ДИПЛОМНА РАБОТА

НА ТЕМА

Изследване на алгоритми за съхранение и заявки за търсене в хетерогенни бази от данни.

Изготвил: .....

Фак.номер: .....

Спец.: .....

Научен ръководител:

2012, София

Съдържание:

ГЛАВА 4 ЕКСПЕРИМЕНТАЛНО ИЗСЛЕДВАНЕ НА ЕФЕКТИВНОСТТА НА МЕХАНИЗЪМ ЗА ПАРАЛЕЛНИ ЗАЯВКИ ЗА ТЪРСЕНЕ.

4.1. Изпълнението на паралелни заявки за търсене в SQL. Структурата, особености и избор на алгоритъм.

4.2. Проектиране на информационната структура на базата данни за тестване.

4.3. Резултати от експерименталните паралелни заявки.

ЗАКЛЮЧЕНИЕ

Литература

Приложения

Въведение

Проблемът за подобряване на работата на системите за управление на бази данни (СУБД) в момента е от все по-голямо значение. Увеличаване на броя потребители работещи с бази данни и увеличаване на размера на обрабваната информация изискват съответното подобрение на базите данни за изпълнение в разумен срок от време за генериране на отговор на заявката на потребителя [41]. Ефективна и икономически жизнеспособна алтернатива на еднопроцесорен СУБД се явява паралелно СУБД, работещо едновременно на множество процесори. Използването на паралелна база данни чрез съчетаване на няколко маломощни машини позволява да се получи същото ниво на изпълнение на производството, както и в случай на една, но по-мощни машина, за получаване на резултати в скалируемост и надеждност на системата в сравнение с еднопроцесор база данни [15]. В момента, бази данни се използват в почти всички области на човешките дейности, свързани със съхраняване и обработка на информация. Напредъкът постигнат в областта на технологиите на бази данни се основава до голяма степен на релационните модели, предложени от Код, Е. [57] в края на 60-те - 70-те години на XX век. По време на своята тридесет-годишна история релационните бази данни произтичат от изследване прототип, най-важните от които са System R [18, 43] и Ingres [91,89], търговски продукти, които могат да съхраняват и обработват терабайти информация. Въпреки това, научната и практическа дейност на човека поставя все по-нови предизвикателства, които изискват обработката на много големи бази данни.

Появата на много големи бази данни е в резултат на разширяването на приложенията и базите данни. Примери за нови приложения на бази данни са електроннаа търговия [72], цифровите библиотеки [14, 63], геоинформационните системи [39], мултимедийни файлове [76], научните база данни [48, 92].

Днес се наблюдава нарастващото търсене на технологии за паралелни и разпределителни бази данни и от големи и от средни компании. Друга логична стъпка в посока на информационни системи от този клас е да се създаде единна база данни, за да даде възможност за бърз достъп и анализ на данни, както и за решаване на специфични за компанията (в зависимост от сферата на дейността) въпроси [27].

В действителност, единственото известно и ефективно днес решение за съхранение и обработка на много големи бази данни е използването на паралелни

системи за бази данни [61, 47], с репликация на съхраняваната информация с оглед обезпечаване на паралелна обработка на информационните търсения в многопроцесорни системи.

Интензивната изследователска дейност в областта на паралелните база данни е започнала през 80-те години. През последните две десетилетия, паралелните системи бази данни са се преориентирали от изследователски прототипи към пълнофункционални търговски продукти, доставяни на пазара на информационни системи с висока производителност. Като примери за успешни проекти, създаване на паралелни системи бази данни могат да причислят гамата от продукти - DB2 Parallel Edition [11], NonStop SQL [37] и NCR Teraciata, [21]. Въпреки това, разходите на такава специализиран софтуер за тези системи е сравним, и в някои случаи по-високи от разходите на хардуерен компонент.

Тези системи интегрират стотици процесори и твърди дискове и състояние да се справят с десетки терабайта. Въпреки това, в паралелни системи бази данни се нуждаят от допълнителни изследвания [97, 10], едно от тях - нататъшно развитие на методи за оптимизиране на ефективността на съхраняване и търсене (заявките) в паралелни и хетерогенни бази данни. В допълнение, много изследователи правилно отбелязва ограниченията в мащаба. С голям брой на процесорите в системите от този тип започват да се появяват конфликти на достъпа до споделената памет, която може да доведе до тежко влошаване на общата производителност на системата [95]. В съответствие с това реалната мащабируемост така се наричат SMP системи с ограничен 32 процесор [98].

Както е посочено в един доклад, написан от водещи експерти в областта на системи за управление на бази данни, по посока на научните изследвания зя развитие на базите данни [46], в близко бъдеще ще има база данни, по-големи от един петабайта.

Целта на дипломната работа е да се разработят методи и алгоритми за паралелно изпълнение за SQL-заявки в хетерогенни архитектури, както и да се посочат начините за оптимизирано съхраняване на информация.

Въз основа на тези изследователски цели, нейните основни цели са:

1. Да извърши анализ на сегашните архитектури на СУБД и методите, използвани тях за паралелно изпълнение на SQL-заявки, обобщителен анализ на

сегашното състояние на научните изследвания в областта на прилагането на паралелна обработка на заявките.

2. Да се разработят ефективни методи за паралелни запитвания в хетерогенна среда с доказване на тяхната истинност.

3. Да се разработят алгоритми и софтуерни инструменти за паралелно изпълнение на SQL-заявки, за да се гарантира тяхната съвместимост със съществуващото СУБД.

4. Въз основа на тези резултати, да се въведе система за паралелно запитване и провеждане на изчислителни експерименти, с помощта на разработените методи и алгоритми.

Използваните методи са характерни за булевата алгебра, релационната алгебра, симулация, математическата статистика, теория на графите, математическия анализ, елементарен системен анализ.

ГЛАВА 1 АНАЛИЗ НА СЪВРЕМЕННИТЕ МЕТОДИ ЗА ОБРАБОТКА НА ЗАЯВКИ ЗА ТЪРСЕНЕ В SQL.

1.1. Архитектура на паралелни системи.

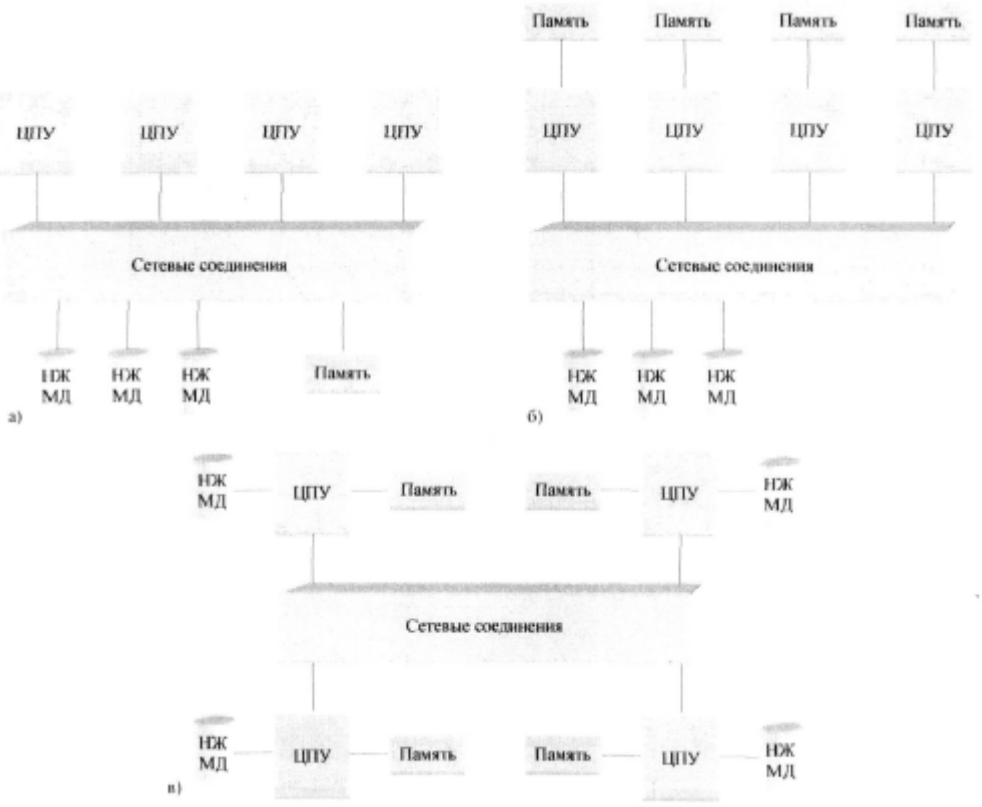
В основата на различните класове паралелни системи стоят структурни различия в подреждането на различните функционални елементи, основните от които са процесора и паметта. В съответствие с този подход са следните четири класа на мултипроцесорни системи [17, 82, 4]:

1.SMP (Symmetric многопроцесорни) - симетрична мултипроцесорна система (фиг. 1 (а));

2.. NUMA (Non-uniform memory architecture) - мулти-процесорни системи с нееднородна памет за достъп (фиг. 1 (б));

3. MPP (масивната паралелна обработка) - масово паралелни системи (фиг. 1 (в));

4. Клъстери - пълен набор от компютри, свързани в една компютърна система.



Фиг.1 Архитектура на паралелните системи

SMP архитектура се състои от малко на брой, свързани процесори с шина или комутатор с обща памет. В такива системи, времето за достъп до една споделена памет е физически еднаква за всички процесори. Ето защо, за SMP системите принадлежат към класа на UMA (Uniform Memory Architecture) [16]. Затрудненията при SMP системи е споделена връзка за достъп до паметта. Поради тази причина, броя на процесорите в една многопроцесорна система рядко надхвърля 32 [16, 98].

NUMA архитектурата се характеризира с наличието на физически разпределена памет, споделена от всички процесорни модули като едно адресно пространство [16, 82]. В същото време отделен процесорен модул може да бъде една многопроцесорна система. NUMA системните модули са свързани в мрежа. В същото време, за да се гарантира еднакъв достъп до споделената памет и синхронизация на промените в кеш процесори в модула за всеки процесор се използват специални устройство [82]. Система с NUMA архитектура може да включва в себе си до 64 процесорни модули с общо до 128 процесора. Но хетерогенността на достъпа до споделената памет води до това, че извадката на данни от паметта на друг модул ще има забавяне от 200% до 750% в различните NUMA системи [82], и тази цифра ще се увеличи с увеличаване на броя процесорните модули. Поради тази причина, ние може само да говорим за мащабността на многопроцесорни системи с NUMA архитектура. Трябва да се отбележи, че разликата във времето за достъп до локална и отдалечена кеш-памет в различни NUMA системи може да бъде в границата на 400-1500% [82]. Следователно, производителността на NUMA системите може да бъде значително намалена при приложения, изискващи често синхронизация на данните. В същото време трябва да се отбележи, че разликите в скоростта на достъп до различни части на паметта в съвременната архитектура става все по-малко видимо място [8].

При MPP архитектура, изградена като комбинация от голям брой хомогенна процесорни модули, свързани с високоскоростна мрежа [17, 19, 8]. В допълнение, всеки модул има своя собствена памет и евентуално собствени дискове. Такива системи могат да включват стотици и хиляди процесорни устройства. Ключов проблем, когато използвате MPP системи е разпределение на данни между процесорнитер модули, които ще разделят проблем на подпроблеми, (по един за всеки модул), с цел минимизиране на обменна между различни подзадачи и да се осигури балансирано натоварване на всички процесорни модули. В много случаи, това е задача от не толково

тривиален характер. Трябва да бъде отбелязано, че съвременните тенденции в развитието на хардуера доведе до де факто сливане на MPP и клъстерите [1,4].

Използване клъстер архитектура - сроден набор на пълноценни компютри като един компютри [1, 82, 33]. В общия случай системата клъстер може да включва дискови масиви, както и други специализирани устройства. Също така клъстера може да използва и еднопроцесорни системи и системи с SMP архитектура, и дори NUMA. Въпреки това, трябва да се отбележи, че за клъстери, характеризиращ се с един и същ общ проблем с разпространението на данни и балансиране на натоварването, както и при MPP системите.

Разделението на многопроцесорни системи, описани по класове функционални характеристики е доста конвенционален смисъл, че истински мулти-процесорни системи, могат да се комбинират функции, присъщи на различните класове. Тези класове на паралелни системи не включва функции, които се случват по време на съхранение, товарене и разтоварване голям обем от данни, така че на практика се изисква специална класификация на архитектурата на паралелна база данни.

1.2. Класификация на паралелни системи база данни.

Най-общата класификация на паралелна система от база данни е системата, предложена от Майкъл Стоунбрейкър [90]. Схематично, тази класификация е показана на фиг. 2 (A, B, C). С развитието на паралелни системи за обработка, бази данни, класификацията се определя като недостатъчна [79], поради появата на мулти- системи, като едновременно съчетава функциите на няколко архитектури [44]. За обозначаването на такива системи е [58] предложено разширяване на класификацията чрез въвеждане от Стоунбрейкър на два допълнителни класа на архитектури на паралелни машини за бази данни (фиг. 2 (г, д)).

В съответствие с класификацията на паралелните системи на Стоунбрейкър, бази данни могат да бъдат разделени в следните пет основни класове в в зависимост от метода на разделяне на хардуерните ресурси (първоначално класифицирани Стоунбрейкър включени първите три класа системи):

1. SE (Shared-Everything) - архитектура с обща памет и дисково пространство (Фиг. 2 (а));

2. SD (shared disdk) - споделената-диск архитектура (фиг. 2 (б));

3. SN (Shared-nothing) - архитектура, без споделяне на ресурсите (Фиг. 2 (в));

4. CE (Clustered-everything) - SE клъстер архитектура, съчетана въз основа на SN (Фиг. 2 (г))

5. CD (Clustered-disk) - SD архитектура с клъстери, комбинирани на принципа на SN (Фиг. 2 (д)).

SE представлява архитектурата на системите бази данни, в която всички дискове са пряко достъпни за всички процесори с еднакво време за достъп, както и всички процесори споделят обща памет. Междупроцесорната комуникация в системите на SE се осъществява чрез обща памет. Достъпът до дискове в SE системи обикновено се прави чрез общ буферен пул. Трябва да се отбележи, че всеки процесор в системата на SE има своя собствена кеш памет.

Има голям брой паралелни системи за бази данни с SE архитектура. По същество всички водещите търговски бази данни вече имат изпълнението на SE-базирана архитектура. Като един от първите примери преход от един процесор система на SE архитектура може да доведе изпълнението на DB2 на IBM3090 с 6 процесора [56]. Трябва да се отбележи, обаче, че огромното множество на търговски системи SE използва само междутранзакционен паралелизъм (т.е. вътрешнотранзакционен паралелизъм отсъства). Въпреки това, към днешна дата, са създадени няколко изследователския прототипа на SE система, която използва вътрепроцесорен паралелизъм.