Тема 4. Методи та інструменти аналізу великих даних

1. Методи аналізу великих даних

2. Програмне забезпечення для аналізу великих даних

3. Платформи великих даних

1. Методи аналізу великих даних

Стандартна бізнес-практика великомасштабного аналізу даних ґрунтується на понятті "корпоративного сховища даних" (*Enterprise Data Warehouse, EDW*), запити до якого надходять від програмного забезпечення "бізнес-аналітики" (*Business Intelligence, BI*). Інструменти BI дають змогу створювати звіти та інтерактивні інтерфейси, узагальнення даних за допомогою агрегатних функцій (наприклад, обчислити кількість або середнє) до різноманітних розподілів ієрархічних даних на групи.

Традиційно вважається, що ретельно спроектоване сховище даних відіграє центральну роль у разі правильного застосування інформаційних технологій. Сховище даних традиційно контролюють спеціально призначені працівники IT, які не тільки супроводжують систему, а й ретельно контролюють доступ до неї, щоб керівні особи могли гарантовано розраховувати на високий рівень обслуговування .

Кількість внутрішньокорпоративних великомасштабних джерел даних істотно зростає: великі бази даних сьогодні виникають навіть на основі єдиного джерела потоків даних про відвідування Web-сайтів (*click-stream*), журналів програмних систем, архівів електронної пошти і форумів тощо. Загальновизнаною стала значущість аналізу даних. Численні компанії демонструють, що складний аналіз даних сприяє зменшенню витрат та навіть прямому зростанню доходів. Результатом цих можливостей є масовий перехід до збирання та використання даних у декількох організаційних одиницях корпорацій.

У цьому змінному кліматі збирання розрізнених великомасштабних даних доцільним є підхід, який називають *могутнім аналізом даних* (МАД; *Magnetic, Agile, Deep (MAD) data analysis*). Акронім MAD походить від трьох аспектів цього середовища, що відрізняють його від ортодоксальних сховищ даних, а саме: Магнетична (*magnetic*); Гнучкість (*agile*); Ґрунтовність (*deep*).

Великі дані (*англ. Big data*) – серія підходів, інструментів і методів опрацювання структурованих та неструктурованих даних величезних обсягів і значного різноманіття для отримання зрозумілих для людини результатів, ефективних в умовах безперервного приросту, розподілу по численних вузлах обчислювальної мережі, що сформувалися в кінці 2000-х років, альтернативних традиційним системам управління базами даних і рішень класу Business Intelligence. Є три типи завдань, пов'язаних з Великими даними (Big Data).

1) *зберігання і управління.* Обсяг даних в сотні терабайт або петабайт не дає змоги легко зберігати їх та керувати ними за допомогою традиційних реляційних баз даних;

2) *неструктурована інформація.* Більшість Великих даних неструктуровані;

3) *аналіз Великих даних*. Як аналізувати неструктуровану інформацію? Як на

основі Великих даних складати прості звіти, будувати та впроваджувати поглиблені прогностичні моделі?

Робота з Великими даними не схожа на звичайний процес бізнес-аналітики, коли просте додавання відомих значень приносить результат. Працюючи з великими даними, результат одержують, очищаючи їх за допомогою послідовного моделювання: спочатку висувається гіпотеза, будується статистична, візуальна або семантична модель, на її підставі перевіряється достовірність висунутої гіпотези і потім пропонується наступна. Цей процес вимагає від дослідника або інтерпретації візуальних значень, або складання інтерактивних запитів на основі знань, або розроблення адаптивних алгоритмів "машинного навчання", здатних отримати потрібний результат. Причому час життя такого алгоритму може бути доволі коротким.

За інтенсивного розвитку бізнесу для збереження конкурентоспроможності підприємства та опрацювання значних обсягів накопичених структурованих та неструктурованих даних допомогти може інформаційна технологія Великі дані. Актуальним є застосування методів і технологій аналізу Великих даних та інтегрованої платформи для бізнес-аналітики. Метою роботи є дослідження особливостей класифікації методів і технологій аналітики Великих даних з урахуванням означення та особливостей застосування технології Великих даних.

Описання методів і технологій аналітики Великих даних (Big Data Analytics) Формальна модель великих даних як інформаційної технології така:

$$\mathbf{BD} = \langle \mathbf{Vol}\text{BD}, \mathbf{Ip}, \mathbf{A}\text{BD}, \mathbf{T}\text{BD} \rangle,$$

де **Vol**BD – множина типів обсягів; **Ip** – множина типів джерел даних (інформаційних продуктів); **A**BD – множина методик аналізу Великих даних; **T**BD – множина технологій обробки Великих даних. На основі означення Великих даних можна сформулювати основні принципи роботи з такими даними: горизонтальна масштабованість; стійкість до відмов; локальність даних. Усі сучасні засоби роботи з Великими даними так чи інакше відповідають цим трьом принципам. Для того, щоб їх дотримуватися, необхідно придумувати якісь методи, способи і парадигми розроблення засобів опрацювання даних.

Сьогодні наявна множина **A**BD ={**A***i*} різноманітних методик аналізу масивів даних, в основу яких покладено інструментарій, запозичений з статистики та інформатики.

Необхідність у нових засобах для аналізу обґрунтована тим, що даних стає більше, більше їх зовнішніх і внутрішніх джерел, тепер вони складніші та різноманітніші (структуровані, неструктуровані та слабкоструктуровані), використовуються різні схеми індексації (реляційні, багатовимірні, noSQL). Колишні способи опрацювання даних вже неефективні – *Big Data Analytics* поширюється на великі й складні масиви, тому ще використовують терміни *Discovery Analytics* (аналітика, що відкриває) і *Exploratory Analytics* (аналітика, що пояснює).

Сьогодні не розмежовують вживання термінів Big Data і Big Data Analytics. Ці терміни описують як самі дані, так і технології управління та методи аналізу.

Big Data Analytics є розвитком концепції Data Mining. Ті самі завдання, сфери застосування, джерела даних, методи і технології. За роки, що минули з моменту

появи концепції Data Mining до настання ери Великих даних, революційно змінилися обсяги даних, що аналізуються, з'явилися системи високопродуктивних обчислень, нові технології, зокрема MapReduce і її численні програмні реалізації. З появою соціальних мереж постали і нові завдання.

Data Mining – це процес підтримки ухвалення рішень, що ґрунтується на пошуку в сирих даних прихованих закономірностей, раніше невідомих, нетривіальних, практично корисних та доступних інтерпретації знань, необхідних для ухвалення рішень в різних сферах людської діяльності. Data Mining – це особливий підхід до аналізу даних. Акцент робиться не тільки на добуванні фактів, а й на генерації гіпотез.

Якщо підхід DataMining доповнити технологією MapReduce і вимогою 4V (Volume (обсяг), Velocity (швидкість), Variety (різноманітність), Veracity (достовірність), то це відобразить функціональні зв'язки Big Data Analytics (рис. 4.1).

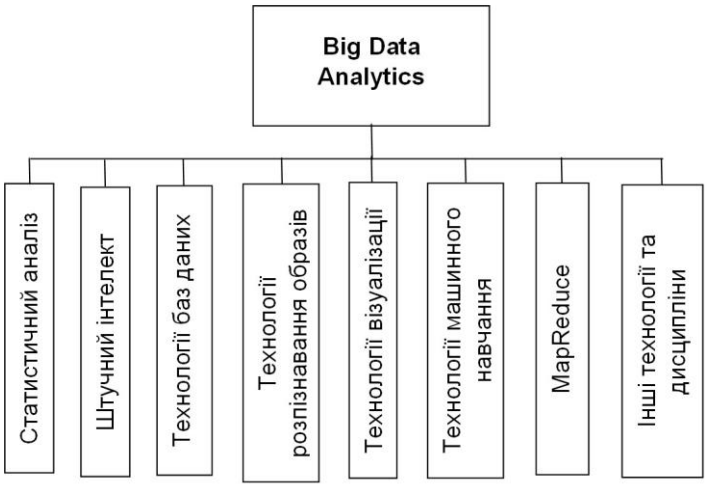

Рис. 4.1. Функціональні зв'язки аналітики Великих даних

Аналіз великих обсягів даних і необхідності зрозуміти значення з індивідуальної поведінки потребує методів оброблення, які виходять за межі традиційних статистичних методів.

Методики і методи аналізу, які застосовують до великих даних, також описано в звіті McKinsey: методи DataMining; краудсорсинг; консолідація та інтеграція даних; машинне навчання; нейронні мережі, мережевий аналіз, оптимізація, зокрема, генетичні алгоритми; розпізнавання образів; аналітика, прогнозування; імітаційне моделювання; просторовий аналіз; статистичний аналіз; візуалізація аналітичних даних.

BothManyika (2011) і Chen (2012) запропонували такий список методів аналітики Великих даних (в алфавітній послідовності): A/B тестування (A/B testing), правило навчання асоціації (Association rule learning), класифікація (Classification), кластерний аналіз (Cluster analysis), злиття і інтеграція даних (Data fusion and data integration), Ансамблі навчання (Ensemble learning), генетичні алгоритми (Genetic algorithms), машинного навчання (Machine learning), обробки природної мови (Natural Language Processing), Нейронні мережі (Neural networks),

мережевий аналіз (Network analysis), розпізнавання образів (Pattern recognition), Прогнозне моделювання (Predictive modelling), регресія (Regression), Настроїв аналіз (Sentiment Analysis), Обробка сигналів (Signal Processing), Просторовий аналіз (Spatial analysis), статистика (Statistics), кероване і некероване навчання (Supervised and Unsupervised learning), моделювання (Simulation), аналіз часових рядів та візуалізації (Timeseries analysis and Visualization).

Опишемо групи методів і технологій аналітики Великих даних, які класифікуються з урахуванням функціональних зв'язків та формальної моделі цієї інформаційної технології, а саме: методи Data Mining, технології Text Mining, технологія MapReduce, візуалізація даних, інші технології та методики аналізу (рис. 4.2).

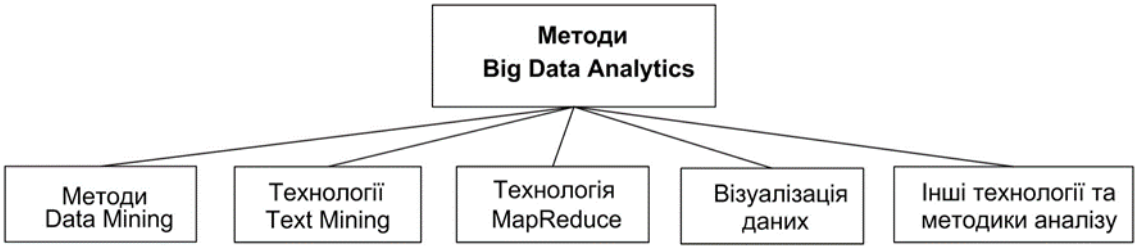

Рис. 4.2. Групи методів аналітики Великих даних

Методи інтелектуального аналізу даних (Data Mining). Застосування методів і технологій Data Mining дає змогу розв'язати такі задачі: класифікація (*Classification*); кластеризація (*Clustering*); асоціація (*Associations*); послідовність (*Sequence*), або послідовна асоціація (*sequential association*); прогнозування (*Forecasting*); визначення відхилень (*Deviation Detection*), аналіз відхилень або викидів; оцінювання (*Estimation*); аналіз зв'язків (*Link Analysis*); візуалізація (*Visualization, Graph Mining*); підбивання підсумків (*Summarization*) – опис конкретних груп об'єктів за допомогою аналізованого набору даних.

Методи Data Mining поділяють на дві групи: навчання з учителем (*Supervised Learning*); навчання без учителя (*Unsupervised Learning*). Інша класифікація поділяє все різноманіття методів Data Mining на дві групи: статистичні й кібернетичні методи. Ця схема поділу ґрунтується на різних підходах щодо навчання математичним моделям.

Опишемо найпридатніші з них для аналізу Великих даних.

Асоціативні правила (*Association Rule Learning*). Набір методик для виявлення взаємо-зв'язків, тобто асоціативних правил, між змінними величинами у великих масивах даних. Для ана-лізу ринкового кошика застосовують **аналіз прихованих закономірностей** (A*ssociation Analysis*).

Класифікація (*Classification*). Набір методик, які дають змогу передбачити поведінку споживачів у певному сегменті ринку (прийняття рішень про покупку, відтік, обсяг споживання тощо).

Метод дерев рішень (*Decision Trees*) є одним з найпопулярніших методів розв'язання завдань класифікації та прогнозування. У найпростішому вигляді дерево рішень – це спосіб подання правил в ієрархічній, послідовній структурі. Метод дерев рішень зазвичай називають "наївним" підходом.

Кластерний аналіз (*Cluster Analysis*). Статистичний метод класифікації об'єктів за групами у результаті виявлення наперед не відомих загальних ознак. Приклад – сегментування ринку.

Для вирішення завдання кластеризації на графах застосовують алгоритм Girvanand Newman методу MLP (Markov Cluster Algorithm).

Для аналізу Великих багатовимірних даних розроблено методологію "Dynamic Quantum Clustering" (DQC), що реалізує парадигму пошуку як "нехай дані говорять про себе самі". Метод DQC (як і багато інших методів аналітики Великих даних) "працює" без попереднього знання про ті "структури", їх тип і топології, які можуть бути "приховані" в даних і виявлені в результаті його застосування. Метод добре працює з багатовимірними даними і час аналізу лінійно залежить від розмірності.

Регресія (*Regression*). Набір статистичних методів для виявлення закономірності між зміною залежної змінної та однієї або декількох незалежних.

Аналіз часових рядів (*Time Series Aanalysis*). Набір запозичених зі статистики та цифрової обробки сигналів методів аналізу повторюваних з плином часу послідовностей даних. **Аналіз викидів** (*Outlieran Aalysis*) застосовують для виявлення шахрайства, особистого маркетингу, медичного аналізу.

Машинне навчання (*Machine Learning*). Напрям в інформатиці (історично за ним закріпилася назва "штучний інтеллект"), який має на меті створення алгоритмів самонавчання на основі аналізу емпіричних даних. Машинне навчання сьогодні використовується: для розпізнавання спаму або не спаму повідомлень електронної пошти; для отримання знань про переваги користувача та надання рекомендацій, що ґрунтуються на цій інформації; для визначення кращого контенту для залучення потенційних клієнтів; для встановлення ймовірності виграшу справи та відповідності юридичним нормам пред'явлених рахунків.

Кероване і некероване навчання (*Supervised and Unsupervised Learning*). Набір методик, що ґрунтуються на технологіях машинного навчання, які дають змогу виявити функціональні взаємозв'язки в аналізованих масивах даних. Некероване навчання має спільні риси з кластерним аналізом.

Ансамблі навчання (*Ensemble Learning*). У цьому методі задіється множина предикативних моделей, за рахунок чого поліпшується якість прогнозів.

Еволюційні алгоритми, генетичні алгоритми (*Evolution Analysis, Genetic Algorithms*). Генетичні алгоритми навіяні природою еволюційних процесів – тобто таких механізмів, як успадкування, мутації та природний добір. Ці механізми використовуються для "еволюціонування" корисного вирішення проблем, які потребують оптимізації. У цій методиці можливі рішення подають у вигляді "хромосом", які можуть комбінуватися і мутувати. Як і в процесі природної еволюції, виживає найпристосованіша особина.

Нейронні мережі (*Neural Networks*) – це клас моделей, що ґрунтуються на аналогії з роботою мозку людини та призначені для розв'язання різноманітних задач аналізу даних після проходження етапу навчання на даних. За допомогою нейронних мереж можна, наприклад, передбачати обсяги продажів, показники фінансового ринку, розпізнавати сигнали, розробляти самонавчальні системи.

Візуалізація даних

Візуалізація (*Visualization*). Методи графічного подання результатів аналізу

великих даних у вигляді діаграм або анімації для спрощення інтерпретації, полегшення розуміння отриманих результатів. Візуалізація аналітичних даних – зображення інформації у вигляді рисунків, графіків, схем і діаграм з використанням інтерактивних можливостей та анімації для результатів, а також вихідних даних для подальшого аналізу.

Наочне представлення результатів аналізу Великих даних має принципове значення для їхньої інтерпретації. Сприйняття людини обмежене, і вчені продовжують вести дослідження у галузі вдосконалення сучасних методів подання даних у вигляді зображень, діаграм або анімацій. Новими прогресивними методами візуалізації є: хмара тегів; кластерограма; історичний потік; просторовий потік.

Технології Text Mining. Підґрунтям технології **Text Mining** – статистичний та лінгвістичний аналіз, методи штучного інтелекту. Ця технологія застосовується для проведення аналізу, забезпечення навігації та пошуку в неструктурованих текстах. Застосування інформаційних систем класу Text Mining дає змогу користувачам набувати нових знань.

Технології Text Mining – набір методів, які призначені для видобування відомостей з текстів на основі сучасних ІКТ, що дає змогу виявити закономірності, які забезпечують користувачам отримання корисних даних та нових знань. Основна мета Text Mining – надати аналітику можливість працювати з великими обсягами початкових даних за рахунок автоматизації процесу здобуття потрібних даних.

Основними методами технології Text Mining є: класифікація (*classification*); кластеризація (*clustering*); побудова семантичних мереж або аналіз зв'язків (*Relationship, Event andFact Extraction*); здобуття феноменів, фактів, понять (*feature extraction*); автоматичне реферування, створення анотацій (*summarization*); відповідь на запити (*question answering*); тематичне індексування (*thematic indexing*); пошук за ключовими словами (*keyword searching*); засоби підтримки та створення таксономії (*oftaxonomies*) і тезаурусів (*thesauri*).

Прикладом ефективного застосовування технологій Text Mining є проведення контент-аналізу. **Контент-аналіз** (*Content Analysis*) – це якісно-кількісне, систематичне опрацювання, оцінювання та інтерпретація форми і змісту тексту.

Інші технології та методики досліджень

Опишемо декілька технологій і дисциплін дослідження даних з погляду технології Великих даних.

A/B тестування (*A/B testing, Splittesting*). Методика маркетингового дослідження, в якій контрольна вибірка по черзі порівнюється з іншими. Метод використовується для оптимізації Web-сторінок відповідно до заданої мети.

Обробка природної мови (*Natural Language Processing (NLP*). Набір запозичених з інформатики та лінгвістики методик розпізнавання природної мови людини.

Аналіз настроїв (*Sentiment Analysis*). В основу методик оцінки настроїв споживачів покладено технології розпізнавання природної мови людини. Аналіз настроїв допомагає дослідникам визначити настрої спікерів або авторів щодо теми.

Мережевий аналіз (*Network Analysis*). Набір методик аналізу зв'язків між вузлами в мережах. Стосовно соціальних мереж дає змогу аналізувати

взаємозв'язок між окремими користувачами, компаніями, спільнотами тощо.

Оптимізація (*Optimization*). Набір числових методів для редизайну складних систем і процесів для поліпшення одного або декількох показників. Допомагає у прийнятті стратегічних рішень, наприклад, складу виведеної на ринок продуктової лінійки, у проведенні інвестиційного аналізу тощо.

Розпізнавання образів (*Pattern Recognition*). Набір методик з елементами самонавчання для передбачення поведінкової моделі споживачів.

Прогнозне моделювання (*Predictive Modeling*). Набір методик, які дають змогу створити математичну модель наперед заданого ймовірного сценарію розвитку подій.

Обробка сигналів (*Signal Processing*). Запозичений з радіотехніки набір методик, який має на меті розпізнавання сигналу на тлі шуму і його подальшого аналізу.

Просторовий аналіз (*Spatial Analysis*). **Просторовий аналіз** – використання топологічної, геометричної та географічної інформації в даних. Набір частково запозичених зі статистики методик аналізу даних. Джерелом великих даних у цьому випадку є геоінформаційні системи (ГІС).

Статистика (*Statistics*). Наука про збирання, організацію та інтерпретацію даних, зокрема розроблення опитувальників і проведення експериментів. Статистичні методи часто застосовують для оцінкових суджень про взаємозв'язки між тими чи іншими подіями.

Моделювання (*Simulation*). Моделювання поведінки складних систем часто використо-вується для прогнозування, передбачення і опрацювання різних сценаріїв під час планування.

Краудсорсинг (*Crowdsourcing*). Методика збирання даних з великої кількості джерел. Краудсорсинг – категоризація та збагачення даних силами широкого, невизначеного кола осіб, з метою використання їхніх творчих здібностей, знань і досвіду із застосуванням інформаційно-комунікаційних технологій.

Злиття та інтеграція даних (*Data Fusion and Data Integration*). Набір технік, що дають змогу інтегрувати різнорідні дані з різноманітних джерел інформації для проведення глибинного аналізу. Цей набір методик дає змогу аналізувати коментарі користувачів соціальних мереж і зіставляти з результатами продажів у режимі реального часу.

Технологія MapReduce. Створення і підтримка сховищ даних обсягом в терабайт, петабайт і більше уможливилась завдяки технологіям розподілених файлових систем. Розподілені системи опрацювання даних, замість зберігання даних в одній файловій системі, зберігають та індексують дані на декількох (навіть тисячах) жорстких дисках і серверах. Створюється також "карта" (*map*), на якій міститься інформація про місцезнаходження тих чи інших даних. Однією з найвідоміших систем, що використовують цей підхід, є **Hadoop**. Щоб опрацювати дані в розподіленій файловій системі, необхідно виконувати низькорівневі обчислення, такі як підсумовування, агрегування тощо, в місці їхнього фізичного розміщення в розподіленій файловій системі. Створити карту (*map*) виконаних обчислювальних алгоритмів і відстежувати локальні результати, а потім акумулювати результати (reduced). Цей підхід і шаблон проведення

обчислювальних алгоритмів отримав назву **MapReduce** . MapReduce – це фреймворк для обчислення деяких наборів розподілених завдань з використанням великої кількості комп'ютерів ("нод"), що утворюють кластер. Опрацьовуватися можуть дані, які зберігаються або в файловій системі (неструктуровано), або в базі даних (структуровано).

Багато практичних завдань можна реалізувати у цій моделі програмування. Є безліч інструментів для проведення такого агрегування даних у розподіленій файловій системі, що дає змогу легко здійснювати цей аналітичний процес.

Наведений опис методів і технологій аналізу Великих даних дає змогу побудувати онтологію відповідно до підходу METHONTOLOGY, який відображає процес ітеративного проектування. За методологією METHONTOLOGY глосарій термінів містить всі терміни (концепти та їхні екземпляри, атрибути, дії), важливі для аналізу Великих даних, і їхні природно-мовні описи.

Глосарій термінів онтології аналізу Великих даних містить означені вище терміни, які можна семантично розділити на три групи: структура завдання (групи технологій аналітики, зв'язки), дані, що наповнюють задачу (методи, що застосовують для кожної групи), і результати обчислень (рекомендації щодо використання Великих даних для підвищення ефективності ухвалення рішень). Онтологія аналізу Великих даних розроблена засобами Protégé-OWL.

Великі дані мають вагоме практичне значення як технологія, призначена для вирішення актуальних повсякденних проблем, але породжує ще більше нових. Великі дані здатні змінити наш спосіб життя, праці й мислення.

Однією з умов успішного розвитку світової економіки на сучасному етапі стає можливість фіксувати й аналізувати величезні масиви і потоки інформації. Є думка, що країни, які оволодіють найефективнішими методами роботи з Великими даними, чекає нова індустріальна революція. Напрям "Big Data" концентрує зусилля в організації зберігання, оброблення, аналізу величезних масивів даних.

Міжнародна консалтингова компанія McKinsey, що спеціалізується на розв'язанні задач, пов'язаних зі стратегічним управлінням, виділяє 11 методів і технік аналізу, що застосовуються до великих даних.

Методи класу Data Mining (видобуток даних, інтелектуальний аналіз даних, глибинний аналіз данних) — сукупність методів виявлення у даних раніше невідомих, нетривіальних, практично корисних знань, необхідних для прийняття рішень. До таких методів, зокрема, належать: навчання асоціативним правилам (association rule learning), класифікація (разгалуження на категорії), кластерний аналіз, регресійний аналіз, виявлення і аналіз відхилень тощо.

Краудсорсінг — класифікація і збагачення даних силами широкого, неозначеного кола особистостей, що виконують цю роботу без вступу у трудові стосунки.

Змішання та інтеграція даних (data fusion and integration) — набір технік, що дозволяють інтегрувати різнорідні дані з розмаїття джерел з метою проведення глибинного аналізу (наприклад, цифрова обробка сигналів, обробка природньої мови, включно з тональним аналізом).

Машинне навчання, включаючи навчання з учителем і без учителя – використання моделей, побудованих на базі статистичного аналізучи машинного навчання для отримання комплексних прогнозів на основі базових моделей.

Штучні нейронні мережі, мережевий аналіз, оптимізація, у тому числі генетичні алгоритми (genetic algorithm — евристичні алгоритми пошуку, що використовуються для розв'язання задач оптимізації і моделювання шляхом випадкового підбору, комбінування і варіації потрібних параметрів з використанням механізмів, аналогічних натуральному відбору у природі)

З точки зору обробки в основу технологій Big Data покладені два основних принципи:

– розподіленого зберігання даних;
– розподіленої обробки, з урахуванням локальності даних.

Розподілене зберігання вирішує проблему великого обсягу даних, дозволяючи організовувати сховище з довільного числа окремих простих носіїв. Зберігання може бути організовано з різним ступенем надмірності, забезпечуючи стійкість до збоїв окремих носіїв Розподілена обробка з урахуванням локальності даних означає, що програма обробки доставляється на обчислювач, що знаходиться якомога ближче до оброблюваних даних. Це принципово відрізняється від традиційного підходу, коли обчислювальні потужності і підсистема зберігання розділені і дані повинні бути доставлені на обчислювач. Таким чином, технології Big Data спираються на обчислювальні кластери з безлічі обчислювачів, забезпечених локальної підсистемою зберігання.

Доступ до даних і їх обробка здійснюються спеціальним програмним забезпеченням. Найбільшвідомим і інтенсивно розвиваються проектом в області Big Data є Apache Hadoop. В даний час на ринку інформаційних систем і програмного забезпечення синонімом Big Data є технологія Hadoop, яка представляє собою програмний фреймворк, що дозволяє зберігати і обробляти дані за допомогою комп'ютерних кластерів, використовуючи парадигму MapReduce. Основними складовими платформи Hadoop є:

– відмовостійка розподілена файлова система Hadoop Distributed File System (HDFS), за допомогою якої здійснюється зберігання;

– програмний інтерфейс Map Reduce, який є основою для створення програмного забезпечення, що обробляють великі обсяги структурованих і неструктурованих даних паралельно на кластері, що складається з тисяч машин;

– Apache Hadoop YARN, що виконує функцію управління даними.

Відповідно до підходу MapReduce обробка даних складається з двох кроків: Map і Reduce. На кроці Map виконується попередня обробка даних, яка здійснюється паралельно на різних вузлах кластера.

На кроці Reduce відбувається зведення попередньо оброблених даних в єдиний результат.

В основі моделі роботи Apache Hadoop лежать три основних принципи. По-перше, дані рівномірно розподіляються на внутрішніх дисках безлічі серверів, об'єднаних HDFS.

По-друге, не дані передаються програмі обробки, а програма - до даних. Третій принцип - дані обробляються паралельно, причому ця можливість закладена архітектурно в програмному інтерфейсі Map Reduce. Таким чином, замість звичної концепції «база даних + сервер» у нас є кластер з безлічі недорогих вузлів, кожен з яких є і сховищем, і обробником даних, а саме поняття «база даних» відсутня.

Платформа Hadoop дозволяє скоротити час на обробку і підготовку даних, розширює можливості по аналізу, дозволяє оперувати новою інформацією та неструктурованими даними.

Компанія Oracle розбиває життєвий цикл обробки інформації на три етапи і використовує для кожного з них власне рішення:

1) Збір, обробка та структурування даних.

В якості вирішення застосовується Oracle Big Data Appliance - це встановлений Hadoop-кластер, Oracle NoSQL Database і засоби інтеграції з іншими сховищами даних. Завдання Oracle Big Data Appliance полягає в зберіганні та первинній обробці неструктурованою або частково структурованої інформації, тобто як раз в тому, що у систем на базі Hadoop виходить найкраще.

2) Агрегація і аналіз даних.

Для роботи зі структурованими даними використовується комплекс Oracle Exadata. Модулі інтеграції Oracle Big Data Appliance дозволяють оперативно завантажувати дані в Oracle Exadata, а також отримувати доступ до даних «на льоту» з Oracle Exadata.

3) Аналітика даних в реальному часі.

2. Програмне забезпечення для аналізу великих даних

Процес інформатизації суспільства та економіки загалом, з кожним роком набирає обертів та проникає у всі галузі та сфери життя. Така тенденція є позитивною для розвитку та функціонування постіндустріального суспільства, однак вона наділена специфічними особливостями. «Big Data» – сукупність великої кількості неструктуризованої інформації яка зростає у геометричній прогресії щороку, та значно ускладнює процес пошуку та аналізу необхідної інформації в мережі. Разом з формуванням великих об’ємів інформації з’являється програмне забезпечення, що дозволяє обробляти та класифікувати необхідну інформацію для спеціалістів з маркетингу. Підбір такого програмного забезпечення є незамінним інструментом майбутнього для великого комплексу аналітичних дій на підприємстві та якості отриманих результатів, що уже є 50% успіху при плануванні будь яких стратегій підприємства.

Сучасне бізнес середовище, як ніколи, є досить турбулентним та інформаційно перевантаженим, кількість різноманітних даних з внутрішнього та зовнішнього середовища постійно зростає, стає складнішою та менш структурованою. Існуючі підходи та методи аналізу інформації уже не виконують повноцінно свої функції та стають менш актуальними, виникає потреба пошуку нових можливостей. Найкраще з такими викликами справляються методики Big Data Analytics.

Big Data Analytics – це комплекс методик та підходів, що направлені на акумулювання, систематизацію та обробку великої кількості різної за своїми характеристиками інформації та формуванні на їх основі відповідних висновків та гіпотез. Відповідно функціональні зв’язки Big Data Analytics є досить розгалуженим та включають у себе такі елементи: технології візуалізації, статистичний аналіз, штучний інтелект, технології баз даних, технології розпізнавання образів, об’єднавши які та класифікувавши можна отримати перелік методів аналітики Великих даних.

До таких методів можна віднести: методи Data Mining, технології Text Mining, технологія MapReduce та візуалізація даних.

Для отримання обґрунтованого управлінського рішення дані проходять крізь послідовність процесів накопичення у сховищі даних, аналітичну обробку та видобуток знань. Для реалізації цього процесу використовують методи та алгоритми, які входять до складу технології Data Mining. Близько 80% роботи над Data Mining полягає в зборі та підготовці даних, що проводиться ще до запуску інструментів видобутку знань. Найбільш поширинеми методами Data Mining є: Базові методи, нечітка логіка, генетичні алгоритми, нейронні мережі.

Технологія MapReduce – модель розподілених обчислень у комп`ютерних кластерах, представлена компанією Google. Згідно з цією моделлю, додаток розділяється на значну кількість однакових елементарних завдань, що виконуються на вузлах кластера і потім, природнім шляхом зводяться у кінцевий результат.

Технології Text Mining. Підґрунтям технології Text Mining – статистичний та лінгвістичний аналіз, методи штучного інтелекту. Ця технологія застосовується для проведення аналізу, забезпечення навігації та пошуку в неструктурованих текстах. Застосування інформаційних систем класу Text Mining дає змогу користувачам набувати нових знань.

Візуалізація (*Visualization*). Методи графічного подання результатів аналізу великих даних у вигляді діаграм або анімації для спрощення інтерпретації, полегшення розуміння отриманих результатів. Візуалізація аналітичних даних – зображення інформації у вигляді рисунків, графіків, схем і діаграм з використанням інтерактивних можливостей та анімації для результатів, а також вихідних даних для подальшого аналізу.

Аналізуючи особливості Big Data Analytics, а саме, розгалуженість, багатофункціональність та складність існуючих методик, можна чітко стверджувати, що даний інструментарій є перевантаженим для практичного використання на підприємствах. Бізнес середовище не може витрачати багато часу на усі етапи аналізу, а тому, потребує автоматизації даних процесів. Відповідно на рику почало з’являтись програмне забезпечення, функції якого дають змогу виконувати ряд специфічних функцій, що до обробки великих даних.

Програмні засоби великих даних можна класифікувати за задачами, які вони вирішують. У процес створення рішення повинні бути інтегровані різні засоби для зберігання, управління та аналізу великих даних. Інструменти великих даних відповідно до їх задач включають наступні групи:

1) ПЗ зберігання великих даних; ·
2) ПЗ управління великими даними; ·
3) ПЗ обробки великих даних;
4) методи та засоби візуалізації великих даних; ·
5) методи та засоби аналітики великих даних.

Повноцінне програмне забезпечення по обробці великих даних повинен відображати інструменти для зберігання, управління та обробки такої інформації, інструменти та методи аналітики, візуалізації та оцінювання у різні етапи процесу побудови рішення.

Досліджуючи такий ринок програмного забезпечення, необхідно відмітити, що за останні п’ять років він значно збільшився і почав кластеризуватись. В процесі

таких згрупувань створилась платформа Business Intelligence. Business Intelligence (BI) – це термін-метафора, який не має дослівного тлумачення та «ієрархічно-синергетичний комплекс автоматизованих засобів нетривіального аналізу первинних даних і візуалізації його результатів для підтримки рішень (Decision Support)».

У бізнес середовищі за допомогою BI можна виконувати такі завдання: – класифікація споживачів;

–виявлення асоціативних правил у споживчому попиті та їх використання для збільшення продажів; – багатомірний аналіз обсягів продажів, маркетингових затрат засобами OLAP;

–оптимізація асортименту;

–прогнозування обсягів продажів та інших показників за допомогою методу регресійного аналізу; –сегментування ринку за допомогою кластерного аналізу

–оцінка ефективності та оптимізація маркетингових кампаній; – оптимізаційне управління ціновою політикою.

Основними операціями, які проводяться з інформацією в бізнес середовищі це – накопичення, аналіз, та побудова на її основі прогнозів та виявлення тенденцій (Табл. 4.1).

Таблиця 1.

Програмне забезпечення, що використовується при роботі з Big Data у бізнес середовищі

Призначення	Продукт
Для збору інформації про внутрішнє та зовнішнє середовище	Marketing Geo, Mapinfo, ArcGI “Infostream corporate”, “Stikler” KonSi-Competitive Intelligence&Benchmarking CRM-системи ERP-системи:1C, SAP ERP, Галактика ERP.
Інтеграція данних	ERP Integration, ETL Integration, Portal Inte-gration, CRM Integration, MS Office Applica-tions and Big-Data Connectors
Аналіз данних та їх уніфікація	SAS Business Intelligence, Microsoft BI, IBM Cognos BI, SAP Business Intelligence, Oracle Business Intelligence. Tableau 9.0, Qlik Sense 2.0 і Microsoft Power BI
Візуалізація данних Integration, Autochartin	Visual Querying, Storyboarding, Geospatial

Аналізуючи вище наведені програми та методи, необхідно відмітити, що усі вони є дієвими та ефективними, та мають місце на практичне застосування, їхня актуальність для бізнесу є досить індивідуальною та залежить від особливостей підприємства та очікуваних результатів від їхнього використання.

В умовах розбудови постіндустріального суспільства, ключовим благом якого є інформація та інтелект, неможливо вести бізнес без роботи з цифровими даними та різноманітними інформаційними технологіями. Актуальність та вагомість, на сьогодні програмного забезпечення, яке використовується для аналізу «Big Data» є очевидним, оскільки дає змогу ефективно та в короткі строки опрацювати та систематизувати великі дані та значно полегшити роботу при планування, аналізі та прогнозуванні.

3. Платформи великих даних

Платформа великих даних – це інструмент, розроблений постачальниками програмного забезпечення (ПЗ) для управління даними з метою покращення масштабованості, доступності, продуктивності та безпеки організацій, які працюють з великими даними.

Платформа призначена для обробки в режимі реального часу об'ємних багатоструктурних даних. Різні користувачі можуть її використовувати для виконання різних задач. Так, наприклад, інженери да-них – для очищення, агрегування та підготовки даних для аналізу, бізнес-користувачі – для запуску запитів, а вчені вважають її корисною при аналізі шаблонів з наборів великих даних за допомогою алгоритмів машинного навчання.

Це платформа інформаційних технологій (IT) класу підприємства, яка забезпечує властивості та функціональність прикладної системи в одному рішенні для розробки, розгортання, обробки та управління великими даними. Програмне забезпечення (ПЗ) аналітики великих даних допомагає розкрити приховані шаблони, невідомі кореляції, ринкові тенденції, вподобання клієнтів та іншу корисну інформацію з широкого різноманіття наборів даних.

Головне питання організації роботи з великими даними на корпоративному рівні: обрати реляційну (SQL) чи нереляційну (NoSQL) базу даних? Головною причиною відмови від SQL баз даних (БД) є не правильна робота з самою базою. Більшість компаній не можуть собі дозволити тримати спеціалістів для постійного налагодження баз даних, а для того, щоб розпочати використовувати NoSQL БД не потрібно додаткових розробок. При розробці NoSQL БД особлива увага приділяється забезпеченню високої масштабованості та гнучкості рішень. NoSQL БД – це, перш за все, швидкий доступ до даних, що зберігаються в оперативній пам'яті, гнучкість використання та можливість швидкого розподілення даних між вузла-ми. Однак можливі такі сценарії, коли дані згодом виходять з-під контроля або вже просто не вміщуються в оперативній пам'яті.

Основні властивості та переваги платформ великих даних

До основних властивостей платформ великих даних можна віднести:

- *забезпечення ефективного зберігання та обробки даних, а також їх інтеграції, управління, витягнення, трансформації, та завантаження* (*ETL*);
- *використання системи Hadoop*: забезпечують функції для масового зберігання даних будь-якого типу, величезну потужність обробки та можливість обробляти практично необмежену кількість паралельних задач;
- *потокові обчислення*: забезпечують функції для *затягування* даних у потік, обробки даних та передачі їх назад єдиним потоком;
- *функції розвиненої аналітики та машинного навчання*;
- *функції управління життєвим циклом контенту та документів;*
- *функції інтеграції великих даних з будь-якого джерела*;
- *управління даними*: містять комплексну систему безпеки, рішення для управління даними та забезпечують дотримання вимог щодо захисту даних.

До головних переваг платформи великих даних та ПЗ аналітики великих даних можна віднести

- *точні дані.* Платформа великих даних пропонує точні дані, що сприяє прийняттю правильних рішень. Її аналітичні засоби зменшують ризик отримання

недостовірних даних, які виникають внаслідок використання сирих, не проаналізованих даних;

 підвищення ефективності праці. Платформа спрощує отримання джерела необхідної інформації. Пропонує також інформацію, що може стати у нагоді в майбутньому, таким чином, зберігаючи час та підвищуючи ефективність роботи користувачів;

 швидкі відповіді на складні питання. Ефективне управління бізнесом вимагає швидких адекватних відповідей на критичні питання, які впливають на успішність бізнес-операції. Платформа великих даних дозволяє робити це більш надійно. Деякі критичні питання, відповіді на які вимагають тижнів або місяців, за наявності правильного інструменту можуть вирішуватись лише за кілька годин або хвилин;

 безпека даних. Забезпечує безпечну інфраструктуру, яка гарантує безпеку даних.

Задачі та ПЗ великих даних

Програмні засоби великих даних можна класифікувати за задачами, які вони вирішують. У процес створення рішення повинні бути інтегровані різні засоби для зберігання, управління та аналізу великих даних. Інструменти великих да-них відповідно до їх задач включають наступні групи:

 ПЗ зберігання великих даних;
 ПЗ управління великими даними;
 ПЗ обробки великих даних;
 методи та засоби візуалізації великих даних;
 методи та засоби аналітики великих даних.

Таким чином, відповідний фреймворк повинен відображати інструменти для зберігання, управління та обробки великих даних, інструменти та методи аналітики, візуалізації та оцінювання у різні етапи процесу побудови рішення. Аналітика великих даних може застосовуватись до виявлення знань та обґрунтованого прийняття рішень.

Зберігання та управління великими даними

Традиційні методи структурованого зберігання та витягування даних, такі як реляційні БД, вітрини або SQL сховища даних, мають певні обмеження, які роблять їх не придатними для роботи з великими даними, а саме вони:

 не дозволяють включати нові джерела даних без їх попереднього очищення та інтеграції,
 не дозволяють швидко виробляти та адаптувати дані,
 не забезпечують можливості синхронізації логічного та фізичного вмісту БД з швидкою еволюцією даних,
 не забезпечують поточні потреби аналізу даних.

Необхідність порівняно недорогого зберігання та обробки гігантських обсягів неструктурованої інформації призвела до створення спеціалізованого ПЗ, яке дозволило розподіляти дані за кластерами з сотень та тисяч вузлів, а також обробляти їх у паралельному режимі. Засоби нового покоління для зберігання та управління не структурованими (не реляційними) даними, а саме NoSQL БД, дозволили використовувати репозиторій даних без додаткових розробок, підготовки або на-лагодження, забезпечили високу масштабованість, розподілення даних між вузла-ми

та швидкий доступ до даних, що зберігаються в оперативній пам’яті. NoSQL БД дозволяють записувати задачі управління даними на прикладному рівні. Кожна база, в даному випадку, є колекцією незалежних документів, де кожний документ підтримує власні дані та схеми та може мати метадані – оглядову інформацію про дані документа. Прикладна про-грама може мати доступ до багатьох БД, розташованих у різних місцях.

Нові вимоги до зберігання, управління та обробки даних обумовили виникнення Hadoop – фреймворка з відкритим кодом під крилом Apache Sotware Foundati-

системи на базі відносно недорогого об-ладнання масового попиту. З часом Hadoop був розширений набором бібліотек та утиліт, та сформував навколо себе екосистему проектів з розподіленої обробки даних. Розглянемо його більш детально.

Apache Hadoop фреймворк

Apache Hadoop забезпечує розподілене зберігання та обробку дуже великих наборів даних на комп’ютерних кластерах з промислового комп’ютерного обладнання. Тобто, замість того, щоб використовувати один великий комп’ютер, Hadoop дозволяє кластеризувати апаратне забезпечення для паралельного виконання аналізу масивних наборів даних. Сервіси Hadoop забезпечують виконання наступних функцій:

1) зберігання даних;
2) обробка даних;
3) доступ до даних;
4) управління даними;
5) безпека та операції з даними.

Екосистема Hadoop складається з багатьох модулів (процедур, бібліотек та властивостей), які розглядаються як частини фреймворка. Кожний модуль виконує певну задачу, необхідну для виконання аналітики великих даних. Ядро Hadoop складається з двох основних модулів: розподіленої файлової системи Hadoop Distributed File System (HDFS) та базового інструменту для обробки даних MapReduce та підтримується майже всіма відомими постачальниками систем великих даних. Також до найбільш використовуваних від-носять планувальник завдань та управлін-ня кластерами YARN і множину загальних утиліт Hadoop Common.

Розподілена файлова система. HDFS дозволяє зберігати дані у простому доступному форматі. Це досягається завдяки використанню великої кількості пов’язаних пристроїв зберігання даних та механізму MapReduce для їх обробки.

"Файлова система" є методом, що застосовується комп’ютером для зберігання даних таким чином, щоб їх можна було знаходити та використовувати. Зазвичай, він визначається операційною системою (ОС) комп’ютера, але система Hadoop використовує власну файлову систему, яка надбудовується "над" файловою системою хост-комп’ютера. Це означає, що доступ до даних можна отримати з будь-якого комп’ютера, на якому встановлена будь-яка підтримувана ОС.

Hadoop розділяє файли на великі блоки та розподіляє їх між вузлами у кластері. Потім він передає пакетований код у вузли для обробки даних у паралельно-му режимі. В даному підході вузли маніпулюють даними, до яких вони мають доступ. Це дозволяє обробляти набір да-них швидше та ефективніше, ніж в більш традиційній

архітектурі суперкомп’ютера, яка спирається на паралельну файлову систему, де обчислення та дані розподіляються у високошвидкісній мережі.

HDFS є розподіленою, масштабованою та портативною файловою системою, що написана на Java для Hadoop фреймворк. Вона забезпечує виконання команд та Java інтерфейсів (API), подібних до інших файлових систем, для зв'язку використовує протокол TCP/IP. Клієнти для спілкування один з одним використовують виклики віддаленої процедури (RPC). Надійність зберігання даних досягається шляхом реплікації між декількома хостами. Щоб зменшити трафік у мережі, Hadoop необхідно знати, які сервери є най-ближчими до даних або інформації, яка може забезпечити встановлення мостів з HDFS.

Hadoop може працювати безпосередньо з будь-якою розподіленою файловою системою, яка може бути встановлена основною операційною системою.

Прикладами файлових систем, що підтримуються Hadoop (окрім HDFS), є:

 FTP (зберігає всі свої дані на віддалених FTP-серверах);

 сховище об’єктів Amazon S3 (Simple Storage Service), орієнтоване на кластери, які розміщені на інфраструктурі Amazon Elastic Compute Cloud типу сервер-на-запит;

 Windows Azure Storage Blobs (WASB), розширення HDFS, яке дозволяє

розподілення Hadoop для доступу до даних в Azure блог-сховищах без постійного пе-реміщення даних у кластер.

Існують також файлові системи, які не розповсюджуються разом з Hadoop, але постачаються як альтернативні, що використовуються за замовченням, з де-якими його комерційними рішеннями. Наприклад: IBM General Parallel File System, Parascale, Appistry (драйвер файлової системи Hadoop для використання з CloudIQ Storage), драйвер файлової системи IBRIX Fusion, альтернативна файлова система MapR FS, що заміщує HDFS системою повністю випадкового доступу для читання/запису файлів.

Модуль MapReduce. MapReduce названий за двома головними операціями, які він виконує, а саме: читання даних з БД і переведення їх у формат, що підходить для аналізу (map), та виконання математичних операцій (reduce).

Функціонування MapReduce забезпечується двома компонентами: JobTracker та TaskTracker. Клієнтські прикладні програми направляють завдання MapReduce до JobTracker, JobTracker працює з доступними у кластері вузлами TaskTracker, щоб наблизитись до потрібних для виконання цих завдань даних. JobTracker відомо, які вузли містять дані, та які інші комп’ютери є поруч (рис. 4.3).

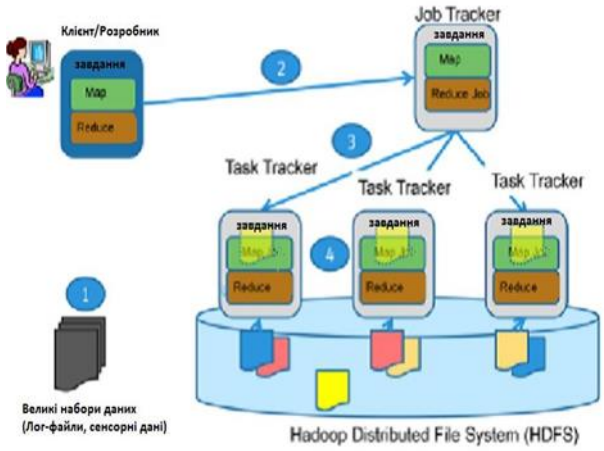

Рис. 4.3. Робота Map Reduce модуля

❶ Дуже великий набір даних. HDFS зберігає репліки даних у вузлах даних.

❷ Клієнт виконує Map та Reduce завдання на конкретному наборі даних та відсилає їх JobTracker.

❸ JobTracker розподіляє завдання серед TaskTracker. TaskTracker запускає механізм відображення (map), результат роботи якого зберігається у HDFS.

❹ Запускається завдання Reduce на даних, що вже оброблені завданням Map

Якщо робота не може бути виконана на тому вузлі, де розміщені дані, пере-вага надається вузлам, що розміщені на тій самій стойці. Таким чином, скоро-чується трафік на головній магістральній мережі. Якщо TaskTracker виходить з ладу або зазнає затримку, здійснюється пере-планування частини завдань. TaskTracker в кожному вузлі породжує окремий процес віртуальної машини Java (JVM). Це дозволяє запобігти виходу з ладу TaskTracker, якщо запущене на виконання завдання зруйнує свою JVM. Кожні кілька хвилин з TaskTracker до JobTracker над-силається імпульс, щоб перевірити його стан. Стани JobTracker і TaskTracker та інформація про їх роботу відображаються у контейнері сервлетів Jetty та її можна також переглядати у веб-браузері.

Слід зазначити, що даний підхід має певні обмеження:

 Алгоритм розподілення роботи TaskTracker є дуже простим. Кожний TaskTracker має множину наявних слотів. Кожна активна задача займає один слот. JobTracker розподіляє роботу на TaskTracker з наявним слотом, найближчий до даних. При цьому не приймається до уваги поточна завантаженість системи призначеної машини – її реальна доступність.

 Якщо один TaskTracker дуже повільний, це може затримати роботу MapReduce в цілому. Однак, коли дозволе-не паралельне виконання, окрема задача може виконуватися на декількох підпорядкованих вузлах.

В первинному варіанті, для впоряд-кування завдань з робочої черги, Hadoop підтримує First-In-First-Out (FIFO) планування та опціональне планування пріоритетів, які використовуються за замовченням. Згодом до планувальника завдань бу ла додана можливість використовувати альтернативні планувальники, такі як Fair (Facebook AI Research) або Capacity. Планувальник Fair є розробкою Facebook. Розробники мали за мету забезпечити швидку відповідь для невеликих завдань та якість сервісу для виробничих завдань. Завдання групуються у пули і ресурси роз-поділяються між цими пулами. За замовченням для кожного користувача є окремий

пул, так що кожний користувач от-римує рівну частку кластера. На відміну від планувальника Hadoop, що використовується за замовченням та формує чергу завдань, Fair дозволяє коротким завданням завершуватись у розумний час, не очікуючи довго своєї черги. Це також є простим способом спільного використання кластера між кількома користувачами, яке також може працювати з пріоритетами завдань. Ці пріоритети використовуються як ваги для визначення частки загального часу обчислення, яке отримує кожне завдання.

Планувальник Capacity, розроблений Yahoo, підтримує декілька властивостей, подібних властивостям Fair, а саме: кожній черзі виділяється частка загального ресурсу, вільні ресурси виділяються чергам за їх потужністю. У черзі завдання з більшим пріоритетом мають пріорітетніший доступ до її ресурсів.

HDFS не обмежується MapReduce завданнями. Вона може працювати з іншими прикладними програмами, багато з яких є розробками Apache, наприклад, база даних HBase, система машинного навчання Apache Mahout, система Apache Hive Data Warehouse. Теоретично, Hadoop може використовуватися для будь якого типу завдань, які є швидше пакетно-орієнтованими, ніж тими, що виконуються у реальному часі, а також для завдань з дуже інтенсивними даними і завдань, для яких корисна паралельна обробка даних. Hadoop також може використовуватися для доповнення системи реального часу, такої як, наприклад, лямбда архітектура, Apache Storm, Flink або Spark.

Hadoop Common та планувальник Yarn. Hadoop Common забезпечує інструменти, які дозволяють комп'ютерним системам користувача читати дані, що зберіга-ються у файловій системі Hadoop.

YARN керує ресурсами систем, які зберігають дані та виконують їх аналіз.

Використання Hadoop. Hadoop також містить безліч інших інструментів з відкритим кодом, призначених для створення додаткових функцій на компонентах ядра Hadoop.

Так, Apache Tez є фреймворком наступного покоління, який може використовуватися замість Hadoop MapReduce, в якості двигуна. Amazon EMR включає конектор EMRFS, який дозволяє Hadoop використовувати для зберігання даних сховище Amazon S3. Amazon EMR також може використовуватися для легкого встановлення та налаштування у кластері таких інструментів, як Hive, Pig, Hue, Ganglia, Oozie та HBase. Окрім Hadoop на Amazon EMR можна запускати інші фреймворки, такі як Apache Spark для обробки даних у пам'яті або Presto для виконання інтерактивних запитів.

Гнучка природа системи Hadoop дозволяє компаніям, коли вони потребують змін, додавати або змінювати власну систему даних, використовуючи дешеві та легко-доступні частини від будь-яких постачальників інформаційних систем. Сьогодні Hadoop є найбільш використовуваною системою для зберігання та обробки даних на виробничому апаратному забезпеченні. Hadoop використовують майже всі великі постачальники он-лайн продуктів, та кожний має можливість його вільно модифікувати відповідно до своїх цілей. Ці зміни, які вносять до ПЗ експерти, наприклад, Amazon чи Google, відсилаються до спільноти розробників, де вони часто використовуються в подальшому для вдосконалення "офіційного" продукту. Така форма колаборативної розробки є ключовою властивістю ПЗ з відкритим кодом.

Слід зазначити, що використання базових модулів Hadoop Apache є складним навіть для фахівця галузі інформацій-них технологій, тому були розроблені комерційні версії продукту такі, як наприклад, Cloudera, що спрощують задачу інсталяції та запуску Hadoop, а також пропонують послуги навчання та підтримки. Завдяки гнучкій природі Hadoop, компанії при розширенні бізнесу мають можливість корегувати та розширювати операції аналізу даних. Підтримка спільноти відкритого коду робить аналіз великих даних доступним для кожного.

Apache Spark

Apache Spark – фреймворк (містить більш ніж 80 операторів для роботи з даними) з відкритим кодом, який був створений для розподіленої обробки великих даних. На відміну від класичного обробника з ядра Hadoop, що реалізує дворівневу концепцію MapReduce з дисковим сховищем, він використовує спеціалізовані примитиви для рекурентної обробки в оперативній пам’яті, завдяки чому дозволяє отримувати значне прискорення роботи для деяких класів задач. Зокрема, можливість багатократного доступу до даних користувача, що завантажені в оперативну пам’ять, робить бібліотеку дуже привабливою для алгоритмів машинного навчання. Фактично, Spark є переосмисленим MapReduce, але працює у 10-100 разів швидше, залежно від того, працює він в пам’яті або на диску. Spark підтримує мови програмування Scala, Python, Java, R.

Головним поняттям у Spark є Resili-ent Distributed Dataset (RDD) – це розподілена структура даних, яка розміщується в оперативній пам’яті (рис. 4.4). Кожний RDD є фрагментом даних, що розподілені по вузлах кластера. RDD є незмінними структурами, тому після виконання перетворень створюються нові RDD. RDD обробляються паралельно за допомогою трансформацій/дій, які виконуються одночасно во всіх розділах (partition). RDD є відмовостійкими: якщо розділ втрачається в результаті відмови вузла, він може бути відновлений з вихідних джерел.

Рис. 4.4. Розподілення RDD

Фактично RDD являє собою набір даних, над яким можна виконувати перетворення двох типів: трансформації та дії. Відповідно, вся робота з цими структурами полягає у послідовності цих перетворень.

Трансформація. Як правило, перетворює якимось чином елементи даного набору даних. Результатом застосування її до RDD є новий RDD. Далі наведений неповний перелік найрозповсюдженіших транс-формацій, кожна з яких повертає новий RDD (рис. 3):

 map(f) – застосовує функцію f до кожного елемента набора даних;

 filter(f) – повертає всі елементи набору даних, на яких функція f повернула істинне значення;

 distinct([numTasks]) – повертає набір даних, який містить унікальні елементи вихідного набору даних;

Також підтримуються наступні операції над множинами:

 union(Dataset) – об’єднання з на-бором даних Dataset,

 intersection(Dataset) – перетин з набором даних Dataset,

 cartesian(Dataset) – результатом операції є новий набір даних, який містить пари (A,B), де A належить вихідному набору даних, а B – набору даних Dataset.

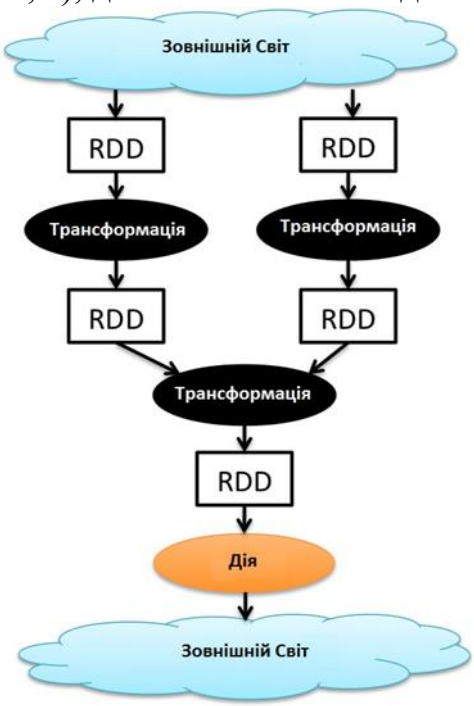

Рис. 4.5.Робота з RDD

Дії. Застосовуються, коли необхідно матеріалізувати результат – як правило, зберегти дані на диску, або вивести частину даних у консоль. Найбільш розповсюдженими діями, які можна застосувати до RDD, є:

 saveAsTextFile(path) – зберігає дані у текстовому файлі (hdfs) на локальну машину або у будь-яку іншу файлову систему, яка підтримується, path визначає шлях для збереження файлу;

 collect() – повертає елементи набора даних у вигляді масива. Як правило, використовується після застосування до набору даних фільтрів та перетворень для візуалізації або додаткового аналізу результату;

 take(n) – повертає у вигляді масива перші n елементів набору даних;

 count() – повертає кількість елементів у наборі даних;

 reduce(f). Функція f (приймає на вхід 2 аргументи, повертає одне значення) повинна обов'язково бути комутативною та асоціативною.

Spark не змушує думати в парадигмі MapReduce, а дозволяє створювати зрозумілий код, який спрямований саме на виконання поставленої бізнес-задачі. Фреймворк бере на себе розподілення та фрагментацію кода та даних, які автоматично передаються на кластер.

Spark має також колекцію бібліотек (набір готових алгоритмів, підходів та практик), що дозволяють комбінувати існуючі рішення в межах одного програмного кода для досягнення поставленої мети. На цей час Spark містить наступні бібліотеки:

 Spark SQL;

 Spark Streaming (аналіз у реальному часі);

 MLib (машинне навчання);

 GraphX (робота з графами).

Spark SQL. Це модуль Apache Spark, який є частиною ядра Spark та інтегрує реляційну обробку даних та процедурний API Spark. Він може працювати ра-зом з Hive (HiveQL/ SQL) або його заміщу- вати. Окрім цього, модуль здатний взаємодіяти з інструментами бізнес-аналітики.

Spark SQL підтримує реляційну обробку даних як в межах програм Spark (через RDD), так і з зовнішніх джерел даних. Він може взаємодіяти з новими джерелами даних, включаючи слабоструктуровані дані та зовнішні бази даних, що підтримують федеративні запити.

Spark SQL реалізує та оптимізує реляційну обробку, підтримуючи наступні підходи:

 перетворення даних у більш ефективні формати (з точки зору сховища, мережі та операцій введення/ виведення), зокрема, в різні формати, що орієнтовані на стовбці (columnar format);

 розбиття даних на секції;

 зменшення кількості операцій читання на основі статистики;

 оптимізація операцій над даними;

 виконання оптимізації наскільки можливо пізніше, коли доступна вся інформація по конвейєрах даних.

Spark SQL використовує оптимізатор запитів Catalyst для інтелектуального планування запитів.

Spark SQL може підтримувати пакетний та потоковий SQL. Ядро Spark забезпечує обробку пакетних навантажень через RDD. RDD можуть посилатися на статичні набори даних, а за допомогою розвиненого API Spark можна маніпулювати RDD в оперативній пам’яті із застосуванням «ледачих» обчислень.

Spark Streaming. Реалізує абстракцію DStream (discretized stream, дискретизований потік), що являє собою безперервний потік даних. DStream може бути створений з потоку вихідних даних; на основі таких джерел, як Kafka або Flume, або за допомогою виконання операцій з іншими DStream. По суті, DStream є послідовністю RDD (рис. 4.6).

Рис. 4.6. Структура DStream

RDD, що створений за допомогою DStream, можна перетворювати у Data-Frame та виконувати SQL запити до нього. Доступ до потоку може надаватися для будь-якої зовнішньої прикладної програми, що підтримує SQL, за допомогою JDBC-драйвера. Пакети потокових даних зберігаються у пам’яті вузла. До цих даних можна будувати інтерактивні запити, використовуючи SQL або API Spark. Для виконання SQL-запитів до Dstream використо-вується StreamSQL, що поєднує Spark Streaming з Catalyst. StreamSQL є розши-ренням SQL, яке додатково забезпечує підтримку наступних потокових операцій:

 виборка (SELECT) з потоку для обчислення функцій або фільтрації даних (за допомогою умови WHERE);

 з’єднання (JOIN) потоку з одним або декількома наборами даних для створення нового потоку;

 застосування віконних функцій та виконання агрегацій. Потік можна налаштувати таким чином, щоб він створював набори даних обмеженого розміру. За допомогою віконних функцій можна виконувати складний відбір повідомлень на основі значень полей. Після створення обмеженого пакета можна виконувати аналітику на ньому.

В основу підходу для реалізації аналітики реального часу покладено лямбда-архітектуру, що застосовується для створення аналітичних систем реального часу в контексті великих потокових даних (рис. 4.7).

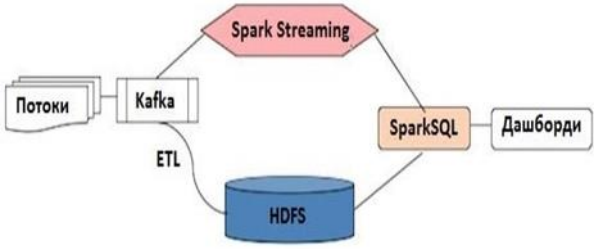

Рис. 4.7. Логічна схема реалізації аналітики великих даних в реальному часі за допомогою Spark SQL

MLib. Бібліотека для машинного навчання. Її метою є зробити машинне навчання масштабованим та простим. Вона містить розповсюджені алгоритми і утіліти машинного навчання, та дозволяє розпаралелювати на кластері алгоритми машинно-го навчання (класифікація, регресія, кластеризація і т. і.) лише за пару строк коду. Окрім цього, SparkMLib якісно працює з локальними даними, використовуючи пакет лінійної алгебри Breeze. MLib має добре продуманий API, працює з даними у будь-якому форматі на базі Hadoop та не потребує попереднього встановлення.

GraphX. Розподілений фреймворк обробки графів на основі Apache Spark. Графи є наявною та простою для розумін-ня моделлю даних. Розподілені обчислення кардинально спростили зберігання та обробку графів.

Головним механізмом ітерації графа в GraphX є розроблений Google алгоритм Pregel. Головною ідеєю цього алгоритму є передача повідомлень між вузлами у графі, які називають супер кроками завдяки послідовності ітерацій. Ітерація часто формується як ”think like a vertex”, тобто стан поточного вузла залежить лише від стану його сусідів. Використання Pregel є особливо доцільним, коли задачу складно вирішити за допомогою звичайного MapReduce.

Головним примитивом для обходу графа у GraphX є триплет: поточний вузол, вузол, до якого здійснюється перехід, та ребро між ними. Pregel вимагає визначення відстані між вузлами за замовченням, як правило, це PositiveInfinity – UDF (user defined function) функція для кожного вуз-ла, що дозволяє обробити вхідне повідомлення та порахувати наступний вузол, а також UDF функції для злиття двох вхідних повідомлень. Ці функції повинні бути комутативними та асоціативними.

GraphX містить статичну та динамічну версії реалізації алгоритму PageRank, який для кожного вузла графа призначає вагу серед решти вузлів. Наприклад, якщо користувач Твітера має велику кількість підписок від інших користувачів, то він буде мати високий рейтинг, тобто, його можна буде легко знайти у пошуковій системі.

Статична версія має фіксовану кількість ітерацій, тоді як динамічна версія буде працювати доки рейтинг не почне зходитися до заданого значення.

Через те що GraphX побудований на основі незмінних RDD, графи теж незмінні, тому GraphX непридатний для роботи з графами, які оновлюються, тим більше транзакціями, як у графових БД.

GraphX надає два окремі API для реалізації масово паралельних алгоритмів (таких як PageRank): Pregel-подібний та більш загальний — MapReduce API.

Основні типи NoSQL сховищ

На сьогодні виділяють чотири основних типи NoSQL сховищ:

 сховище «ключ-значення». В ньому є велика хеш-таблиця, що містить ключі та значення. (Приклади: Riak, Amazon DynamoDB);

 документоорієнтоване сховище. Зберігає документи, які складаються з тегованих елементів. (Приклад: CouchDB);

 стовпчикове сховище. У кожному блоці зберігаються дані лише з однієї колонки. (Приклади: HBase, Cassandra);

 сховище на основі графів. Мережеве сховище, яке використовує вузли та ребра для відображення та зберігання да-них. (Приклад: Neo4J).

Сховище типу «ключ-значення». Відсутність схеми у сховищах типу «ключ-значення» є важливою перевагою для зберігання великих даних. Ключ може бути синтетичним або автосгенерованим, а значення може бути представлено строкою, JSON, блобом (BLOB, Binary Large Object) тощо. Такі сховища, як правило, використовують хеш-таблицю, яка містить унікальний ключ та посилання на певний об’єкт даних. Існує поняття блока – логіч-ної групи ключів, що фізично не поєднують дані у групи. У різних блоках можуть бути ідентичні ключи.

Продуктивність обробки даних значно збільшується за рахунок механизмів хешування, що працюють на основі маппінгів. Щоб прочитати значення, необхідно знати ключ та блок, оскільки насправді ключ є хешем (блок + ключ).

Модель «ключ-значення» проста в реалізації. Такі сховища є доступними та толерантними до розділення, але явно програють у питаннях погодженості даних. В якості недоліків сховищ типу «ключ-значення» можна зазначити:

 модель не надає стандартних можливостей баз даних таких, як атомарність транзакцій або погодженість даних при одночасному виконанні декількох транзакцій. Такі можливості повинні надаватися самою прикладною програмою.

 при збільшенні об’ємів даних, підтримка унікальних ключей може стати проблемою. Для її вирішення необхідно якось ускладнити процес генерації строк, щоб вони залишалися унікальними серед дуже великої множини ключей.

Документоорієнтоване сховище. Дані, які представлені парами ключ-значення, стискаються як сховище документів, що є схожим зі сховищем «ключ-значення». Але на відміну від сховища «ключ-значення», документи, які зберігаються, мають визначену структуру та кодування даних. Деякі зі стандартних розповсюджених кодировок, що використовуються – це XML, JSON та BSON.

Однією з ключових відмінностей між сховищами «ключ-значення» та документоорієнтованим є те, що останнє включає метадані, які пов’язані зі вмістом, що зберігається. Це надає можливість робити запити на основі цього вмісту. Найпопулярнішими прикладами документоорієнтованих сховищ є CouchDB та MongoDB. CouchDB використовує JSON для зберігання даних, JavaScript в якості

мови запитів з використанням MapReduce та HTTP для API. Дані та відношення не зберігаються в таблицях так, як це відбувається у традиційних реляційних БД, а за сутністю є набором незалежних документів. Той факт, що такі сховища працюють без схеми, спрощує задачу додавання полів до JSON-документа без необхідності попереднього заявлення про зміни.

Стовпчикове сховище. У стовпчикових NoSQL сховищах дані зберігаються у комірках, що згруповані у стовпчики, а не строки даних. Стовпчики логічно групуються у стовпчикові сімейства. Стовпчикові сімейства можуть складатися з практично необмеженої кількості стовпчиків, які можуть створюватися під час роботи програми або під час визначення схеми. Читання та запис відбувається із використанням стовпчиків, а не строк.

Порівняно зі зберіганням даних у строках, як у більшості реляційних БД, переваги зберігання у стовпчиках полягає у швидкому пошуці /доступі та агрегації даних. Реляційні БД зберігають кожну строку як безперервний запис на диску. Різні строки зберігаються у різних місцях на диску, в той час як стовпчикові сховища зберігають всі комірки, що відносяться до стовпчика, як безперервний запис, що прискорює пошук/доступ.

Стовпчикові сховища використовують наступну модель даних:

- стовпчикове сімейство – структура, яка може легко групувати колонки та суперколонки;
- ключ – постійне ім’я запису. У ключів може бути різна кількість стовпчиків, тому сховище може розширюватися нерівномірно;
- простір ключів – визначає зов-нішній рівень організації, як правило, ім’я прикладної програми/БД.
- стовпчик – має впорядкований список елементів – кортежів з іменами та значеннями.

Найвідомішими представниками стовпчикових сховищ є Google BigTable та HBase з Cassandra.

BigTable є високопродуктивним, стислим та пропрієтарним сховищем даних від Google. Воно має наступні атрибути:

- розрідженість – деякі комірки можуть бути порожніми;
- розподіленість – дані розділені між багатьма вузлами;
- постійність – дані зберігаються на диску;
- багатомірність – більш ніж одне вимірювання;
- співставлення – ключ та значення;
- відсортованість.

На стовпчики можна посилатися за допомогою стовпчикового сімейства.

Графове сховище. У графовому сховищі немає строгого формату SQL або представлення таблиць та стовпчиків, замість цього використовується гнучке графічне представлення, яке ідеально підходить для вирішення проблем масштабованості. Графові структури використовуються разом із ребрами, вузлами та властивостями, що забезпечує безіндексну суміжність. При використанні графового сховища дані можуть бути легко перетворені з однієї моделі в іншу (рис. 4.8).

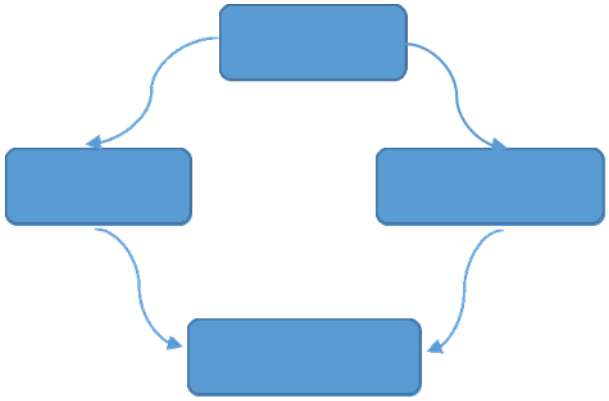

Рис. 4.8. Принципи використання графової моделі

 Такі сховища використовують ребра та вузли для представлення даних.

 Вузли пов’язані між собою певними відношеннями, які представлені ребрами між ними.

 Вузли та відношення мають деякі властивості.

Розмічений, спрямований, атрибутований мультиграф (рис. 4.9) – це граф, який містить вузли, які помічені певними властивостями та які мають зв’язки один з одним, що представлені спрямованими ребрами. Наприклад, зв’язок «Аліса знає Боба» виражена ребром з відповідними властивостями. Будь-який рейтинг «вам рекомендовано», представлений на різних сайтах, часто вираховується виходячи з того, як інші користувачі оцінили продукт. Графові БД відмінно підходять для вирішення такого типу задач.

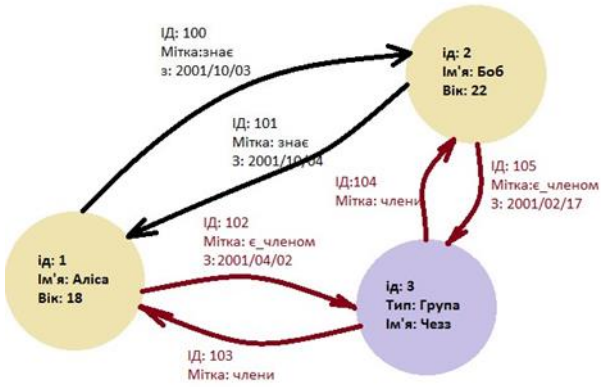

Рис. 4.9. Приклад атрибутованого мультиграфу

Прикладами найпопулярніших графових сховищ є InfoGrid та Infinite Graph. InfoGrid дозволяє з’єднувати множини ребер та вузлів, що спрощує представлення набора інформації зі складними взаємними посиланнями. InfoGrid пропонує два типи сховищ:

 MeshBase — підходить для автономного розгортання;

 NetMeshBase — підійде для великих розподілених графів та має додаткові можливості для взаємодії з іншими подібними сховищами.

Постачальники та ПЗ великих даних

В області розробки підходів до ро-боти з пам’яттю найбільшим чемпіоном є SAP зі своєю Hana платформою, але, слід зазначити, що зараз Microsoft та Oracle також вводять спеціальні опції для роботи з пам’яттю для своїх провідних баз даних. Постачальники ПЗ, що фокусуються на використанні аналітичних БД, включаючи Actian, HP Vertica та Teradata ввели спеціальні опції для співвідношення високих-ОЗУ-дисків, а також пропонують інструменти для розміщення конкретних даних у пам’яті для виконання ультра-швидкого аналізу. Прогрес, що має місце у зростанні пропускної здатності та обчислювальної потужності, вдосконалив й можливості потокової обробки та проведення аналізу в реальному часі.

До великих постачальників сервісів обробки даних можна віднести IBM, Microsoft, Oracle, SAP, які пропонують все від ПЗ інтеграції даних та систем керування базами даних (DBMS) до ПЗ для бізнес-аналізу та аналітичної обробки, а також Hadoop опцій для роботи з пам’яттю та потокової обробки. Teradata більш вузько зосереджений на керуванні даними, та подібно Pivotal, він має тісні зв’язки з лідером аналітичного ринку SAS.

Багато постачальників пропонують реалізовані окремі опції хмарних технологій, але такі розробники, як 1010data та Amazon Web Services (AWS) використовують хмарну модель у повному обсязі в своєму ПЗ. З них двох Amazon має найширшу вибірку продуктів і є очевидним вибором для тих, хто працює з великими навантаженнями та зберігає велику кількість даних на AWS платформі. 1010data має високо масштабований сервіс БД та підтримує можливості управління інформацією, бізнес-аналіз та аналітику, що обслуговуються в стилі приватної хмари.

Hadoop довів свою користь та переваги у вартості там, де є екстремальними об’єм та різноманітність даних. На сього-дні це найбільш відомий та поширений програмно-апаратний комплекс для роботи з великими даними. Він виявився на-стільки гарним, що став фундаментом декількох комерційних реалізацій на його основі, а саме: Cloudera, MapR та Horton-works, кожна з яких пропонує власний дистрибутив. На сьогодні всі постачальники традиційних BI-систем, як Micro-Strategy або SAS, забезпечують інтерфейс з Hadoop. Виробники MPP-систем (масово-паралельних архітектур) у свою чергу забезпечують суттєво більш міцну інтеграцію з Hadoop, коли дані, що зберігаються і в Hadoop, і в реляційній СКБД, можуть оброблятися в одному SQL-запиті. Oracle, IBM, Teradata. Cloudera, Hortonworks та MapR, що також включили Hadoop до своїх продуктових лінійок, роблять все можливе, щоб перемістити Hadoop з високо-масштабованого зберігання даних та Map Reduce обробки у світ аналітики.

Менші постачальники такі як Ac-tian, InfiniDB/Calpont, HP Vertica, Info- bright та Kognitio, навпаки фокусуються загалом скоріше на аналітиці, ніж на обробці транзакцій.

Такі постачальники аналітики, як Alpine Data Labs, Revolution Analytics та SAS, працюють з платформами, які забезпечуються сторонніми постачальниками СКБД та дистриб’ютерами Hadoop, хоча, зокрема, SAS розмиває це розмежування зі зростаючою підтримкою для середовищ SAS-керованих рядів даних «у-пам’яті» та Hadoop. NoSQL та NewSQL СКБД фокусуються на високо-масштабованій обробці транзакцій, а не на аналітиці.

Взагалі програмні засоби для роботи з великими даними не заміщують решту інструментів обробки, бізнес-аналітики, візуалізації та прогнозування, а лише допомагають підтримати терабайти нових даних та спрямовують їх у потрібне русло.

Так, відповідно до аналітичних платформ для великих даних, деякі експерти вважають найбільш універсальною платформу Pentaho, а для вирішення задач машинного самонавчання, таких як, на-приклад, кластеризація, класифікація, регресія та інші, краще підходять Mahout та Spark. Серед найбільш технологічних MPP – платформ спеціалісти виділяють Vertica та Teradata Aster. Останнім часом з'явилася множина платформ, які підтримують швидку аналітику для великих да-них, наприклад, MemSQL або Splice Machine.

Окремої уваги заслуговує Intel платформа з відкритим кодом для Hadoop. Привабливість рішення Intel для Hadoop обумовлює також й фактор "апаратного забезпечення", а саме – оптимізація, що виконана Intel з урахуванням архітектури процесорів Xeon та специфіки роботи твердотільних накопичувачів з контролерами Intel, дозволяє досягти значного приросту продуктивності. Процесори Xeon прискорюють операції шифрування або дешифрування за алгоритмом AES (Advanced Encryption Standard), що реалізується за допомогою додаткового набору команд AES-NI (New Instruction). Окрім цього, платформа Intel для Hadoop також пропонує розширені можливості у галузі обробки потокових даних.

Різноманіття платформ для роботи з великими даних доповнюється величезною кількістю прикладних програмних продуктів, комерційних чи безкоштовних, для аналітичної обробки таких даних. Нижче наведений невеликий перелік найпоширеніших прикладів такого ПЗ:

 Cluvio – сучасна платформа аналітики даних, що дозволяє виконувати SQL запити, обробляти дані, візуалізувати результати та створювати гарні, інтерактивні дашборди за лічені хвилини. Підтримує потужне вбудовування, що дозволяє додавати аналітичні властивості до будь-якого веб-сайту або веб-застосунку.

 IBM SPSS Statistics дозволяє виявляти нові зв'язки між даними та будувати прогнози. Він дозволяє отримувати легкий доступ до даних, управляти та аналізувати набори даних, не маючи попереднього статистичного досвіду. Це дозволяє практично виключити довготривалу підготовку даних та швидко створювати, маніпулювати та розповсюджувати інформацію для прийняття рішень.

 Qlik Sense Desktop – безкоштовний продукт, що надає можливість інтерактивного створення звітів та дашбордів з діаграмами та графіками. Програма візуалізації спрощує аналіз даних та допомагає створювати інформовані бізнес-рішення швидше, ніж будь-коли раніше. Перетворення електронних таблиць у більш чіткі візуалізації робить процес аналізу простішим та швидшим для перегляду всіх користувачів.

 Elasticsearch – розповсюджений пошуковий та аналітичний движок на базі Apache Logstash, Kibana та Beats складають "Elastic Stack", розроблений фірмою Elastic. Також Elasticsearch забезпечується хостінг Elastic Cloud.

 Cyfe – бізнес-панель для управління даними компанії за допомогою звітів, попередньо побудованих віджитів тощо.

 Forestpin Analytics – платформа аналізу даних для знаходження нерівностей, кореляцій та дублювань за допомогою простого дашборда.

Кількість підприємств, що використовують великі дані, безперервно зростає. Практика останніх років продемонструвала, що застосування результатів аналізу великих масивів даних може принести реальний ефект. Але, окрім переваг існує велика кількість проблем, вирішення яких вимагає застосування досить значних ресурсів.

Для систем, що отримують аналітичні дані в масштабі, близькому до реального часу, ключовими є вимоги не лише до продуктивності, але й до часу відгуку (наприклад, IBM каже про час відгуку, менший за мілісекунди). Це дуже обмежує вибір аналітичних платформ. Неможливо використовувати колосальні обчислювальні можливості Hadoop, якщо накладні витрати на ініціювання та завершення тривіальної MapReduce-програми складають десятки секунд. Забезпечити прийнятний час відгуку можуть або досить недешеві MPP-платформи (такі як Netezza, Teradata, Greenplum), або розподілені системи з розвиненою індексацією або високим рівнем резидентності даних в оперативній пам’яті.

Багато аналітичних систем все ще використовують реляційну модель даних, внаслідок чого вибір платформ обмежується такими рішеннями, як GridGain або Gigaspaces XAP. Для роботи з потоковими даними в режимі он-лайн були створені технології Storm, Spark Streaming та Akka. Але аналіз даних за допомогою SQL на Hadoop не дозволяє досягти того максиму-му, який пропонує платформа.

Компанії обирають Hadoop, щоб збирати складні та різноманітні дані: історія відвідувань веб-сайтів, логи, дані про використання мобільних пристроїв й інформації з соцмереж та багато іншого. Цими даними складно оперувати у СКБД. Можна витягувати структуровані дані з Hadoop для SQL-аналізу, але більш перспективними є такі підходи як машинне самонавчання та інші, що дозволяють спів-віднести нові дані зі вже накопиченою, проаналізованою та структурованою інформацією. BI та SQL системи досить добре себе проявили, але постійно виникають нові потреби та нові питання, що виходять за межі поточних можливостей. Уже не достатньо просто управляти даними. Окрім цього, компанії не можуть покладатися лише на аналітику, вони потребують також рішень зі сфери BI, системи збору та передачі оперативної інформації та інше. Межа між цими поняттями почала стиратися, а SAS, Alpine Data Labs та інші стали підтримувати кластеризовані серверні середовища, вимогливі до пам’яті та Hadoop.