

3943.233

Жуков І. А. д-р техн. наук  
Іванкевич О. В. канд. техн. наук

## МЕТОДИ ПРИСКОРЕННЯ ОБРОБКИ ВЕЛИКИХ БАЗ ДАНИХ ІЗ ЗАСТОСУВАННЯМ БАГАТОПРОЦЕСОРНИХ МОДУЛІВ

Інститут комп'ютерних технологій Національного авіаційного університету

*У статті запропоновані методи розпаралелювання запитів до баз даних, які дозволяють досягти в багатопроцесорних обчислювальних системах, що базуються на нових модулях розширення, оптимального балансування завантаження процесорів при відносно низьких витратах на міжпроцесорні комунікації. Запропоновано апаратні засоби та алгоритми, що дозволяють оптимізувати обробку трудомістких операцій сортування, з'єднання, вибірки й ін., що дозволяє значно скоротити час обробки запитів у системі.*

**Вступ.** У наш час все більш важливою стає можливість оперативного використання накопичених людством знань в основних сферах державної, суспільної, виробничої й управлінської діяльності. Інформаційні системи, які застосовуються в багатьох галузях науки і техніки, обробляють задачі в режимі реального часу чи при мінімальному проміжку між запитом та відгуком і все частіше зіштовхуються з проблемою недостатньої швидкодії. Проблема виникає як за рахунок збільшення обсягу інформації в базах даних (БД), так і зростаючою тенденцією до застосувань систем розподілених БД, що використовують інформацію, яка міститься у вузлах локальних комп'ютерних мереж.

**Аналіз досягнень в галузі обробки великих БД.** Існує коло науково-технічних і комерційних задач обробки надвеликих масивів даних, для реалізації яких вже недостатньо звичайного комп'ютера з однопроцесорною архітектурою [1]. Прикладом таких задач є використання БД лінійного прискорювача для експерименту BABAR (вивчення зіткнення субатомних часток з метою визначення впливу поводження матерії й антиматерії на формування всесвіту) обсягом 895 Тбайт [2]. Науковцями також використовується надвелика база системи спостереження Землі EOS/DIS (*Earth Observation System / Data and Information System*) [3], яка містить у собі дані безлічі супутників, що збирають інформацію для вивчення довгострокових тенденцій стану атмосфе-

ри, океанів і земної поверхні з обсягом надходження інформації 1 Пбайт на рік.

Розвиток технологій і теорії систем БД (СБД) обумовлено повсякденним використанням комп'ютерів різних типів для вирішення великого кола задач, пов'язаних зі збереженням і обробкою великих і надвеликих обсягів інформації. Індустрія БД активно розвивається як у напрямку створення досконалих програмних засобів, що дозволяють прискорити обробку даних на універсальних комп'ютерах, так і в напрямку розробки спеціалізованих комп'ютерів і СБД, архітектура яких орієнтована на переробку нечислової інформації. Під СБД мається на увазі комп'ютеризована система збереження записів, основна мета якої – містити інформацію і надавати її за вимогою. Сучасна СБД складається з БД, системи керування базою даних (СКБД), що визначає дані, здійснює обробку, захист, забезпечує цілісність даних тощо; а також апаратного забезпечення і користувачів.

Дослідження по першому напрямку розвитку індустрії БД передбачають вирішення задачі оптимальної організації збереження і пошуку інформації в БД, що складають інформаційне ядро обчислювальних систем (ОС). Відомі СКБД, як комерційні, так і з відкритим кодом, дозволяють оптимізувати операції з БД за рахунок використання індексних таблиць, кешування, попереднього читання даних і т. ін., однак така оптимізація ефективна при роботі з невеликими БД, а її ефект з ростом обсягу даних зменшується.

Розробка другого напрямку дозволяє створювати ОС на основі універсальних і спеціалізованих процесорів, машин БД, комп'ютерних мереж, що зараз стають домінуючими інструментами для створення систем інтенсивної обробки даних.

Ефективним рішенням проблеми істотного підвищення продуктивності СБД є використання багатопроцесорних ОС (БОС), що забезпечують паралельну обробку даних. Такі БОС орієнтовані на вирішення задач, пов'язаних з виконанням великого обсягу як обчислювальних операцій, так і процедур обробки великих та складних структур даних. Розвиток БОС пройшов шлях від експериментів до повнофункціональних комерційних продуктів, що постачаються на ринок високопродуктивних систем обробки БД. Створено пристрої розпаралелювання, які включають, крім декількох типів багатопроцесорних платформ, шинні адаптери, RAID-масиви, програмно-керовані комутатори і великий набір готових конструктивів, що дозволяють оперативно створювати складні багатопроцесорні комплекси. З'явилися архітектури, засновані на розподіленій пам'яті, розподілених дисках, неоднорідній пам'яті (NUMA) та кластерах комп'ютерів без загальних ресурсів (*sharing-nothing*). Сучасні СБД найкращим чином пристосовані для систем без розподілених ресурсів, тому що такі системи мають найкращі характеристики масштабованості. Крім того, в таких кластерах застосовуються масові компоненти і тому вони стають значно дешевшими. Широке розповсюдження також отримали багатопроцесорні графічні системи. Зараз БОС застосовуються в інформаційно-пошукових системах [4,5], зокрема – в Інтернет-проектах *Google, Rambler, Yandex, Altavista*, електронній комерції, електронних бібліотеках, геоінформаційних системах, мультимедійних архівах, наукових БД та ін. Прикладами спішних проектів створення паралельних СБД є системи *NCR Teradata* [6], *DB2 Parallel Edition* [7] і *NonStop SQL* [8], що поєднують до тисячі процесорів і магнітних дисків і здатні обробляти БД у десятки і сотні Тбайт.

**Мета роботи.** Незважаючи на значні досягнення в галузі систем обробки надвеликих БД, залишається ряд проблем, що вимагають додаткових наукових досліджень, зокрема подальший розвиток алгоритмів обробки великих масивів даних за допомогою БОС, а також розробка нових підходів до створення високопродуктивних архітектур, орієнтованих на роботу в СБД і здатних забезпечити масштабованість системи, відмовостійкість і високу доступність даних при збереженні чи невеликому збільшенні вартості системи.

Метою статті є висвітлення нових ефективних методів і апаратно-програмних засобів побудови високопродуктивних СБД. У роботі розглядаються методи і засоби прискорення обробки великих БД, до яких умовно можливо віднести бази обсягом від 10 до 30 Гбайт, до надвеликих віднесемо бази понад 30 Гбайт [9].

**Підходи прискорення СБД.** Зараз існують два можливих підходи для прискорення СБД – організація спеціального програмного забезпечення та створення апаратного устаткування. До недоліків програмного підходу необхідно віднести велику вартість розробок і складність СКБД, трудомісткість експлуатації громіздких БД, низьку надійність і значне навантаження на центральний процесор, складність синхронізації виконання функцій керування файлами в розподілених системах, низьку ефективність використання універсальних комп'ютерів [10, 11].

При використанні апаратного підходу прискорення СБД можливе підвищення швидкості обробки запитів у сотні і тисячі разів; паралельна обробка запитів та паралельне виконання операцій, що значно збільшує продуктивність; підвищення надійності за рахунок динамічної реконфігурації багатопроцесорної системи, прискорення обчислень за рахунок реалізації функцій керування мікропрограмним або апаратним способами; розвантаження центрального процесора комп'ютера від функцій СКБД і операцій вводу-виводу; спеціалізація апаратних засобів відповідно до характеру обробки і збереження даних у формі, що найближча до конкретного представлення; – ефективна підтримка моделей даних.

Апаратну частину СБД пропонуємо побудувати, базуючись на трирівневій ієрархічній схемі, що використовує спеціальні багатопроцесорні модулі – акселератори СБД (АСБД) [12]. На першому рівні

ієрархії ОС, що використовуються для обробки великих масивів даних, поєднуються між собою в єдиний кластер за допомогою високошвидкісної обчислювальної мережі (рис. 1).

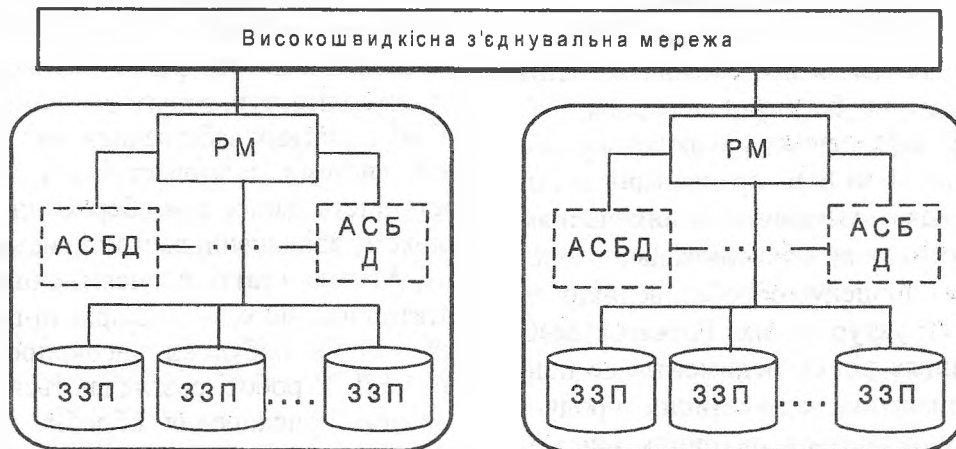


Рис. 1. Перший рівень ієрархії. Багатопроцесорна система з використанням АСБД

Другий рівень ієрархії зображений на рис. 2. У запропонованій структурі другого рівня СБД можна виділити дві основні частини:

- керуючий *Host*-комп'ютер, реалізований на основі стандартної ОС;
- один чи декілька зовнішніх багатопроцесорних модулів, під'єднані до *Host*-комп'ютера за допомогою внутрішніх системних інтерфейсів, що виконують основні операції по обробці даних і обчислювальні операції.

Багатопроцесорні модулі є третім, найнижчим рівнем ієрархії, що дозволяє на основі звичайної серверної ОС створювати швидкодіючу БОС. У СБД може

бути використаний один чи декілька однакових модулів прискорення, лінк-порти яких з'єднуються між собою для забезпечення швидкісного обміну даними. При використанні такого підключення з'являється можливість нарощування процесорів (залежно від конфігурації модулів) від 3 до 48, що дозволяє робити великі обсяги обчислень.

Як основа побудови багатопроцесорного модуля використовуються *DSP*-процесори, об'єднані між собою відповідно до архітектури, що забезпечує паралельність виконання операцій по обробці даних, а також обчислювальних операцій.

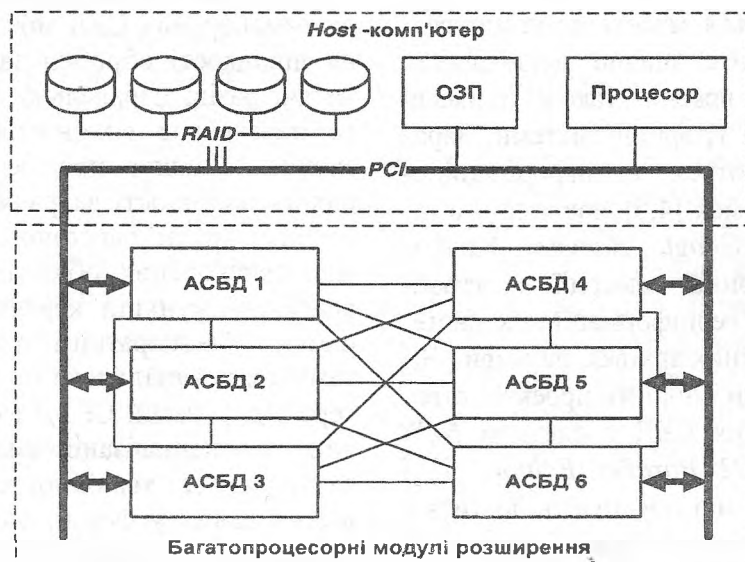


Рис. 2. Другий рівень ієрархії

Процесор *DSP* характеризується складною високопродуктивною обчислювальною структурою, що дозволяє реалізувати різні алгоритми обробки інформаційних потоків. Процесори мають порівняно невисоку вартість, а також розвинуті засоби розробки програмного забезпечення, що дозволяє застосовувати їх при побудові СБД із масовим паралелізмом. Також *DSP* характеризується високим ступенем спеціалізації; широким використанням методів скорочення тривалості командного циклу, характерні для універсальних *RISC*-процесорів, такі як конвеєризація на рівні окремих мікрокоманд і команд; розміщення операндів більшості команд у регістрах; використання "тіньових" регістрів для збереження стану обчислень при переключенні контексту, поділ шин команд і даних. У сигнальних процесорах використовується апаратний помножувач, який дозволяє виконувати множення як мінімум двох чисел за один командний такт.

Основними функціональними елементами АСБД є модуль сигнальних процесорів, робоча пам'ять, пам'ять програм,

а також блок керування, який може бути реалізований на основі спеціалізованого керуючого сигнального процесора, програмованій логічній інтегральній схемі (ПЛІС) чи мати розподілену структуру, при якій функції загального керування розподілені між *DSP*-процесорами. При розробці й аналізі базових структур АСБД за основу прийнятий процесор *ADSP-TS101S*. У восьмипроцесорній конфігурації АСБД містяться такі компоненти апаратного забезпечення (рис. 3): процесори *ADSP-TS101S*; блоки динамічної пам'яті загальним обсягом до 1 Гбайт; блоки статичної пам'яті загальним обсягом до 512Кх64; блок керування; контролер для підключення до *PCI*-шини *Host*-комп'ютера; лінк-порти для з'єднання з іншими подібними пристроями та їх синхронізації.

У АСБД використовується загальна внутрішня шина (64 біт, 66 МГц), яка дозволяє з'єднувати процесор з комутаційним блоком, а також зі статичною і динамічною пам'яттю. Розмір динамічної та статичної пам'яті може варіюватися залежно від поставлених перед АСБД задач.

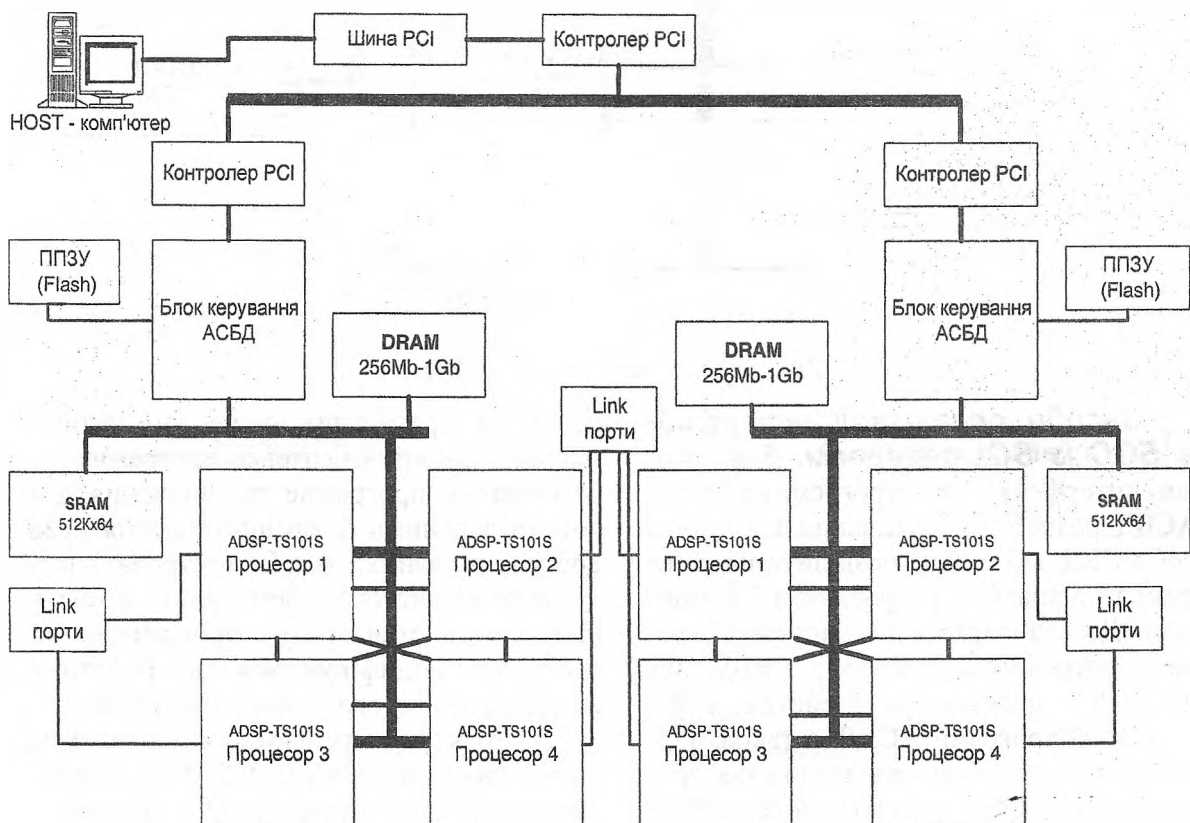


Рис. 3. Третій рівень ієрархії. Конфігурація АСБД з використанням восьми *DSP* процесорів

Пропонується використовувати в АСБД спеціальний блок керування, який призначено для фізичного з'єднання ОС через шину *PCI* з процесорами і пам'яттю АСБД, керування цими пристроями і відпрацьовування протоколів обміну. Функціональна схема блоку керування представлена на рис. 4. АСБД містить чотири, вісім або більше процесорів, які складають основу модуля, і об'єднані в єдиний кластер, підключений до внутрішньої шини модуля, до якої також мають доступ статична, динамічна пам'ять і блок комунації.

Зв'язок з *PCI*-шиною комп'ютера здійснюється за допомогою контролера *PCI9056*, що дозволяє забезпечити вихід на шину *PCI* як у режимах *Master* (ведучий), так і *Slave* (відомий). У пристрої ви-

користовується швидка статична пам'ять з однотактним доступом і обсягом до 512Кх64 (4 Мбайт) і поділена динамічна пам'ять (обсягом від 256 Мбайт до 1 Гбайт), що дозволяє здійснювати запис-зчитування як з боку будь-якого процесора, так і шини *PCI*. Доступ до динамічної пам'яті здійснюється за три процесорних такти. До апаратного забезпечення *Host*-комп'ютера в складі СБД висувуються наступні вимоги. Основною вимогою можна вважати ємність використовуваних ЗП, що особливо важливо для варіанта АСБД, при якому обмін даними в багатопроекторній системі із зовнішніми ЗП (ЗЗП) – найбільш інтенсивний. Тому доцільно використовувати *Host*-комп'ютер з оперативним ЗП (ОЗП) ємністю 128 Мбайт і вище.

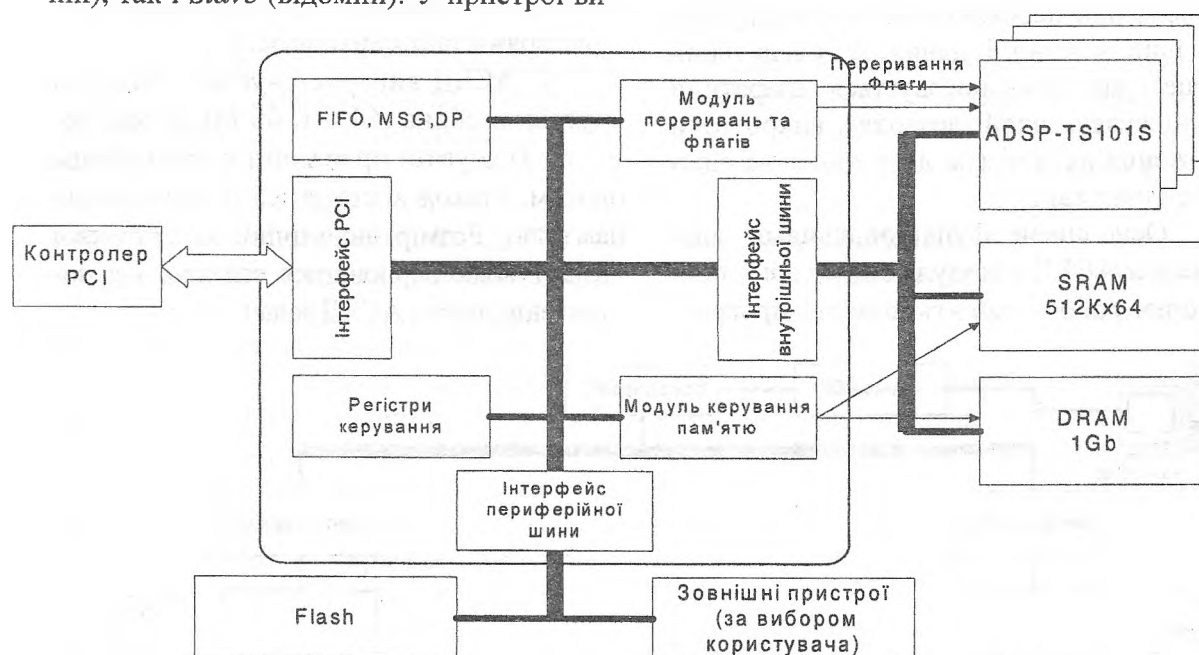


Рис. 4. Блок керування АСБД

**Засоби організації інтерфейсів БОС із SQL-сервером.** Для створення інтерфейсів між програмами СКБД і АСБД розроблені спеціальні програмні засоби СБД, які можна розділити на стандартні і спеціально розроблені компоненти. До стандартного відносяться програми операційної системи, СКБД *MS SQL Server*, драйвери *DSP*-систем, внутрішні мікропрограми АСБД, а також фірмові засоби розробки і налагодження програмного забезпечення *DSP*-системи. Розроблювальними компонентами є керуючі

програми віртуальних процесорів даних і масової пам'яті, клієнтське програмне забезпечення, програмне забезпечення для роботи з даними й адміністрування прав доступу до даних, засоби контролю і тестування апаратного забезпечення, а також компоненти стандартних програмних засобів, що модифікуються для роботи в складі програмного забезпечення СБД.

Структуру програмного комплексу, заснованого на СКБД *MS SQL Server*, представлено на рис. 5. При роботі у складі СБД роль СКБД полягає, головним



чином, в обробці невеликих за часом виконання чи обчислення обсягів даних, а також в інструментальній підтримці функцій адміністратора БД, тоді як великі запити, критичні за часом виконання, краще оформляти, використовуючи запуск спеціальних процедур сервера, які викликають програми, що працюють із системою співпроцесорів. У розробленій СБД система блоків розширення виконує лише окремі операції над даними, а нескладні операції з базами, у тому числі адміністративні, виконуються *MS SQL Server* на основному процесорі. Тому у користувача існує можливість вибору способу вико-

нання операцій СКБД цілком на *MS SQL Server* чи з використанням співпроцесорів.

До засобів, що були впроваджені в СКБД, відноситься препроцесор вхідної мови (компоновник вихідного потоку для АСБД), що здійснює виділення і переклад у внутрішнє представлення операторів, які виконуються співпроцесором. Велими важливою для СБД є система контролю і тестування апаратного забезпечення. Система одержує дані про конфігурацію системи співпроцесорів і виконує початкове завантаження й оперативне тестування співпроцесорів за допомогою

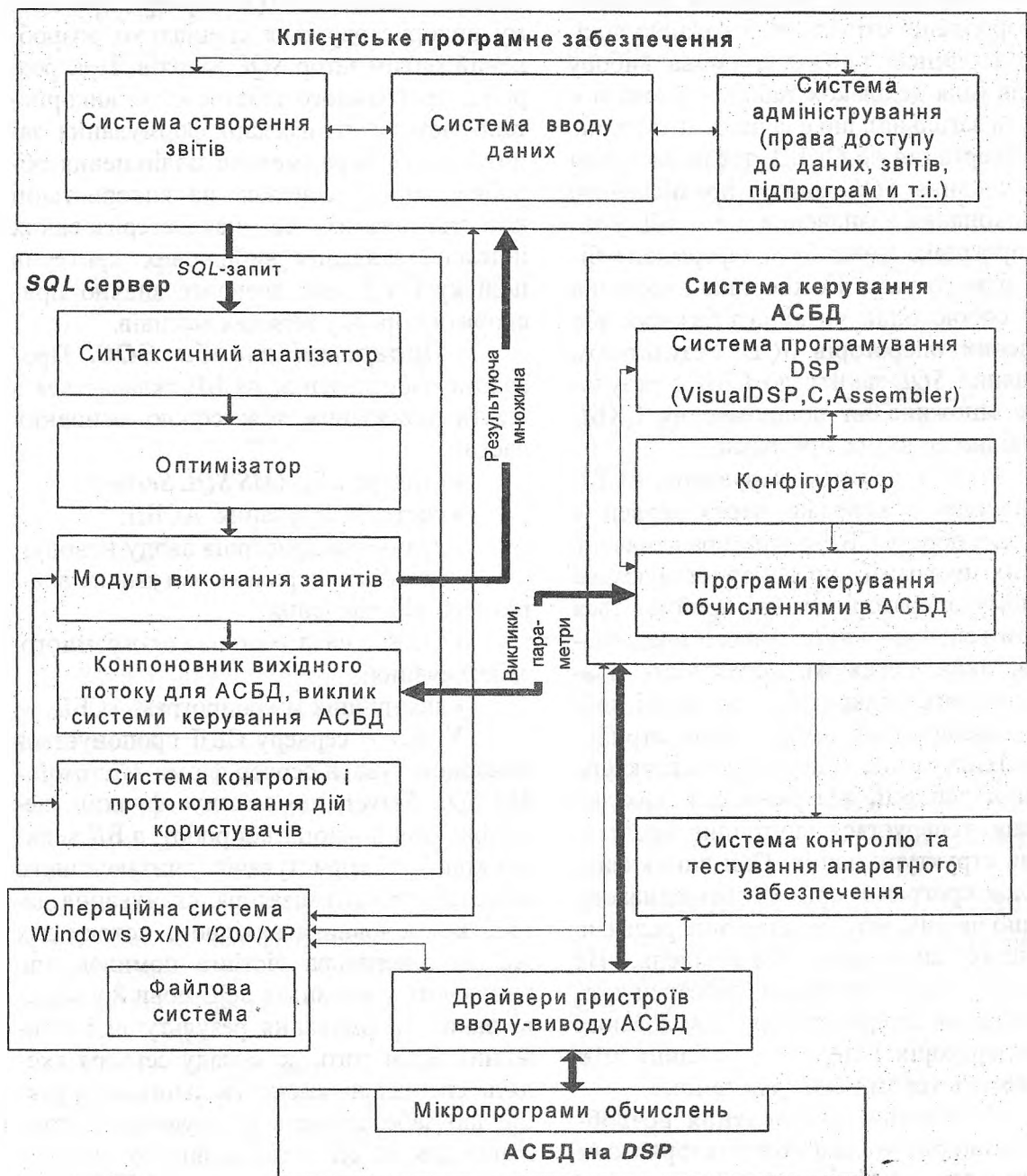


Рис. 5. Структура програмного комплексу для обробки великих БД

існуючих конфігураційних файлів. Крім того, у її функції входить пошук резервних конфігураційних файлів і передача їх на виконання при зміні конфігурації системи співпроцесорів, пов'язаної, наприклад, з відмовою елементів.

Для організації інтерфейсів між програмами керування БОС і СКБД запропоновано використовувати спеціальні функції *SQL*-сервера, які викликають програми керування БОС і передають до них дані. Запропоновано структури мови СБД для зв'язку БОС із *SQL*. Внутрішня мова забезпечує представлення базових операторів вхідної мови СБД. Інформація про кожну з таблиць зосереджена в спеціально розробленій структурі. З кожною вихідною таблицею пов'язана умова вибору записів (для декількох таблиць умова може бути загальною), що є фільтром таблиці.

Звертання до СКБД, тобто завдання на виконання *SQL*-запиту, призначеного для виконання визначених дій із БД, у тексті програми може бути оформлене більше одного разу. Кожне таке звертання являє собою один чи кілька базових або складених операторів *SQL*. Результатом виконання *SQL*-запиту до СБД є результуюча множина чи повідомлення СКБД відповідно до вимог програми.

Робота програм керування АСБД починається з передачі через сервер у комплекс образу СБД-оператора і значень змінних програми, що використовуються у даному операторі. Змінні приймаються блоком генерації внутрішнього представлення, який перевіряє коректність значень змінних, відкриваються вхідні таблиці, перевіряється правильність атрибутів та їхніх типів, генерується структура вихідної таблиці, відкривається вихідна таблиця, генерується внутрішнє представлення структури даних. При виникненні помилок програми організують однакову реакцію на них, візуалізацію попереджень та інших системних повідомлень. На останній стадії програми забезпечують пересилання інтерпретатору закодованих СБД-операторів і передачу таблиць між *Host*-комп'ютером і *DSP*- системою.

Програмне забезпечення розроблене з використанням компілятора мови *C* стандарту *ANSI*. Компілятор входить

до складу системи *VisualDSP++* компанії *Analog Devices*, яка містить у собі модулі мови *C*, що призначені для обробки даних на *DSP*-процесорах і містять наступні розширення мови *C*: операції з масивами і векторами, комплексні типи даних, циклічні покажчики і масиви перемінної розмірності. Програмне забезпечення для розроблювачів включає також наступні елементи: бібліотеку *C* зі стандартними функціями роботи з *DSP* процесорами, налагоджувач програм, написаних на *C* та асемблері, асемблер, компоновник і імітатор.

Авторами статті було організовано обробку запитів на *SQL*-сервері, при цьому використовується спеціально розроблений оптимізатор *SQL* запитів. При розробці програмного забезпечення використано методи оптимізації формування запитів до сервера, методи поліпшення обробки даних, засновані на використанні кластеризованих та некластеризованих індексів, завдання вибіркового критерію пошуку і т.д., що дозволяє значно прискорити обробку великих масивів.

**Програмні засоби БОС.** Програмне забезпечення АСБД складається з шести пов'язаних між собою основних частин:

- сервера СБД (*MS SQL Server*);
- системи керування АСБД;
- драйверів пристроїв вводу-виводу;
- засобів контролю і тестування апаратного забезпечення;
- засобів налагодження програмного забезпечення;
- внутрішніх мікропрограм АСБД.

У якості серверу СБД пропонується використовувати сервер фірми *Microsoft - MS SQL Server*, що виконує функції, необхідні для повноцінної роботи з БД великої кількості користувачів (синтаксичного аналізатора, оптимізатора, сканування даних, компоновника вихідного потоку для АСБД, генератора лістинга помилок, що виникають у командах *SQL*-мови й у масиві даних, формування результуючої множини). Крім того, до складу сервера входять спеціальні користувальницькі функції, що забезпечують формування потоку даних для АСБД, а також виклик програм системи керування і контролю АСБД.

У СБД можливі два варіанти виклику функцій АСБД. Перший варіант припускає, що програміст знає про наявність у складі апаратного забезпечення АСБД і використовує спеціальні користувальницькі функції SQL-сервера, що у свою чергу викликають програми паралельної обробки для підвищення продуктивності сервера. Другий варіант полягає в тому, що коли на SQL-сервер надходять стандартні оператори SQL, для використання можливостей АСБД використовуються створені для DSP-процесорів драйвери Windows.

Первинною ланкою інтерфейсу, заснованого на користувальницьких функціях, є збережена процедура *xp\_cmdshell*, що дозволяє програмісту писати власні програми основними мовами програму-

вання, і викликати їх безпосередньо з SQL-запиту.

Система керування АСБД забезпечує передачу потоку даних від сервера у свою внутрішню пам'ять та запуск процесу обчислення і керування цим процесом. Програми, що входять у цю систему, формують вихідний потік оброблених даних, що потім передається на сервер СБД. Система забезпечує середовище на етапі виконання користувальницької програми, дозволяючи підпрограмам АСБД зв'язуватися з Host-комп'ютером та її файловою системою для обміну даними. Система забезпечує також контроль працездатності АСБД і діагностику компонент, з яких вона складається. Для цього використовується спеціальне тестове програмне забезпечення.

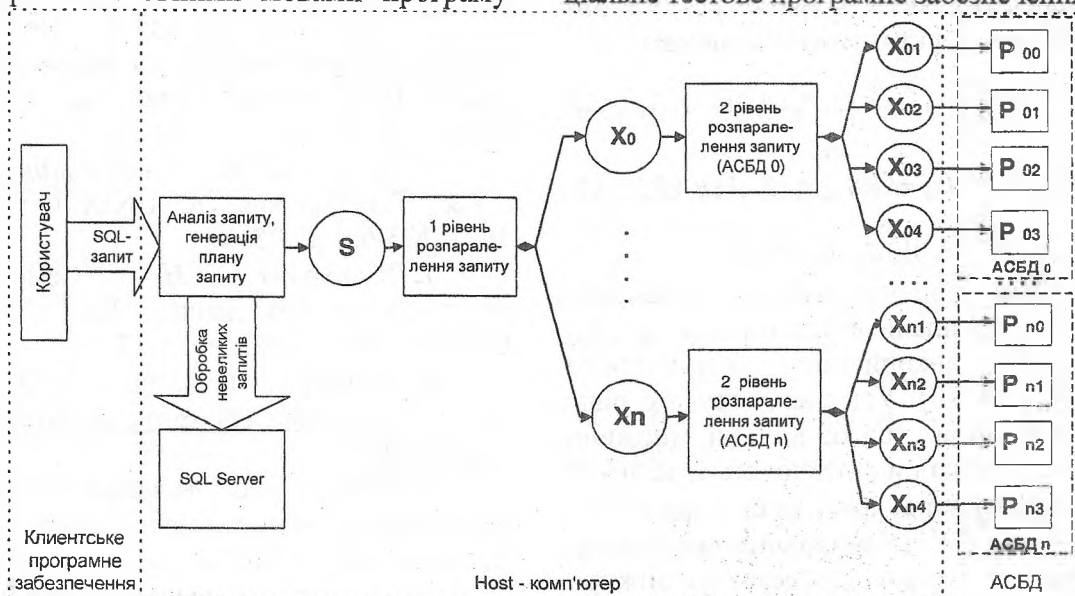


Рис. 6. Схема обробки запитів у СБД, на базі АСБД

Запропоновано принципи розбивки реляційної структури даних на багато-процесорній системі (рис. 6.). Кожне відношення реляційної таблиці поділяється на фрагменти, що розташовуються в різних ЗП (чи ділянках ЗП, що відповідають кожному процесору). Запит застосовується не до відношення в цілому, а до даних фрагментів. Кожен фрагмент обробляється на окремому процесорі й отримані результати поєднуються в загальну результуючу множини.

Цикл обробки запиту в системі, що використовує модулі АСБД, виглядає у такий спосіб. Наприклад, необхідно обчислити  $S = A \bowtie B$  - з'єднання двох відношень

$A = \{a_1, a_2, \dots, a_j, \dots, a_i\}$  і  $B = \{b_1, b_2, \dots, b_j, \dots, b_i\}$  по деякому загальному атрибуту  $Y$ . Припустимо, що у системі використовується два блоки АСБД. Для того, щоб обробити відношення  $A$ , розіб'ємо його на два фрагменти  $A_0 = \{a_1, a_2, \dots, a_j\}$  і  $A_1 = \{a_{j+1}, a_{j+2}, \dots, a_i\}$ , для яких

$A = A_0 \cup A_1, A_0 \cap A_1 = \emptyset, \pi_Y(A_0) \cap \pi_Y(A_1) = \emptyset$ , де  $\pi$  - операція проєкції. Для відношення  $B$  аналогічно розіб'ємо його на  $B_0 = \{b_1, b_2, \dots, b_j\}$  і  $B_1 = \{b_{j+1}, b_{j+2}, \dots, b_i\}$ , для яких  $B = B_0 \cup B_1, B_0 \cap B_1 = \emptyset, \pi_Y(B_0) \cap \pi_Y(B_1) = \emptyset$ .

Послідовний фізичний план запиту  $S$  перетвориться в паралельний план першого рівня, що містить у собі дві частини  $X_0$  і  $X_1$ , що виконуються на двох блоках



АСБДО і АСБД1 відповідно. План  $X_0$  на відміну від  $X_1$  має додаткову операцію *MERGE*. Операція має два вхідних потоки й один вихідний. Вхідними потоками *MERGE* є область пам'яті АСБД, асоційована з операцією *join* і канал зв'язку між АСБД, що йде від  $X_1$ . Вихідним потоком *MERGE* є стандартний потік виводу даних на *Host*-комп'ютер.

При перетворенні плану першого рівня  $X_1$  у паралельний план другого рівня множини  $A_1$  і  $B_1$  розбиваються усередині АСБД на чотири підмножини  $A_{10}=\{a_1, a_2, \dots, a_{j1}\}$ ,  $A_{11}=\{a_{j1+1}, a_{j1+2}, \dots, a_{j2}\}$ ,  $A_{12}=\{a_{j2+1}, a_{j2+2}, \dots, a_{j3}\}$ ,  $A_{13}=\{a_{j3+1}, a_{j3+2}, \dots, a_i\}$  і  $B_{10}=\{b_1, b_2, \dots, b_{j1}\}$ ,  $B_{11}=\{b_{j1+1}, b_{j1+2}, \dots, b_{j2}\}$ ,  $B_{12}=\{b_{j2+1}, b_{j2+2}, \dots, b_{j3}\}$ ,  $B_{13}=\{b_{j3+1}, b_{j3+2}, \dots, b_i\}$  по атрибуту з'єднання  $Y$ , причому  $A_1$  і  $B_1$  розбиваються на підмножини однаково.

$$A_1 = \bigcup_{n=0}^3 A_{1n}, \bigcap_{n=0}^3 A_{1n} = \emptyset, \forall_{i \neq n} \pi_Y(A_{1i}) \cap \pi_Y(A_{1n}) = \emptyset;$$

$$B_1 = \bigcup_{n=0}^3 B_{1n}, \bigcap_{n=0}^3 B_{1n} = \emptyset, \forall_{i \neq n} \pi_Y(B_{1i}) \cap \pi_Y(B_{1n}) = \emptyset;$$

$$\forall_{i \neq n} \pi_Y(A_{1i}) \cap \pi_Y(B_{1n}) = \emptyset.$$

План першого рівня  $X_1$  перетворюється в чотири плани другого рівня  $X_{10}, X_{11}, X_{12}, X_{13}$  (при використанні чотирьохпроцесорного АСБД). Усі плани другого рівня однакові, різні тільки вихідні множини. Плани другого рівня виходять із плану  $X_1$  при вставленні в нього оператора обміну *CHANGING*. У чотирьохпроцесорній конфігурації цей оператор забезпечує міжпроцесорний обмін. Для кожного оператора *CHANGING* зазначена функція розподілу, що визначає, на якому процесорі АСБД повинна бути оброблена множина.

**Висновок.** Запропоновані методи розпаралелювання запитів до БД, дозволяють досягти в БОС, що базуються на нових модулях розширення, оптимального балансування завантаження процесорів при відносно низьких витратах на міжпроцесорні комунікації. Авторами було запропоновано алгоритми, що дозволяють оптимізувати обробку трудомістких операцій сортування, з'єднання, вибірки й ін., що дозволяє значно скоротити час обробки запитів у системі. Також було проведено оцінку ефективності паралельних алгоритмів і пристроїв, що входять до

складу СБД, по часових витратах, числу процесорів і значенню прискорення.

### Список літератури

1. Жуков І.А., Іванкевич А.В., Кременецкий Г.Н. Методы организации параллельной обработки в системах больших баз данных // Проблемы информатизации та управління. – К.: НАУ, 2002. – Вип.5. – С. 99-105.
2. The BaBar Homepage (<http://www.slac.stanford.edu/BFROOT/>).
3. The Earth Observing System Project Science Office Homepage (<http://spsosun.gsfc.nasa.gov/eosinfo/Welcome/index.html>).
4. Храмов П. Информационно-поисковые системы в Internet // Открытые системы. – 1996. – №3. – С. 46-49.
5. Steve Lawrence and C. Lee Giles. Context and page analysis for improved web search. IEEE Internet Computing, 1998. – N2(4). – P. 38-46.
6. Лисянский К., Слободяников Д. СУБД Teradata для ОС UNIX // СУБД. 1997. №5-6. – С. 25-46.
7. Игнатович Н. Семейство реляционных баз данных IBM DB2 // СУБД. – 1997. – №2. – С.5-17.
8. Compaq NonStop SQL/MP. ([http://www.tandem.coni/prod\\_des/nssqlpd/nssqlpd.htm](http://www.tandem.coni/prod_des/nssqlpd/nssqlpd.htm))
9. Дейт К. Дж. Введение в системы баз данных, 7-е изд.: Пер. с англ. – М.: Вильямс, 2002. – 1072 с.
10. Жуков І.А., Іванкевич О.В. Апаратне і програмне забезпечення неоднорідних багатопроцесорних систем великих баз даних // Вісник НАУ. – К.: НАУ, 2002. – №3. – С. 132-138.
11. Жуков І.А., Іванкевич А.В. Методы и средства повышения производительности SQL сервера при работе с большими базами данных // Збірник наукових праць Інституту проблем моделювання в енергетиці НАН України ім. Г.Є.Пухова. – Вип. 17. – К.: Інститут проблем моделювання в енергетиці, 2002. – С. 43-49.
12. Жуков І.А., Іванкевич А.В. Апаратные методы организации высокоскоростной обработки больших массивов конфиденциальной информации в системах баз данных // Захист інформації. – К.: НАУ, 2004. – №3. – С. 22-30.