

ПРИНЦИПИ ПРОЕКТУВАННЯ ТА ОСОБЛИВОСТІ СИСТЕМ ОБРОБКИ НАДВЕЛИКИХ ОБ'ЄМІВ ДАНИХ

У статті проведено аналіз принципів проектування та основні особливості систем для обробки надвеликих об'ємів даних. Проведений аналіз орієнтований на системи обробки надвеликих об'ємів даних, а саме, на область, яка називається OLAP (Online Analytical Processing) – аналітична обробка в реальному часі, на відміну від оперативної обробки транзакцій – OLTP (Online Transaction Processing). В статті обговорюються вимоги до аналітики даних, що розвивалися в період ранніх робіт з паралельними системами баз даних. Розглянуті основні технологічні інновації, кожна з яких породила окрему категорію систем для обробки та аналізу надвеликих об'ємів даних. Проведена класифікація ряду аспектів, по яких можна оцінити і порівняти кожен з категорій систем для обробки та аналізу надвеликих об'ємів даних.

Ключові слова: надвеликі об'єми даних, колонкові бази даних, OLTP, OLAP, MapReduce, Dataflow.

Вступ. В даний час наукова і практична діяльність людини висуває все нові масштабні завдання, що вимагають обробки надвеликих об'ємів даних. Згідно з прогнозами аналітичної компанії IDC до 2020р. кількість даних у світі досягне 40 зеттабайт. У зв'язку з появою задач, що вимагають обробки надвеликих баз даних, необхідні нові ефективні методи паралельної обробки та аналізу таких обсягів даних на багатопроцесорних обчислювальних системах. Фактично єдиним ефективним вирішенням проблеми зберігання і обробки надвеликих об'ємів даних є використання паралельних систем баз даних, що забезпечують паралельну обробку запитів на багатопроцесорних обчислювальних системах.

Великі компанії завжди відчували необхідність у вирішенні задач аналізу даних, які перетворюють великі обсяги вихідних даних в інформацію потрібну для своєчасного прийняття рішень. Паралельні бази даних, наприклад, «Gamma» [1] і «Teradata» [2] були одними з перших систем для вирішення цієї проблеми. За останні десять років, з'явилося все більше і більше джерел великих наборів даних - Big Data (великі дані). Можна виділити багато джерел, які генерують величезні обсяги даних серед яких мережевий серфінг, соціальні медіа, наукові експерименти, і моніторинг центрів обробки даних.

Багато інновацій та покращень в продуктивності потребують в своєчасному і економічно ефективному аналізі великих обсягів даних. За останнє десятиліття ця потреба привела до появи нових значних інновацій в системах аналітики надвеликих об'ємів даних. Паралельні бази даних додали такі методи, як колонкове зберігання і обробка даних [3]. Одночасно з цим були розроблені нові розподілені масиви даних та обчислювальні системи, такі як MapReduce [4] і Bigtable [3].

Постановка задачі. Паралельні бази даних, які можна віднести до категорії класичних систем, були першими системами для паралельної обробки даних, доступних для широкого класу користувачів за допомогою інтуїтивно зрозумілої моделі програмування високого рівня. Паралельні бази даних були засновані переважно на реляційній моделі даних. Декларативний SQL-був використаний в якості мови запитів для вираження завдань обробки даних, що зберігаються у вигляді записів таблиці.

Паралельні бази даних досягли високої продуктивності і масштабованості секціонування таблиць по вузлах в "shared-nothing cluster" (розподілена обчислювальна архітектура), в якій кожен вузол є незалежним і самодостатнім, і немає жодної точки неузгодження у всій системі. Зокрема, жоден з вузлів не розділяє пам'ять або дисковий простір. Така горизонтальна схема розбиття включає реляційні операції, такі як фільтри

з'єднання і агрегування для паралельного запуску з різних розділів якоїсь таблиці збереженої на різних вузлах.

Три тенденції, що викликали питання про перевагу класичних паралельних баз даних:

– Все більше і більше компаній почали зберігати стільки інформації, скільки могли зібрати. Класичні паралельні бази даних почали створювати серйозні перешкоди в плані масштабованості і сукупної вартості володіння, як і виникнення необхідності обробки цих постійно зростаючих обсягів даних.

– Дані що надходять і зберігаються в компанії були різноманітні по своїй структурі. Наприклад, стало звичайною практикою для збору добре структурованих даних, таких як дані про продажі і користувацьку демографію поряд з менш структурованими даними, таких як журнали пошукових запитів і веб-вміст сторінок. Було важко підігнати такі різноманітні дані до жорсткої моделі даних, що підтримуються класичними паралельними базами даних.

– Потреби бізнесу почали вимагати більш короткі інтервали між часом, коли будуть зібрані дані (як правило, в OLTP (англ. Online Transaction Processing) – онлайнна обробка транзакцій) і час, коли результати аналізу даних були б доступні для ручного або алгоритмічного прийняття рішень.

Виклад основного матеріалу досліджень

Розглянуті вище тенденції стимулювали два типи інновацій:

а) інновації, спрямовані на рішення недоліків класичних паралельних баз даних при збереженні їх переваг, таких як висока продуктивність і декларативний запит мови;

б) інновації, спрямовані на створення альтернативної архітектури системи, що може підтримати вище описані тенденції більш економічно ефективним чином.

Ці нововведення, в поєднанні з категорією класичної паралельної системи управління базами даних, дають наступні унікальні категорії систем для аналізу великомасштабних даних (рис.1а):

1. Категорія колонкових систем. Колонкові системи були піонерами нової концепції зберігання даних у таблицях, розташовуючи всі стовпці разом, замість розташування по рядкам, як це зроблено в класичних паралельних базах даних. Системи з колонкоподібним зберіганням та обробкою даних, такі як Vertica [1], показали використання процесора, пам'яті і ресурсів введення/виведення більш ефективно у великомасштабному аналізі даних в порівнянні з рядково-орієнтованими системами. Деякі з основних переваг отримуються від зменшення операцій введення/виводу в колонкових системах, читаючи тільки необхідні стовпці при обробці запиту.

2. Категорія MapReduce систем. MapReduce – це програмна модель та програмний каркас, що її реалізує, розроблені компанією Google для обробки величезних масивів даних шляхом використання дуже великого скупчення кластерів зі звичайних вузлів [3]. Системи в класичній категорії традиційно намагалися масштабуватись до таких рівнів. MapReduce системи стали піонером концепції множино автономної побудови масштабованих розподілених систем, і можливості компонувати дві або більше з цих систем разом для того, щоб запустити аналітичні завдання по вибірці з великих наборів даних. Популярні системи у цій категорії, такі як Hadoop [4], зберігають дані в автономну блок-орієнтовану розподілену файлово-систему і запускають обчислювальні завдання в іншій розподіленій системі, яка підтримує програмну модель MapReduce.

3. Категорія систем потоків даних (Dataflow). Деякі недоліки в MapReduce були визначені тим, як ці системи використовувались для завдань аналізу великих наборів даних. Програмна модель MapReduce є занадто обмеженою щоб легко виразити певні завдання аналізу даних, наприклад, з'єднання двох наборів даних разом. Що ще більш важливо, методика виконання яка використовується в MapReduce системах неоптимальна для багатьох поширених типів завдань аналізу даних, таких як реляційних операцій, ітераційного машинного навчання та опрацювання графа. Більшість з цих проблем можуть бути вирішені шляхом заміни MapReduce на більш гнучку модель виконання на основі потоків даних, яка може виражати широкий спектр доступу до даних і комунікаційних моделей. Різні моделі

виконання, основані на потоках даних, були використані в системах цієї категорії, в тому числі спрямовані ациклічні графи в «Dyad», дерева обслуговування в «Dremel», і масова синхронно паралельна обробка в «Pregel».

Стало зрозуміло, що з плином часу нові системи можуть бути побудовані шляхом об'єднання принципів проектування з різних категорій систем. Наприклад, методи, використовувані для високоефективної обробки в класичних паралельних базах даних можуть бути використані разом з методами що використовуються для дрібномодульної відмовостійкості в MapReduce системах [3]. Кожна система в цій об'єднаній категорії надає єдиний інтерфейс системи, що забезпечує загальний набір функцій, які традиційно асоціюються з різними категоріями системи.

Необхідність скоротити розрив між генерацією даних і генерацією результатів аналітики цих даних потребувало системних розробників постійно підвищувати планку в великомасштабних аналізах даних. З одного боку, ця потреба спонукала появу масштабованих розподілених систем зберігання, які забезпечують різні ступені транзакційних можливостей. Підтримка для транзакцій дозволяє цим системам служити, як сервіс сховища даних онлайн, а ці дані робить доступними одночасно в тих самих системах для аналітики. Ця потреба призвела до появи паралельних систем управління базами даних, які підтримують як OLTP і OLAP в одній системі. Ми ставимо обидва типи систем в категорію названу змішаними системами через їх здатності виконувати ефективно змішані робочі навантаження, такі як містять транзакційні завдання, а також завдання аналітики.

Можна виділити вісім ключових особливостей систем, які описують основні характеристики систем для аналітики великомасштабних даних (рис. 1б):

1. Модель даних і інтерфейси. Модель даних забезпечує чіткість і логічну структуру даних, і визначає в якому форматі дані можуть бути збережені, організовані і використовуватись системою. Найбільш популярний приклад моделі даних є реляційна модель (яка використовує формат таблиці в своїй основі), в той час як більшість систем в MapReduce і потоках даних дозволяють щоб дані були в будь-якій довільній формі, що зберігається в одноманітних файлах. Модель даних, використовувана кожною з систем тісно пов'язана з інтерфейсом запиту, доступного в системі, що дозволяє користувачам управляти і маніпулювати збереженими даними.

2. Storage Layer (рівні збереження). На високому рівні, Storage Layer просто відповідає за збереження даних, а також надання методів для доступу й зміни даних. Тим не менш, розробка, впровадження та особливості умов використання рівнів збереження, в різних категоріях систем сильно розрізняються, особливо коли ми починаємо порівняння систем з різних категорій. Наприклад, класичні паралельні бази даних використовують інтегровані і спеціалізовані сховища даних, які тісно взаємопов'язані з їх механізмом виконання, в той час як системи MapReduce зазвичай використовують незалежні розподілені файлові системи для доступу до даних.

3. Механізм виконання. Коли система отримує запит на виконання, вона, як правило, перетворює його в схему виконання для доступу й обробки вхідних даних запиту. Механізм виконання несе відповідальність за те як фактично працює дана схема виконання в системі і генеруються результати запиту. У системах, які ми обговорюємо, виконавчий механізм також несе відповідальність за розпаралелювання обчислень по великомасштабних кластерах машин, обробки відмов машини, і створення міжмашинного зв'язку для ефективного використання дискової пропускної здатності та мережі.

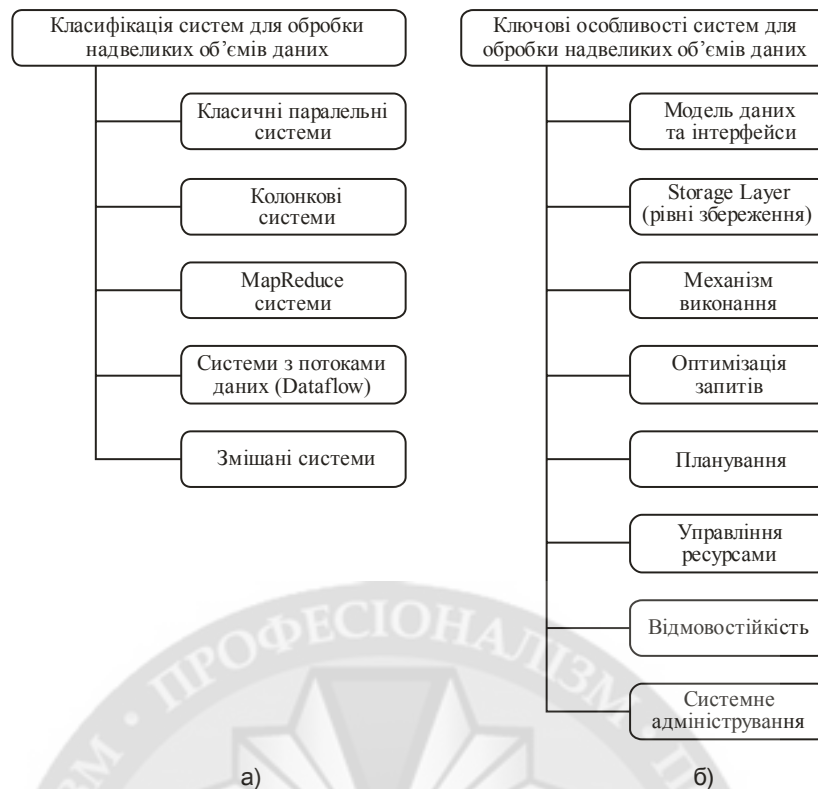


Рис. 1. Категорії систем: а) категорії систем для обробки надвеликих об'ємів даних; б) ключові особливості систем обробки надвеликих об'ємів даних

4. Оптимізація запитів. В цілому, оптимізація запитів процес який система використовує для визначення найбільш ефективнішого способу виконання даного запиту з урахуванням кількох альтернативних еквівалентних, схем виконання. Методи, використовувані для оптимізації запитів в системах, ми розглядаємо з різних точок зору: (I) простір можливих схем виконання (наприклад, реляційні оператори в базах даних у порівнянні з настройками параметрів конфігурації в MapReduce систем), (II) тип оптимізації запитів (наприклад, на основі витрат у порівнянні з основаних на правилах), (III) типу технік моделювання витрат (наприклад, аналітичні моделі в порівнянні з моделями навченими за допомогою методів машинного навчання) і (IV) зрілість методів оптимізації (наприклад, повністю автоматизована проти ручного налаштування).

5. Планування. Враховуючи розподілений характер більшості систем аналізу даних, планування виконання схеми запиту є важливою частиною системи. В даний час системи повинні приймати кілька рішень планування, в тому числі планування, де запустити кожне обчислення, планування передачі даних між вузлами, а також планування поновлення виконання та роботи з технічного обслуговування.

6. Управління ресурсами. Управління ресурсами в першу чергу відноситься до ефективного і результативного використання ресурсів кластера на основі вимог до ресурсів запитів додатків, запущених в системі. Крім того, багато систем сьогодні пропонують гнучкі властивості, які дозволяють користувачам динамічно додавати або видаляти ресурси, як необхідно відповідно до вимог робочого навантаження.

7. Відмовостійкість. Апаратні збої є досить поширеним явищем у великих кластерах. Таким чином, більшість систем мають вбудовані функціональні можливості відмовостійкості, що дозволить їм продовжувати надання послуг, можливо, з поступовим погіршенням, на фоні небажаних подій, таких як апаратних збоїв, помилки в програмному забезпеченні, а також псування даних. Приклади типових особливостей допустимих збоїв, включають перезапуск невдалих завдань, як через прикладні або апаратні збої, відновлення даних в результаті збою машини або псування, і використання спекулятивного виконання, щоб уникнути відсталості.

8. Системне адміністрування. Системне адміністрування відноситься до діяльності де додаткові зусилля людини можуть бути необхідні для підтримки системи працюючою, система обслуговує потреби різних користувачів і додатків. Загальні заходи по системному адмініструванню включають моніторинг та налаштування, діагностика причин поганого виконання або невдачі, планування потужностей, і відновлення системи при безповоротних збогах (наприклад, відмови диска) або стихійного лиха.

Саме ці ключові особливості допоможуть виявити найбільш вузькі місця для модифікації систем для обробки надвеликих об'ємів даних. Удосконалення в певній системі хоча б одної з цих особливостей, може призвести до відчутного збільшення продуктивності при роботі з великими об'ємами даних та вирішення ряду проблем, які існують на цей час.

Основні проблеми при аналізі надвеликих об'ємів даних сьогодні виявляються з чистого обсягу доступних даних для обробки. Обсяги даних, які багато компаній хочуть обробити вчасно і економічно ефективними способами, постійно росли від діапазону мультігігабайта до терабайт і тепер до багатьох петабайт. Дану проблему контакту з надвеликими об'ємами даних назвали проблемою обсягу. Існує дві інших пов'язаних проблеми, а саме, проблеми зі швидкістю обробки та різноманітністю даних.

До проблеми швидкості відносяться вимоги до часу відгуку для збору, зберігання і обробки даних. Більшість систем, є пакетними системами. Для чутливих до затримок додатків, таких як ідентифікація потенційного шахрайства та рекомендація персоналізованого контенту, пакетної обробки даних недостатньо. Дані, повинні бути оброблені, під час передачі їх потоком в систему для вилучення максимальної користі з даних. Наразі існує потреба в збільшенні швидкості отримання результатів запиту.

Проблема різноманітності відображається на зростаючому списку типів даних - реляційного, часового ряду, тексту, графіків, аудіо, відео, зображень, генетичних кодів, а також зростаючого списку аналітичних методів до таких даних. Нові ідеї знаходяться при аналізі більше ніж одного з цих типів даних разом. Методи зберігання і обробки переважно націлені на обробку даних, які можуть бути представлені за допомогою реляційної моделі (рядки і стовпці) і оброблені операторами типу запитів як фільтри, з'єднання і агрегації. Однак нові типи даних, що з'являються, не можуть бути легко отримані в реляційній моделі даних або легко проаналізовані програмним забезпеченням, що залежать від роботи операторів як фільтрів, з'єднання і агрегації. Замість цього для нових типів даних потрібно безліч аналітичних методів, таких як лінійна алгебра, статистичне машинне навчання, текстовий пошук, обробка сигналів, обробка природної мови та ітеративна обробка графіків.

Ці проблеми формують нові тенденції досліджень в обробці даних з масовим паралелізмом. Потреба зменшити розрив між генерацією даних і генерацією результатів аналітики за цими даними привела до систем, які можуть підтримувати і OLTP і робочі навантаження OLAP в єдиній системі. Масштабовані розподілені системи зберігання, що забезпечують різні ступені транзакційних можливостей, вже розробляються. Підтримка транзакцій дозволяє цим системам служити сховищем даних для онлайн-служб при створенні доступних даних одночасно в тій же системі для аналітики. Найбільш видатним прикладом тут є система Google Bigtable, яка є розподіленою, що має версію, яка орієнтована на колонкову систему, що зберігає багатовимірні і сортовані набори даних [3]. Bigtable забезпечує атомарність на рівні окремих кортежів.

Традиційно, паралельні бази даних використовували різні системи для підтримки OLTP і OLAP [4, 3]. Робочі навантаження OLTP характеризуються з'єднанням запитів на читання і записів до кількох кортежів за один раз, зазвичай через індексні структури як B-дерева. Робочі навантаження OLAP характеризуються масовими оновленнями і великими послідовними скануваннями, однак тільки для читання декількох стовпців за один раз. Однак більш нові робочі навантаження баз даних все більш і більш є поєднанням традиційного OLTP і робочих навантажень OLAP. Наприклад, "available-to-promise" додатки, вимагають запитів стилю OLTP при агрегації рівнів запасів в режимі реального часу за допомогою запитів стилю OLAP, щоб визначити, чи може замовлення бути виконано.

Висновки. На основі проведеного аналізу принципів побудови та особливостей систем обробки надвеликих об'ємів даних приведена класифікація ряд аспектів за якими можна оцінити та порівняти кожен з категорій систем для аналізу даних. Саме в цих аспектах можна виявити найбільш проблемні місця систем для обробки надвеликих об'ємів даних. В свою чергу удосконалення хоча б одного з цих ключових особливостей, може призвести до відчутного збільшення продуктивності при роботі з великими об'ємами даних та вирішення ряду проблем, які існують на даний час.

Виявлений ряд проблем, породжених сучасними вимогами до аналізу даних та стрімким збільшенням об'ємів самих даних, допоможе зорієнтуватися в якому саме напрямку на даний момент найбільш затребувано удосконалення систем аналізу надвеликих об'ємів даних.

ЛІТЕРАТУРА:

1. David J DeWitt. The Gamma Database Machine Project. IEEE Trans. on Knowledge and Data Engineering. Shahram Ghandeharizadeh, Donovan A. Schneider, Allan Bricker, H-I Hsiao, Rick Rasmussen. 2(1):44–62, 1990.
2. Виктор Майер-Шунбергер. Большие данные. Революция, которая изменит то как мы живем, работаем и мыслим. / Виктор М., Кеннет К. – М.:Манн, Иванов и Фербер, 2012.
3. Andrew Lamb, Matt Fuller, Ramakrishna Varadarajan, - The Vertica Analytic Database: C-store 7 Years Later. Proc. of the VLDB Endowment, 5(12):1790–1801, 2012.
4. Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 51(1):107–113, 2008.

Рецензент: д.т.н., проф. Сбітнев А.І., провідний науковий співробітник науково-дослідного центру Військового інституту Київського національного університету імені Тараса Шевченка

д.т.н., доц. Джулий В.Н., к.т.н. Ленков Е.С., Копачовец А.М.

ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ И ОСОБЕННОСТИ СИСТЕМ ОБРАБОТКИ СВЕРХБОЛЬШИХ ОБЪЕМОВ ДАННЫХ

В статье проведен анализ принципов проектирования и основные особенности систем для обработки сверхбольших объемов данных. Проведенный анализ ориентирован на системы обработки сверхбольших объемов данных, а именно, на область, которая называется OLAP (Online Analytical Processing) - аналитическая обработка в реальном времени, в отличие от оперативной обработки транзакций - OLTP (Online Transaction Processing). В статье обсуждаются требования к аналитике данных, которые развивались в период ранних работ с параллельными системами баз данных. Рассмотрены основные технологические инновации, каждая из которых породила отдельную категорию систем для обработки и анализа сверхбольших объемов данных. Проведена классификация ряда аспектов, по которым можно оценить и сравнить каждую из категорий систем для обработки и анализа сверхбольших объемов данных.

Ключевые слова: сверхбольшие объемы данных, колонку базы данных, OLTP, OLAP, MapReduce, Dataflow.

Ph.D. Julie V.M., Ph.D. Lenkov E.S., Kopachovets A.M.

DESIGN PRINCIPLES AND CHARACTERISTICS OF HANDLING SUPER LARGE AMOUNTS OF DATA

The article analyzes the principles of design and key features of systems to handle extra-large amounts of data. The analysis focused on the processing system Extra Large amounts of data, namely, the area called OLAP (Online Analytical Processing) - analytical processing in real time, as opposed to operational transaction processing - OLTP (Online Transaction Processing). The paper discusses the requirements for data analysts that developed during the early work on parallel database systems. Rozlyanuti major technological innovations, each of which bore a category of systems for the processing and analysis of super large amounts of data. The classification of a number of aspects, which can assess and compare each of the categories of systems for the processing and analysis of super large amounts of data.

Keywords: very large volumes of data, core database, OLTP, OLAP, MapReduce, Dataflow.