$C_к$ – теплоємність теплоносія, кДж/(кг·К);

Δt – різниця температур, ^{0}С

Висновки:

Застосування опалювальних систем із тепловими насосами – один із перспективних і ефективних способів енергозбереження.

Теплові насоси значно ефективніші та економічніші, ніж звичайне опалення на газі та вугіллі. Тому, застосування теплових насосів – це заощадження невідновлювальних джерел енергії і захист навколишнього середовища, у тому числі, шляхом скорочення викидів CO_2 в атмосферу.

Список літератури

1. Бродянский В.М. Эксергетический метод термодинамического анализа / В.М. Бродянский. – М.: Энергия, 1973. – 296 с.
2. Соколов Е.Я., Бродянский В.М. Энергетические основы трансформации тепла и процессов охлаждения. – М.: Энергиздат, 1981. – 320 с.
3. Босий М.В. Перспективи застосування відновлювальних джерел енергії у системах теплопостачання / М.В. Босий // Наукові записки. – Кіровоград: КНТУ, 2014. - № 15. – С.89-92.
4. Трубаев П.А. Тепловые насосы: / П.А. Трубаев, Б.М. Гришко – Белгород: БГТУ. – 2009. – 142 с.
5. Рей Д.А. Тепловые наосы: пер. с англ. / Д.А. Рей, Д. Макмайки. – М.: Энергоиздат, 1982. – 220 с.
6. Боровков В.М. Энергосберегающие теплонасосные системы теплоснабжения / В.М. Боровков // Известия высших учебных заведений. Проблемы энергетики. – 2007. - № 12. – С. 42-47.

Одержано 28.04.15

УДК 004.93

І.О. Мастерний, студ. ФПМ
Національний технічний університет України «Київський політехнічний інститут»
Л.В. Рибакова, доц.
Кіровоградський національний технічний університет

Організація архітектури систем розподілених обчислень для обробки надвеликих об’ємів даних

В статті описується дослідження основних проблем сучасних систем розподілених обчислень. Наводиться модель такої системи з певними характеристиками, які пропонується задовільнити шляхом побудови трискладової архітектури і показана необхідність застосування такої архітектури.

системи розподілених обчислень, BIG DATA, високопродуктивні системи, архітектура розподілених систем

За останні десятиліття IT технології розвинулися і продовжують розвиватися з величезною швидкістю. За такого стрімкого розвитку постають все нові і нові проблеми обробки даних. Якщо найпотужніший комп’ютер 1951-го року – UNIVAC I (акронім від англ. *UNIVersal Automatic Computer I*) мав змогу обробляти сотні записів за

хвилину, то зараз потреби, як бізнесу так і наукових систем, – мільйони записів за секунду. Оскільки потужність, навіть сучасних процесорів, недостатня для такої швидкості обробки даних, для вирішення подібних задач створюються системи розподілених обчислень. Декілька років назад високопродуктивні системи об'єднували десятки серверів, мали час відгуку, що вимірювався в секундах, могли бути офлайн декілька годин та обробляли гігабайти даних. Сьогодні такі системи об'єднують тисячі мультиядерних процесорів, працюючи 24 години 7 днів на тиждень, а їх користувачі очікують не більше ніж мілісекундні відгуки.

Метою дослідження є аналіз основних проблем сучасних розподілених систем та моделі трискладової архітектури для побудови аналітичної системи, що оперує надвеликими об'ємами даних.

Об'єктом дослідження є системи обробки надвеликих об'ємів даних – Big Data (з англ. великі дані). Власне «великі дані» в таких системах використовують для прогнозування і ефективність цих систем досягається саме через надходження в систему надвеликих об'ємів даних, за рахунок яких будуються результати прогнозування. В наш час за рахунок аналізу надвеликих об'ємів даних стали можливими навіть такі амбітні проекти, як ancestry.com, що намагається побудувати сімейну історію людства базуючись на всіх доступних типах даних сьогодення (починаючи від рукописів в облікових книгах закінчуючи аналізами ДНК). Цьому порталу вдалося зібрати 5 мільярдів (!) профілей користувачів, що жили в різних історичних епохах і 45 мільйонів геніалогічних дерев з описами сімейних зв'язків [1,2].

Іншим прикладом використання надвеликих об'ємів даних є проект eHarmony. Це сайт знайомств, на якому зареєстровано 40 мільйонів користувачів з анкетами, що підтримують 1000 різних записів. Кожен день система робить 100 мільйонів припущень про сумісність двох користувачів. Такі припущення базуються не тільки на тривіальному пошуку відповідностей між заповненими полями користувачів. Наприклад завдяки дослідженням eHarmony, виявилось що площа обличчя на фотографії може впливати на ймовірність контакту між користувачами [2,3].

Піонерами технологій Big Data були компанії Google (з їх розробками парадигми MapReduce та розподілених файлових систем), Amazon, що створив інноваційне розподілене сховище типу «ключ-значення» - Dynamo. В свою чергу Open Source (з англ. відкритий вихідний код) спільнота розвинула починання Google та Amazon проектами: Hadoop, Hbase, MongoDB, Cassandra, RabbitMQ і багатьма іншими. Сьогодні жодна з провідних IT компаній не може обійтись без застосування технологій Big Data.

Предметом дослідження є модель розподіленої BigData системи з наступними характеристиками:

1. Відмовостійкість – здатність системи не зупинятися і адекватно діяти у випадку виникнення помилок в деяких її частинах. При цьому можливе падіння якості експлуатації має відбуватися пропорційно до серйозності помилки. Відновлення після помилок можна характеризувати як прогін або відкат. Коли система визначила, що сталася помилка, відновлення прогоном виправляє поточний стан системи до можливості продовження роботи. Відкочування повертає стан системи до деякої попередньої, "правильної" версії, наприклад, із використанням контрольних точок, продовжуючи виконання звідти. Відкочування вимагає, щоб операції між контрольними точками і станом із виявленою помилкою могли бути виконані ідемпотентно.

2. Оперативність – система реагує вчасно, якщо це взагалі можливо. Оперативність означає, що проблеми можуть бути виявлені швидко і ефективно вирішені. Оперативна система зосереджена на наданні швидких відгуків за деякий сталий час, тобто за деякою надійною верхньою межею. Така послідовна поведінка, в

свою чергу спрощує обробку помилок, зміцнює довіру кінцевих користувачів системи і закликає до подальшої взаємодії.

3. Еластичність. Система продовжує працювати при будь-яких навантаженнях. Система може реагувати на зміни навантаження шляхом збільшення або зменшення обсягу ресурсів, що виділяються для її обслуговування (горизонтальне масштабування системи). Це також означає, що система має бути побудована без, так званих, точок конкурування, або «вузьких місць», має мати змогу певним чином розподіляти дані між усіма її ресурсами. Еластична система повинна підтримувати прогнозуючі алгоритми горизонтального масштабування, використовуючи для цього відповідні показники продуктивності її роботи в реальному часі.

Задачею дослідження є розробка системи аналізу інформації, що надходить із соціальних мереж, має задовільняти вище наведеним характеристикам і бути здатною обробляти надвеликі об’єми даних.

На рисунку 1 представлена стандартна архітектура такої системи, де ГПЗРБД – геопозиційно розподілена база даних [4].

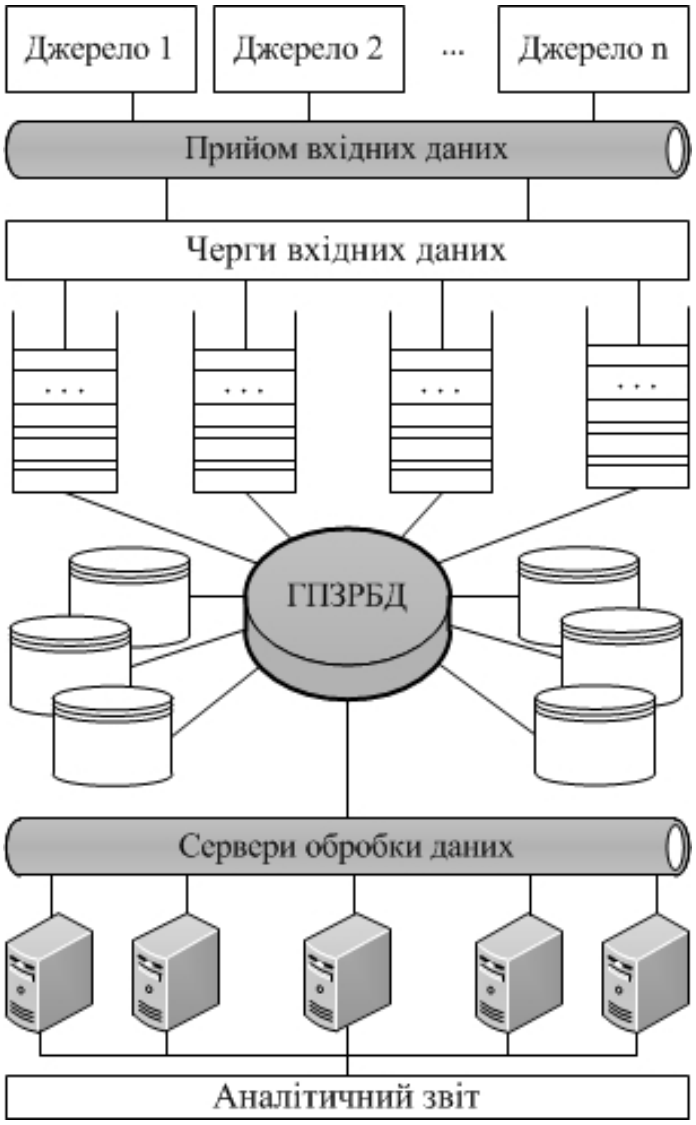

Рисунок 1 – Архітектура аналітичної розподіленної системи

Така архітектура певний час була досить дієвою, але при суттєвому збільшенні обсягів оброблюваних даних в ній відразу виникнуть труднощі із зберіганням інформації [5]. Алгоритми розробки розподілених БД за геолокаційним принципом лише частково вирішують проблему зберігання, породжуючи додаткову складність в її адмініструванні, а також вони значно знижують відмовостійкість системи. До проблем відмовостійкості можна віднести і програмні помилки, що призводять, як правило, до невідворотнього пошкодження інформації в БД. Якщо програміст зробить помилку при написанні сентиментного аналізу весь потік інформації за деякий час буде зіпсовано, що безумовно вплине на результати загального аналізу (за весь час роботи системи). Беручи до уваги те, що ми маємо справу з гігабайтами даних за тиждень, а аналіз даних не зупиняється ні на мить, в такій системі ми не зможемо своєчасно робити резервні копії даних.

Для вирішення вище наведених задач, по-перше, пропонується використовувати нереляційні моделі зберігання даних, які будуються відповідно до вимог аналітики, тобто запитів що робляться до БД, та накладають менше обмежень на структуру даних. Накладання меншої кількості обмежень дає нам змогу зберігати сировинні дані (англ. Raw data), тобто дані в їх початковому стані [5]. Вся інформація, що зберігається нашою системою, має велику цінність для проведення аналізу, зберігаючи дані незмінними ми отримаємо можливість повторно обробити їх для задоволення інших аналітичних потреб.

Продемонструємо важливість збереження незмінюваних даних на прикладі. Припустимо, користувач соціальної мережі через певний час змінює своє місце проживання, роблячи відповідні зміни у користувацькому інтерфейсі програми. Аналітична система не повинна змінювати місце проживання користувача, оскільки тоді вона втратить попередню інформацію, тому вона повинна додати нове місце знаходження і дату його зміни у профіль користувача. З цього прикладу випливає інша вимога до аналітичної системи – це модель даних, що базується на фактах. Для побудови такої моделі необхідно наступне:

- кожен запис до БД є окремим фактом, що має час виникнення (час коли користувач змінив місце проживання, тощо);
- кожний факт не може бути поділений на дрібніші факти.

Роботу запропонованої системи, розглянемо на прикладі зберігання даних, отриманих із соціальної мережі Twitter. Для забезпечення кращої інформативності системи, необхідно зберігати не тільки загальну кількість цитувань повідомлення окремого користувача, а також кожну вагому її зміну (наприклад кількість цитувань збільшилася на 100), що дає змогу побудувати звіт, в якому користувач зможе побачити залежність зростання цитування повідомлення від інших подій за цей день.

Серед інших принципів побудови аналітичної системи слід зазначити, що програмне забезпечення для БД повинне просто масштабуватися (при збільшенні об’ємів даних знадобляться додаткові ресурси для їх зберігання). Програмне забезпечення має ефективно і гнучко поєднувати операції швидкого запису та параллельного зчитування даних [6, 7].

Разом з вище наведеними вдосконаленнями підсистеми зберігання даних, до архітектури системи слід додати наступні підсистеми: сервісна підсистема та підсистема швидкого аналізу. В результаті отримаємо наступну архітектуру (рис. 2).

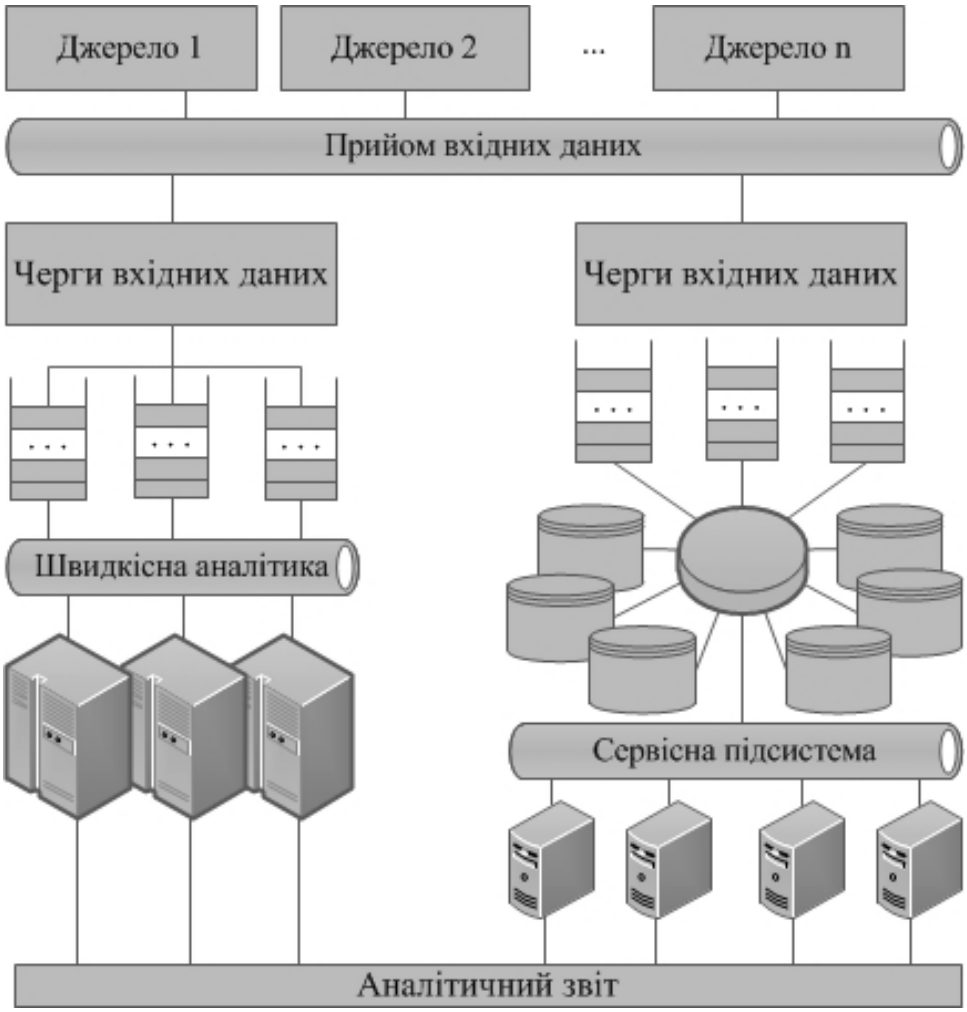

Рисунок 2 – Трискладова архітектура розподіленної аналітичної системи

Сервісна підсистема буде виконувати функції індексації сировинних даних, що зберігаються в БД. Оскільки сировинні дані поступають безперервно і в надвеликих об’ємах, а операція вставки даних для вже створених індексів є досить довготривалою, тому доцільно постійно проводити переіндексацію всіх сировинних даних по запланованому графіку в залежності від тривалості самого процесу переіндексації. Постійна переіндексація БД створює проблему застарілих даних, оскільки, як було зазначено, сам процес побудови індексів може займати декілька годин, а за цей час в систему потрапить багато нових даних, що призведе до того, що аналітичний звіт буде некоректним. Для вирішення цієї проблеми розглянемо підсистему швидкої аналітики.

Як ми бачимо з рисунку 2, вхідні дані зберігаються в сховищі БД, а також ці ж самі дані надходять до підсистеми швидкої аналітики. Аналітичні звіти, що складає система, використовують обидва джерела надходження інформації (один із проіндексованої БД, інший – із підсистеми швидкої аналітики). Завдяки підсистемі швидкісної аналітики ми маємо змогу обробити дані, що надходять під час проведення індексації БД. В результаті, для отримання кінцевого анлітичного звіту, інформація із проіндексованих таблиць БД поєднується з обробленими даними системи швидкісного аналізу. Таке поєднання швидкісної обробки даних з історичними даними БД надає переваги і у відмовостійкості всієї системи, так як будь-які програмні помилки в підсистемі швидкої аналітики не призведуть до псування частини даних, оскільки

сировинні дані вже збережені в БД і будуть знову оброблені під час наступного процесу переіндексації. Помилки складання алгоритмів побудови індексів так само можуть бути виправлені при плановій переіндексації, і користувач таким чином отримає коректний звіт.

Висновок: в результаті дослідження були проаналізовані основні проблеми сучасних розподілених систем та запропонована модель трискладової архітектури аналітичної системи аналізу інформації, що надходить із соціальних мереж, яка задовільняє вимогам відмовостійкості, еластичності та оперативності і здатна обробляти надвеликі об'єми даних. Також проведено опис основних складових запропонованої архітектури та показана доцільність використання такої побудови системи.

Перспективи подальших досліджень пов'язані з використанням моделі трискладової архітектури аналітичної системи аналізу інформації у інтелектуальних системах автоматизованого контролю стану біооб'єктів, які потребують збору і обробки великих об'ємів індивідуальних даних, з метою підвищення інформативності процесу та забезпечення необхідної якості.

Список літератури

1. Genealogy, Family History Records at Ancestry.com [Електронний ресурс] — Режим доступа: http://www.ancestry.com — Назва з екрану.
2. Лекція Андрія Себранта в Яндекс [Електронний ресурс] — Режим доступа: http://www.habrahabr.ru/company/yandex/blog/214217/ — Назва з екрану
3. eHarmony #1 Trusted Singles Online Dating Site More than Personals [Електронний ресурс] — Режим доступа: http://www.eharmony.com/ — Назва з екрану.
4. Академия Microsoft Поддержка разработки распределенных приложений в Microsoft .NET Framework [Електронний ресурс] — Режим доступа: http://www.intuit.ru/studies/courses/1115/177/lecture/4778/ — Назва з екрану.
5. Nathan Marz and James Warren Big Data (2015), Principles and best practices of scalable realtime data systems , Manning Publications Co, 425p.
6. Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills (2014), Advanced Analytics with Spark: Patterns for Learning from Data at Scale, O'Reilly Media, 276 p.
7. Zaharia M. An architecture for fast and general fata processing on large clusters./ Zaharia M. – EECS Department University of California, Berkeley Technical Report No. UCB/EECS-2014-12. – Feb 3, 2014.

Одержано 28.04.15