операций – объединение, пересечение, отрицание и др.

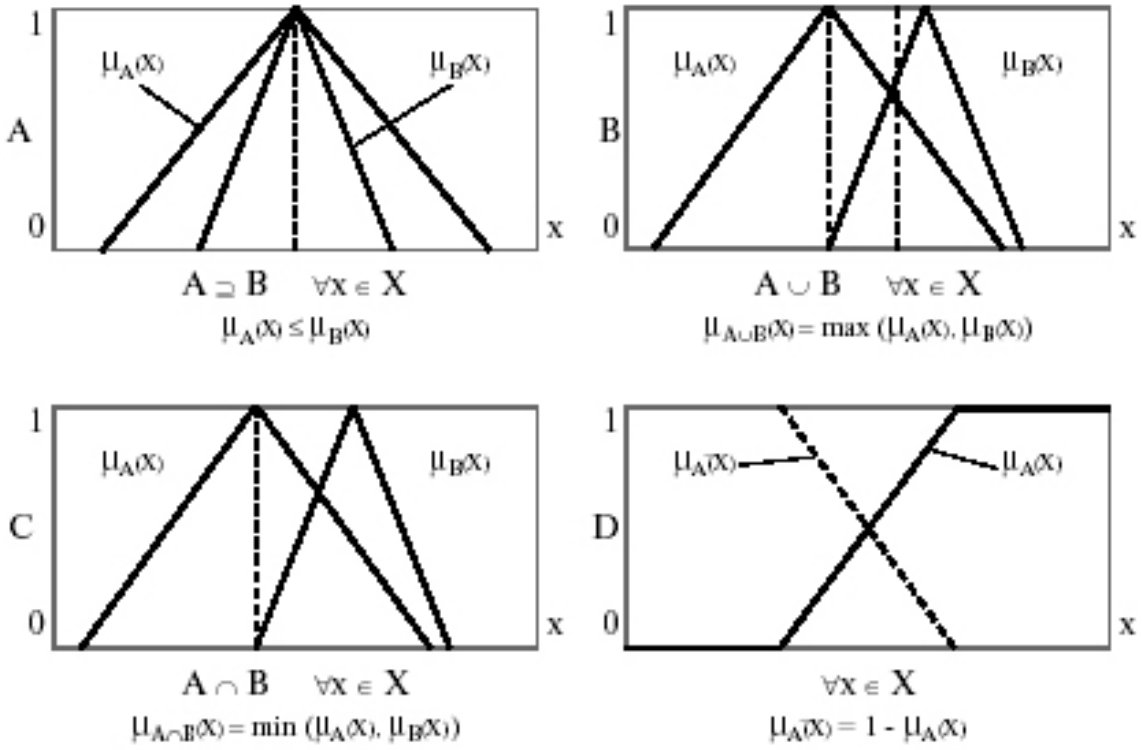

Рисунок 9.4 - Операції включення (A), об'єднання (B), перетину (C) і доповнення (D) нейронної мережі

Згідно знаменитої теореми FAT (Fuzzy Approximation Theorem), доведеною Коско, будь-яка математична система може бути апроксимована системою, заснованої на нечіткій логіці. Своє друге народження теорія нечіткої логіки пережила на початку вісімдесятих років, коли відразу кілька груп дослідників (в основному в США і Японії) зайнялися створенням електронних систем різного застосування, що використовують нечіткі керуючі алгоритми. Використовуючи переваги нечіткої логіки, які полягають в простоті

змістовного уявлення, можна спростити проблему, представити її в більш доступному вигляді і підвищити продуктивність системи.

Завдання за допомогою нечіткої логіки вирішуються за наступним принципом:

- чисельні дані (показання вимірювальних приладів, результати анкетування) фаззіруются (переводяться в нечіткий формат);

- обробляються за певними правилами;

- дефаззіруются і у вигляді звичної інформації подаються на вихід.

Виявилося можливим створення нечіткого процесора, що дозволяє виконувати різні нечіткі операції і наближені міркування (нечіткий висновок) відповідно до правил логічного висновку. У 1986 році в AT and T Bell Labs створювалися процесори з "прошитої" нечіткою логікою обробки інформації. На початку 90-х компанія Adaptive Logic з США випустила кристал, зроблений по аналогово-цифрової технології. Він дозволить скоротити терміни конструювання багатьох вбудованих систем управління реального часу, замінивши собою традиційні схеми нечітких мікроконтролерів. Апаратний процесор нечіткої логіки другого покоління приймає аналогові сигнали, переводить їх у нечіткий формат, потім, застосовуючи відповідні правила, перетворює результати у формат звичайної логіки і далі - в аналоговий сигнал. Все це здійснюється без зовнішніх запам'ятовуючих пристроїв, перетворювачів і якого б то не було програмного забезпечення нечіткої логіки. Цей мікропроцесор відносно простий в порівнянні з громіздкими програмними забезпеченнями. Але так як його основу становить комбінований цифровий/ аналоговий кристал, він функціонує на дуже високих

швидкостях (частота відліків вхідного сигналу - 10 кГц, а швидкість розрахунку - 500 тис. Правил/с), що в багатьох випадках призводить до кращих результатів в системах управління в порівнянні з більш складними, але повільними програмами.

У Європі та США ведуться інтенсивні роботи по інтеграції fuzzy-команд в асемблери промислових контролерів вбудованих пристроїв (чіпи Motorola 68HC11.12.21). Такі апаратні засоби дозволяють у декілька разів збільшити швидкість виконання додатків і компактність коду в порівнянні з реалізацією на звичайному ядрі. Крім того, розробляються різні варіанти fuzzy-сопроцессоров, які контактують з центральним процесором через загальну шину даних, концентрують свої зусилля на розмивання / ущільненні інформації та оптимізації використання правил (продукти Siemens Nixdorf).

Ідеї нечіткої логіки не є панацеєю і не зможуть зробити переворот в комп'ютерному світі. Нечітка логіка не вирішить тих завдань, які не вирішуються на основі логіки двійковій, але в багатьох випадках вона зручніше, продуктивніше і дешевше. Розроблені на її основі спеціалізовані апаратні рішення (fuzzy-обчислювачі) дозволять отримати реальні переваги у швидкодії. Якщо каскадировать fuzzy-обчислювачі, ми отримаємо один з варіантів нейропроцесора або нейронної мережі. У багатьох випадках ці поняття просто об'єднують, називаючи загальним терміном "neuro-fuzzy logic".

В даний час перспективою використовувати процесори, засновані на нечіткій логіці, всерйоз зацікавилися військові. Відомо, що NASA розглядає можливість застосування (якщо ще не застосовує) нечітких систем для управління процесами стикування космічних апаратів.

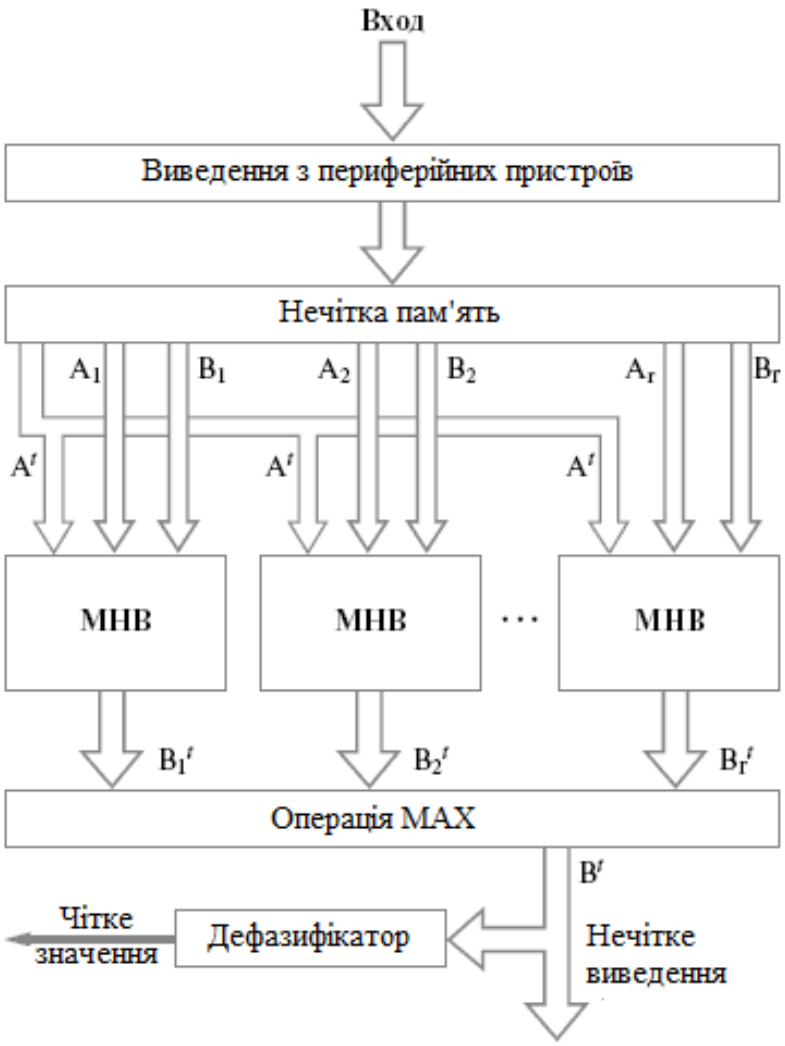

Рисунок 9.5 - Архітектура нечіткого комп'ютера
(МНВ-механізм нечіткого введення)

У Європі та США ведуться інтенсивні роботи по інтеграції fuzzy-команд в асемблери промислових контролерів вбудованих пристроїв (чіпи Motorola 68HC11.12.21). Такі апаратні засоби дозволяють у декілька разів збільшити швидкість виконання додатків і компактність коду в порівнянні з реалізацією на звичайному ядрі. Крім того, розробляються різні варіанти fuzzy-сопроцессоров, які контактують з центральним процесором через загальну шину даних,

концентрують свої зусилля на розмивання / ущільненні інформації та оптимізації використання правил (продукти Siemens Nixdorf).

Ідеї нечіткої логіки не є панацеєю і не зможуть зробити переворот в комп'ютерному світі. Нечітка логіка не вирішить тих завдань, які не вирішуються на основі логіки двійковій, але в багатьох випадках вона зручніше, продуктивніше і дешевше. Розроблені на її основі спеціалізовані апаратні рішення (fuzzy-обчислювачі) дозволять отримати реальні переваги у швидкодії. Якщо каскадировать fuzzy-обчислювачі, ми отримаємо один з варіантів нейропроцесора або нейронної мережі. У багатьох випадках ці поняття просто об'єднують, називаючи загальним терміном "neuro-fuzzy logic".

В даний час перспективою використовувати процесори, засновані на нечіткій логіці, всерйоз зацікавилися військові. Відомо, що NASA розглядає можливість застосування (якщо ще не застосовує) нечітких систем для управління процесами стикування космічних апаратів.

10. КОМУТАТОРИ ДЛЯ БАГАТОПРОЦЕСОРНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ. ПРОСТІ КОМУТАТОРИ

10.1. Склад комунікаційних середовищ обчислювальних систем

Комунікаційні середовища обчислювальних систем складаються з адаптерів обчислювальних модулів і комутаторів, що забезпечують з'єднання між ними. Використовуються як прості комутатори, так і складені, зкомпоновані з набору простих. Прості комутатори можуть з'єднувати лише мале число обчислювальних модулів в силу фізичних обмежень, проте забезпечують при цьому мінімальну затримку при встановленні з'єднання. Складені комутатори, зазвичай будуються з простих у вигляді багатокаскадних схем за допомогою ліній "точка-точка", долають обмеження на малу кількість з'єднань, проте збільшують і затримки

Типи простих комутаторів:

- з тимчасовим поділом;
- з просторовим розділенням.

Переваги простих комутаторів: простота управління і високу швидкодію.

Недоліки: мала кількість входів і виходів.

Приклади використання: прості комутатори з тимчасовим поділом використовуються в системах SMP Power Challenge від SGI, прості комутатори з просторовим розділенням (Gigaplane) використовуються в сімействі Sun Ultra Enterprise.

10.2. Прості комутатори з тимчасовим поділом

Прості комутатори з тимчасовим поділом називаються також шинами або шинними структурами. Всі пристрої підключаються до загальної інформаційної магістралі, використовуваної для передачі інформації між ними (рис. 10.1). Зазвичай шина є пасивним елементом, управління передачами здійснюється передають і приймають пристроями.

Процес передачі виглядає наступним чином.

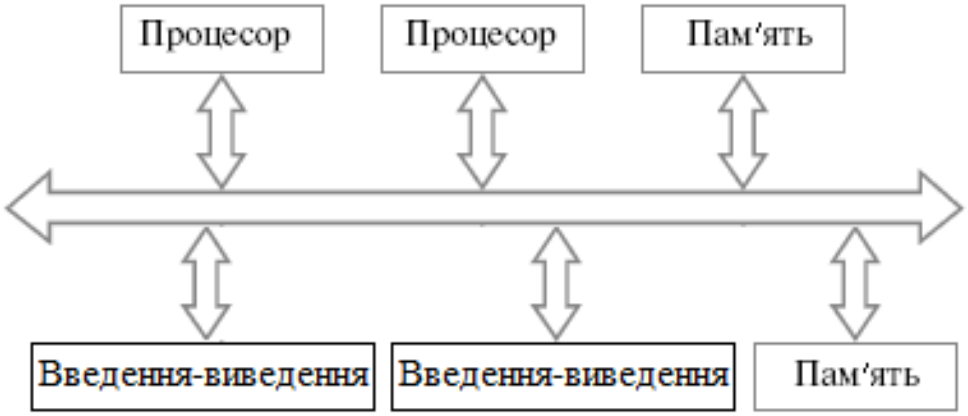

Рисунок 10.1 - Загальна схема шинної структури

Передавальний пристрій спочатку отримує доступ до шини, далі намагається встановити контакт з пристроєм-адресатом і визначити його здатність до прийому даних. Приймаючий пристрій розпізнає свою адресу на шині і відповідає на запит передавального. Далі передавальний пристрій повідомляє, що насамперед має зробити приймаючий пристрій в ході взаємодії. Після цього відбувається передача даних.

Так як шина є загальним ресурсом, за доступ до якого змагаються підключені до неї пристрої, необхідні методи управління

наданням доступу пристроїв до шини. Можливе використання центрального пристрою для управління доступом до шині, однак це зменшує масштабованість і гнучкість системи.

10.3. Вирішення конфліктів запиту

Для вирішення конфліктів, що виникають при одночасному запиті пристроїв на доступ до шини, використовуються різні прийоми, зокрема:

- призначення кожному пристрою унікального пріоритету (статичного чи динамічного);
- використання черги запитів FIFO;
- виділення фіксованих тимчасових інтервалів кожному пристрою.
- алгоритми арбітражу
- статичні пріоритети

Кожен пристрій в системі отримує унікальний пріоритет - при одночасному запиті декількох пристроїв на передачу доступ до шини надається пристрою з найвищим пріоритетом. На практиці часто використовується з'єднання пристроїв в ланцюг, при якому пріоритет пристрою визначається місцем його підключення до шини. Для контролю доступу до шині використовується окремий блок управління.

- динамічні пріоритети

Так само, як і в попередньому алгоритмі, пристрої отримують унікальні пріоритети, проте на відміну від нього ці пріоритети непостійні в часі. Пріоритети динамічно змінюються, надаючи пристроїв більш-менш рівні шанси отримання доступу до шини.

Найчастіше застосовуються такі способи зміни пріоритетів: надання найвищого пріоритету пристрою, найбільш довго не використовував шину, і циклічна зміна пріоритетів. Контроль доступу до шині здійснює пристрій, що одержав доступ до шини в попередньому циклі арбітражу.

- Фіксовані тимчасові інтервали

Всі пристрої по порядку отримують однакові часові інтервали для здійснення передачі. Якщо пристрій не має даних для передачі, то інтервал проте наступного пристрою не надається.

- черга FIFO

Створюється черга запитів "перший прийшов - перший пішов", проте зберігається проблема арбітражу між майже одночасними запитами, а також виникає необхідність підтримки черги запитів достатньої довжини. Перевагою даного алгоритму є можливість досягнення максимальної пропускної здатності шини.

10.4. Особливості реалізації шин

Усередині мікросхем шини використовуються для об'єднання функціональних блоків мікропроцесорів, мікросхем пам'яті, мікроконтролерів. Шини використовуються для об'єднання пристроїв на друкованих платах і друкованих плат в блоках. Останнім часом широко застосовуються шини наступних стандартів:

ISA - Industry Standard Architecture

EISA - Extended ISA

VESA - Video Electronics Standards Association

PCI - Peripheral Computer Interconnect

I2C - Inter Integrated Circuit

AGP - Accelerated Graphic Port

Шини використовуються також в мезонін технології, де на великій платі встановлюється один або кілька шинних роз'ємів для установки менших плат, так званих мезонінів.

Шини, які об'єднують пристрої, з яких складається обчислювальна система, є критичним ресурсом, відмова якого може привести до відмови всієї системи. Шини володіють також рядом принципових обмежень. Можливість масштабованості шинних структур обмежується часом, що витрачається на арбітраж, і кількістю пристроїв, підключених до шини. При цьому чим більше підключених пристроїв, тим більше часу витрачається на арбітраж. Час арбітражу обмежує і пропускну здатність шини. Крім того, в кожен момент часу шина використовується для передачі тільки одним пристроєм, що стає вузьким місцем при збільшенні кількості підключених пристроїв. пропускна здатність шини обмежується її шириною - кількістю провідників, використовуваних для передачі даних, - і тактовою частотою її роботи. Дані величини мають фізичні обмеження.

10.5. Прості комутатори з просторовим розділенням

Прості комутатори з просторовим розділенням дозволяють одночасно сполучати будь-який вхід з будь-яким одним виходом (ординарні) або декількома виходами (неординарні). Такі комутатори являють собою сукупність мультиплексорів, кількість яких відповідає кількості виходів комутатора, при цьому кожен вхід комутатора повинен бути заведений на всі мультиплексори. Структура цих комутаторів показана на рис. 10.2.

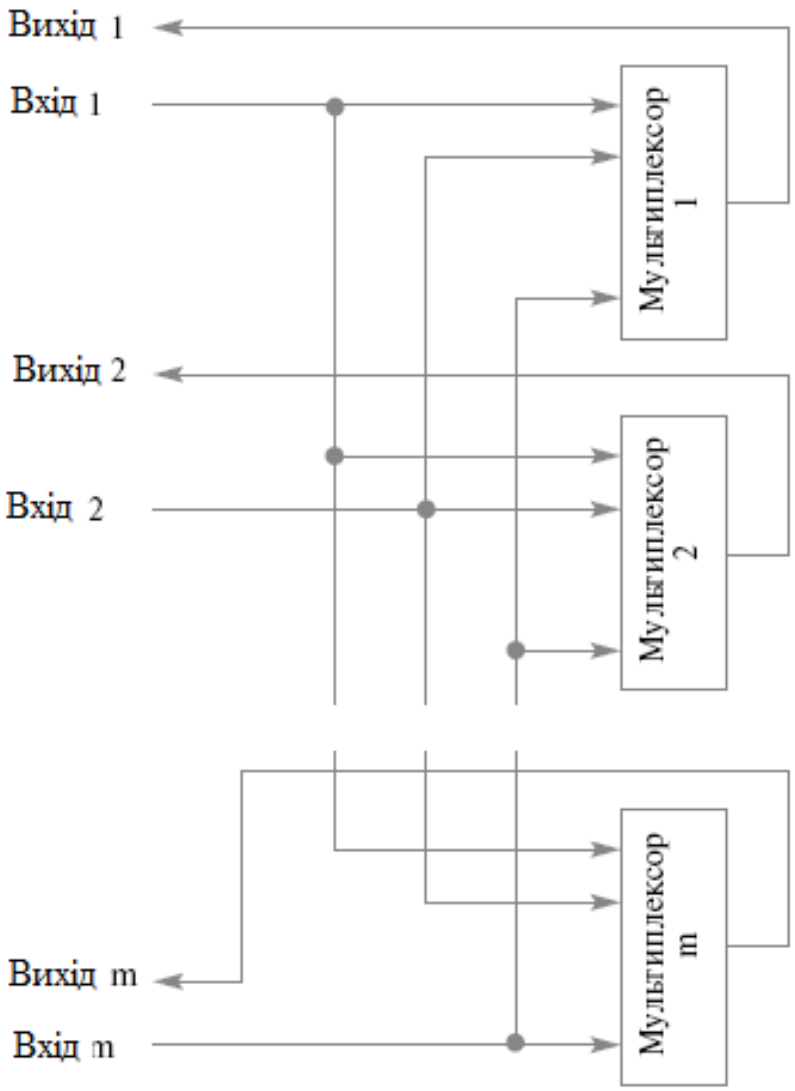

Рисунок **10.2 -** Простий комунікатор із просторовим поділом

Переваги:

можливість одночасного контакту з усіма пристроями;

• мінімальна затримка;

Недоліки:

• висока складність порядку n x m, де n – кількість входів; m – кількість виходів;

• складність забезпечення надійності.

11. КОМУТАТОРИ ДЛЯ БАГАТОПРОЦЕСОРНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ. СКЛАДЕНІ КОМУТАТОРИ. РОЗПОДІЛЕНІ СКЛАДЕНІ КОМУТАТОРИ

11.1. Складені комутатори

Прості комутатори мають обмеження на число входів і виходів, а також можуть вимагати великої кількості обладнання при збільшенні цього числа (в разі просторових комутаторів). Тому для побудови комутаторів з великою кількістю входів і виходів використовують сукупність простих комутаторів, об'єднаних за допомогою ліній "точка-точка".

Складальні комутатори мають затримку, пропорційну кількості простих комутаторів, через які проходить сигнал від входу до виходу, тобто числу каскадів. Однак обсяг обладнання складеного комутатора менше, ніж простого з тією ж кількістю входів і виходів.

Найчастіше складові комутатори будуються з прямокутних комутаторів 2 х 2 з двома входами і виходами. Вони мають два стани: пряме пропускання входів на відповідні виходи і перехресне пропускання. Комутатор 2 х 2 складається з власне блоку комутації даних і блоку управління. Блок управління в залежності від вступників на нього керуючих сигналів визначає, який тип з'єднання слід здійснити в блоці комутації - прямий або перехресний. При цьому якщо обидва входи хочуть з'єднатися з одним виходом, то комутатор дозволяє конфлікт і пов'язує з даними виходом тільки один вхід, а запит на з'єднання з боку другого блокується або відкидається.

11.2. Комутатор Клоза

Комутатор Клоза може бути побудований в якості альтернативи для прямокутного комутатора з (m x d) входами і (m x d) виходами. Він формується з трьох каскадів комутаторів: m комутаторів (d x d) у вхідному каскаді, m комутаторів (d x d) в вихідному і d проміжних комутаторів (m x m).

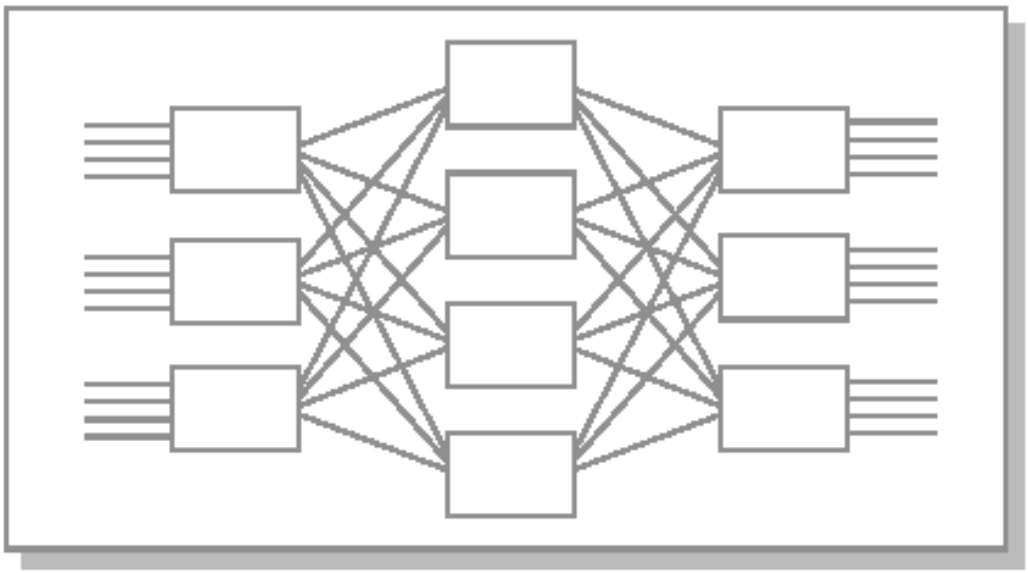

Рисунок 11.1 - Комутатор Клоза 3 x 4

З'єднання всередині комутатора влаштовані таким чином:

- i-й вихід i-ого комутатора вхідного каскаду з'єднаний з i-им входом j-ого проміжного комутатора;

- j-й вхід k-ого комутатора вихідного каскаду з'єднаний з k-им виходом j-ого проміжного комутатора.

Даний тип складових комутаторів дозволяє з'єднувати будь-який вхід з будь-яким виходом, однак при встановлених з'єднаннях додавання нового з'єднання може зажадати розриву і переустановленія всіх з'єднань.

11.3. Баньян-мережі

Комутатори цього типу будуються на базі прямокутних комутаторів таким чином, що існує тільки один шлях від кожного входу до кожного виходу.

Найбільш важливою різновидом баньян-мереж є дельта-мережу. Вона формується з прямокутних комутаторів (a x b) і являє собою n-каскадний комутатор з an входами і bn виходами. Складові комутатори з'єднані так, що для з'єднання будь-якого входу і виходу утворюється єдиний шлях однаковою для всіх пар входів і виходів довжини.

Основна властивість цих мереж:

– вони складаються з n=log2N та N/2 вузлів на рівень;

– вони мають властивість, що самоналаштовується – унікальна n-бітна адреса призначення може використовуватися для передачі осередку від будь-якого входу до будь-якого виходу, по одному біту на кожний рівень;

– їх регулярність та взаємозв'язна схема дуже приваблива для застосування в VLSI (VLSI - понад великий ступінь інтеграції).

11.4. Розподілені складові комутатори

У розподілених обчислювальних системах ресурси поділяються між завданнями, кожна з яких виконується на своєму підмножині процесорів. У зв'язку з цим виникає поняття близькості процесорів, яка є важливою для активно взаємодіючих процесорів. Зазвичай близькість процесорів виражається в різній каскадності з'єднань, різних відстанях між ними.

Один з варіантів створення складових комутаторів полягає в

об'єднанні прямокутних комутаторів (v + 1 x v + 1), v> 1 таким чином, що один вхід і один вихід кожного складового комутатора служать входом і виходом складеного комутатора. До кожного внутрішнього комутатора під'єднуються процесор і пам'ять, утворюючи обчислювальний модуль з v-каналами для з'єднання з іншими обчислювальними модулями. Вільні v-входів і v-виходів кожного обчислювального модуля з'єднуються лініями "точка-точка" з входами і виходами інших комутаторів, утворюючи граф міжмодульних зв'язків.

Найбільш ефективним графом міжмодульних зв'язків з точки зору організації обміну даними між обчислювальними модулями є повний граф. У цьому випадку між кожною парою обчислювальних модулів існує пряме з'єднання. При цьому можливі одночасні з'єднання між довільними обчислювальними модулями.

Однак зазвичай створити повний граф міжмодульних зв'язків неможливо з ряду причин. Обмін даними доводиться робити через ланцюжки транзитних модулів. Через це збільшуються затримки, і обмежується можливість встановлення одночасних з'єднань. Таким чином, ефективний граф міжмодульних зв'язків повинен мінімізувати час міжмодульних обмінів і максимально збільшити кількість одночасно активізованих з'єднань. Крім того, на вибір графа міжмодульних зв'язків впливає облік відмов і відновлень обчислювальних модулів і ліній зв'язку.

11.5. Граф міжмодульних зв'язків Convex Exemplar SPP1000

Як приклад реального графа міжмодульних зв'язків розглянемо структуру системи Convex Exemplar SPP1000. В основі кожного складового блоку системи лежить прямокутний комутатор (5 x 5), до 16 подібних блоків об'єднуються каналами "точка-точка" в кільце

(одновимірний тор), що складається з чотирьох незалежних подканалов.

Усередині кожного блоку чотири входи і виходи прямокутного комутатора (5 х 5) використовуються для взаємодії пристроїв всередині блоку (при цьому в кожному блоці розташовується по два процесори), п'яті вхід і вихід використовуються для об'єднання блоків в кільце. При цьому кожен з чотирьох кільцевих каналів розглядається як незалежний ресурс, і система зберігає працездатність до тих пір, поки існує хоча б один функціонуючий кільцевої канал.

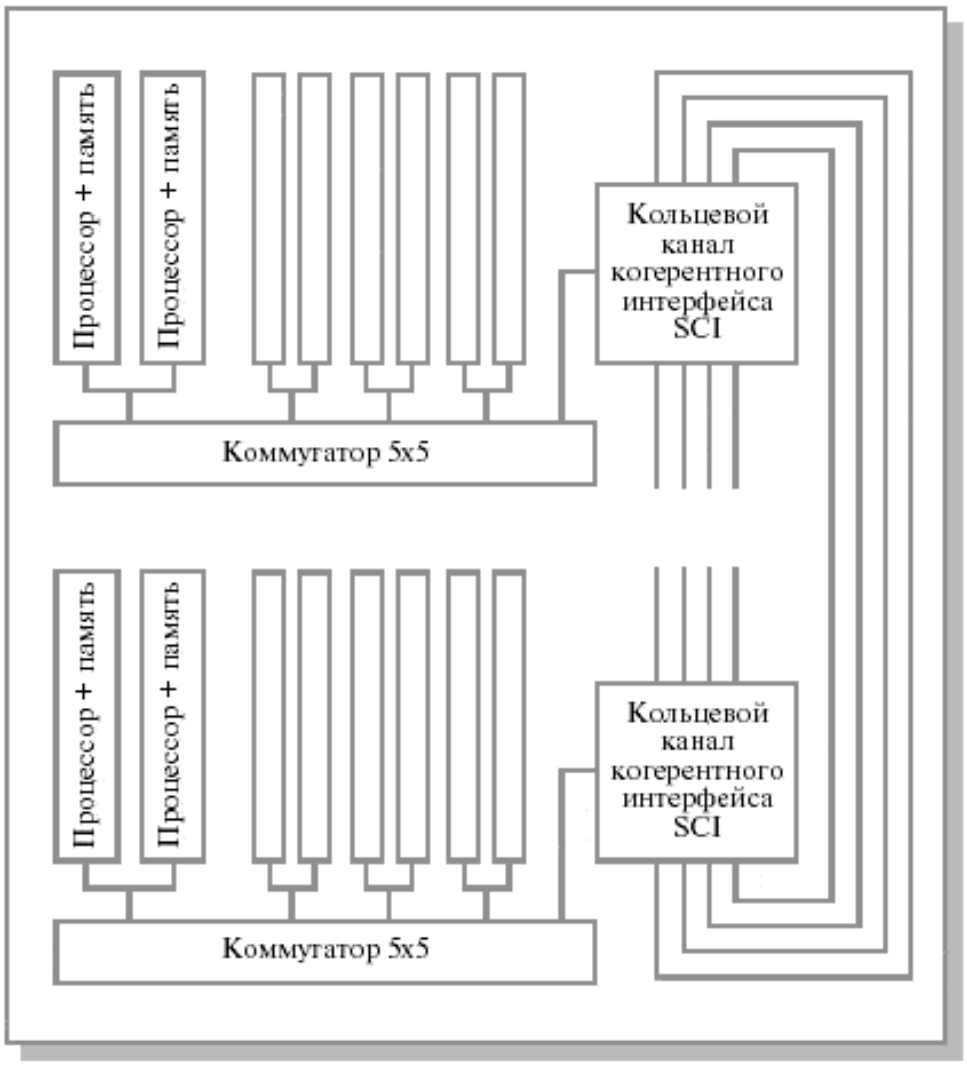

Рисунок 11.2. Граф міжмодульних зв'язків Convex Exemplar SPP1000

11.6. Граф міжмодульних зв'язків БКС-100

Система БКС-100 пропонує блоковий підхід до побудови архітектури паралельної обчислювальної системи. Структурний модуль системи складається з 16 обчислювальних вузлів, що утворюють матрицю 4х4 (рис. 10.3). Кутові вузли з'єднуються попарно по діагоналі, таким чином, максимальна довжина шляху між будь-якою парою елементів дорівнює трьом. У початковій ж матриці 4 х 4 ця довжина дорівнює шести. Кожен блок має 12 виходів, що дозволяє об'єднувати їх в більш складні структури.

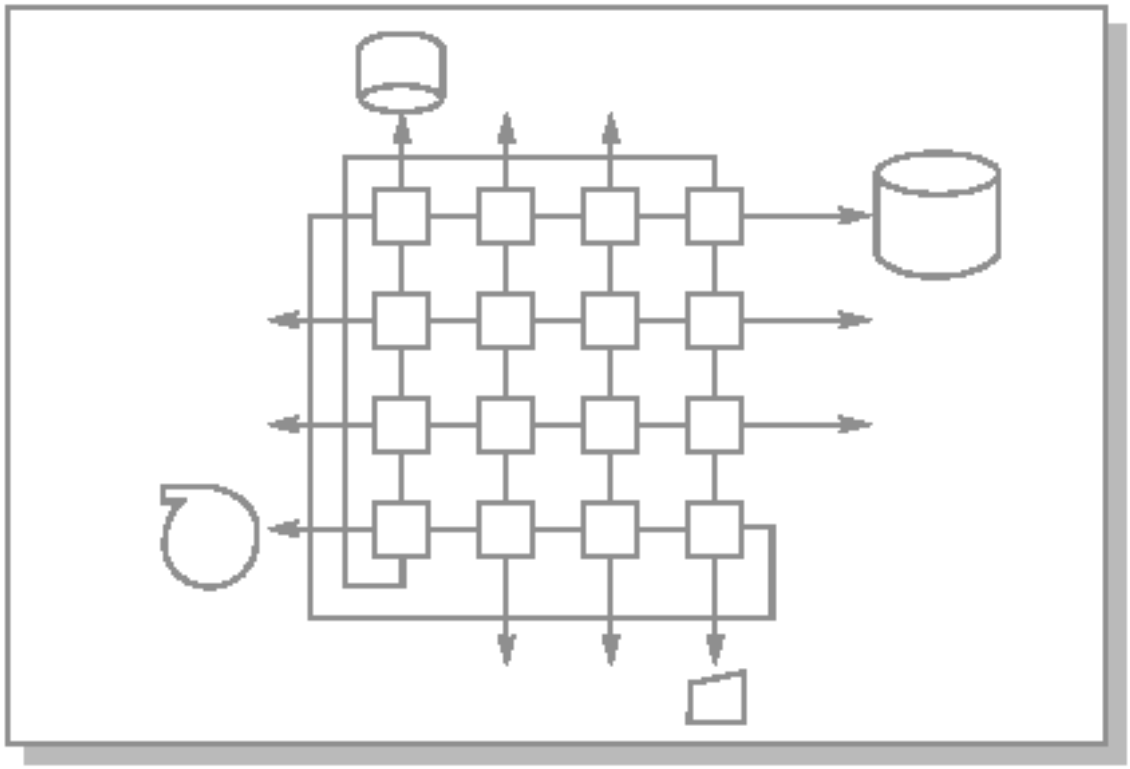

Рисунок 11.3. Структурний модуль БКС-100

Для БКС-100 базовий обчислювальний блок містить 32 вузла. Такий блок будується з двох структурних модулів відповідно до схеми, наведеної на рис. 10.4. В цьому випадку максимальна довжина шляху

між будь-якою парою обчислювальних вузлів дорівнює п'яти. При цьому залишається 16 вільних зв'язків, що дозволяє продовжити об'єднання. При об'єднанні двох базових блоків за схемою, наведеною на рис. 10.4 (64 обчислювальних вузла) максимальна довжина шляху складе 6, як і в гіперкубе, а кількість вільних зв'язків дорівнюватиме 16.

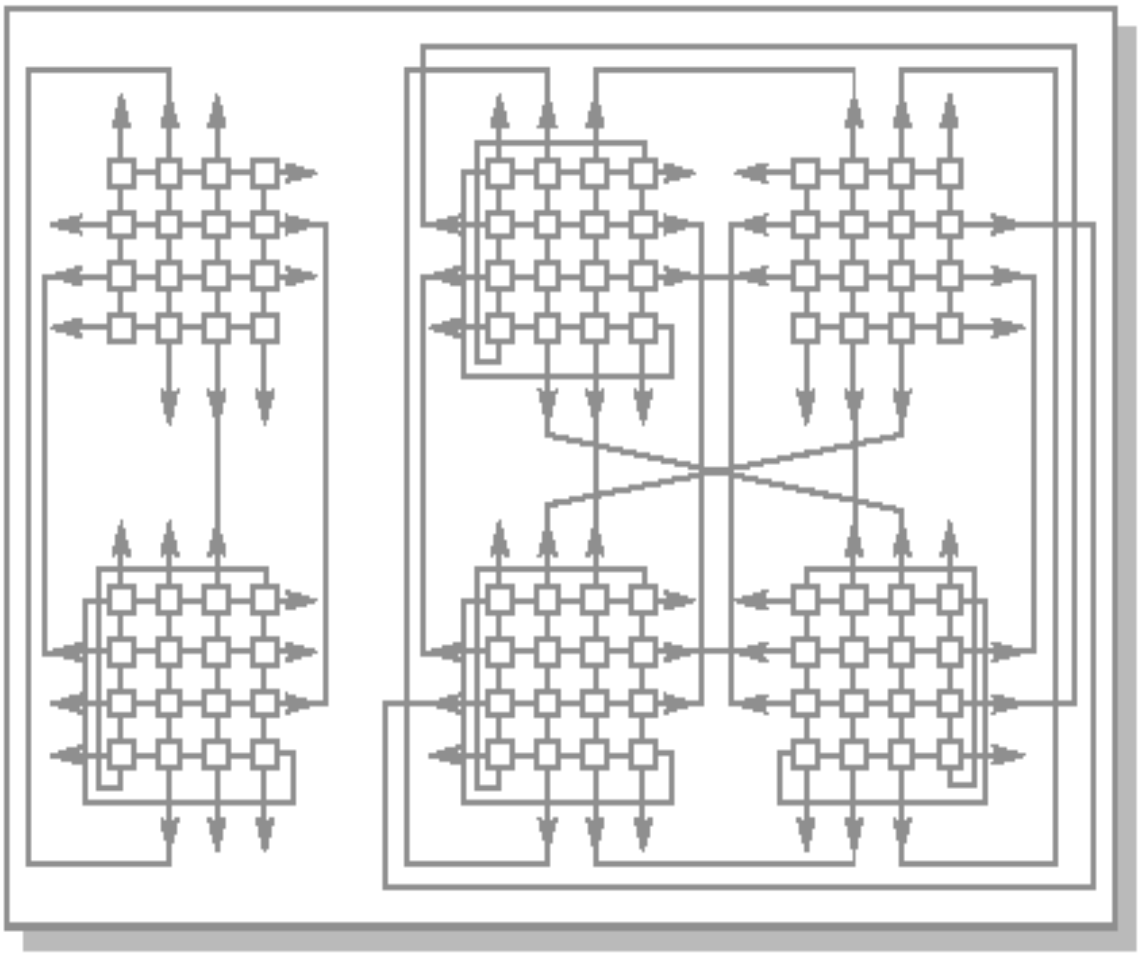

Рисунок 10.4. Варіанти об'єднання структурних модулів БКС-100

11.7. Граф міжмодульних зв'язків БКС -1000

Архітектура системи БКС-1000 аналогічна архітектурі БКС-100. Основою системи є масштабований масив процесорних вузлів. Кожен

вузол містить обчислювальний мікропроцесор Alpha 21164 з продуктивністю 2 GFLOPS при тактовій частоті 500 MHz і оперативну пам'ять об'ємом 128 MB, з можливістю розширення. Процесорні вузли взаємодіють через комунікаційні процесори TMS320C44 виробництва Texas Instruments, що мають по 4 зовнішніх каналу (лінка) із загальною пропускною спроможністю 80 Мбайт/с (20 Мбайт/с кожен). Також розробляється варіант системи з використанням комунікаційних процесорів SHARC (ADSP 21060) компанії Analog Devices, що мають по 6 каналів із загальною пропускною спроможністю до 240 Мбайт/с (40 Мбайт/с кожен).

Процесорні вузли пов'язані між собою за оригінальною схемою, подібною до топологією двомірного тора (для 4-лійкових вузлів). Аналогічно БКС-100, структурний модуль БКС-1000 складається з 16 обчислювальних модулів, що утворюють матрицю 4 х 4, в якій чотири кутових елементи поєднуються через транспьютерном лінки по діагоналі попарно. Решта 12 лінків призначаються для під'єднання зовнішніх пристроїв (4 линка кутових ВМ) і з'єднань з подібними ВМ.

Конструктивним утворенням БКС-1000 є базовий обчислюваль-ний блок, що містить 32 обчислювальних модуля. Максимальна довжина шляху між будь-якими з 32 обчислювальних модулів дорівнює п'яти, при цьому число вільних лінків після комплектації блоку становить 16, що дозволяє продовжити процедуру об'єднання. Можлива схема об'єднання чотирьох базових блоків в 128-процессорную систему приведена на рис. 11.5.

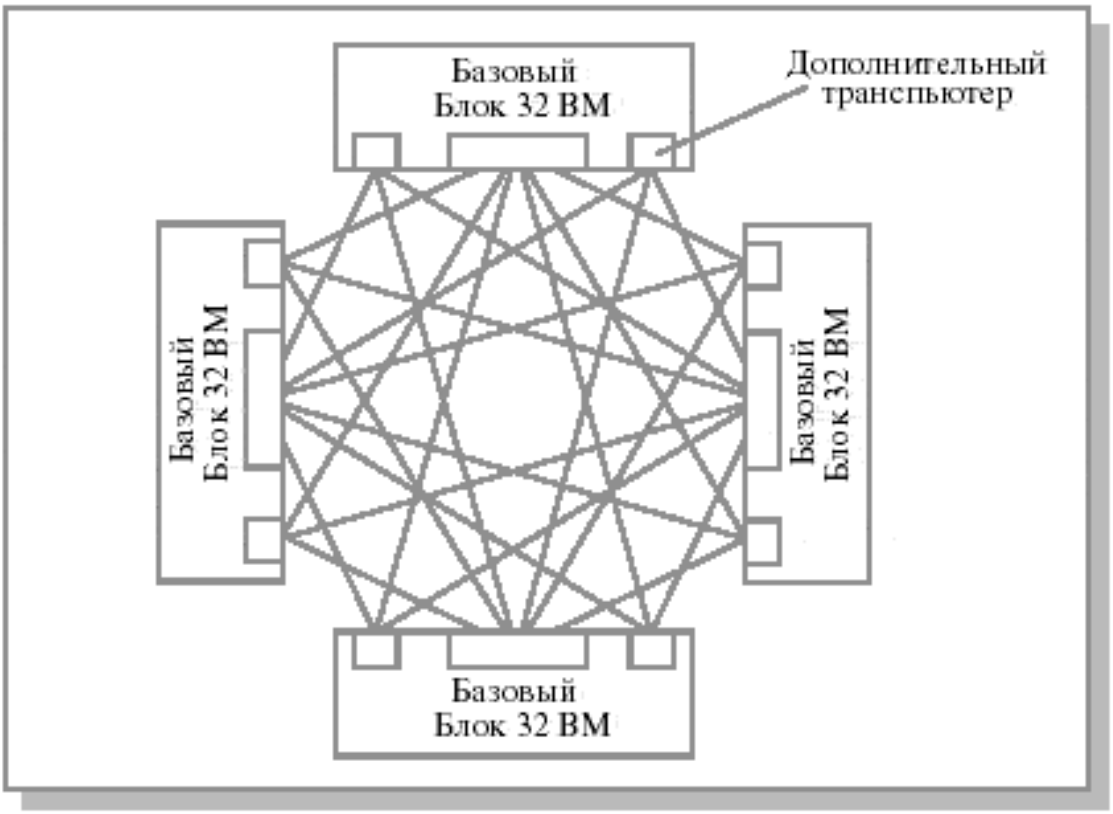

Рисунок 11.5. Структура 128-процесорної системи БКС-1000, 4 базових блоки

12. ВИМОГИ ДО КОМПОНЕНТІВ БАГАТОПРОЦЕСОРНИХ КОМП'ЮТЕРНИХ СИСТЕМ

Вимоги до компонентів бігітопроцесорних комп'ютерних систем потрібно розуміти в більш широкому сенсі, ніж просто набір вимог до технічних характеристик компонентів обчислювальної системи: процесору, дисковим масивам, пам'яті, комутаторів і тому подібним апаратних засобів. В якійсь мірі надійна робота компонентів систем мається на увазі апріорі: компоненти повинні працювати настільки довго, наскільки це необхідно і підтримувати при цьому задані значення параметрів системи (ясно, що такий стан є ідеалізацією). Досягається така надійність шляхом поліпшення технологій створення компонентів, складання систем і їх експлуатації і т.п. прийомами. Велике значення мають науково-технічні дослідження в області створення принципово нових підходів в розробці і створенні як відомих компонентів, так і принципово нових приладів. Але не менше, якщо не більше значення мають вимоги, що пред'являються до обчислювальної системи, яку планується побудувати для реалізації конкретних цілей, як єдиного цілого: для вирішення завдань певного кола (наукових, економічних і т.п.), як базової основи для обробки великих потоків даних (інформаційні системи), оптимальної реалізації моделі програмування і т.д. Звідси, в результаті проведеного аналізу, випливає вибір архітектури багатопроцесорної комп'ютерної системи.

Розробникам систем необхідно, перш за все, проаналізувати такі пов'язані між собою питання:

- відношення "ціна / продуктивність";
- надійність і відмовостійкість системи;
- масштабованість системи;

- сумісність програмного забезпечення.

12.1. Відношення "вартість/продуктивність"

Домогтися додаткового підвищення продуктивності в багатопроцесорних комп’ютерних систем ах складніше, ніж зробити масштабування всередині вузла. Основним бар'єром є труднощі організації ефективних міжвузлових зв'язків. Комунікації, які існують між вузлами, повинні бути стійкі до затримок програмно підтримуваної когерентності. Додатки з великою кількістю взаємодіючих процесів працюють краще на основі SMP-вузлів, в яких комунікаційні зв'язки більш швидкі. У кластерах, як і в MPP-системах, масштабування додатків більш ефективно при зменшенні обсягу комунікацій між процесами, що працюють в різних вузлах. Це зазвичай досягається шляхом розбиття даних.

Саме такий підхід використовується в найбільш відомому додатку на основі кластерів OPS (Oracle Parallel Server).

Поява будь-якого нового напряму в обчислювальній техніці визначається вимогами комп'ютерного ринку. Тому у розробників комп'ютерів немає єдиної мети. Велика універсальна обчислювальна машина (мейнфрейм) або суперкомп'ютер коштують дорого. Для досягнення поставлених цілей при проектуванні високопродуктивних конструкцій доводиться ігнорувати вартісні характеристики.

Суперкомп'ютери фірм Cray Inc., NEC і високопродуктивні мейнфрейми компанії IBM, суперкластери фірми SGI відносяться саме до цієї категорії комп'ютерів. Іншим протилежним прикладом може служити порівняно недорога конструкція, де продуктивність принесена в жертву для досягнення низької вартості. До цього напряму належать персональні комп'ютери IBM PC. Між цими двома крайніми

напрямами знаходяться конструкції, засновані на відношенні "вартість / продуктивність", в яких розробники знаходять баланс між вартістю і продуктивністю. Типовими прикладами такого роду комп'ютерів є мінікомп'ютери і робочі станції.

Для порівняння різних комп'ютерів між собою зазвичай використовуються стандартні методики вимірювання продуктивності. Ці методики дозволяють розробникам і користувачам задіяти отримані в результаті випробувань кількісні показники для оцінки тих чи інших технічних рішень, і, врешті-решт, саме продуктивність і вартість дають користувачеві раціональну основу для вирішення питання, який комп'ютер вибрати.

Наприклад, в якості критерію вимірювання продуктивності використовується тест LINPACK. Даний тест був обраний через його доступності майже для всіх розглянутих систем. Тест LINPACK був введений Джеком Донгарра (Jack Dongarra) в 1976 р Даний тест заснований на вирішенні щільної системи лінійних рівнянь. Як один з варіантів LINPACK використовується версія тесту, яка дозволяє користувачеві змінювати розмірність завдання і оптимізувати програмне забезпечення для досягнення оптимальної роботи для даної машини. Така продуктивність не відображає загальну продуктивність цієї системи. Однак вона відображає її продуктивність при вирішенні щільної системи лінійних рівнянь.

Для оцінки продуктивності обчислювальних систем використовуються також тести SPECfp_rate_base2000: SPEC, SPECfp і SPECrate, які є зареєстрованими торговими марками Standard Performance Evaluation Corporation. Для оцінки швидкості роботи пам'яті системи використовується тест STREAM Triad.

12.2. Масштабованість

Масштабованість є можливість нарощування числа і потужності процесорів, обсягів оперативної і зовнішньої пам'яті і інших ресурсів обчислювальної системи. Масштабованість повинна забезпечуватися архітектурою і конструкцією комп'ютера, а також відповідними засобами програмного забезпечення.

Так, наприклад, можливість масштабування кластера обмежена значенням відношення швидкості процесора до швидкості зв'язку, яке не повинно бути занадто великим (реально це відношення для великих систем не може бути більше 3-4, в іншому випадку не вдається навіть реалізувати режим єдиного образу операційної системи) . З іншого боку, останні 10 років історії розвитку процесорів і комунікаторів показують, що розрив в швидкості між ними все збільшується. Додавання кожного нового процесора в дійсно масштабованої системі повинно давати прогнозоване збільшення продуктивності і пропускної здатності при прийнятних витратах. Однією з основних завдань при побудові масштабованих систем є мінімізація вартості розширення комп'ютера і спрощення планування. В ідеалі додавання процесорів до системи повинно приводити до лінійного росту її продуктивності. Однак це не завжди так. Втрати продуктивності можуть виникати, наприклад, при недостатню пропускну спроможність шин через зростання трафіку між процесорами і основною пам'яттю, а також між пам'яттю і пристроями введення / виводу. Насправді реальне збільшення продуктивності важко оцінити заздалегідь, оскільки воно в значній мірі залежить від динаміки поведінки прикладних задач.

Можливість масштабування системи визначається не тільки архітектурою апаратних засобів, але залежить від властивостей

програмного забезпечення. Масштабованість програмного забезпечення впливає на кожну рівні, від простих механізмів передачі повідомлень до роботи з такими складними об'єктами як монітори транзакцій і все середовище прикладної системи. Зокрема, програмне забезпечення повинно мінімізувати трафік межпроцессорного обміну, який може перешкоджати лінійному зростанню продуктивності системи. Апаратні засоби (процесори, шини і пристрої введення / виводу) є тільки частиною масштабованої архітектури, на якій програмне забезпечення може забезпечити передбачене зростання продуктивності. Важливо розуміти, що, наприклад, простий перехід на більш потужний процесор може призвести до перевантаження інших компонентів системи. Це означає, що дійсно масштабована система повинна бути збалансована за всіма параметрами.

12.3. Сумісність і мобільність програмного забезпечення

Концепція програмної сумісності вперше в широких масштабах була застосована розробниками системи IBM / 360. Основне завдання при проектуванні всього ряду моделей цієї системи полягала в створенні такої архітектури, яка була б однаковою з точки зору користувача для всіх моделей системи, незалежно від ціни і продуктивності кожної з них. Великі переваги такого підходу, що дозволяє зберігати існуючий розділ програмного забезпечення при переході на нові (як правило, більш продуктивні) моделі, були швидко оцінені як виробниками комп'ютерів, так і користувачам, і починаючи з цього часу практично всі фірми-постачальники комп'ютерного устаткування взяли на озброєння ці принципи, поставляючи серії сумісних комп'ютерів. Слід зауважити, однак, що з часом навіть сама передова архітектура неминуче застаріває і виникає потреба внесення

радикальних змін і в архітектуру, і в способи організації обчислювальних систем.

В даний час одним з найбільш важливих факторів, що визначають сучасні тенденції в розвитку інформаційних технологій, є орієнтація компаній-постачальників комп'ютерного обладнання на ринок прикладних програмних засобів. Це пояснюється, перш за все, тим, що для кінцевого користувача, врешті-решт, важливо програмне забезпечення, що дозволяє вирішити його завдання, а не вибір тієї чи іншої апаратної платформи. Перехід від однорідних мереж програмно сумісних комп'ютерів до побудови неоднорідних мереж, що включають комп'ютери різних виробників, в корені змінив і точку зору на саму мережу: з порівняно простого засобу обміну інформацією вона перетворилася в засіб інтеграції окремих ресурсів - потужну розподілену обчислювальну систему, кожен елемент якої (сервер або робоча станція) найкраще відповідає вимогам конкретної прикладної задачі.

Цей перехід висунув ряд нових вимог. Перш за все, така обчислювальна середовище повинне дозволяти гнучко змінювати кількість і склад апаратних засобів і програмного забезпечення відповідно до нових вимог вирішуваних завдань. По-друге, вона повинна забезпечувати можливість запуску одних і тих же програмних систем на різних апаратних платформах, тобто забезпечувати мобільність програмного забезпечення. По-третє, це середовище має гарантувати можливість застосування одних і тих же людино-машинних інтерфейсів на всіх комп'ютерах, що входять в неоднорідну мережу. В умовах жорсткої конкуренції виробників апаратних платформ та програмного забезпечення сформувалася концепція

відкритих систем, що представляє собою сукупність стандартів на різні компоненти обчислювальної середовища, призначених для забезпечення мобільності програмних засобів в рамках неоднорідною, розподіленої обчислювальної системи.

Одним з варіантів моделей відкритої середовища є модель OSE (Open System Environment), запропонована комітетом IEEE POSIX. На основі цієї моделі національний інститут стандартів і технології США випустив документ "Application Portability Profile (APP). The US Government's Open System Environment Profile OSE / 1 Version 2.0", який визначає рекомендовані для федеральних установ США специфікації в області інформаційних технологій, що забезпечують мобільність системного і прикладного програмного забезпечення. Всі провідні виробники комп'ютерів і програмного забезпечення в США в даний час дотримуються вимог цього документа.

13. НАДІЙНІСТЬ І ВІДМОВОСТІЙКІСТЬ БАГАТОПРОЦЕСОРНИХ КОМП'ЮТЕРНИХ СИСТЕМ

Однією з основних проблем побудови обчислювальних систем залишається завдання забезпечення їх тривалого функціонування.

Найважливішою характеристикою обчислювальних систем є надійність, тобто робота системи без збоїв в певних умовах протягом певного часу. Підвищення надійності засноване на принципі запобігання несправностей шляхом зниження інтенсивності відмов і збоїв за рахунок застосування електронних схем і компонентів з високої і надвисокої ступенем інтеграції, зниження рівня перешкод, полегшених режимів роботи схем, забезпечення теплових режимів їх роботи, а також за рахунок вдосконалення методів збірки апаратури .

Поняття надійності включає не тільки апаратні засоби, а й програмне забезпечення, яке використовується, зокрема, для аналізу продуктивності систем і управління конфігураціями. Головною метою підвищення надійності систем є цілісність зберігаються в них даних. Одиницею вимірювання надійності є середній час напрацювання на відмову (MTBF - Mean Time Between Failure), інакше - середній час безвідмовної роботи.

Відмовостійкість - це здатність обчислювальної системи продовжувати дії, задані програмою, після виникнення несправностей. Введення відмовостійкості вимагає надлишкового апаратного і програмного забезпечення. Напрями, пов'язані із запобіганням несправностей і з отказоустойчивостью, - основні для забезпечення надійності. Концепції паралельності і відмовостійкості обчислювальних систем природним чином пов'язані між собою,

оскільки в обох випадках потрібні додаткові функціональні компоненти. Тому на паралельних обчислювальних системах досягається як найвища продуктивність, так і, в багатьох випадках, дуже висока надійність. Наявні ресурси надмірності в паралельних системах можуть гнучко використовуватись як для підвищення продуктивності, так і для підвищення надійності. Структура багатопроцесорних і багатомашинних систем пристосована до автоматичної реконфігурації і забезпечує можливість продовження роботи системи після виникнення несправностей.

В даний час ці два поняття - надійності та відмовостійкості - при описі комп'ютерних систем часто змішують. Багато в чому це пояснюється тим, що користувача (не обов'язково індивідуального) цікавить головне: обчислювальна система повинна працювати протягом необхідного часу і надавати певний набір послуг. Для досягнення безвідмовної роботи використовуються різні прийоми, частина з яких ми тут розглядаємо, не акцентуючи уваги на те, до якого з вищевказаних понять ці прийоми відносяться.

Для підвищення надійності інформаційно-обчислювальної системи ідеальною схемою є кластерні системи. Завдяки єдиному уявленню, окремі несправні вузли або компоненти кластера можуть бути без зупинки роботи і непомітно для користувача замінені, що забезпечує безперервність і безвідмовну роботу обчислювальної системи навіть в таких складних додатках як бази даних.

Основа надійності кластера - це деякий надлишок відмовостійких серверів (вузлів), в залежності від конфігурації кластера і його завдань.

Кластерна конфігурація вузлів, комунікаційного обладнання і

пам'яті може забезпечити віддзеркалення даних, резервування компонентів самоконтролю і попередження, а також спільне використання ресурсів для мінімізації втрат при відмові окремих компонентів.

Рішення, що забезпечує підвищену відмовостійкість сервера, повинне включати:

- компоненти з "гарячої" заміною;

- диски, вентилятори, зовнішні накопичувачі, пристрої PCI, джерела живлення;

- надлишкові джерела живлення і вентилятори;

- автоматичний перезапуск і відновлення системи;

- пам'ять з корекцією помилок;

- функції перевірки стану системи;

- превентивне виявлення і аналіз несправностей;

- засоби віддаленого адміністрування системи.

У багатьох випадках кластер, як типовий представник багато-процесорної комп’ютерної системи, представляється користувачеві та адміністратору як єдина система. Спостереження за системою включає збір, зберігання та вилучення таких показників як використання центрального процесора і пам'яті, температура системи і процесора, швидкість обертання вентиляторів; ці та інші параметри допомагають користувачам і адміністраторам розуміти загальний стан системи і ефективність її використання.

Єдине управління системами кластера дозволяє максимально збільшити період безвідмовної роботи, контроль і управління додатками, операційними системами та апаратними засобами. При цьому всі вузли кластера управляються з єдиного центру контролю.

Програми-утиліти забезпечують поліпшення захисту і можливості відновлення даних, а також згладжують наслідки збоїв в роботі обладнання для кінцевого користувача. Операційна система кластера служить для управління всіма функціями кластера.

Програмне забезпечення дає можливість організувати ефективну службу супроводу і моніторингу рішення, дозволяючи збирати дані на рівні вузла, використовуючи плату управління. Важливим напрямком є вдосконалення та розвиток бібліотеки MPI і розвиток системи налагодження паралельних програм, що працюють на багатопроцесорних комп'ютерних системах. До неї відносяться отладчики, профілировщики, що забезпечують контроль над проходженням завдань.

В операційній системі HP-UX11i, створеної компанією Hewlett-Packard і призначеної для обслуговування критично важливих завдань в Internet, для підвищення надійності передбачена можливість підключення додаткових процесорів без перезавантаження ОС. Застосування файлової системи Veritas дає можливість резервного копіювання в режимі online і дефрагментації дисків без виключення системи. Операційна система може відключати непрацездатні процесори і блоки пам'яті без виключення системи.

Системи зберігання повинні бути представлені RAID-системами високої готовності. Надлишкові з'єднання повинні забезпечувати доступність даних навіть у разі виходу з ладу вузлів, контролерів або кабелів. З'єднання з системами зберігання даних в кластері може бути реалізовано як з використанням інтерфейсів SCSI, так і на основі Fibre Channel технології.

Для синхронізації і спільної роботи серверів в якості кластера

необхідні надлишкові з'єднання між серверами, звані "системним сполученням" (private interconnect). Системне з'єднання використовується для передачі сигналів про стан серверів, а також застосовується паралельними базами даних для передачі даних.

Катастрофостійкі рішення створюються на основі того, що розноситься вузлів кластерної системи на сотні кілометрів і забезпечення механізмів глобальної синхронізації даних між такими вузлами.

Як приклад підвищення надійності кластерної системи наведемо рішення фірми Hewlett-Packard. У цих рішеннях, в залежності від потрібного рівня відмовостійкості, серверні вузли кластера розміщуються таким чином:

- централізовано (локальний кластер);
- по сусіднім будівлям (кампусні кластер);
- по декількох територій в межах міста (метро кластер);
- в різних містах, країнах або континентах (два пов'язаних кластеру - континентальний кластер).

На додаток до дубльованого центрального комутатора, все апаратні компоненти: системний контролер, джерела живлення, системи охолодження, годинник - повністю дубльовані. Система не має одиничної точки збою. Для порівняння - якщо такий простий елемент як системний годинник вийде з ладу в дорогому сервері HP Superdome або IBM p680, вся система припинить роботу.

У систему повинні бути заздалегідь встановлені або сконфігуровані запасні модулі, так що при відмові одного з модулів запасний модуль може замінити його практично негайно. Відмовив модуль може ремонтуватися автономно, в той час як система

продовжує працювати.

Принцип швидкого прояву несправності зазвичай реалізується за допомогою двох методів - самоконтролю і порівняння. Засоби самоконтролю припускають, що при виконанні деякої операції модуль робить і деяку додаткову роботу, що дозволяє підтвердити правильність отриманого стану. Прикладами цього методу є коди виявлення несправності при зберіганні даних і передачі повідомлень. Метод порівняння грунтується на виконанні однієї і тієї ж операції двома або більшою кількістю модулів і зіставленні результатів компаратором. У разі виявлення розбіжності результатів робота припиняється.

Методи самоконтролю були основою побудови відмовостійких систем протягом багатьох років. Вони вимагають реалізації додаткових схем і часу розробки і, ймовірно, будуть домінувати в пристроях пам'яті і пристроях зв'язку завдяки простоті і ясності логіки. Однак для складних пристроїв обробки даних економічні міркування, пов'язані із застосуванням стандартних масових компонентів, нав'язують використання методів порівняння. Оскільки компаратори порівняно прості, їх застосування дає деяке збільшення логічних схем при істотному скороченні часу розробки. Слід зазначити, що в більш ранніх відмовостійких конструкціях 30% логічних схем процесорів і 30% часу розробки йшло на реалізацію засобів самоконтролю. З цієї точки зору схеми порівняння додають лише універсальні схеми з простою логікою. В результаті скорочуються загальні витрати на розробку і логіку.

Ще одним засобом побудови відмовостійкої архітектури є принцип дублювання дуплексних модулів, який передбачає створення

деякої комбінації двох модулів ("супермодуля"), побудованих на принципах швидкого прояву несправності. Такий "супермодуль" продовжує працювати, навіть коли відмовляє один з субмодулів.

Дублювання дуплексних модулів вимагає більшого обсягу обладнання, але дозволяє робити вибір одного з режимів роботи: організацію або двох незалежних обчислень на принципах швидкого прояву несправності, що виконуються на двох парах модулів, або одного високонадійного обчислення, що виконується на всіх чотирьох модулях.

Необхідно пам'ятати, що сама по собі надмірність тільки знижує надійність в разі дублювання і троірования. Для істотного збільшення рівня готовності надмірна конструкція повинна забезпечувати можливість ремонту і заміни відмовили модулів.

14. КЛАСТЕРИ ТА МАСИВНО-ПАРАЛЕЛЬНІ СИСТЕМИ РІЗНИХ ВИРОБНИКІВ.

Розвиток мережевих технологій привело до появи недорогих, але ефективних комунікаційних рішень. Це і зумовило появу кластерних обчислювальних систем, які фактично є одним з напрямків розвитку комп'ютерів з масовим паралелізмом. Класичні суперкомп'ютери, що використовують спеціалізовані процесори таких виробників як Cray або NEC (векторно-паралельні або масивно-паралельні), недешеві, тому і вартість подібних систем незрівняна з вартістю систем, що знаходяться в масовому виробництві.

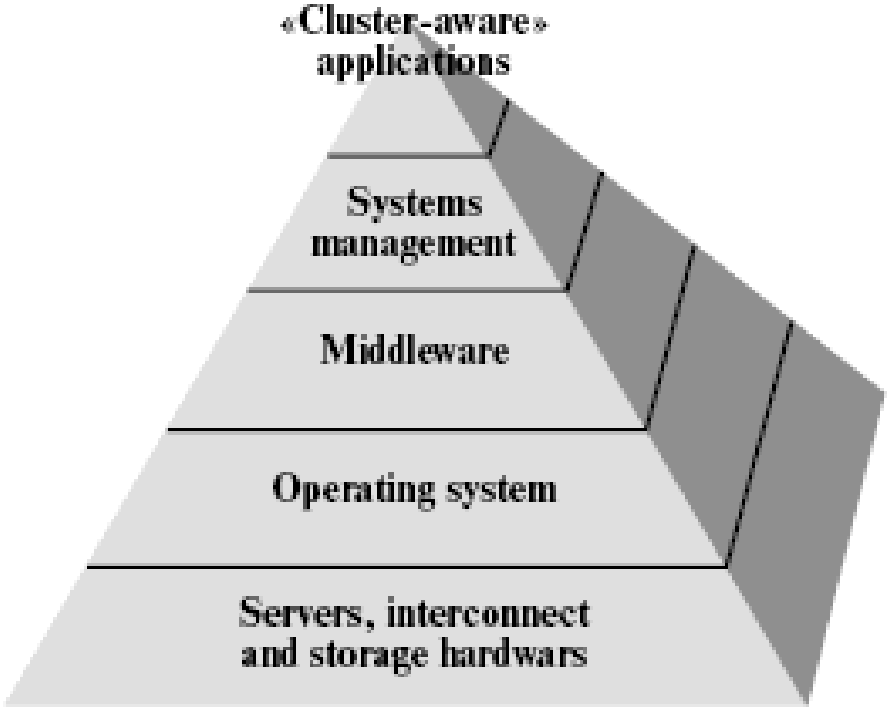

Рисунок 13.1. Піраміда рівнів кластерної системи

Обчислювальні системи (ВС), створювані з масово випускаються компонентів, стали альтернативою традиційним

суперкомп'ютерних систем. При виконанні багатьох прикладних задач такі ВС, навіть з невеликим або середнім (до 128-256) числом обчислювальних модулів, показують продуктивність, не поступається або навіть перевершує продуктивність традиційних суперкомп'ютерів як з розподіленою, так і з пам'яттю, що. При цьому такі ВС мають ряд переваг, серед яких: більш низька вартість, короткий цикл розробки і можливість оперативно використовувати найбільш ефективні обчислювальні і комунікаційні компоненти з наявних на ринку під час створення системи. Тому не дивно, що провідні розробники високопродуктивної техніки приступили до створення кластерних систем.

14.1. Приклади кластерних рішень IBM

На початку 2000 р фахівці IBM створили Linux-кластер з встановлених в стійці серверів IBM xSeries, об'єднавши їх з відповідними мережами, системами управління (апаратне і програмне забезпечення) і необхідними послугами. Після випуску в 2001 р кластера 1300 компаній IBM представила кластер 1350 на процесорах Intel Xeon.

Схема Linux-кластера або суперкластери нетривіальна. У ній є кілька логічних шарів, і рівень складності зростає при збільшенні розміру системи. Варто відзначити, що на великих системах просте відтворення у великій кількості малих кластерів майже ніколи не приводить до успіху.

Хоча число вузлів, необхідних для вирішення задачі, досить легко оцінюється для будь-якої програми, необхідне число вузлів в дійсності виявляється більше через необхідність мати сервісні вузли, обслуговуючі інфраструктуру кластера. Так, для кожних 32-64 вузлів, в

залежності від компоновки, потрібен центральний вузол. Якщо такий вузол використовується як обчислювальний, у нього повинна бути відповідна конфігурація.

У будь-якій системі повинен бути керівник вузол, який, зокрема, може бути і одним з головних вузлів. Для організації введення / виведення також необхідні окремі вузли, які працюють або з пристроями зберігання інформації, або з мережевими Рутер.

Основою Linux-кластера є щільно упаковані системи з процесорами Intel, встановлені в стійці. Найбільш часто використовуваним модулем є стандартна 19 "стійка. Усередині стійок встановлюються вузли, апаратура для ефективного з'єднання компонентів, така як комутатори або хаби, апаратура управління внутрішньої мережею системи, термінальні сервери і т.п.

Вузли можуть бути функціонально об'єднані в дві групи:

- Обчислювальні вузли, які виконують основні обчислювальні завдання, для яких спроектована система.

- Вузли інфраструктури, такі як вузли введення-виведення, вузли управління і вузли запам'ятовуючих пристроїв. Вузли інфраструктури забезпечують управління системами і заданими функціями, необхідними для об'єднання комп'ютерних вузлів в систем.

Упаковка обчислювальних вузлів, наскільки це можливо, повинна бути щільною і мати достатні можливості для ефективного з'єднання компонентів. Істотно включення сервісного процесора для функцій управління системами. Стандартним обчислювальним вузлом для кластера 1350 є IBM xSeries 335. Це дозволяє один або два процесори Intel Pentium 4 (Xeon) зі швидкою динамічною пам'яттю і диском розміщувати в стандартному корпусі розміром "1U". Символ

1U позначає 1,75 дюймів висоти в стандартному 19-дюймовому корпусі. X335 має вбудований сервісний процесор і два слота для з'єднання з іншими компонентами системи.

Головні вузли, вузли управління і вузли запам'ятовуючих пристроїв забезпечують особливі функції для управління кластером (забезпечення завантаження, управління пристроями, зовнішній введення / виведення і т.д). Сервер 2U IBM xSeries 345, заснований на процесорах Xeon, в кластері 1350 використовується як вузол управління і зберігання даних і може також застосовуватися як обчислювальний вузол. Комутатори використовуються для межпроцессорного з'єднання в паралельному програмуванні і для різних функцій управління.

Для паралельного програмування в якості межпроцессорного з'єднання зазвичай використовується комутатор Myrinet фірми Myricom. Пропускна здатність каналу становить приблизно 200 Мбайт / с в кожному напрямі з часом затримки 6-8 мкс.

Якщо паралельне програмне оточення вимагає менше міжпроцесорних з'єднань, то високошвидкісні з'єднання можна замінити на більш дешеві продукти типу Ethernet. Для замовника можуть бути обрані GigaNet, Quadratics, SCI або ServerNet. Надалі, після доопрацювання, можна вибрати InfiniBand.

Комутатори використовуються для побудови внутрішніх мереж для систем управління і інтерфейсу зовнішніх мереж. В якості альтернативних рішень замовнику пропонуються різні комутатори від Cisco.

Термінальні сервери забезпечують віддалений доступ до консолей ОС вузлів через послідовну мережу. Додаткові функціональні

можливості додаються за допомогою клавіатури, миші, монітора.

Зовнішні пристрої введення / виводу, такі як SCSI Raid, повинні бути в стійках з вузлами, комутаторами і т.д.

Основна відмінність полягає в збірці (інтеграції). У той час як кластер для замовників може бути зібраний в будь-якій установі або навіть на підлозі у замовника (невдала ідея), то апаратна частина кластера 1 350 виробляється (тобто інтегрується) і тестується на заводі IBM. Процес складання кластера в заводських умовах має ряд переваг.

Приклад конфігурації кластера 1350 наведено в таблиці 13.1.

Програмне забезпечення (ПО) для кластера 1350 істотно залежить від вимог замовника. Додаткові технічні умови можуть зажадати особливих програмних пакетів. Комерційний програмний пакет, наприклад, може включати в себе WebSphere, DB2, MySQL і т.д. HPC пакет може включати MPICH, PVM, Maui Scheduler, математичні бібліотеки, транслятори, профіліровщики і т.д.

Операційна система Linux інстальована на кожному вузлі кластера. Кластер 1350 запускається під Red Hat Linux. Надалі планується ставити ОС SuSE (4Q02).

Управління системами кластера для Linux (CSM) - це ліцензійний програмний продукт IBM. Він забезпечує функції управління системами, подібними за формою з програмами підтримки паралельних систем (Parallel System Support Programs - PSSP) для AIX-систем рівня підтримки на RS / 6000 SP. CSM - це стандартний програмний продукт для кластера 1350.

CSM для Linux включає технологію, витягнуту з PSSP, і зараз доступну на AIX для управління кластерами, зібраними з серверів xSeries і запускаються під Linux, серверами IBM pSeries, керованими

AIX, або комбінацією обох операційних систем.

Таблиця 13.1. Конфігурація кластера 1350.

Клас	Кількість вузлів кластера	Швидкість процесора, ГГц	Пам’ять системи, Гбайт	Внутрішня пам’ять, Гбайт	З’єднання кластера, Мбіт/с
Початковий	8	2,0	0,512	18	10/100 Ethernet
Середній	32	2,4	1	18	10/100 Ethernet
Професиіний	128	2,8	1	36	Gigabit Ethernet
Високо-продуктивний	64	2,8	1	36	Myrinet-2000

14.2. Приклади кластерних рішень HP

Злиття в 2002 р компаній Hewlett Packard і Compaq забезпечило HP міцне становище лідера з продажу Linux-систем, що відповідають індустріальним стандартам на базі архітектур IA-32 і IA-64. Ця технологія доповнюється потужною підтримкою розробок ядра Linux на базі сімейства Itanium, а також розробок з відкритим кодом в цілому.

Кластери HP будуються шляхом об'єднання комп'ютерів в групи, які називаються "кластерами підприємства". Кожен вузол кластера має принаймні один процесор, оперативну пам'ять і образ операційної системи. Для зв'язку між вузлами використовуються спеціальні протоколи зв'язку і системні процеси.

Підтримка ОС Linux з боку HP охоплює все сімейство серверів

HP, заснованих на архітектурі Intel (IA-32 і IA-64), включаючи всі сервери промислового стандарту HP ProLiant, сверхплотную (blade) архітектуру, робочі станції HP, настільні комп'ютери Evo, окремі портативні комп'ютери, сервери ProLiant для застосування в якості міжмережевих екранів і навіть портативні пристрої iPAQ. HP також продовжує підтримувати технологію ОС Linux для архітектури AlphaServer, розроблену компанією Compaq. ОС Linux працює на Alpha системах, починаючи з 1994 р Це був перший приклад 64-розрядної системи з підтримкою Linux. Саме з нього почалися сучасні розробки ОС Linux на базі сімейства Itanium. HP підтримує на своїх серверах дистрибутиви Red Hat і SuSE, плануючи здійснювати підтримку дистрибутивів операційної системи UnitedLinux після її випуску. HP пропонує замовникам можливість попередньо встановити будь-яку ОС Linux на обрані сервери ProLiant і робочі станції Evo.

HP підтримала лабораторію Sandia в її планах по розгортанню кластерної системи Cplant на базі ОС Linux з найвищою на сьогодні продуктивністю. Зараз HP спільно з Pacific Northwest National Laboratories працює над створенням обчислювальної системи, заснованої на 1400 процесорах Itanium 2 з оптоволоконними міжз'єднаннями на основі рішень Quadrics.

Програмне забезпечення HP може підтримувати більшість сучасних засобів розробки і настройки продуктивності для кластерних рішень на базі системи Linux. При виборі цих засобів діють обмеження, пов'язані з типами процесорів і міжвузлових з'єднань. У число програмних продуктів входять:

- Транслятор Intel C ++ Compiler для Linux;
- Транслятор Intel Fortran Compiler для Linux;

- Бібліотека Intel Math Kernel Library;

- Intel Vtune Performance Analyzer - засіб оптимізації програмного коду.

14.3. Приклади кластерних рішень SGI

Сьомого січня 2003 року компанія SGI представила нове сімейство 64-розрядних Linux-серверів і суперкластери, названих SGI Altix 3000 (сервери SGI Altix 3300 і суперкластери SGI Altix 3700). Система SGI Altix 3000 використовує процесори Intel Itanium 2 і заснована на архітектурі глобальної пам'яті, що SGI NUMAflex, яка є реалізацією архітектури неоднорідного доступу до пам'яті (NUMA). NUMAflex з'явилася в 1996 р і з тих пір використовувалася в відомої серії серверів і суперкомп'ютерів SGI Origin, заснованих на процесорах MIPS і 64-розрядної операційної системи IRIX. Дизайн NUMAflex дозволяє поміщати процесор, пам'ять, систему введення / виведення, з'єднувальні дроти, графічну підсистему в модульні компоненти, інакше звані блоками або цеглою. Ці цеглинки можуть комбінуватися і конфигурироваться з великою гнучкістю, щоб задовольняти потреби клієнта в ресурсах і робочого навантаження. Використовуючи цей дизайн третього покоління, компанія SGI змогла створити систему SGI Altix 3000 на основі традиційних блоків введення/ виведення (IX- і PX-блоки), зберігання даних (D-блоки) і сполучних компонентів (маршрутизуючі блоки/R-блоки). Основною відмінністю цієї нової системи є процесорний блок (C-блок), який містить процесори Itanium 2.

Ключовою особливістю системи є використання каскадіруемих комутаторів в маршрутизирующих елементах. Каскадуючі комутатори забезпечують системі малий час затримки, або уповільнення доступу до пам'яті, незважаючи на модульну конструкцію. Це критично для

машин, що використовують архітектуру неоднорідного доступу до пам'яті (NUMA). Затримки завжди були проблемою в архітектурі NUMA, так як пам'ять розподіляється між вузлами, а не зосереджена в одному місці. Каскадіруемие комутатори використовують каталогізіруемую схему пам'яті для відстеження даних, що знаходяться в різних кешах. В результаті менші обсяги даних пересилаються між частинами пам'яті, що виливається в зниження затримок в порівнянні з традиційними системами, заснованими на шинах.

Системне ПО для SGI Altix 3000 складається з стандартного дистрибутива Linux для процесорів Itanium і SGI ProPack - продукту, який додає особливі властивості Linux. SGI ProPack включає ядро 2.4, HPC-бібліотеки, налаштовані для використання особливостей архітектури SGI, утиліти для роботи з NUMA і драйвери. Також SGI ProPack включає додаткові інструменти і бібліотеки для поліпшення роботи великих NUMA-систем, особливо при одночасному виконанні декількох ресурсномістких додатків. Це дозволяє ефективно використовувати системні ресурси і доставляти результати в розумний час: характеристики, критичні для середовищ високопродуктивних обчислень.

Утиліти роботи з NUMA, бібліотеки HPC і додаткове ПО, встановлені на стандартний дистрибутив Linux, створюють програмне оточення для високопродуктивних обчислень, ефективне при великих обчислювальних навантаженнях і навантаженнях, пов'язаних з передачею даних. SGI ProPack створює проміжний шар ПО, яке дозволяє на основі Linux створювати блоки для побудови великих середовищ високопродуктивних обчислень.

Високопродуктивні програми вимагають балансу між

процесором і підсистемою пам'яті для підтримки постійного рівня потоку даних. Кластери SGI Altix 3000 були протестовані за допомогою тестів STREAM Triad, які вимірюють швидкість роботи пам'яті. 64-процесорна система досягла рівня продуктивності пам'яті в 125 Гбайт / с на єдиному образі операційної системи: перевага в 460% над 64-процесорної системою HP Superdome, яка показала продуктивність 27 Гбайт / с. У порівнянні з 32-процесорним сервером IBM eServer p690, система SGI Altix показує подвоєну продуктивність, а при вдвічі меншій вартості - поліпшення показника ціна/ продуктивність на 640%. Результати також показують, що Linux може добре масштабироваться за рамками обмеження в 8 процесорів.

Сімейство SGI Altix 3000 демонструє пропускну здатність системи введення/ виводу більш ніж 2 Гбайт / с при використанні єдиного способу Linux - кращий результат для Linux-систем. При постійному збільшенні обсягів оброблюваних даних можливості переміщення інформації з диска в пам'ять грають все більш важливу роль в загальній продуктивності системи. Це досягнення дозволяє додаткам Linux вирішити проблеми оперування великими обсягами даних в середовищах високопродуктивних обчислень.

15. КЛАСТЕРИ ТА МАСИВНО-ПАРАЛЕЛЬНІ СИСТЕМИ РІЗНИХ ВИРОБНИКІВ

15.1. SMP Power Challenge фірми Silicon Graphics

Компанія Silicon Graphics (SGI) була створена в 1981 р Основним напрямком її роботи протягом багатьох років було створення високопродуктивних графічних робочих станцій. В даний час інтереси SGI поширюються на ринок високопродуктивних обчислень як для технічних, так і для комерційних додатків. Зокрема, вона концентрує зусилля на розробці і впровадженні сучасних технологій візуалізації обчислень, тривимірної графіки, обробки звуку і мультимедіа.

Рисунок 15.1. SMP Power Challenge

Свого часу розробка процесора R10000 дозволила компанії перейти до об'єднання своїх серверів Challenge (на базі процесора R4000) і PowerChallenge (на базі процесора R8000) в єдину лінію продуктів. Завдяки підвищеній продуктивності цього процесора на цілочисельних операціях і плаваючою точки, обидві лінії продуктів можуть бути об'єднані без втрати продуктивності.

Сервери Silicon Graphics працюють під управлінням операційної системи IRIX (ОС UNIX реального часу), побудованої відповідно до вимог стандартів SVID (System V Interface Definition) і XPG4. Вона підтримує можливість роботи декількох машин на одному шлейфі SCSI (multi-hosted SCSI), чотириразове віддзеркалення і 128-кратне розщеплення дискових накопичувачів. На платформі підтримуються багато продуктів компаній Oracle, Informix і Sybase.

Комп'ютери CHALLENGE DM / L / XL орієнтовані в першу чергу на комерційні застосування, а POWER CHALLENGE L / XL - на роботу з плаваючою комою. Системи CHALLENGE DM відносяться до середнього класу.

POWER CHALLENGE відноситься до класу симетричних мультипроцесорних ЕОМ (SMP-системи), що базуються на поколінні суперскалярні процесорів MIPS R8000 фірми Silicon Graphics.

Відмінними рисами цих систем є:

- масштабованість суперкомпьютінга;

- використання великої динамічної пам'яті (адресація у POWER CHALLENGE до 16 Гбайт - в два рази вище, ніж у Cray T90 / C90 / J90);

- 64-розрядна архітектура;

- двоичная сумісність з усім сімейством комп'ютерів SGI, включаючи робочі станції Indy.

Таблиця 15.1. Продуктивність комп'ютерів POWER CHALLENGE

Комп’ютери	Число відсот.	Пікова продуктивність, Гфлоп	Продуктив-ність I/O, Гбайт/с	RAM, Гбайт	Диски, Тбайт
POWER CHALLENGE XL	1-18	0.36-6,5	до 1,2	0,064-16	до 6,3
POWER CHALLENGE array	до 144	до 51,8	до 4	до 128	до 63

За пікової продуктивності POWER CHALLENGE поступається векторних комп'ютерів Cray (старим, природно, моделям). Однак процесор R8000 на 90 МГц виявився швидшим, ніж процесори в Cray Y-MP (333 МФлоп) і комп'ютері Cray J90 (200 МФлоп). Дані тестів LINPACK показують, що при однаковій кількості процесорів SMP-комп'ютери від SGI випереджають Cray J90 і поступаються Cray C90 як на середніх (N = 100), так і на довгих (N = 1000) векторах всього в 3 рази. Рівень розпаралелювання (відношення продуктивності n процесорів до продуктивності одного процесора) в серверах SGI трохи вище, ніж в DEC AlphaServer 8400, і, в свою чергу, трохи нижче, ніж у Cray. Відомо, що з ростом числа процесорів ефективність розпаралелювання сильно зменшується. Завдання, які добре распараллелівать при великому числі процесорів, часто можуть

ефективно виконуватися і в рамках моделі розподіленої пам'яті, в тому числі в кластерних системах. Такі додатки бувають, наприклад, в динаміці рідини, обробці сейсмоданих і т.д. Silicon Graphics пропонує користувачам кластер POWER CHALLENGEarray, який може містити до 8 SMP-серверів POWER CHALLENGE L / XL. Він має до 144 процесорів з піковою продуктивністю 52 GFLOPS і до 128 Гбайт RAM, і, таким чином, перевершує по ряду показників машини Cray. Сервери зв'язуються в кластер через FDDI або HiPPI.

POWER CHALLENGE XL перевершує старші моделі Cray за розміром оперативної пам'яті, поступаючись по продуктивності введення-виведення і максимальним розміром дискового простору. Більший можливий розмір пам'яті в системах SGI пов'язаний з використанням дешевої DRAM-технології в порівнянні з дорогою високопродуктивної SRAM-пам'яттю в Cray C90 / T90.

15.2. Сімейство SUN Ultra Enterprise компанії SUN

Sun Ultra Enterprise - це серія масштабованих, зручних в управлінні і надійних серверів. Її можна розділити на наступні групи:

- сервери робочих груп: Sun Enterprise 10s, 250, 220R, 450 і 420R;

- сервери відділу підприємства: Sun Enterprise 3500 і 4500;

- сервери масштабу підприємства: Sun Enterprise 5500, 6500 і 10000.

Sun Enterprise 10s - сервер Sun початкового рівня, який взаємодіє з усіма клієнтами мережі: персональними комп'ютерами (ПК), системами Macintosh, робочими станціями UNIX, мережевими комп'ютерами. Завдяки процесору UltraSPARC-IIi, що працює з тактовою частотою 440 МГц, який підтримує до 1 Гб пам'яті, 4 PCI

слота розширення і до 18,2 Гб на зовнішньому носії, ця система надає недорогу платформу для роботи критичних мережевих додатків. Сервер Sun Enterprise 10s призначений для Web, електронної пошти, файлових служб і служб друку. Крім того, Solaris для Intranet - розширення операційної системи Sun Solaris - забезпечує надання Web-послуг.

Рисунок 15.2. Sun Enterprise 10s

Sun Microsystems представляє двохпроцесорний SMP-сервер масштабу робочої групи. У лінійці продуктів Sun сервер Sun Enterprise 250 позиціонується між системами Sun Enterprise Ultra 5s і 10s та сервером Sun Enterprise 450 і заміщає собою сервер Sun Enterprise 150. Сервер Sun Enterprise 250 поставляється у вигляді підлогового рішення або стійки.

Рисунок 15.3. Сервер Sun Enterprise 250

Сервер Sun Enterprise 250 зручний для невеликих, можливо, віддалених офісів, розрахованих на 50-100 користувачів. Сервер Sun Enterprise 250 підтримує щоденну роботу офісу: виконання додатків баз даних, фінансових додатків; підтримка обміну повідомленнями, надання сервісу додатків, а також підтримка стандартних додатків робочих груп (документообіг, розподіл і планування товарів і послуг і т.д.). Він також підходить для ринку провайдерів Internet-послуг і офісних додатків.

Сервер Sun Enterprise 250 підтримує від 128 Мбайт до 2Гбайт оперативної пам'яті, при цьому швидкість передачі даних досягає 1,6 Гбайт / с. Він також вміщує в себе до шести 1-дюймових або 1,6-дюймових дисків UltraSCSI зі швидкістю зовнішнього інтерфейсу 40 Мбайт / с, що допускають роботу в режимі "гарячої заміни". Використовуються диски на 4,2 Гбайт, 9 Гбайт або 18 Гбайт, таким чином, максимальний обсяг збережених даних дорівнює, відповідно,

25 Гбайт, 54 Гбайт або 108 Гбайт.

Рисунок 15.4. Сервер Sun Enterprise 220R

Сервер Sun Enterprise 450 призначений для організації електронної пошти, роботи з базами даних, створення кластерних комплексів, надання Web-сервісу, забезпечення управління ресурсами підприємства. У складі сервера працюють до чотирьох процесорів UltraSPARC з тактовою частотою 300 МГц або 400 МГц, внутрішня шина UPA зі швидкістю 1,6Гбайт / с і підсистема вводу-виводу на базі шини PCI, що забезпечує пропускну здатність 600 Мбайт / с. Оперативна пам'ять розширюється до 4 Гбайт, вбудовані диски Ultra SCSI ємністю до 84 ГБ з режимом швидкої "гарячої заміни" і можливості підключення зовнішніх накопичувачів загальною ємністю до 6 Тбайт.

Сервер може взаємодіяти з будь-якими робочими станціями, Intel і Macintosh. Надійність закладена в сервер Sun Enterprise 450 спочатку: такі можливості як корекція помилок ECC на внутрішній шині і в пам'яті, автоматичне відновлення системи після збою і резервуються з режимом "гарячої заміни" блоки живлення і дискові накопичувачі присутні тут в стандартній конфігурації.

Рисунок 15.5. Сервер Sun Enterprise 450

Сервер Sun Enterprise 420R - чотирипроцесорний сервер масштабу робочої групи, призначений для проведення мережевих обчислень і заснований на технології UltraSPARC. Сервер надає можливості мультипроцессорной системи, диски UltraSCSI, шину введення / виводу PCI, що є стандартом в галузі, в компактному корпусі, призначеному для монтажу в промислові стійки.

Рисунок 15.6. Сервер Sun Enterprise 420R

Сервер Sun Enterprise 420R призначений для Internet-провайдерів і провайдерів послуг, фінансових установ, для організації високопродуктивних обчислень і для будь-яких галузей, де потрібно потужний сервер обробки даних, що займає небагато місця. Надзвичайно важливо, що сервер Sun Enterprise 420R забезпечує високий рівень продуктивності при повній двійковій сумісності вгору і вниз по всій лінійці серверів і робочих станцій.

Сервер Sun Enterprise 3500 дає можливість виконувати складні ділові додатки і надавати Internet / Intranet послуги з тією ж продуктивністю і доступністю, що і вельми дорогі великомасштабні системи. Час простоїв значно знижено завдяки стійкій архітектурі сервера, засобів системного управління та новим програмним можливостям, таким як динамічна реконфігурація і вибір альтернативного маршруту (Dynamic Reconfiguration and Alternate Pathing).

Рисунок 15.7. Сервер Sun Enterprise 3500

Поєднання масштабируемой операційної системи Sun Solaris, яка встановлюється на всіх серверах Sun, і модульних апаратних компонентів дозволяє легко нарощувати продуктивність і розширювати можливості системи. Як сервер додатків Enterprise 3500 демонструє високу продуктивність, об'єднуючи можливості восьми UltraSPARC процесорів, пов'язаних високопродуктивної системою Gigaplane. Загальна архітектура сімейства серверів Sun Enterprise 3500-6500 дозволяє виробляти локальне нарощування обчислювальних потужностей шляхом підключення до 30 процесорів. Програмне забезпечення Sun Enterprise SyMON полегшує управління системою, завдяки простому у використанні інтерфейсу і систему попередження

збоїв апаратного забезпечення.

Сервер Sun Enterprise 4500 - це компактний сервер середнього рівня з обчислювальними можливостями, практично в два рази перевершує можливості наращиваемости сервера Sun Enterprise 3500. Сервер забезпечує доступність для критично важливих додатків і зручні засоби керування системою для ділових додатків баз даних і електронної комерції. Загальна модульна архітектура сімейства серверів від Sun Enterprise 3500 до Sun Enterprise 6500 полегшує модифікацію. Завдяки використанню в серверах Sun операційної системи Solaris можна працювати з більш ніж 12000 додатків.

Рисунок 15.8. Сервер Sun Enterprise 4500

У сервері Sun Enterprise 4500 можлива установка до 14 процесорів. До 4 серверів Sun Enterprise 4500 можна встановити в

стійку центру даних. Програмне забезпечення Sun Enterprise management Center полегшує управління системою, завдяки простому у використанні інтерфейсу і систему попередження збоїв апаратного забезпечення. Загальна архітектура сімейства серверів від Sun Enterprise 3500-6500 дозволяє виробляти локальне нарощування обчислювальних можливостей шляхом підключення до 30 процесорів.

Сервер Sun Enterprise 5500 призначений для великих систем центрів даних. Число процесорів змінюється від 1 до 14. Основна пам'ять розширюється від 256 Мбайт до 14 Гбайт. Можливі конфігурації наповнення одного банку від 256 Мбайт до 1 Гбайт (групи з 8 модулів SIMM). Є зовнішній масив зберігання даних. Підтримується більше 6 Тбайт даних.

Рисунок 15.9. Сервер Sun Enterprise 5500

Сервер Sun Enterprise 6500 створений для виконання критично важливих промислових додатків, таких як сховища даних і ERP-системи. Модульні компоненти цього високомасштабіруемого сервера дозволяють збільшувати продуктивність і можливості введення / виведення. Такі можливості як динамічна реконфігурація і вибір альтернативного маршруту (Dynamic Reconfiguration and Alternate Pathing), введені в операційну систему Solaris, дозволяють додавати, видаляти, модифікувати і обслуговувати системні компоненти, не припиняючи роботи сервера. Використання в сервері Sun Enterprise 6500 операційної системи Solaris гарантує йому масштабування до максимальних можливостей. Кількість процесорів змінюється від 1 до 30. Основна пам'ять масштабується від 256 Мбайт до 30 Гбайт. Можливі конфігурації наповнення одного банку від 256 Мбайт до 1 Гбайт (групи з 8 модулів SIMM). Підтримується більше 10 Тбайт даних.

Рисунок 15.10. Сервер Sun Enterprise 6500

Сервер Enterprise 10000 націлений на роботу з важливими додатками: інформаційними сховищами, системами прийняття рішень, консолідованих ЛВС (LAN) або високооб'ємних додатків з онлайнової обробкою транзакцій (OLTP).

Рисунок 15.11. Сервер Enterprise 10000

Enterprise 10000 - це єдина UNIX-система, що забезпечує роботу з незалежними розділами, як на мейнфрейми, що вельми важливо для ефективного використання обчислювального центру. Система забезпечує масштабування в тому, що стосується продуктивності, числа користувачів, ємності додатків, розширюючись до 64 процесорів UltraSPARC з тактовою частотою 400 МГц, і включає

дискову підсистему в тому ж єдиному корпусі. Властивості забезпечення постійної працездатності SunTrust роблять систему Enterprise 10000 найбільш надійною в своєму класі.

Система Enterprise 10000, що вміщає до 64 Гбайт пам'яті, що, з шириною пропускання внутрісистемної магістралі до 12 Гбайт/с для швидкої пересилання даних і фіксованого часу затримки, а також з підтримкою RAID 0, RAID 1 і RAID 5, перевершує по продуктивності всі інші масштабовані системи . Крім того, забезпечується підтримка до 20 Тбайт дискового простору, що істотно для проектів найбільших обчислювальних центрів. Можливості "гарячої заміни" дозволяють легко проводити оновлення і заміну компонентів в існуючих системах без необхідності їх перезавантаження або виключення живлення.

16. КЛАСТЕРИ ТА МАСИВНО-ПАРАЛЕЛЬНІ СИСТЕМИ РІЗНИХ ВИРОБНИКІВ

16.1. Сімейство масивно-паралельних машин ВС БКС-100 і БКС -1000

Масивно-паралельні масштабовані системи багатопроцесорних комп’ютерних систем призначені для вирішення прикладних завдань, що вимагають великого обсягу обчислень і обробки даних. Супер-комп'ютерна установка системи багатопроцесорної комп’ютерної системи є мультипроцесорний масив, об'єднаний з зовнішньої дискової пам'яттю і пристроями введення-виведення інформації під загальним управлінням персонального комп'ютера або робочої станції.

Основні області фактичного застосування суперкомп'ютерів БКС-100/1000:

- рішення задач розрахунку аеродинаміки літальних апаратів, у тому числі явища інтерференції при груповому русі;
- розрахунок тривимірних нестаціонарних течій в'язкого стиснення газу;
- розрахунки течій з локальними тепловими неоднородностями в потоці;
- розробка квантової статистики моделей поведінки речовини при екстремальних умовах;
- розрахунки структуроутворення біологічних макромолекул;
- моделювання динаміки молекулярних і біомолекулярних систем;
- рішення задач лінійних диференціальних ігор. Динамічні задачі конфліктів управління;Сімейство масивно-паралельних машин

ВС БКС-100 і БКС -1000

– рішення задач механіки деформівних твердих тіл, в тому числі з урахуванням процесів руйнування.

Програмне забезпечення, встановлене на обчислювальних системах, по суті, мінімально за своїм обсягом: транслятори з мов FORTRAN і C (C ++); додаткові кошти опису паралельних процесів; програмні засоби PVM і MPI; засоби реалізації розрахованих на багато користувачів режимів і віддаленого доступу.

Міжвідомчий суперкомп'ютерний центр (МСЦ) був відкритий 5 листопада 1999 р ньому в якості основної машини була встановлена 16-процесорна система фірми HP V2250 продуктивністю 15 ГОпер / с. Паралельно з цією системою в МСЦ працював також 96-процесорний варіант вітчизняної системи БКС-1000. Сумарна продуктивність всіх систем центру досягала 230 ГОпер / с (2,3х1011).

16.2. Мультипроцессорна обчислювальна система БКС-100

Підхід, який використовується при створенні вітчизняних суперкомп'ютерів, полягає в закупівлі новітніх комплектуючих виробів, створення на цій основі суперкомп'ютерних систем, їх інтеграції в інформаційно-обчислювальні мережі і необхідних зусиль в області застосування, тобто в розробці прикладних програм і методів математичного моделювання. Така концепція реалізована в мультипроцессорной обчислювальної системі БКС-100.

БКС-100 - це вітчизняна мультипроцессорная система другого покоління. В даний час вона замінюється на БКС-1000. Система поставляється у вигляді типових конструктивних модулів по 32, 64 і 128 процесорів. Число процесорів в модулях і число модулів може варіюватися. Для основного обробітку застосовуються мікропроцесори

Intel 860 з продуктивністю до 100 Мфлоп (64 розряду при подвійної точності) і приєднаної оперативною пам'яттю, змінною від 8 до 32 Мбайт. Для межпроцессорного обміну в кожному вузлі використовується трансп'ютер, що працює з тією ж оперативною пам'яттю, а також пам'яттю зовнішнього обміну обсягом 2-8 Мбайт. Загальна пропускна здатність 4 транспьютерном каналів для зовнішнього обміну - 20 Мбайт / с. Система БКС-100 експлуатується в ряді інститутів РАН і промисловості. Установки БКС-100 із сумарним подаванням понад 50 ГОпер / с експлуатуються в обчислювальних центрах РАН (в Москві, Єкатеринбурзі, Новосибірську, Владивостоці) і в галузевих ВЦ. Показана можливість ефективного розпаралелювання обчислень і обробки даних.

16.3. Мультипроцессорна обчислювальна система БКС-1000

БКС-1000 - система третього покоління, заснована на використанні мікропроцесорів Alpha 21164 (розробка компанії DEC-Compaq; випускається також заводами фірм Intel та Samsung) з продуктивністю до 1-2 Тфлоп та приєднаною оперативною пам'яттю обсягом 0,1-2 Гбайт. Система БКС-1000 із продуктивністю до 1 Тфлоп складається з 8 стійок (512 вузлів).

В основному виконанні системи міжпроцесорний обмін структурно аналогічний використовуваному в системі БКС-100 і реалізується у двох модифікаціях: на базі <трансп'ютероподібного> зв'язного мікропроцесора TMS320C44 (фірма Texas Instruments), що має 4 канали з пропускною здатністю кожного в 20 Мбайт/ зв'язного мікропроцесора SHARC ADSP 21060 (фірма Analog Devices), що має 6 зовнішніх каналів із пропускною здатністю кожного 40 Мбайт/с.

Виконання БКС-1000К відрізняється використанням між про-

цесорного обміну комутаційної мережі MYRINET (фірма Myricom, США) з пропускною здатністю каналу в дуплексному режимі 2x160 Мбайт/с. Крім того, передбачено підключення до кожного процесора пам'яті на жорсткому диску об'ємом 2-9 Гбайт. У стандартній стійці знаходиться до 64 процесорів системи БКС-1000 або 24 процесори системи БКС-1000К. Передбачені засоби системного об'єднання стійок для установок із великою кількістю процесорів.

Процесорні вузли пов'язані між собою за схемою, подібною до топології двомірного тора (для 4-лінкових вузлів). Конструктивною освітою БКС-1000 є базовий обчислювальний блок, що містить 32 обчислювальні модулі.

Для управління масивом процесорів та зовнішніми пристроями, а також для доступу до системи ззовні використовується так званий хост-комп'ютер (керуюча машина). Зазвичай це робоча станція AlphaStation із процесором Alpha та операційною системою Digital Unix (Tru64 Unix) або ПК на базі Intel із операційною системою Linux.

Починаючи з 1999 р., всі новостворені БКС-1000 будуються як кластери виділених робочих станцій. Це означає, що, на відміну від ранніх версій БКС-1000, як обчислювальний модуль використовуються не спеціалізовані ЕОМ, призначені тільки для застосування як деталі суперкомп'ютерної установки, а звичайні персональні комп'ютери. Як комунікаційна апаратура використовуються звичайні мережеві плати і комутатори, що застосовуються для побудови офісних локальних мереж.

Як базова ОС вузла використовується Linux, що фактично є загальноприйнятим світовим стандартом для побудови систем такого класу. Це дозволило багаторазово розширити та спростити, порівняно з

ранніми версіями БКС-1000, адаптацію різноманітного програмного забезпечення, як вільно розповсюджуваного, так і комерційного.Розробники БКС -1000 пропонують користувачам набір єдиних архітектурних рішень, що охоплює три класи паралельних обчислювальних систем:

– великі і надвеликі системи, що складаються з сотень вузлів. Прикладом такої системи може служити БКС-1000М;

– середні і малі системи, що поставляються на замовлення. В даний час серійно випускаються БКС-1000/16 і БКС-1000/32, з 16 і 32 процесорів, відповідно. Крім того, аналогічні системи успішно експлуатуються в цілому ряді наукових центрів країни;

– віртуальні паралельні системи на базі офісної локальної мережі.

Для освоєння і вивчення паралельних технологій, а також для постійної експлуатації в умовах гострого дефіциту фінансових коштів пропонується програмне забезпечення, що дозволяє організувати повноцінну паралельну систему на базі обладнання вузівського комп'ютерного класу, або аналогічної офісної локальної мережі. При цьому з точки зору користувача система не тільки не відрізняється від <справжньою> за способом роботи, але і дозволяє отримувати очевидний виграш в швидкості при виконанні реальних програм. На відміну від реалізованих апаратно суперкомп'ютерів БКС-1000, такий повністю віртуальний суперкомп'ютер називається БКС-900.

Обчислювальні системи з розподіленою пам'яттю компанії Sequent

У 1999 р комп'ютерні корпорації IBM і Sequent оголосили про своє злиття, і Sequent фактично стала підрозділом IBM. Після цього IBM зупинила значне оновлення лінії серверів NUMA Q, що

випускалися Sequent.

Sequent був постачальником масштабованих NUMA-серверів (серії NUMA-Q 1000 і NUMA-Q 2000), що включають до 64 процесорів Intel і призначених в основному для комерційних систем онлайнової обробки транзакцій і підтримки СУБД.

Ртсунок 16.1. Система Sequent NUMA-Q

Крім того, NUMA-технології, розроблені фахівцями Sequent, використовуються і в серверах від IBM. Розробники IBM стверджують, що NUMA стане визначальною технологією для UNIX- і NT-серверів вже на початку 21 століття.

Sequent була, мабуть, другий після IBM компанією, що здійснила поставки UNIX-кластерів баз даних в середини 1993 г. Вона пропонувала рішення, відповідні середньому і високому рівню готовності своїх систем. Спочатку Sequent Hi-Av Systems забезпечували дублювання систем, які поділяли загальні диски. Користувачі могли вибирати ручний або автоматичний режим перемикання на резерв в разі відмови. Hi-Av Systems забезпечує також гаряче резервування IP-адрес і дозволяє кластеру, до складу якого входить до чотирьох вузлів, мати єдиний мережеву адресу.

Компанія Sequent однією з перших освоїла технологію Fast-Wide SCSI, що дозволило їй досягти значного збільшення продуктивності систем при обробці транзакцій. Компанія підтримує дискові підсистеми RAID рівнів 1, 3 і 5. Крім того, вона пропонує в якості ресурсу, стрічкові накопичувачі SCSI. Модель SE90 підтримує кластери, до складу яких можуть входити два, три або чотири вузли, що представляють собою багатопроцесорні системи Symmetry 2000 або Symmetry 5000 в будь-якій комбінації. Це досить потужні системи. Наприклад, Sequent Symmetry 5000 Series 790 може мати від 2 до 30 процесорів Pentium 66 МГц, оперативну пам'ять ємністю до 2 Гбайт і дискову пам'ять ємністю до 840 Гбайт.

При роботі з Oracle Parallel Server всі вузли кластера працюють з єдиною копією бази даних, розташованої на загальних поділюваних дисках.

Серед розробок архітектура cc-NUMA виділяється принципово. Це архітектура симетричного багатопроцесування (SMP), що володіє безліччю достоїнств: проста модель програмування, відмінна масштабованість, можливість роботи з кількістю процесорів більше

восьми, переносимість додатків і т. Д. Але є і недоліки у вигляді високої вартості і спеціального програмного забезпечення. Біля витоків створення архітектури cc-NUMA стояла компанія Sequent, що реалізувала власну версію NUMA-Q.

Таблиця 16.1. Характеристики комп’ютерної системи NUMA-Q 2000

Производитель	IBM (раніше Sequent)
Класс архітектури	Багатопроцесорна система із загальною пам'яттю (cc-NUMA). Використовується в організацію складних інформаційних систем.
Модифікації	Model E410/E330/E320/E300/E200
Процесори	Intel Pentium III Xeon (700 MHz в моделі E410)
Вузол	Від 4 до 64 процесорів, до 64 Гб оперативної пам'яті; вузол складається з базових плат по 4 процесори (quads), з'єднаних між собою комутатором IQ-Link
Масштабуємість	Можлива організація кластерів, що включають до 4 вузлів
Системне ПЗ	Використовується операційна система DYNIX/ptx: версія UNIX від Sequent. Усередині однієї системи можуть одночасно виконуватись UNIX та Windows NT

Першою NUMA-системою була машина Butterfly, розроблена BBN в 1981 р Першої діючою системою з архітектурою cc-NUMA стала Stanford DASH. Група, яка працювала з нею, використала можливість вивчити функціонування операційної системи SGI IRIX на 32-процесорному вузлі cc-NUMA в 1992 р

16.4. Обчислювальна система з розподіленою пам'яттю фірми DATA GENERAL

Компанія Data General Corp. була куплена в 1999 р EMC Corporation і стала її підрозділом.

До цього Data General поставляла багатопроцесорні SMP-сервери серій AV 5500, AV 8500 і AV 9500. Ці сервери підтримують роботу з відмовостійкими дисковими і стрічковими підсистемами CLARiiON, засоби автоматичної діагностики AV / Alert, ініційовані оператором або автоматичні засоби перемикання на резервну систему, управління зовнішньої пам'яттю в режимі on-line, управління вводом / виводом і швидке відновлення файлів.

У разі відмови процесора, пам'яті або компонента введення / виведення система автоматично починає процес виключення і потім здійснює власну перезавантаження з виключенням відмовили компонентів. Стандартним засобом зазначених систем є наявність надлишкових джерел живлення.

Максимальний ступінь готовності досягається при підключенні двох серверів до високонадійних дискового масиву CLARiiON. Дискові масиви CLARiiON Series C2000 Disk Array забезпечують RAID рівнів 0, 1, 3 і 5 в будь-якому поєднанні до 20 накопичувачів в одному шасі загальною ємністю до 80 Гбайт і можливість заміни накопичувача без виключення живлення. У конструкції дискового масиву використовуються надлишкові інтелектуальні контролери з дубльованими зв'язками, що забезпечують відмовостійкість. Стрічковий масив CLARiiON Series 4000 підтримує відмовостійке резервне копіювання і відновлення. У складі масиву використовується спеціальний процесор, який реалізує схему розщеплення даних,

подібну RAID рівня 5. Стрічковий масив забезпечує не тільки високу пропускну здатність, але і реалізує захист від відмов носія і накопичувача. Насправді, навіть при відмові накопичувача або картриджа, операції резервного копіювання або відновлення даних тривають без втрати даних. У масив можна встановлювати 3, 5 або 7 накопичувачів. При двократної компресії даних загальна ємність стрічкового масиву може досягати 48 Гбайт.

16.5. Кластери DIGITAL

Як і Sequent і Data General, фірма Digital (DEC) припинила своє самостійне існування, спочатку ставши відділенням компанії Compaq, а потім разом з нею увійшовши до складу Hewlett-Packard. Компанія DEC відома тим, що вона була розробником серверів AlphaServer, на основі яких будувалися Alpha-кластери. Перейшовши в іншу компанію, колектив Digital під маркою HP продовжує роботу по оновленню цієї лінії серверів.

Архітектура систем на основі процесора Alpha була розроблена в 1988-1991 рр. з перспективою розвитку на 20-25 років. Випускає в даний час четверте покоління процесорів містить чотирьохканальна суперскалярное ядро, 80 регістрів для цілочисельних операцій, 72 регістра з плаваючою комою, причому в процесі обробки одночасно може перебувати до 80 інструкцій. Для архітектури процесора Alpha характерно позачергове виконання команд, логіка передбачення розгалужень, повністю інтегрована на кристалі кеш-пам'ять першого рівня, багатоканальні пристрої доступу до оперативної пам'яті. У сімейство Alpha-сервер входить чотири серії серверів: AlphaServer DS, AlphaServer ES, AlphaServer GS і AlphaServer SC. Якщо в моделях AlphaServer DS використовується не більше 64 процесорів, то в

серверах AlphaServer SC їх може бути кілька сотень (до 512 і більше). Максимальна тактова частота процесорів в даний час складає 1,25 ГГц. Всі комп'ютери сімейства побудовані по комутованій технології, що дозволяє уникнути недоліків, властивих системної шини.

З кожною системою поставляється пакет HP Insight Manager - інструментальне засіб, що забезпечує моніторинг і управління серверами платформи Alpha. Системи AlphaServer GS розроблені спеціально для підтримки найпотужніших бізнес-додатків. Модульна архітектура систем забезпечує високу масштабованість.

В даний час на системах AlphaServer початкового рівня підтримуються операційні системи Tru64 Unix, OpenVMS і Linux, а в серверах середнього рівня і високопродуктивних серверах - Tru64 Unix і OpenVMS.

Одним з переваг платформи Alpha стали розвинені рішення в області побудови кластерів. Операційна система OpenVMS Cluster дозволяє об'єднувати до 95 вузлів з практично необмеженим територіальним розподілом, з повністю кооперативним спільним використанням ресурсів - систем зберігання даних, файлів і навіть окремих записів. При цьому кожна система має окреме обладнання, може запускатися і зупинятися незалежно. Додатки та програмні продукти допускають динамічне оновлення, нові пристрої зберігання даних також можна додавати і замінювати в <гарячому> режимі, а оновлення ОС OpenVMS розгортається у всьому кластері поетапно, зупиняти його роботу для цього не потрібно. OpenVMS Galaxy дозволяє виконувати в одній системі декілька примірників OpenVMS. Кожен екземпляр OpenVMS Galaxy можна запускати і зупиняти окремо. Обчислювальне середовище OpenVMS Galaxy пропонує

широкі можливості масштабування в SMP-конфігураціях. Сервери AlphaServer нового покоління мають високий ступінь масштабованості, дозволяючи будувати системи з числом процесорів від 2 до 128 з використанням всього двох типів <блоків>: 2-процесорних і 8-процесорних.

Таблиц1 16.2. Серии DS, ES, GS80, GS160, GS320

	Серія DS	Серія ES	GS80	GS160	GS320
Кількість моделей	6	3	1	1	1
Підтримувані процесори	EV67 600 МГц, EV68 833 МГц, EV68 1.25 ГГц	EV68 833 МГц, EV68 1 ГГц, EV68 1.25 ГГц	EV68 1.224 ГГц	EV68 1.25 ГГц	EV68 1.25 ГГц
Кількість процесорів	1-2	1-4	1-8	1-16	1-32
Кэш-пам'ять 2-го рівня на один процесор, Мбайт	2-4-8	8-16	16	16	16
Максимальний об'єм оперативної пам'яти, Гбайт	4	32	64	128	256
Максимальний об'єм внутрішніх дисків, Гбайт	436	2900	252	504	504
Максимальна кількість PCI слотів	5	10	56	112	224
Пропускнаа	0,532	1,85	3,2	6,4	12,8

спроможність шини в/в, Гбайт/с					
Максимальне число апаратних розділів	1	1	2	4	8
Максимальне число програмних розділів (Open VMS)	-	-	8	16	32

17. КЛАСТЕРИ ТА МАСИВНО-ПАРАЛЕЛЬНІ СИСТЕМИ РІЗНИХ ВИРОБНИКІВ

17.1. Серія Hitachi SR8000

Серія SR8000, або Супертехніческій сервер, була розроблена для чисельного моделювання складних науково-технічних завдань (структурний аналіз, динаміка рідини, прогноз погоди і т.п.). Серія об'єднує можливості як векторного суперкомп'ютера S-3000, так і паралельного комп'ютера SR2201.

Рисунок 17.1. Hitachi SR8000

Високопродуктивний 64-розрядний RISC-процесор розроблений і створений Hitachi з використанням CMOS-технології 0,14

мікрометрового довжини логічних елементів. Для максимальної ефективності мікропроцесорів на великомасштабних завданнях використовуються можливості псевдовекторной обробки. Це дозволяє даними вибиратися з оперативної пам'яті конвеєрним способом без затримки змінюваних процесів. В результаті дані подаються з пам'яті в арифметичні пристрої також ефективно, як в суперкомп'ютері векторного типу. Випускаються моделі SR8000 і SR8000 E1 / F1 / G1.

Таблиця 17.1. Конфігурація вузла Hitachi SR8000

Модель	SR8000	SR8000 E1	SR8000 F1	SR8000 G1
Пікова продуктивність, Гфлоп	8	9,6	12	14,4
Пам'ять	2/4/8	2/4/8/16	2/4/8/16	2/4/8/16

Для 144-вузловий конфігурації моделі G1 (450 МГц) при вирішенні повної системи лінійних рівнянь розмірністю 141000 була досягнута швидкість в 1709 Гфлоп / (теоретично можлива - 2074 Гфлоп / с), що дало ефективність 63%. На 112-вузловий моделі F1 (375 МГц) досягнута швидкість в 1035 Гфлоп / с з 1344 Гфлоп / с (ефективність - 77%). На окремому вузлі при вирішенні повної лінійної системи і симетричної задачі на власні значення (порядок 5000) процесорні швидкості були вище 6,2 і 4,1 Гфлоп / с, відповідно.

Таблица 17.2. Конфигурация системы Hitachi SR8000

Число вузлів		4	8	16	32	64	128	256	512
Продуктивність, Гфлоп	SR8000	32	64	128	256	512	1024	-	-
	SR8000	38,4	76,8	153,	307,	614,	1228,	2457,	4915,2

	E1			6	2	4	8	6	
	SR8000 F1	48	96	192	384	768	1536	3072	6144
	SR8000 G1	57,6	115, 2	230, 4	460, 8	921, 6	1843, 2	3686, 4	7372,8
Максимальний об'єм общей пам'яті, Гбайт	**SR8000 E1/F1/G 1**	64	128	256	512	1024	2048	4096	8192
Зонішній інтерфейс		Ultra SCSI, Ethernet/Fast Ethernet, Gigabit Ethernet, ATM, HIPPI, Fibre Channel							

17.2. Серія Fujitsu VPP5000

Серія VPP5000 є наступником колишніх систем VPP700 / VPP700E (остання система має тактовий цикл 6,6 нс замість 7 нс). Глобальні зміни в архітектурі щодо серій VPP700 малі. Тактовий цикл був зменшений наполовину. Архітектура вузлів VPP5000 майже ідентична вузлів VPP700. Кожен вузол в системі, званий процесорним елементом (ПЕ), є потужним векторних процесором (9,6 Гфлоп / с пікової швидкості і тактовий цикл 3,3 нс). Векторний процесор доповнено RISC-скалярним процесором з піковою швидкістю 1,2 Гфлоп / с. Формат скалярних команд має 64 розряду і може виконувати до 4 операцій паралельно. Кожен ПЕ має пам'ять до 16 Гбайт і кожен ПЕ безпосередньо з'єднується з іншими ПЕ зі швидкістю передачі 1,6 Гбайт / с.

Рисунок 17.2. Обчислювальна система Fujitsu VPP5000

Fujitsu VPP5000U - це однопроцесорна машина без мережі і розширень передачі даних, які потрібні для VPP5000.

Скалярний пристрій підтримує RISC-архітектуру <дуже довгого командного слова> (VLIW - Very Long Instruction Word), одночасно виконуючи до 4 команд за один тактовий цикл. Висока скалярная продуктивність досягається за допомогою як першого і другого кешей, так і асинхронного виконання звернення до пам'яті, команд з плаваючою комою і векторних команд.

Векторне пристрій складається з 4 конвеєрів, векторного регістра і регістра маски (mask register) зі швидкістю векторних операцій до 9,6 Гфлоп / ПЕ. Конвеєр квадратного кореня збільшує продуктивність в операціях, включаючи квадратні корені. Векторні операції виконуються зі швидкістю 2,4 Гфлоп.

Всі ПЕ з'єднуються через високошвидкісну мережу з поперечної комутацією. Особливе пристрій зв'язку між ПЕ, зване DTU (Data transfer unit), робить можливим одночасне виконання з'єднань між процесорами і обчислення. Це дозволяє виконувати передачу і прийом даних зі швидкістю 615 Мбайт / с в кожному напрямі, в той час як ПЕ виконують обчислення.

Система VPP5000 має додаткові можливості для операцій з плаваючою комою розширеної точності і непрямого доступу до пам'яті, що виникає в різних обчислювальних алгоритмах.

Компоненти ПЕ є високопродуктивними енергозберігаючими CMOS (complementary metal oxide semiconductor) LSI-мікросхемами, зробленими по 0,22 мк технології і містять до 33 мільйонів транзисторів кожна, з часом вентильной затримки (gate delay time) в 24 пікосекунди. Для оперативної пам'яті використовується 128-розрядна SDRAM (synchronous dynamic RAM) з часом довільного доступу в 45 наносекунд.

Проведені тести показали, що для системи з 32-х процесорів при вирішенні повної лінійної системи близько 170 880 швидкість склала 296,1 Гфлоп / с (ефективність - 96%). Для окремого процесора швидкість в 6,04 Гфлоп / с була досягнута при вирішенні системи близько 2 000. При обчисленні многочлена 10-го порядку була визначена швидкість в 8,68 Гфлоп / с (ефективність - більше 90%).

Основні технічні характеристики:

рік випуску - листопад 1999;

9,6 Гфлоп векторної продуктивності на ПЕ;

1,2 Гфлоп скалярною продуктивності;

масштабується від 1 до 128 ПЕ (512 ПЕ для особливого

розміщення) і досягає пікової продуктивності 1,228 Тфлоп;

4, 8 або 16 Гбайт оперативної пам'яті SDRAM на ПЕ (максимум 2 Тбайта на систему);

76,8 Гбайт / с пропускна здатність пам'яті (memory transfer bandwidth) на ПЕ;

64-розрядна архітектура;

операційна система UXP / V Unix System V Release 4.

Специфікації системи VPP5000U:

число процесорів — 1;

теоретична пікова продуктивність - 9,6 Гфлоп .;

оперативна пам'ять - 4-16 Гбайт;

Специфікації системи VPP5000:

число процесорів - від 4 до 128 (512 ПЕ для особливого розміщення);

теоретична пікова продуктивність - від 38,4 Гфлоп до 1,229 Тфлоп (4,915 Тфлоп для 512 ПЕ);

оперативна пам'ять - від 16 Гбайт до 2 048 Тбайт (8 192 Тбайт для 512 ПЕ);

пропускна здатність шини - 1,6 Гбайт / с / ПЕ.

17.3. Сучасні суперкомп'ютери - Cray T3E-1200

Системи Cray T3E - це масштабовані паралельні системи, які використовують DECchip 21164 (DEC Alpha EV5) RISC-процесори з піковою продуктивністю 600 Мфлоп і 21164A для машин Cray T3E-900 і Cray T3E-1200. Кожен процесорний елемент (ПЕ) Cray T3E має свою власну DRAM-пам'яті об'ємом від 64 Мбайт до 2 Гбайт. На відміну від системи CRAY T3D, в якій виконується завдання запитує фіксовану кількість процесорів на весь час виконання, в CRAY T3E вільні

процесори можуть використовуватися іншими завданнями. Моделі T3E, T3E-900, T3E-1200, T3E-1350.

Рисунок 17.3. Cray T3E

Кожен вузол в системі містить один процесорний елемент (ПЕ), що включає процесор, пам'ять і засіб комутації, яке здійснює зв'язок між ПЕ. Система конфігурується до 2048 процесорів. Пікова продуктивність становить 2,4 Тфлоп. Колективна, високопродуктивна, глобально адресується підсистема пам'яті робить можливим звернення до локальної пам'яті кожного ПЕ в Cray T3E. Процесорні елементи в системі Cray T3E пов'язані в тривимірний тор двобічної високошвидкісною мережею з малим часом затримки, яка в шість разів перевершує за швидкістю аналогічну мережу в Cray T3D. Також додана адаптивна маршрутизація, при якій можливий обхід ділянок з

високою ефективністю передачі.

Системи Cray T3E виконують операції введення / виводу через численні порти на один або більше каналів GigaRing. Канали вводу / виводу інтегровані в тривимірну міжвузловими мережу і пропорційні розміру системи. При цьому при додаванні ПЕ пропускна здатність каналів введення / виведення збільшується, і масштабовані додатки можуть виконуватися на системах з великим числом процесорів так само ефективно, як на системах з меншим числом процесорів.

Для Cray T3E була створена масштабована версія операційної системи ОС UNICOS - ОС UNICOS / mk. Операційна система UNICOS / mk розділена на програми-сервери, розподілені серед процесорів Cray T3E. Це дозволяє управляти набором ресурсів системи як єдиним цілим. Локальні сервери обробляють запити ОС, специфічні для кожного ПЕ. Глобальні сервери забезпечують загальносистемні можливості, такі як управління процесами і файлові операції.

У додаванні до призначених для користувача ПЕ, які виконують програми та команди, системи Cray T3E включають спеціальні системні ПЕ, які виконують глобальні сервери UNICOS / mk. Так як глобальні сервери розташовані на системних ПЕ і не дублюються по всій системі, UNICOS / mk ефективно масштабуєма, повнофункціональна і обслуговує від десятків до тисячі ПЕ з мінімальною перевантаженням.

UNICOS / mk забезпечує наступні програмні функції:

- розподіл серверів управління файлами. Функції файлового сервера розподіляються, використовуючи локальні файлові програми-сервера, для забезпечення максимальної продуктивності і ефективності;

– ПЕ може генерувати не тільки послідовну, а й паралельну передачу даних, використовуючи деякі або навіть всі ПЕ даної програми;
– безліч глобальних файлових серверів: система управління файлами розподілена на безліч системних ПЕ, які дозволяють повністю використовувати паралельні дискові канали, підтримувані на Cray T3E.

17.4. Система T3E-1200

Швидкодія серії Cray T3E-1200 в два рази перевищує продуктивність систем Cray T3E при зменшеній вдвічі вартості за Мфлоп. Конфігурації в повітряно-рідинному охолодженні мають від 6 процесорів, а в рідинному - від 32 процесорів. Кожен процесор має продуктивність в 1,2 Гфлоп; для всієї системи пікова продуктивність змінюється від 7,2 Гфлоп до 2,5 Тфлоп. Масштабується до тисяч процесорів. Серія випущена в 1997 р.

Система призначена для найбільш важливих наукових і технічних завдань в аерокосмічній, автомобільній, фінансової, хіміко-фармацевтичної, нафтової і т.д. галузях промисловості, а також в широких областях прикладних досліджень, включаючи хімію, гідродинаміку, прогноз погоди та сейсмічні процеси.

Для підтримки масштабованості використовується операційна система UNICOS/mk -масштабіруемая версія UNICOSR. Системи T3E-1200 підтримують як явне розпаралелювання розподіленої пам'яті за допомогою CF90 і C/C ++ з передачею повідомлень (MPI, MPI-2 і PVM) і передачу даних, так і неявне розпаралелювання за допомогою можливостей HPF і Cray CRAFT.

На системах T3E кожен інтерфейс GigaRing має максимальну

пропускну здатність в 500 Мбайт/с.

На додаток до високої продуктивності і пропускної здатності процесорних елементів і високою масштабованості, системи Cray T3E-1200 мають дві унікальні особливості: STREAMS і E-Регістри. STREAMS доводять до максимуму пропускну здатність локальної пам'яті, дозволяючи процесору запускати при повній швидкості для посилання на векторoподобние дані. E-Регістри надають операції gather/scatter (з'єднання/роз'єднання) для посилань на локальну і віддалену пам'ять і використовують повну пропускну здатність у внутрішньому сполученні для віддаленого читання і запису окремого слова.

Оцінка продуктивності обчислювальної системи проводилася при вирішенні щільною лінійної системи рівнянь порядку 148800 на машині T3E-1200 з 1200 процесорами. Була досягнута швидкість в 1,127 Тфлоп/с, що становить 63% ефективності.

Таблиця 17.3. Оцінка продуктивності системи T3E-1200

Число процесорів	6 - 128 32 - 2048
Тактова частота процесора, МГц	600
Пікова продуктивність, Тфлоп	2,4+
Розмір пам'яті на процесор, Гбайт	0,256 - 2
Топологія внутрішнього з'єднання	3D двоспрямований тор
Максимальна двійкова пропускна здатність, Гбайт/с	122
Максимальне число каналів GIGARING	128
Пікова пропускна здатність введення/виведення, Гбайт	128

17.5. Комп’ютерна система ASCI White

Проект ASCI (Accelerated Strategic Computing Initiative - прискорена стратегічна обчислювальна ініціатива) ініційований оборонними програмами Міністерства енергетики США у співпраці з лабораторіями Lawrence Livermore і Los Alamos (США) для переходу від ядерних випробувань до методів, заснованим на чисельному моделюванні створення ядерної зброї, оцінки його продуктивності і т.п. Ініціатива ASCI є ключовим елементом програми обслуговування арсеналів Stockpile Stewardship, спрямованої на забезпечення безпеки і надійності ядерних арсеналів країни при відсутності випробувань ядерної зброї.

В кінці червня 2000 року компанія IBM повідомила, що вона побудувала найшвидший суперкомп'ютер у світі (на той момент), що виконує до 12 трильйонів обчислень в секунду, що в тисячу разів швидше, ніж продуктивність <Deep Blue ». Суперкомп'ютер RS / 6000 SP, відомий як ASCI White, що займає площу розміром в два баскетбольні поля, використовується Міністерством енергетики США в програмі щодо забезпечення безпеки і надійності запасів ядерної зброї без проведення натурних випробувань.

Система ASCI White є третім кроком в плані Міністерства енергетики США, згідно з яким продуктивність суперкомп'ютерної системи в 2004 р повинна складати 100 Топер / с. В рамках ASCI-проекту протягом кількох років передбачається створити серію суперкомп'ютерів продуктивністю в 1, 3, 10, 30 і 100 Тфлоп.

При перевірці можливостей суперкомп'ютера ASCI White показав обчислювальну продуктивність в 12,28 Тфлоп, перевищивши розбіжностей між вимогами контрактів в цьому пункті на 23%.

Система інстальована в каліфорнійській національної Ліверморської лабораторії.

Система складається з 8 192 мікропроцесорів, має оперативну пам'ять об'ємом в 6 Тбайт і дискову пам'ять в 160 Тбайт, що досить для шестиразового зберігання всіх книг бібліотеки Конгресу США.

Апаратне оточення ASCI White включає в себе систему IBM RS / 6000 SP з 512 симетричними мультіпроцесорними машинами (SMP-вузлами). Кожен вузол має 16 процесорів, а для системи в цілому - 8192 процесора, що забезпечується пікова продуктивність складає не менше 12 Топер / с. Система має загальну пам'ять 4 Тбайт і дискову пам'ять 150 Тбайт.

Додатково система IBM SP оснащена зовнішньої дискової пам'яттю, паралельної файлової системою GPFS, архівної пам'яттю і засобами візуалізації. Спеціалізована високошвидкісна мережа утворює магістраль і з'єднує всі компоненти системи ASCI White.

Система IBM SP, яка формує ядро ASCI White, утворена з багатьох пакетів, в більшості своїй містять чотири вузли. Всі вузли є симетричними Мультіпроцесори IBM RS / 6000 POWER3 з 64-розрядної архітектурою. Кожен вузол є автономною машиною, яка має власну пам'ять, операційною системою, локальним диском і 16 процесорами. IBM виробляє кілька різновидів вузлів POWER3. Вузли ASCI White відомі як вузли Nighthawk-2 (NH-2).

Процесори POWER3 є суперскалярного (одночасне виконання багатьох команд) 64-розрядними чіпами конвеєрної організації з двома пристроями з обробки команд з плаваючою комою і трьома пристроями з обробки цілочисельних команд. Вони здатні виконувати до восьми команд за тактовий цикл і до чотирьох операцій з

плаваючою комою за такт. Всі вузли з'єднані внутрішньої комутаційної мережею SP.

Загальна паралельна файлова система IBM GPFS (General Parallel File System) забезпечує обслуговування файлової системи для паралельних і послідовних додатків, що запускаються в оточенні RS / 6000 SP. GPFS розроблена аналогічно файлової системи UNIX: майже всі програми запускаються під GPFS так само, як вони запускаються в інших файлових системах. Це означає, що користувачі можуть продовжувати застосовувати звичайні команди UNIX для простих операцій над файлами.

GPFS надає спільний доступ до файлів, який може охоплювати багато дисководів на багатьох вузлах SP. Окремі файли зберігаються як ряд <блоків>, розподілених через диски на різних вузлах пам'яті. Також підтримується одночасне читання і запис різних файлів.

Для захисту обчислювальних засобів (Secure Computing Facility - SCF) використовується архівна система зберігання даних HPSS (High Performance Storage System).

Система ASCI White побудована таким чином, щоб підтримувати змішані моди програмування кластерної розподіленої пам'яті з SMP загальної пам'яті. MPI зазвичай використовується для з'єднання розподіленої пам'яті від вузла до вузла.

Операційна система, як і на машині ASCI Blue-Pacific, є версією UNIX IBM AIX. AIX підтримує як 32-розрядні, так і 64-розрядні системи RS / 6000. Номер поточної версії - AIX 4.3.

Підтримка паралельного коду на ASCI White включає паралельні бібліотеки, отладчики, профіліровщики, утиліти IBM і сервісні програми, які виконують аналіз ефективності виконання.

Підтримуються MPI, OpenMP, потоки POSIX і транслятор директив IBM. Доступні: паралельний відладчик IBM, засоби профілювання і TotalView.

ЛІТЕРАТУРА

1. Антонов А.С. Технологии параллельного программирования MPI и OpenMP: Учеб. пособие. Предисл.: В.А.Садовничий. – М.: Издательство Московского университета, 2012. - 344с. - (Серия “Суперкомпьютерное образование”). ISBN 978-5-211-06343-3.

2. Богданов А.В., Корхов В.В., Мареев В.В., Станкова Е.Н. Архитектуры и топологии многопроцессорных вычислительных систем. Интернет-университет информациогнных технологий – ИНТУИТ.ру, 2004.

3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ - Петербург, 2002. – 608 с.

4. Вычислительные системы, сети и телекоммуникации / В.Бройдо – СПб.: Питер, 2002. – 688 с.

5. Гергель В.П. Высокопроизводительные вычисления для многоядерных многопроцессорных систем. Учебное пособие – Нижний Новгород; Изд-во ННГУ им. Н.И.Лобачевского, 2010

6. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. Учебное пособие – Нижний Новгород: Изд-во ННГУ им. Н.И.Лобачевского, 2003.- 184 с.

7. Головкин Б.А. Вычислительные системы с большим числом процессоров. - М.: Радио и связь, 1995. – 320 с.

8. Душкин Р.В. Квантовые вычисления и функциональное программирование. ДМК Пресс, 2015. - 232 с.

9. Орлов С.А., Цилькер Б.Я. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.:– Питер, 2011. – 688 с.

10. Паттерсон Д., Хеннеси Дж. Архитектура компьютера и

проектирование компьютерных систем. 4-е изд. – СПб.: Питер, 2012. – 784 с.

11. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стеен. — СПб.: Питер, 2003. — 877 с: ил. — (Серия «Классика computer science»).

12. Харрис Д.М., Харрис С.Л. Цифровая схемотехника и архитектура компьютера: 2-е изд. - МК.: 2013 -1619 с.

13. Хеннеси Дж., Паттерсон Д.Л. Компьютерная архитектура. Качественный подход. 5-е изд. – М.:Техносфера, 2016. – 936 с.

14. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем: Учебник для вузов. – Питер, 2006. – 668 с.