МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ВОЛОДИМИРА ДАЛЯ

КОНСПЕКТ ЛЕКЦІЙ
з дисципліни
«Методи обробки великих даних»

(для магістрів спеціальності 126 «Інформаційні системи і технології»

(Електронне видання)

ЗАТВЕРЖДЕНО
на засіданні кафедри
програмування та математики
Протокол № 10 від 18.06. 2021 р.

Северодонецьк- 2021

УДК 512.64

Конспект лекційі з дисципліни «Методи обробки великих даних» *(для магістрів спеціальності 126 «Інформаційні системи і технології» (Електронне видання)*/ Уклад.: В.Г. Іванов - Северодонецьк: вид-во СНУ ім. В. Даля. 2021. – 55 с.

Конспект лекцій спрямований на вивчення і засвоєння здобувачами вищої освіти самостійно та на підставі лекційного матеріалу теоретичної основи та практичного матеріалу. Конспект лекцій містить теоретичні відомості, необхідні для виконання самостійної роботи за змістом робочої програми дисципліни, перелік рекомендованої літератури.

Конспект лекцій розрахований на студентів вищих навчальних закладів.

Укладач: В.Г. Іванов., к.т.н., доц.

Рецензент: Д.В. Ратов., к.т.н., доц.

Зміст:

Вступ

Термін Big Data з'явився порівняно недавно. Google Trends показує початок активного росту вживання словосполучення починаючи з 2011 року

Big Data це не якийсь конкретний обсяг даних і навіть не самі дані, а методи їх обробки, які дозволяють распредёлено обробляти інформацію. Ці методи можна застосувати як до величезних масивів даних (таким як зміст усіх сторінок в інтернеті), так і до маленьких (таким як вміст цієї текст).

У розділі 1 вводиться поняття даних, дається короткий огляд підходів і визначень, розповідається про життєвий цикл даних. Це важливо для розуміння того, що є дані. Найчастіше робота з Великими даними істотно спрощується за рахунок роботи з їх метаданими, тому в розділі 2 наводяться короткі відомості про метаданих і про життєвий цикл метаданих. У розділі 3 представлений огляд екосистеми Великих даних, розкривається тема систем управління Великими даними. Глава 4 містить короткий огляд архітектур систем обробки Великих даних. У розділі 5 дається уявлення про Hadoop / MapReduce і про паралельних алгоритмах для роботи з даними. У розділі 6 в стислій формі розповідається про програмних платформах і системах для Великих даних. У розділі 7 викладається матеріал про обладнання для обробки Великих даних.

Всі глави забезпечені питаннями для перевірки засвоєних знань.

У бібліографічному списку наведені інформаційні джерела, посилання на які зустрічаються в тексті даного навчального посібника. Окремо також наведено список інформаційних джерел для самостійного вивчення, що дозволяє зануритися в предметну область Великих даних і отримати більш ґрунтовне уявлення про цю області знань.

1. Дані. Підходи і визначення

1.1. Визначення даних

Дані - подання інформації у формалізованому вигляді, придатному для передачі, інтерпретації та обробки.

Дані - інформація, оброблена і подана у формалізованому вигляді для подальшої обробки.

У Кембриджському словнику наводяться такі визначення даних: дані - це інформація, особливо факти і числа, зібрані для подальшого використання при прийнятті рішень. Дані - це інформація в електронній формі, придатна для зберігання і використання комп'ютером. [13]

На сьогоднішній день також досить широко поширена формулювання, яка полягає в тому, що дані є нафтою цифрової економіки. [12]

1.2. Філософський підхід

Початково поняття даних - філософське, воно виникає в епістемології при розгляді основою проблеми гносеології - пізнаваності світу, пошуку і осмислення істини. Процедури верифікації або фальсифікації даних створюють інформацію, осмислення істини створює знання.

Філософія розглядає перетворення відомостей в дані, даних в інформацію, а інформації - в знання. Істинність відомостей суб'єктивна. Відомості, виражені в формальному поданні, є даними. Обробка даних дозволяє визначити скільки в них міститься інформації. При осмисленні інформації експертом створюються знання.

1.3. Формальний підхід

Визначення інформації як такої, а також пов'язані з інформацією об'єкти і дії:

1) інформація - відомості (повідомлення, дані) незалежно від форми їх подання;
2) інформаційні технології - процеси, методи пошуку, збору, зберігання, обробки, надання, поширення інформації та способи здійснення таких процесів і методів;
3) інформаційна система - сукупність міститься в базах даних інформації і забезпечують її обробку інформаційних технологій і технічних засобів;
4) інформаційно-телекомунікаційна мережа - технологічна система, призначена для передачі по лініях зв'язку інформації, доступ до якої здійснюється з використанням засобів обчислювальної техніки;
5) власник інформації - особа, яка самостійно створила інформацію або отримала на підставі закону або договору право дозволяти чи обмежувати доступ до інформації, яка визначається за будь-якими ознаками;

6) доступ до інформації - можливість отримання інформації та її використання;
7) конфіденційність інформації - обов'язкове для виконання особою, яка одержала доступ до певної інформації, вимога не передавати таку інформацію третім особам без згоди її власника;
8) надання інформації - дії, спрямовані на отримання інформації певним колом осіб або передачу інформації визначеному колу осіб;
9) поширення інформації - дії, спрямовані на отримання інформації невизначеним колом осіб або передачу інформації невизначеному колу осіб;
10) електронне повідомлення - інформація, передана або отримана користувачем інформаційно-телекомунікаційної мережі;
11) документована інформація - зафіксована на матеріальному носії шляхом документування інформація з реквізитами, що дозволяють визначити таку інформацію або в установлених законодавством України випадках її матеріальний носій;

11.1) електронний документ - документована інформація, представлена в електронній формі, тобто у вигляді, придатному для сприйняття людиною з використанням електронних обчислювальних машин, а також для передачі по інформаційно-телекомунікаційних мереж та обробки в інформаційних системах;
12) оператор інформаційної системи - громадянин або юридична особа, які здійснюють діяльність з експлуатації інформаційної системи, в тому числі по обробці інформації, що міститься в її базах даних;
13) сайт в мережі "Інтернет" - сукупність програм для електронних обчислювальних машин та іншої інформації, що міститься в інформаційній системі, доступ до якої забезпечується за допомогою інформаційно-телекомунікаційної мережі "Інтернет" (далі - мережа "Інтернет") по доменних іменах і (або) по мережевих адрес, що дозволяє ідентифікувати сайти в мережі "Інтернет";
14) сторінка сайту в мережі "Інтернет" (далі також - інтернет-сторінка) - частина сайту в мережі "Інтернет", доступ до якої здійснюється за вказівником, що складається з доменного імені і символів, визначених власником сайту в мережі "Інтернет";
15) доменне ім'я - позначення символами, призначене для адресації сайтів в мережі "Інтернет" в цілях забезпечення доступу до інформації, розміщеної в мережі "Інтернет";
16) мережеву адресу - ідентифікатор в мережі передачі даних, що визначає при наданні телематичних послуг зв'язку абонентський термінал або інші засоби зв'язку, що входять в інформаційну систему;
17) власник сайту в мережі "Інтернет" - особа, яка самостійно і на свій розсуд визначає порядок використання сайту в мережі "Інтернет", в тому числі порядок розміщення інформації на такому сайті;
18) провайдер хостингу - особа, яка надає послуги з надання обчислювальної

потужності для розміщення інформації в інформаційній системі, постійно підключеною до мережі "Інтернет";

19) єдина система ідентифікації і аутентифікації - федеральна державна інформаційна система, порядок використання якої встановлюється Кабінетом Міністрів України, і яка забезпечує у випадках, передбачених законодавством Російської Федерації, санкціонований доступ до інформації, що міститься в інформаційних системах;

20) пошукова система - інформаційна система, що здійснює за запитом користувача пошук в мережі "Інтернет" інформації певного змісту і надає користувачеві відомості про покажчику сторінки сайту в мережі "Інтернет" для доступу до запитуваної інформації, розташованої на сайтах в мережі "Інтернет", що належать іншим особам, за винятком інформаційних систем, використовуваних для здійснення державних і муніципальних функцій, надання державних та муніципальних послуг, а також для здійснення інших публічних повноважень, встановлених федеральними законами.

Прийняті в інших країнах терміни можуть відрізнятися, наприклад, в США електронний документ позначає будь-яку інформацію в цифровій формі, де "інформація" може включати в себе дані, текст, звуки, коди, комп'ютерні програми, програмне забезпечення або бази даних. "Дані" в цьому контексті відносяться до обмеженого набору елементів даних, кожен з яких складається з вмісту або значення разом з розумінням того, що означає контент або значення; де електронний документ містить дані, це розуміння того, що елемент даних або значення елемента даних має бути явно включено в сам електронний документ або бути легко доступним для одержувача електронного документа. [4]

1.4. Життєвий цикл даних

Життєвий цикл даних - це послідовність етапів, яку конкретна порція даних проходить від початкового етапу створення або отримання до моменту архівації або видалення. [14]

Основні етапи життєвого циклу даних представлені на рис. 1.

Розглянемо ці етапи докладніше.

1.4.1. Створення даних (Data Generation / Data Capture)

На цьому етапі дані генеруються або захоплюються. Цей етап зазвичай ще ділять на три типи отримання даних:

1. Придбання даних (Data Acquisition)

   Отримання організацією даних, вже згенерованих поза підприємства.

2. Запис даних (Data Entry)

   Створення нових даних оператором або комп'ютером. Дані мають цінність для підприємства.

3. Реєстрація сигналів (Signal Reception)

Захоплення даних пристроями. Особливо важливо в системах управління, але останнім часом особливо цінно при використанні такого підходу, як Інтернет речей.

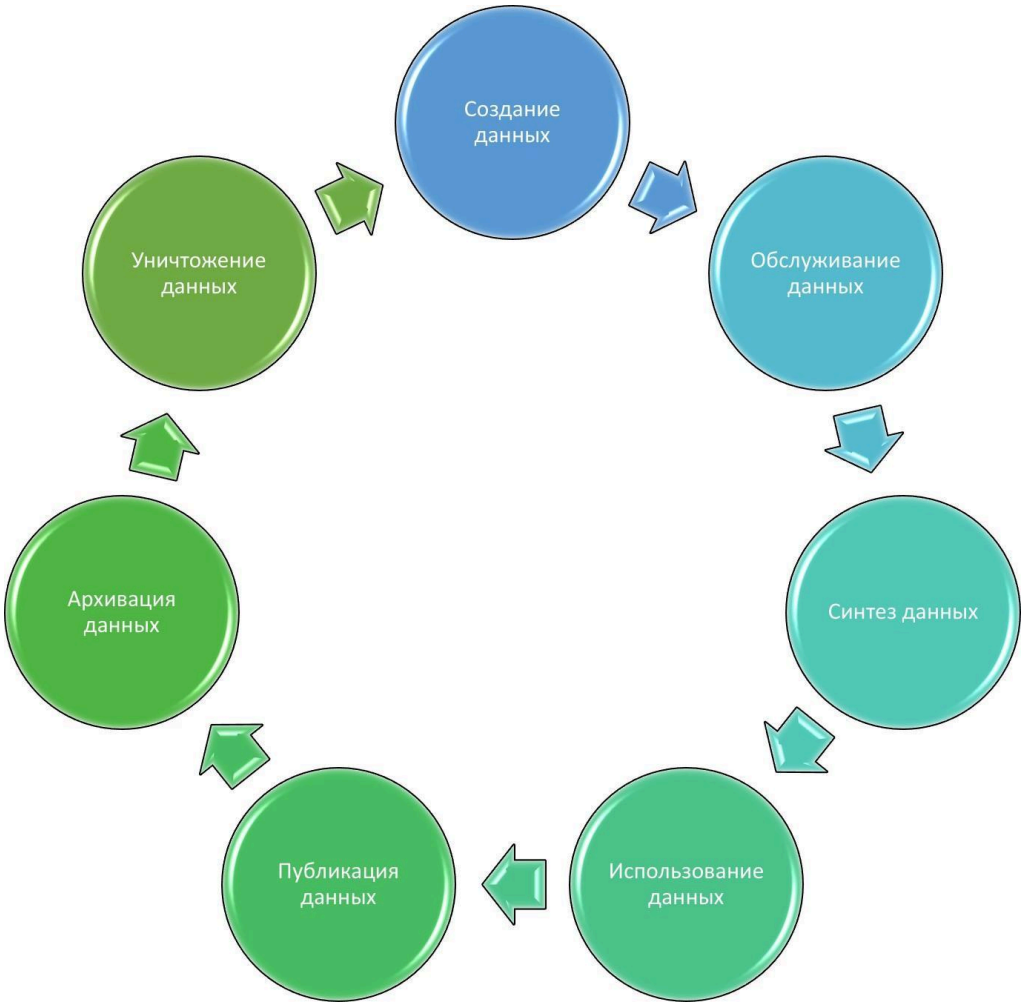

Рисунок 1 - Життєвий цикл даних

Всі три варіанти отримання даних дуже важливі в рамках розгляду процесу управління даними. [17]

1.4.2. Обслуговування даних (Data Maintenance)

Після того, як дані були створені, необхідно їх зберігати і обслуговувати. Необхідно здійснювати доставку даних в точку, де їх будуть використовувати або виробляти над ними маніпуляції (наприклад, операції синтезу).

Можна говорити про те, що обслуговування даних - це обробка даних без

отримання з вилучення з них корисної інформації для підприємства.

Найчастіше обслуговування даних включає в себе такі дії з даними, як переміщення, інтеграція, очищення, збагачення та ETL-процеси (Extract, Transform, Load).

Обслуговування даних зазвичай має на увазі застосування широкого спектра методів з області управління даними (Data Management). [17]

1.4.3. Синтез даних (Data Synthesis)

Це порівняно недавно з'явилася стадія в життєвому циклі даних. Використовується не у всіх моделях життєвих циклах даних.

Синтез даних - це процес отримання додаткової цінності з даних за допомогою використання індуктивної логіки і сторонніх інформаційних джерел.

Це стадія, на якій з даними працюю аналітики, причому вони можуть використовувати в своїй роботи методи моделювання ризиків, актуарного моделювання, моделювання для прийняття інвестиційних рішень і ін.

На цій стадії використовується індуктивна логіка, що не дедуктивна. Індуктивна логіка вимагає використання експертної думки, тому що саме компетенції експертів необхідні для побудови моделей скорингу і ін. [17]

1.4.4. Використання даних (Data Usage)

До сих пір йшлося про використання даних усередині одного підприємства, які можливо були схильні до очищення і збагачення на стадії обслуговування даних і використовувалися спільно з додатковим третіми джерелами даних на стадії синтезу даних.

На стадії використання даних вони застосовуються в якості корисної інформації для задач, які повинні виконуватися і управлятися на основі даних.

Ці завдання можуть бути поза життєвого циклу даних. Проте, дані стають все більш значущою частиною бізнес-процесів підприємств. Дані самі можуть бути продуктом або послугою (або бути частиною продукту або послуги), пропонованої підприємством.

Використання даних має спеціальні завдання в рамках управління даними (Data Governance). Одне із завдань полягає в законному використанні даних в необхідному вигляді. Це називається "дозволене використання даних" (permitted use of data). Можуть існувати регулюють або договірні обмеження на те, як фактично можна використовувати дані, а частина ролі управління даними (Data Governance) полягає в забезпеченні дотримання цих обмежень. [17]

1.4.5. Публікація даних (Data Publication)

При використанні даних можлива ситуація, коли дані відправляються за межі підприємства. У цьому випадку говорять про публікацію даних. Публікація даних - це винесення даних за межі підприємства.

Прикладом цього процесу може бути маклер, що розсилають щомісячні звіти клієнтам. Всі дані, які були розіслані, вже не можуть бути відкликані. Якщо були розіслані дані з невірними значеннями, то такі дані не можуть бути виправлені, оскільки вони вже стають недоступні для підприємства. Управління даними (Data Governance) може знадобитися, щоб допомогти прийняти рішення про те, як будуть оброблятися невірні дані, які були відправлені з підприємства. [17]

1.4.6. Архівація даних (Data Archival)

Дані можуть бути використані як одноразово, так і кілька разів.

Але потім рано чи пізно життєвий цикл даних починає підходити до кінця.

Перша стадія цього стану полягає в архівації даних.

Архівація даних - це копіювання даних в пасивну середу, в якій вони зберігається, для тих випадків, коли вони знадобляться знову в активній виробничому середовищі, і видалення цих даних з усіх активних виробничих середовищ.

Архів даних - це просто місце, де зберігаються дані, без їх обслуговування, використання або публікації. У разі необхідності дані можуть бути відновлені з архіву. [17]

1.4.7. Знищення даних (Data Purging)

Знищення даних - це послідовність операцій для виконання незворотного видалення даних, що робить неможливим як відновлення даних, так і отримання залишкової інформації (Data Remanence) про них. Це одна з найбільш складно реалізованих процедур управління даними.

Навіть з теоретичної точки зору існує команда запису значення в комірку пам'яті, але команди стирання значення як такої немає. Для знищення даних необхідно виготовити високопродуктивний джерело випадкових чисел і перезаписати ними весь носій інформації (перезапису області зберігання недостатньо, так як зберігається інформація про вихідний кількості даних).

Інакше кажучи, при знищенні даних необхідно не тільки зробити недоступними від відновлення на фізичному рівні самі дані, але і пов'язану з ними інформацію в інших наборах даних і протоколи роботи встановлюються на рівні керівних документів

Питання для перевірки

1. Що таке дані?
2. Що таке життєвий цикл даних?
3. Перерахуйте етапи життєвого циклу даних.
4. Для яких цілей потрібен етап "Синтез даних" (один з етапів життєвого циклу даних)?
5. Для яких цілей потрібен етап "Використання даних" (один з етапів життєвого циклу даних)?
6. Для яких цілей потрібен етап "Публікація даних" (один з етапів життєвого циклу даних)?
7. Для яких цілей потрібен етап "Архівація даних" (один з етапів життєвого циклу даних)?

2. Метадані

2.1. Поняття метаданих

При зборі даних виникають метадані, що містять будь-яку інформацію про зібраних даних. Наприклад, час створення набору даних, авторство і першоджерело, розмір і кодування даних - все це метадані.

Метадані - це відомості про даних.

Структуровані метадані називають онтологією або схемою метаданих.

У методичному посібнику "Онтологічний інжиніринг знань в системі PROTÉGÉ" [11] написано, що "онтологія визначає загальний словник для вчених, яким потрібно спільно використовувати інформацію в предметної області. Вона включає машинно-інтерпретовані формулювання основних понять предметної області і відносини між ними ".

Онтології зазвичай використовуються в таких областях, як штучний інтелект, семантична павутина, системна інженерія, біомедична інформатика, бібліотечна справа, інформаційна архітектура та ін. Все онтології потрібні для організації інформації.

2.2. Життєвий цикл метаданих

Життєвий цикл метаданих можна, можливо розділити на чотири стадії,представлених на рис. 2. [20]

1. Оцінка вимог і аналіз контенту.
2. Специфікація системних вимог.
3. Система метаданих.
4. Сервіс та оцінка.

Розглянемо ці стадії докладніше.

2.2.1. Оцінка вимог і аналіз контенту

Цей етап складається з чотирьох частин.

1. Виявлення та отримання базових вимог до метаданих.
2. Огляд релевантних стандартів і проектів по метаданих.
3. Дослідження вимог до глибоких метаданих.
4. Ідентифікація стратегій для схем метаданих.

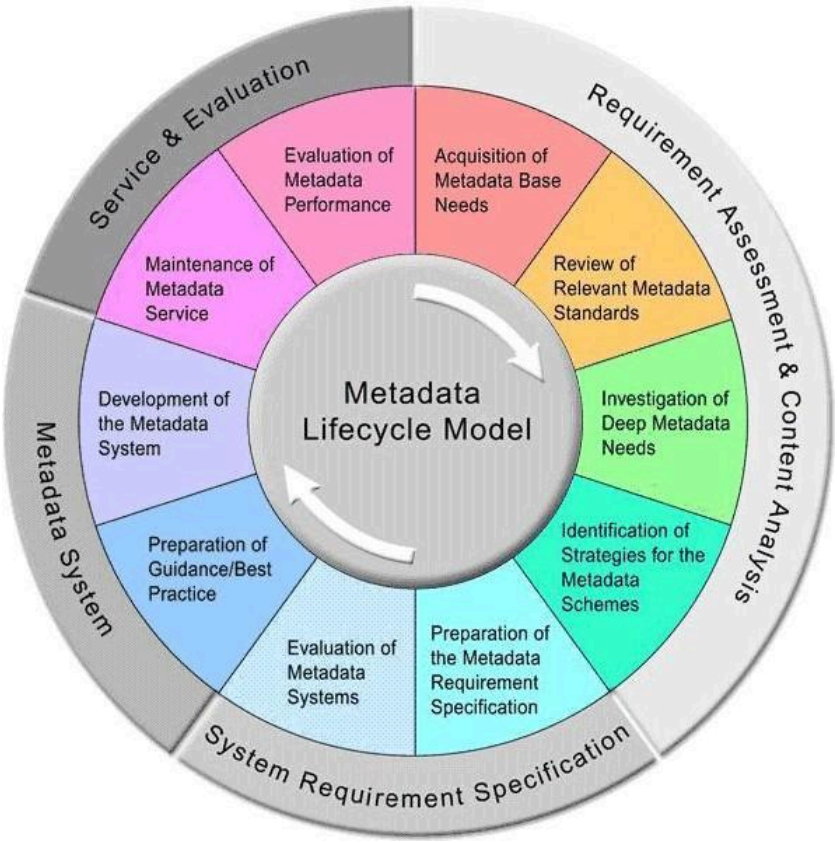

Рисунок 2 - Життєвий цикл метаданих

Виявлення та отримання базових вимог до метаданих

На цьому етапі життєвого циклу метаданих проводиться опитування експертів і фахівців з предметної області на тему вимог до метаданих в конкретному проекті і аналіз атрибутів. Метою опитування або інтерв'ювання експертів є отримання попередньої інформації про проект і встановлення контакту між фахівцями по метаданих та провайдерами даних (або контенту). На цій стадії повинні бути уточнені і роз'яснена і уточнена область метаданих, контекст метаданих, діюча система метаданих, роль і функція метаданих. [20]

Огляд релевантних стандартів і проектів по метаданих

Цей етап включає визначення стандартів метаданих, потенційно корисних для проекту, вивчення існуючих схем метаданих і варіантів їх використання. Цей етап потрібен для того, щоб можна було краще дізнатися, які відмінності існують серед інших аналогічних проектів, а також переглянути цілі проекту. [20]

Дослідження вимог до глибоких метаданих

На цьому етапі визначаються вимоги до метаданих більш детально і глибоко. Для цього використовується концепція контент-аналізу. Тут важливо уточнити область і контент метаданих, а також опрацьовано питання планованих в роботі СУБД і інформаційних систем, в яких планується використовувати метадані. Надалі ці напрацювання можуть бути використані в якості методики для масштабування. [20]

Ідентифікація стратегій для схем метаданих

На цьому етапі формулюється стратегія використання метаданих на основі всіх, хто має попередніх результатів. Стратегія включає прийняття одного або декількох існуючих стандартів метаданих та розробку на основі цих стандартів схеми метаданих. [20]

2.2.2. Специфікація системних вимог

Цей етап складається з двох частин.

1. Підготовка специфікації вимог до метаданих
2. Оцінка систем метаданих

Підготовка специфікації вимог до метаданих

Специфікація вимог до метаданих (Metadata Requirement Specification, MRS) є сполучною ланкою між учасниками проекту, фахівцями по метаданих та розробниками систем. [20]

Оцінка систем метаданих

На цьому етапі проводиться оцінка систем метаданих, які потенційно можуть використовуватися в проекті надалі. Учасники проекту можуть вибрати з існуючих систем метаданих. [20]

2.2.3. Система метаданих

Цей етап складається з двох частин.

1. Підготовка керівництва по кращій практиці.
2. Розробка системи метаданих.

Підготовка керівництва по кращій практиці

Цей етап включає в себе створення документації і розробку керівних принципів "кращої практики" для окремих елементів метаданих. Керівництво може використовуватися як контрольний список або як засіб забезпечення контролю якості записів метаданих в проекті. [20]

Розробка системи метаданих

Розробники системи розробляють інструменти і системи метаданих на основі специфікації вимог до метаданих. [20]

2.2.4. Сервіс та оцінка

Цей етап складається з двох частин.

1. Підтримка служби метаданих.
2. Оцінка якості метаданих.

Підтримка служби метаданих

Мета розробки служб метаданих - гарантувати якість метаданих і механізмів роботи з ними. Модель сервісу метаданих складається з трьох основних елементів: службовий механізм, ролі і відносини між ролями. [20]

Оцінка ефективності метаданих

Останній етап життєвого циклу метаданих спрямований на розгляд результатів всього процесу роботи з метаданими і якості самих метаданих. Інтегральна оцінка складається з оцінки якості метаданих, оцінки ефективності роботи схеми метаданих, оцінки результатів використання інструментів створення метаданих та оцінки застосування моделі життєвого циклу метаданих на кожному етапі. [20]

Для більш детального ознайомлення з метаданими і роботу з ними прочитайте книги [25] - [28] зі Списку інформаційних джерел для самостійної роботи, наведеного в кінці даного навчального посібника.

Питання для перевірки

1. Що таке метадані?
2. Які ГОСТи є для метаданих?
3. Що таке онтологія?
4. Що таке життєвий цикл метаданих?
5. Які етапи життєвого циклу метаданих ви знаєте?
6. Навіщо потрібен етап "Оцінка вимог і аналіз контенту" (один з етапів життєвого циклу метаданих)?
7. Навіщо потрібен етап "Специфікація системних вимог" (один з етапів життєвого циклу метаданих)?
8. Навіщо потрібен етап "Система метаданих" (один з етапів життєвого циклу метаданих)?
9. Навіщо потрібен етап "Сервіс і оцінка" (один з етапів життєвого циклу метаданих)?

3. Великі дані. Системи управління Великими даними

Якщо давати коротке визначення, то Великі дані - це дані, які не поміщаються в оперативну пам'ять комп'ютера.

По суті це визначення позначає те, що властивість "бути великим" є не самостійним властивістю даних, а залежить від характеристики системи, яка застосовується для їх обробки.

Наприклад, звичайній людині важко запам'ятати яка саме температура була в нашому місті кожен день за минулий місяць. Таким чином, три десятка значень цілком можуть бути прикладом Великих даних. Однак ось людина впевнено повідомляє "минулий місяць був холодним". Це повідомлення несе інформацію про оброблених даних: на думку співрозмовника, середня температура за минулий місяць була нижче, ніж зазвичай в цьому місяці за кілька десятків років.

Іншим прикладом можуть бути дані про об'єкти, які теоретично несуть важливу інформацію, але мають такий розмір, що ці дані практично неможливо не тільки обробити або зберегти, але навіть зібрати. Розглянемо приміром набір даних, що містить координати і швидкості молекул в повітряному стовпі над територією аеропорту. Є також метадані з описом в який момент проводилось вимірювання і що це за молекула. Такий набір даних несе інформацію про погодні умови над аеропортом, включаючи температуру, тиск, вологість, хмарність, особливі погодні умови
- проходить торнадо або падаючий град. З іншого боку, для коректної обробки дані для всіх молекул повинні бути досить повні і репрезентативні для статистичної обробки.

В результаті такого уявного експерименту ми розуміємо, що для ефективної роботи з великими даними потрібна модель даних, що дозволяє сформувати методи роботи з даними.

Дані можуть бути різних типів. Інформацію, отриману в результаті обліку або вимірювання будь-яких об'єктів або параметрів, називають майстер-даними (Master Data). Наприклад, облік кількості, заміри координат і швидкостей конкретних молекул - це майстер-дані.

Транзакційні дані (В англомовній літературі застосовуються терміни Transactional Data, Application Specific Data, Operational Data) - це дані, що відображають результат виконання будь-яких операцій. Наприклад, дані про взаємодію молекул між собою, а саме про перетин кордонів розглянутої області, про траєкторію конкретної молекули, про випаровуванні крапель дощу - це транзакційні дані. Транзакційні дані описують взаємодію об'єктів один з одним або з навколишнім світом, які можна отримати за допомогою обробки майстер-даних.

Ретроспективні дані(Historical data) - це дані, забезпечені мітками часу. Наприклад, з одного боку ми можемо зберігати дані про координаті і векторі швидкості кожної молекули, але якщо у нас є набір координат в залежності від часу, то швидкість молекули стає зайвою, вона обчислюється виходячи з моделі, описуваної ньютонівської механікою.

Довідкові дані(Довідники, НДІ, нормативно-довідкова інформація, Reference Data, Lookup Data, Dictionaries) - це базові незмінні дані, заздалегідь відомі з зовнішніх джерел, такі як нормативи, скорочення, акроніми, словники, стандарти. Наприклад, питомі ваги молекул, залежність температури замерзання і кипіння від тиску, залежність середньої швидкості молекул (швидкості звуку) від температури.

Формат даних. Структуровані дані мають заздалегідь визначений формат. *полуструктурірованние* або *слабоструктуровані дані* - це дані, найчастіше зібрані з різних джерел. Структура даних документована, але в залежності від джерела даних конкретний формат подання інформації може бути різним. Неструктуровані дані вимагають обов'язкової обробки і подальшої валідації перед використанням. Наприклад, дані про координати і швидкостях молекул, в яких деякі координати пропущені або деякі записи повторюються, є полуструктурированного. Нам потрібно зрозуміти, чому так сталося і перед використанням або виключити такі дані (що може привестидо систематичної помилку), або, виходячи з моделі даних, відновити пропущені значення.

Дані, в яких координати вимірюються в різних одиницях виміру, числа іноді записані словами, іноді латинськими цифрами, а іноді у вигляді відсканованого зображення почерку лаборанта, є*неструктурованими даними*.

Зазвичай Великі дані описуються за допомогою наступних характеристик. [3]

1. *Об `єм* (Volume) - кількість згенерованих і збережених даних. Розмір даних визначає значимість і потенціал даних, а також те, чи можуть вони бути розглянуті як Великі дані.
2. *Різноманітність* (Variety) - тип даних. Великі дані можуть складатися з тексту, зображень, аудіо, відео. Великі дані при зіставленні один з одним можуть доповнювати відсутні дані.
3. *Швидкість* (Velocity) - швидкість. Тут мається на увазі швидкість, з якою дані генеруються і обробляються. Дуже часто Великі дані використовуються в режимі реального часу.
4. *Мінливість*(Variability) - суперечливість наборів даних може перешкоджати їх обробці й керуванню ними.

5. *Достовірність* (Veracity) - якість даних безпосередньо впливає на точність проведення аналізу даних.

Великі дані можуть бути класифіковані відповідно до кількома головними компонентами. Інтелект-карта, представлена на рис. 3, була складена на основі [2].

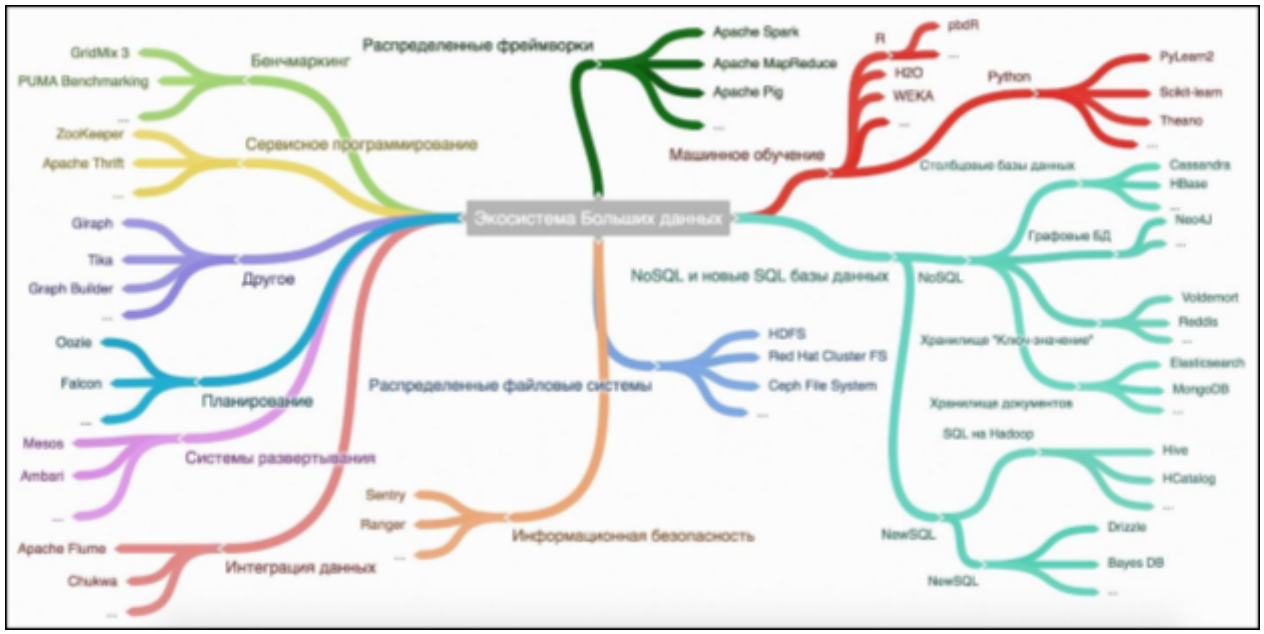

Рисунок 3 - Інтелект-карта екосистеми Великих даних

На рис. 3 за допомогою інтелект-карти показані компоненти екосистеми Великих даних. Розглянемо ці компоненти докладніше.

3.1. Розподілені файлові системи

Для зберігання і обробки Великих даних створені розподілені системи зберігання даних, в тому числі розподілені файлові системи, що дозволяють використовувати зовнішнє файловий простір системи зберігання для обробки даних на нодах, що входять в обчислювальних кластер.

Найчастіше зручно використовувати розподілені файлові системи, орендовані як окремий хмарний сервіс, наприклад, Google Colossus[5], Amazon S36.

3.2. Розподілені фреймворки

Обробка знаходяться на розподілених системах зберігання даних ведеться паралельно на комп'ютерах, що становлять вузли (nodes)

[5] Google Colossus. https://cloud.google.com/bigtable/docs/overview

[6] Amazon S3. https://aws.amazon.com/ru/s3/

обчислювального кластера. Для організації обчислень розробники систем обробки використовують розподілені фреймворки. Більшість фреймворків доступні за ліцензією Apache і орієнтовані на роботу в кластерах на базі Linux. Існують також хмарні фреймворки, орендовані як окремий хмарний сервіс. Деякі з них описані в розділі 4 "Архітектура систем обробки Великих даних" і в розділі 6 "Програмні платформи і системи для Великих даних".

3.3. Бенчмаркінг

Цей клас інструментів був розроблений для оптимізації інсталяції Великих даних за допомогою використання стандартизованих профілів (Profiling suites). Бенчмаркінг і оптимізація інфраструктури Великих даних часто не є сферою відповідальності дата-вчених, це сфера відповідальності для окремих професіоналів, що спеціалізуються на IT- інфраструктури. Використання оптимізованої інфраструктури може істотно знизити вартість використовуваного обладнання. [2]

3.4. Серверне програмування

Припустимо, що ви зробили додаток для прогнозування результатів футбольних матчів світового класу на платформі Hadoop, і ви хочете дозволити іншим використовувати прогнози, зроблені вашим додатком. Тим не менш, ви не маєте уявлення про архітектуру або технології всіх, хто прагне використовувати ваші прогнози. Сервісні інструменти дозволяють надавати додатки на Великих даних іншим програмам в якості служби. Дата- вченим іноді доводиться надавати свої моделі через служби. Найбільш відомим прикладом тут є REST-сервіс; REST означає репрезентативну передачу стану (REpresentational State Transfer, REST). Вона часто використовується в якості обміну даними з веб-сайтами. [2]

3.5. Планування

Інструменти планування дозволяють автоматизувати повторювані задачі і запускати завдання на основі таких подій, як додавання нового файлу в папку. Вони схожі на такі інструменти, як CRON в Linux, але спеціально розроблені для роботи в відмов кластеру. Ви можете використовувати їх, наприклад, для запуску завдання MapReduce щоразу, коли в каталозі є новий набір даних. [2]

3.6. Системи розгортання

Налаштування інфраструктури Великих даних - непросте завдання, і розгортання нових додатків в кластері Великих даних - це зона відповідальності інженерів по Великим даними. Вони в значній мірі автоматизують установку і настройку компонентів Великих даних. [2]

3.7. Інтеграція даних

Припустимо, що вже є розподілена файлова система, і тепер необхідно перенести дані з одного джерела в інший. У таких випадках використовують фреймворки для інтеграції даних, такі як Apache Sqoop і Apache Flume. Цей процес схожий на процес вилучення, перетворення і завантаження (Extract, Transform and Load, ETL) в традиційному сховище даних. [2]

3.8. Інформаційна безпека

Засоби забезпечення безпеки Великих даних дозволяють здійснювати централізований контроль доступу до даних. Безпека Великих даних стала самостійною дисципліною, і дата-вчені зазвичай стикаються з нею тільки як споживачі даних. Безпекою Великих даних займаються експерти з інформаційної безпеки. [2]

3.9. Машинне навчання

Якщо у вас є Великі дані, то було б непогано отримати з них корисний контент. Це можна зробити за допомогою використання методів машинного навчання, статистики та прикладної математики. [2]

Ще перед Другою світовою війною багато трудомісткі обчислення проводилися вручну, що природним чином обмежувало можливості аналізу даних. Після Другої світової війни стала активно розвиватися обчислювальна техніка і наукові обчислення. З'явилася можливість писати програми з формулами і алгоритмами, а потім завантажувати в програми різні дані.

На сьогоднішній день, коли з'явилося величезна кількість даних, один комп'ютер вже не в змозі впоратися із завданням їх обробки. Деякі алгоритми, розроблені в минулому столітті, на жаль, не зможуть впоратися з цим завданням, навіть якщо теоретично можна було б підключити до вирішення завдання все комп'ютери Землі. Це пов'язано з тимчасовою складністю алгорітма[8].

[8] Тимчасова складність алгоритму.
https://ru.wikipedia.org/wiki/Временная_сложность_алгоритма

З прикладами тимчасової складності можна ознайомитися на Stack Overflow9.

Одна з найбільших проблем зі старими алгоритмами полягає в тому, що вони недостатньо масштабируются. З огляду на обсяг даних, які необхідно аналізувати сьогодні, це стає проблематичним. Для обробки цього обсягу даних потрібні спеціалізовані структури та бібліотеки. Наприклад, в мові Python є такі бібліотеки: Scikit-learn (бібліотека машинного навчання), PyBrain (для роботи з нейронними мережами), NLTK (для обробки природної мови), Pylearn2 (ще одна бібліотека машинного навчання), TensorFlow (бібліотека глибокого навчання, є програмний інтерфейс API для мови Python), Keras (бібліотека для роботи з нейронними мережами) та інші. [2]

Існує також Apache Spark - програмний каркас з відкритим вихідним кодом для реалізації розподіленої обробки неструктурованих і слабоструктурованих данних10.

3.10. Бази даних NoSQL і нові SQL бази даних

Використання реляційних баз даних для обробки Великих даних вкрай неефективно через високі накладних витрат. Традиційно для обробки Великих даних використовуються бази даних типу "ключ - значення" (Key value database або HashDB). Одна з перших баз цього типу DBM була реалізована Ken Thompson для AT & T Unix 7 в 1979 році.

База даних виду "ключ - значення", по суті, являє собою асоціативний масив (Hash, Dict), тобто множина, що складається з пар (Key, Value). У деяких реалізаціях на безлічі ключів вводиться відношення порядку, і ми можемо отримати значення послідовно в міру зростання ключа. В інших випадках сортування по ключу нестійка при однакових ключах і при неодноразових вибірках можна отримати різну послідовність пар.

Велика кількість баз даних можна розділити на наступні типи:

- *Столбцовую бази даних*(Column databases). Дані зберігаються в стовпцях, що дозволяє алгоритмам виконувати набагато швидші запити.
- *сховища документів*(Document stores) Сховища документів більше не використовують таблиці, але зберігають кожне спостереження в документі. Це дозволяє використовувати набагато більш гнучку схему даних.

---

[9] Real-world example of exponential time complexity.http://stackoverflow.com/questions/7055652/real-world-example-of-exponential-time- complexity

[10] Apache Spark.http://spark.apache.org/

- *Потокові дані*(Streaming data). Дані збираються, перетворюються і агрегуються не в партіях, а в реальному часі.
- *Сховища для ключів*(Key-value stores). Дані не зберігаються в таблиці; для кожного значення призначається ключ (як розказано про це вище).
- *SQL на Hadoop*- пакетні запити на Hadoop, що використовують фреймворк MapReduce в фоновому режимі.
- *новий SQL* (New SQL). Цей тип поєднує масштабованість баз даних NoSQL з перевагами реляційних баз даних. Тут використовується інтерфейс SQL і реляційна модель даних.
- *Графові бази даних*(Graph databases). Це тип баз даних, що використовують графові структури для семантичних запитів з вузлами і ребрами і властивостями для представлення і збереження даних. Класичним прикладом цього типу є соціальна мережа.

Доступ до баз даних можна орендувати у вигляді сервісу над розподіленимисистемами зберігання, наприклад, у Google це PostgreSQL11 і NoSQL Datastore12.

[11] Google PostgreSQL. https://cloud.google.com/sql/

[12] NoSQL Datastore. https://cloud.google.com/datastore/

Питання для перевірки

1. Що таке Великі дані?
2. Які п'ять характеристик притаманні Великим даними?
3. Які існують базові принципи обробки Великих даних?
4. Оцініть скільки 10 Тб HDD необхідно для зберігання набору даних, що містить координати, швидкості і метаінформацію (тип молекули і час вимірювання по конкретній молекулі) для всіх молекул на території аеропорту.
5. Що таке столбцовую бази даних?
6. Що таке сховища документів?
7. Що таке потокові дані?
8. Що таке сховища для ключів?
9. Що таке SQL на Hadoop? 10.Що таке новий SQL?
11. Что таке графові бази даних?

4. Архітектура системи обробки Великих даних

Для роботи з Великими даними використовуються складні системи, в яких можна виділити кілька компонентів або шарів (Layers). Зазвичай виділяють чотири рівні компонентів таких систем: прийом, збір, аналіз даних і представлення результатів (рис. 4). Цей поділ є значною мірою умовною так як, з одного боку, кожен компонент в свою чергу може бути розділений на подкомпоненти, а з іншого деякі функції компонентів можуть перерозподілятися в залежності від розв'язуваної задачі і використовуваного програмного забезпечення, наприклад, виділяють зберігання даних в окремий шар.

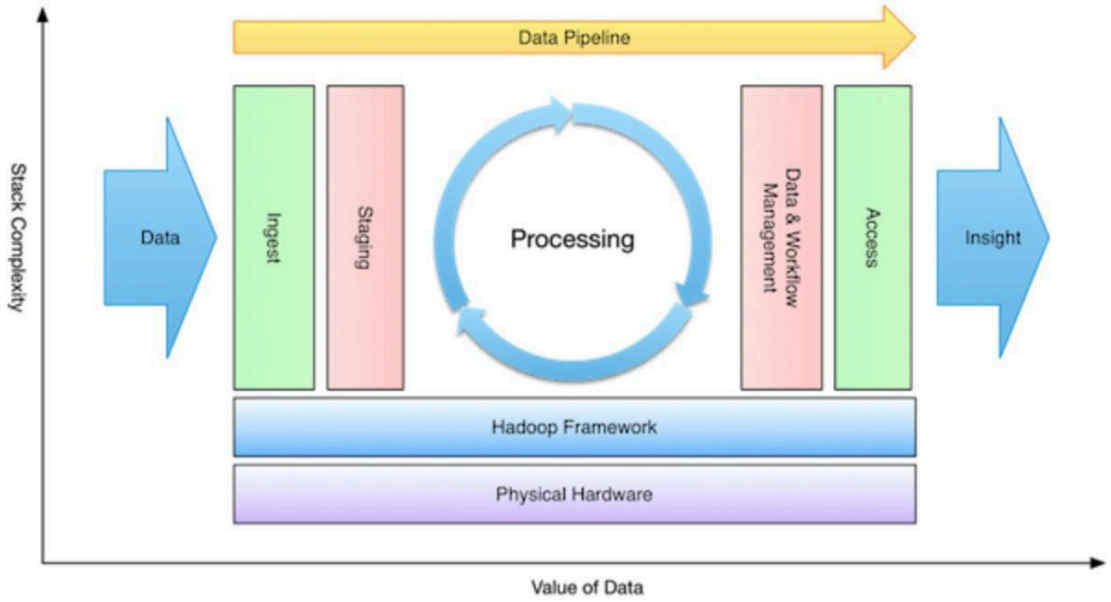

Рисунок 4 - Стек роботи з Великими даними. [15]

Для роботи з Великими даними розробниками систем створюються моделі даних, змістовно пов'язані з реальним світом. Розробка адекватних моделей даних є складною аналітичну задачу, виконувану системними архітекторами і аналітиками. Модель даних дозволяє створити математичну модель взаємодій об'єктів реального світу і включає в себе опис структури даних, методи маніпуляції даними і аспекти збереження цілісності даних. Опис розробки моделей даних не є завданням даного посібника.

Для зберігання даних використовуються розподілені системи різних типів. Це можуть бути файлові системи, бази даних, журнали, механізми доступу до загальної віртуальної пам'яті. Більшість систем зберігання орієнтовані виключно на роботу з Великими даними, вони мають вкрай обмежене число функцій (наприклад, може бути відсутнім можливість

не тільки модифікації, але і видалення даних, що надійшли) що пояснюється внутрішньої складністю створення високоефективних розподілених систем. В кінці тексту наведені посилання на кілька використовуваних зараз систем.

Для того, щоб робота з даними відбувалася швидше системи зберігання і обробки даних распараллелівать в кластері (cluster, група комп'ютерів, об'єднаних мережею для виконання єдиного завдання). Однак, відповідно до гіпотези Брюєра неможливо забезпечити одночасну узгодженість (несуперечність) даних, доступність даних і стійкість системи до відокремлення окремих вузлів. Гіпотеза доведена для транзакцій типу ACID (Atomic, Consistent. Isolated, Durable) і відома під назвою CAP теореми (Consistency, Availability, Partition tolerance).

4.1. Прийом даних (Data Ingestion)

Джерела даних мають різні параметри, такі як частоту надходження даних з джерела, обсяг порції даних, швидкість передачі даних, тип даних, що надходять і їх достовірність.

Для ефективного збору даних необхідно встановити джерела даних. Це можуть бути сховища даних, постачальники агрегованих даних, API будь-яких датчиків, системні журнали, згенерований людиною контент в соціальних мережах, в корпоративних інформаційних системах, геофізична інформація, наукова інформація, успадковані дані з інших систем. Джерела даних визначають вихідний формат даних.

Наприклад, ми можемо самостійно проводити погодні на території аеропорту, використовувати дані, що надходять з злітають і сідають літаків, закупити дані з супутників, що пролітають над аеропортом і у місцевої метеослужби, а також знайти їх десь в мережі в іншому місці. У загальному випадку для кожного джерела необхідно створювати власний складальник (Data Crawler для збору інформації в мережі і Data Acquisition для проведення вимірювань).

Прийом даних полягає в початковій підготовці даних від джерел з метою приведення даних до загального формату представлення даних. Цей єдиний формат вибирається відповідно до прийнятої моделі даних. Виконуються перетворення систем вимірювання, типів (типізація), верифікація. Обробка даних змістовно не впливає на наявну в даних інформацію, але може змінювати її уявлення (наприклад, приводити координати до єдиної системи координат, а значення до єдиної розмірності).

4.2. Збір даних (Data Staging)

Етап збору даних характеризується безпосереднім взаємодією з системами зберігання даних. Встановлюється точка збору, в якій зібрані дані забезпечуються локальними метаданими і поміщаються в сховище або передаються для подальшої обробки. Дані, з яких-небудь причин не пройшли точку збору, ігноруються.

Для структурованих даних проводиться перетворення з вихідного формату по заздалегідь заданими алгоритмами. Це найбільш ефективна процедура в разі, якщо структура даних відома. Однак якщо дані представлені в двійковому вигляді, структура і зв'язку між даними загублені, то розробка алгоритмів і заснованого на них програмного забезпечення для обробки даних може виявитися вкрай скрутній.

Для напівструктурованих даних потрібно інтерпретація даних, що надходять і використання програмного забезпечення, що вміє працювати до мови опису даних. Істотним плюсом напівструктурованих даних є те, що в них часто містяться не тільки самі дані, але метадані у вигляді інформації про зв'язки між даними і способах їх отримання.

Розробка програмного забезпечення для обробки напівструктурованих даних являє собою досить складну задачу. Однак є значна кількість готових конвертерів, які можуть, наприклад, витягти дані з формату XML в сформоване табличне представлення.

Найбільшого обсягу робіт вимагає обробка неструктурованих даних. Для їх переведення до заданого формату може знадобитися створення спеціального ПО, складна ручна обробка, розпізнавання і вибірковий ручний контроль.

На етапі збору проводиться контроль типів даних і може виконуватися базовий контроль достовірності даних. Наприклад, координати молекул газу, що містяться в будь-якій області, не можуть лежати за межами цієї області, а швидкості - істотно перевищувати швидкість звуку. Для того, щоб уникнути помилок типізації, необхідно перевіряти правильність налаштувань одиниці виміру. Наприклад, в одному наборі даних висота може вимірюватися в кілометрах, а в іншому - в футах. У цьому випадку необхідно зробити перетворення висоти в ті одиниці виміру, які прийняті в використовуваної моделі.

При зборі дані систематизуються і забезпечуються метаданими, збереженими в пов'язаних метаданих. При наявності великої кількості джерел даних може знадобитися управління збором даних для того, щоб збалансувати обсяги інформації, що надходять з різних джерел.

Зібрані дані або зберігаються в системах зберігання, або (особливо, для потокових даних) передаються для аналізу в реальному часі.

4.3. Аналіз даних (Analysis Layer)

Аналіз даних, на відміну від збору даних, використовує інформацію, що міститься в самих даних. Аналіз може проводитися як в реальному часі, так і в пакетному режимі. Аналіз даних становить основну по трудомісткості завдання при роботі з Великими даними.

Існує безліч методик обробки даних: предіктивне аналіз, запити та звітність, реконструкція з математичної моделі, трансляція, аналітична обробка та інші. Методики використовують специфічні алгоритми в залежності від поставлених цілей. Наприклад, аналітична обробка може бути аналізом зображень, соціальних мереж, географічного розташування, розпізнавання за ознаками, текстовим аналізом, статистичною обробкою, аналізом голосу, транскрибування.

Алгоритми аналізу даних також, як і алгоритми обробки даних, спираються на модель даних. При цьому під час аналізу може бути використано кілька моделей, які задають загальний формат даних, але по-різному моделюють змістовні процеси, дані про яких ми обробляємо. При використанні при аналізі методів штучного інтелекту, зокрема нейронних мереж, проводиться динамічне навчання моделей на різних наборах даних.

При аналізі даних проводиться ідентифікація сутностей, описуваних даними на підставі наявної в даних інформації і використовуваних моделей. Сутністю аналізу є аналітичних механізм, який використовує аналітичні алгоритми, управління моделями та ідентифікацію сутностей для отримання нової змістовної інформації, що є результатом аналізу.

Для аналізу даних також використовуються методи штучного інтелекту на нейронних мережах, не аналізовані в даному навчальному посібнику.

4.4. Представлення результатів (Consumption Layer)

Результати аналізу даних надаються на рівні споживання. Є кілька механізмів, що дозволяють використовувати результати аналізу великих даних.

- Моніторинг метаінформації.

Підсистема відображення в реальному часі істотних параметрів роботи системи, завантаженості обчислювачів, розподіл завдань в кластері, розподіл інформації в сховищах, наявність вільного місцяв сховищах, надходження даних від джерел, активності користувачів, відмов обладнання і тд.

- Моніторинг даних.

Підсистема відображення в реальному часі процесів прийому, збору і аналізу даних, навігація за даними.

- Генерація звітів, запити до даних, подання даних у вигляді візуалізації на дешбордах (Dashboard), в форматі PDF, інфографіку, зведених таблицях і коротких довідках
- Перетворення даних і експорт в інші системи, інтерфейс з BI-

системами.

Питання для перевірки

1. Які рівні можна виділити в системах обробки Великих даних?
2. Навіщо комп'ютери об'єднують в кластер?
3. У чому полягає прийом даних від джерел?
4. У чому може бути складність обробки структурованих даних?
5. Наведіть приклад напівструктурованих даних.
6. Які завдання може вирішувати аналіз Великих даних?
7. З якою метою використовується шар / рівень компонентів системи "Прийом даних"?
8. З якою метою використовується шар / рівень компонентів системи "Збір даних"?
9. З якою метою використовується шар / рівень компонентів системи "Аналіз даних"?
10. В яких цілях використовується шар / рівень компонентів системи "Подання результату"?

5. Паралельні алгоритми для роботи з даними

5.1. Оператори Map і Reduce

Для паралельної обробки даних в інтерфейсі MPI (Message Passing Interface), що є поширеним стандартом для обміну даними при організації паралельних обчислень, була запропонована нині широко використовувана в безлічі реалізацій парадигма паралельної обробки наборів даних за допомогою використання операторів Map і Reduce.

5.2. Оператор Reduce (згортка)

Згортка в найпростішому випадку отримує на вході функцію від двох параметрів з набору даних, що складається з елементів одного типу. На виході вона повертає один елемент такого ж типу. У загальному випадку від функції потрібно коммутативность і асоціативність на безлічі визначення. У цьому випадку порядок обчислень є несуттєвим, алгоритм дозволяє ефективно распараллелівать, так як є можливість вибрати довільні пари елементів набору даних, від обраних пар за допомогою переданої в Reduce функції паралельно обчислити значення і вихідні пари елементів замінити на обчислені значення. Алгоритми, в яких необхідна нам функція згортки коммутативна і асоціативна, високоефективна при обробці Великого невпорядкованого (або упорядкованого, для обчислення це несуттєво) набору даних.

Для асоціативної, але некомутативної функції згортки визначена для набору даних, елементи якого впорядковані. У цьому випадку ми можемо розбити вихідний набір на кілька впорядкованих між собою послідовних піднаборів, наприклад, за кількістю наявних обчислювачів в кластері. З математичної точки зору ця процедура відповідає розстановці дужок. Потім, отримавши по одному елементу даних з кожного поднабора, ми маємо проміжний упорядкований набір даних, і виконуємо згортку над ним і далі рекурсивно до отримання підсумкового значення.

У більш складному випадку ми хочемо при роботі Reduce отримати дані іншого типу, наприклад, на вході ми маємо набір з масою і швидкостями молекул, а на виході нам потрібно отримати сумарну масу і імпульс (щоб дізнатися середню швидкість вітру розділивши імпульс на масу). Для вирішення такого завдання функція, передана в згортку, істотно видозмінюється наступним чином: вона отримує один параметр результуючого типу (званий акумулятором) і один параметр вихідного типу (итератор). На самому початку акумулятор має деяке початкове значення. Для нашого прикладу з молекулами передане в згортку початкове значення - це структура з чотирьох елементів: маса дорівнює нулю і три компонента (за координатами x, y і z) імпульсу теж дорівнюють нулю.

Потім елементи даних перебираються і виробляються обчислення, певні функцією. У нашому прикладі маса молекули, переданої в Ітератор, додається до маси в акумуляторі, далі обчислюється імпульс молекули (вектор з творів маси на компоненти швидкості) і додається до компонентів імпульсу акумулятора.

Розбивши вихідний набір даних на піднабори ми для кожного обчислимо масу і імпульс. Однак далі у нас з'являється проблема: наявна функція вміє додавати до даних мають типу акумулятора (маса, імпульс) дані мають тип ітератора (маса, швидкість). Щоб зробити підсумкові обчислення, нам необхідно провести перевизначення функції: в тому випадку, якщо передається два параметри, що мають тип акумулятора, то функція починає вести себе зовсім по-іншому. А саме складає маси і імпульси. Інакше кажучи, по суті у нас з'являються дві функції: одна працює з майстер-даними, а інша з проміжними даними. Таким чином, ми можемо провести ефективне розпаралелювання. Звернемо увагу що цей метод також допускає роботу з некоммутірующімі, але впорядкованими даними в разі асоціативних функцій.

Іноді для зручності обчислень, крім описаної вище прямої (лівої) згортки, при якій функція обчислюється від перших двох елементів (або від початкового значення і першого елемента даних), потім від отриманого акумулятора і наступного елемента даних (ітератора) і так далі до завершення обчислення , вводять також праву (зворотний) згортку. При зворотному згортку функція обчислюється спочатку від початкового значення (якщо воно задане) і останнього елемента, потім від акумулятора і передостаннього елемента, і так далі, поки не дійдемо до обчислення від акумулятора першого елемента.

Ефективне розпаралелювання обчислення згортки від некомутативних неассоціативное функцій в загальному випадку неможливо. Зауважимо, що Гвідо ван Россум (BDFL, Benevolent Dictator For Life, великодушний довічний диктатор мови Python) свідомо перевів згортку із загального простору імен в модуль functools оскільки в більшості додатків її використовували або для таких тривіальних речей, як підсумовування масивів, або для отримання абсолютно незрозумілого нечитабельною коду.

5.3. Оператор Map

Цей оператор відомий у багатьох мовах програмування під різними іменами, крім Map використовується також назви Select і Transform. У найпростішому випадку Map створює з переданої йому функції і упорядкованого набору даних інший упорядкований набір даних, в якому кожен елемент є значенням функції від відповідного елемента вихідного набору даних.

Існують розширення оператора Map на кілька наборів даних, при цьому функція повинна бути визначена від відповідної кількості елементів. Повертається набір даних, елементи якого є значеннями функції від відповідних наборів. При цьому є два способи визначити поведінку розширеного оператора Map в разі

відмінності в обсягах переданих наборів даних - або результуючий набір має розмір рівний розміру найменшого набору з переданих, або розміром максимального, при цьому функції передаються невизначені значення.

5.4. Лямбда-архітектура

Архітектура, що дозволяє проводити роботу з даними за допомогою еквівалентних алгоритмів одночасно в пакетному режимі і в режимі реального часу. Аналітика в реальному часі може бути неточною, виконується в оперативній пам'яті, але надаються швидко. Розрахунки в пакетному режимі забезпечують зберігання отриманих результатів і видає достовірні дані, але виконується довго.

Питання для перевірки

1. Наведіть приклад асоціативної, але некомутативної функції,визначеної на множині цілих чисел
2. Наведіть приклад комутативність, але неассоціативное функції,визначеної на множині цілих чисел.
3. Який тип даних може повертати оператор згортки, застосований до початкового значення, функції та набору даних?
4. В якому випадку згортка даних неефективна?
5. Який тип даних повертає оператор Map, застосований до функції і набору даних?
6. Яка обробка даних забезпечує більше високу достовірність результатів - в реальному часі або в пакетному режимі?

6. Програмні платформи і системи для Великих даних

В даний час використовується значна кількість платформ і систем Великих даних. Системи обробки великих даних є фреймворками, тобто каркасами, для використання яких необхідно зістикувати їх з іншими фреймворками, прикладним програмним забезпеченням користувача і системою зберігання даних.

В аналітичному звіті Big Data Analytics Market Study 2017 Edition [18] наводиться така діаграма інфраструктур Великих даних, впроваджених на підприємствах, представленав розрізі розмірів підприємств (рис. 5).

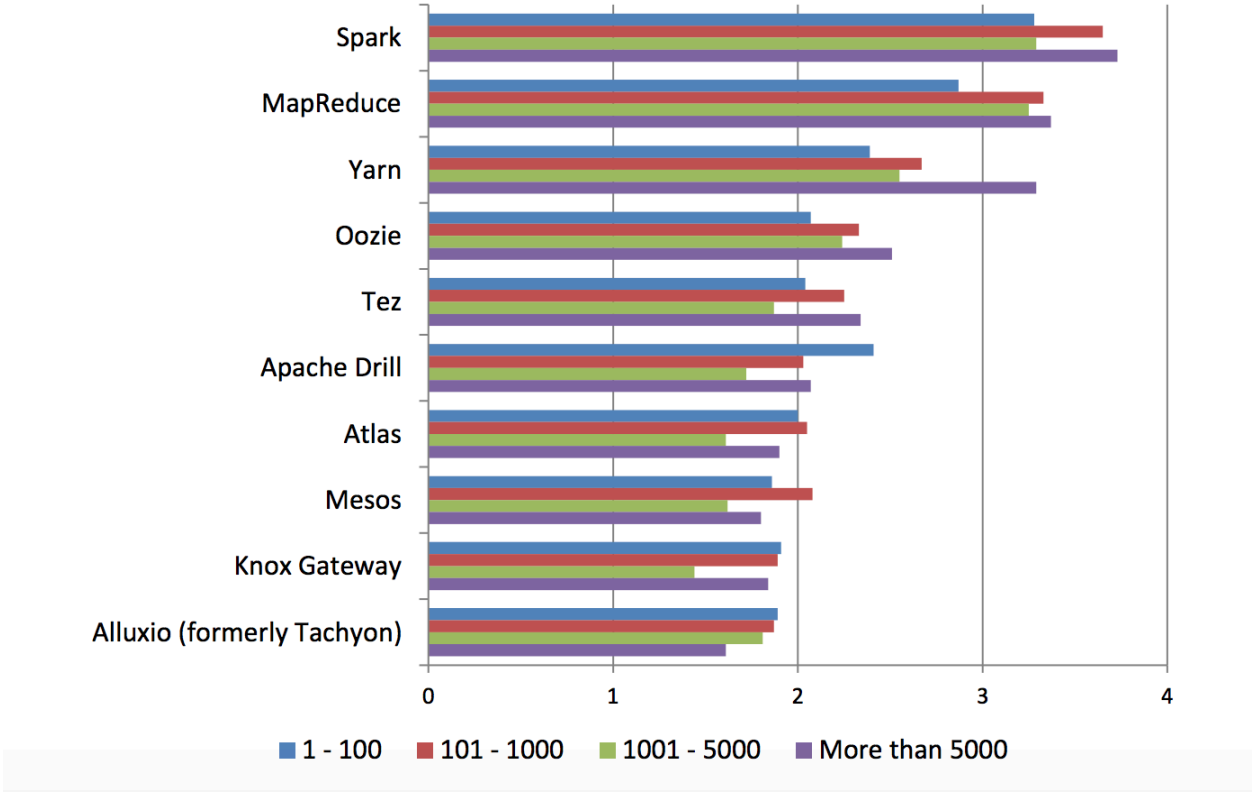

Рисунок. 5 - Інфраструктури Великих даних в розрізі розміру підприємств [18]

Більшість використовуваних платформ є за ліцензією Apache 2.0 і розташовані на сайті фонду програмного забезпечення Apache.

6.1. Системи управління потоками даних

Flume

https://flume.apache.org/
Розроблено в 2017 році.
Система управління потоками даних.

Apache Kafka

https://kafka.apache.org/
Розроблено в LinkedIn в 2011 році.
Масштабований відмовостійкий журнал коммітов.

Niagara Files (NiFi)

https://nifi.apache.org/
Розроблено в NSA в 2014.
Система управління потоками даних.

6.2. Системи зберігання Великих даних

HDFS (Hadoop Distributed File System)

https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
Файлова система, що входить в проект Hadoop.

OpenStack Swift

https://docs.openstack.org/swift/latest/
Платформа зберігання, що входить в проект OpenStack.

Cassandra

http://cassandra.apache.org/
Таблична СУБД, написана на мові Java.

HBase

https://hbase.apache.org/
Таблична СУБД, написана на мові Java.

Apache Drill

https://drill.apache.org/

SQL-інтерфейс до NoSQL баз даних (HBase, MongoDB, MapR-DB, HDFS, MapR-FS, Amazon S3, Azure Blob Storage, Google Cloud Storage, Swift, NAS and local files).

6.3. Платформи Великих даних

Hadoop

http://hadoop.apache.org/

Надає інтерфейс до Java (а через коннектори і до інших мов), вільно розповсюджується під ліцензіями Apache License 2.0 і GNU GPL пакет програмного забезпечення, що складається з керуючого модуля Hadoop Common, розподіленої файлової системи HDFS, планувальника завдань YARN і обчислювальної платформи Hadoop MapReduce. Розвивається з 2005 року.

Spark https://spark.apache.org/

Надає інтерфейси до Scala, Java, Python і R, поширюється під ліцензією Apache License 2.0. обчислювальна платформа, що розвивається з 2014 року.

Elasticsearch

https: //www.elastic.co/products/elasticsearch

Спільно з системою збору Logstash і платформою аналітики Kibana складають інтегровану систему збору, зберігання, пошуку та аналітики даних.

Solr http://lucene.apache.org/solr/

Ще одна система пошуку і аналізу в базах Великих даних.

Hortonworks Data Platform (HDP)https://hortonworks.com/products/data-platforms/hdp/

Платформа керування даними, що включає HDFS, Hadoop, HBase, HCatalog, Pig, Hive, Oozie, Zookeper, Ambari, WebHDFS, TalentOS, Sqoop, Flume, і Mahout.

Windows Azure HDInsight

https://azure.microsoft.com/en-ca/services/hdinsight/

Система від Microsoft для розгортання hadoop на Azure.

6.4. Обробка даних в реальному часі

Apache Storm

https://storm.apache.org/

За попередньо створеної топології обчислень керуючий вузол створює в кластері спаути (spout), згенеровано кортежі (tuple) ключ-значення, і болти (bolt), що виконують обробку. Будь-які мови (JVM, Ruby, Python, Perl).

Apache Spark

https://spark.apache.org/streaming/

Потоки розбиваються на пакети DStream (Дискретизований потік) складаються з декількох RRD (resilient distributed dataset, відмовостійкий розподілений набір даних). Обробка за допомогою технології sliding window (що ковзають вікон). Підтримує Scala, Java і Python.

Apache Samza

http://samza.apache.org/

Розподіл повідомлень на розділи, кожен з яких незалежно обробляється в міру їх отримання. Підтримує JVM: Scala і Java.

6.5. Системи управління Великими даними

Ambari

https://ambari.apache.org/

webUI провізіонінга, управління і моніторингу кластера Hadoop.

Atlas

https://atlas.apache.org/

Система обміну метаданими для Hadoop-стека.

Cloudbreak

https://hortonworks.com/open-source/cloudbreak/

Система розгортання платформи Hortonworks в комерційному хмарі.

Knox

https://knox.apache.org/

Система аутентифікації і авторизації для Hadoop-кластера.

Ranger

https://ranger.apache.org/

Система забезпечення безпеки даних на платформі Hadoop.

ZooKeeper

https://zookeeper.apache.org/

Централізований сервіс для управління конфігураційною інформацією, ім'ям, розподіленої синхронізацією і груповими сервісами.

6.6. Аналітичні платформи

RapidMiner

https://rapidminer.com/

Система прогнозної аналітики, підтримує глибинний аналіз, перевірку, оптимізацію і візуалізацію, має графічний інтерфейс програмування.

IBM SPSS Modeler

https: //www.ibm.com/products/spss-modeler

Комерційна аналітична система автоматизованого моделювання, геопросторової аналітики, аналізу текстової інформації. Погано підходить для великих обсягів інформації.

KNIME

https: //www.knime.com/

Безкоштовна система аналізу даних, що має глибинний аналіз, веб-аналіз,обробку зображень, аналіз соціальних мереж, обробку текстів.

Qlik Analytics Platform

https: //www.qlik.com/us/products/qlik-analytics-platform

система візуальної аналітики, надає доступ до асоціативноїмашині індексації даних QIX Engine.

STATISTICA Data Miner

http://statsoft.ru/products/STATISTICA_Data_Miner/data-mining-tools.php

Система від російського виробника

IBM Watson Analytics https:

//www.ibm.com/watson-analytics

Потужна хмарна система від IBM (застосовується в тому числі Twitter'ом)

Dell EMC Analytic Insights Module https://www.dellemc.com/en-us/big-data/index.htm

Багатокомпонентна система від Dell EMC

SAP Predictive Analytics https://www.sap.com/products/predictive-analytics.html
Система від світового лідера ERP, інтегрується з SAP HANA

Oracle Big Data Preparation
https://cloud.oracle.com/bigdatapreparation Хмарне рішення від світового лідера баз даних

Питання для перевірки

1. Які мови програмування використовуються для роботи з фреймворками даних?
2. Чи дозволяє ліцензія Apache 2.0, під якою випущені деякі фреймворки, вносити власні виправлення в код програмного забезпечення?
3. Перерахуйте кілька фреймворків, які забезпечують обробку даних в реальному часі.
4. Перерахуйте кілька фреймворків, які забезпечують аналітичну обробку даних.
5. Перерахуйте кілька фреймворків, які забезпечують зберігання даних.
6. Перерахуйте кілька фреймворків, які забезпечують управління потоками даних.

7. Устаткування для обробки Великих даних

Комплект обладнання для обробки Великих даних монтується в ЦОД. Основними компонентами системи є система управління, обчислювальні ресурси, система зберігання даних, локальна мережа. Електроживлення, моніторинг, доступ до інтернету і інші зовнішні ресурси надають центри обробки даних (ЦОД).

Система управління призначена для загального управління системою, зовнішнього доступу, забезпечення аутентифікації, авторизації та аккаунтинга і представлення результатів користувачам. Виконується на базі звичайної серверної платформи, специфічних вимог не має.

На рис. 6 представлений кластер Hadoop від Yahoo 10 річної давності. [16]

Рисунок 6 - Кластер Hadoop від Yahoo 10 річної давності

Обчислювальні ресурси кластера складаються з вузлів (nodes), основними параметрами яких є обсяг оперативної пам'яті з контролем парності (ECC) і максимальну кількість ядер CPU. Розрахунковим параметрами є розмір оперативної пам'яті на ядро і швидкість роботи процесора. Висока відмовостійкість вузлів бажана, але в цілому не потрібно, деяка кількість відмов є нормальним і компенсується за допомогою програмного забезпечення - відмовив вузол автоматично виводиться з експлуатації, і його робота перерозподіляється на інші вузли.

У деяких випадках для обробки Великих даних, особливо при використанні нейронних мереж, використовують обчислювачі на базі GPU, аналогічні використовуваним для HPC.

Розподілена система зберігання даних складається з дискових полиць, що забезпечують максимально швидкий доступ до даних. Резервування виконується за допомогою програмного забезпечення систем зберігання і в загальному

випадку не потрібно. Найчастіше також використовуються комп'ютери, в яких підключено велику кількість локальних дисків.

Рисунок 7 - Центр зберігання даних виробництва Titan Power [21]

Мережева інфраструктура пов'язує обчислювальні ресурси, систему зберігання і систему управління в єдине ціле. Основна вимога до локальної мережі - низькі затримки. При великій кількості вузлів використовуються мережеві комутатори, початківці передачу пакета даних відразу після обробки заголовка пакета даних. При малій кількості вузлів поширене пряме з'єднання комп'ютерів за схемою гіперкуб, де розмірність куба відповідають числу портів на інтерфейсних платах. Раніше масово використовувалися мережі на базі різних варіантів інтерфейсу Infiniband, зараз, в основному, використовується 100 Gigabit Ethernet. Порівняно недавно з'явився 200 і 400 Gigabit Ethernet13.

[13]ConnectX®-6 EN Dual-Port Adapter Supporting 200Gb / s Ethernet.http://www.mellanox.com/page/products_dyn?product_family=266&mtag=connectx_6_ en_card

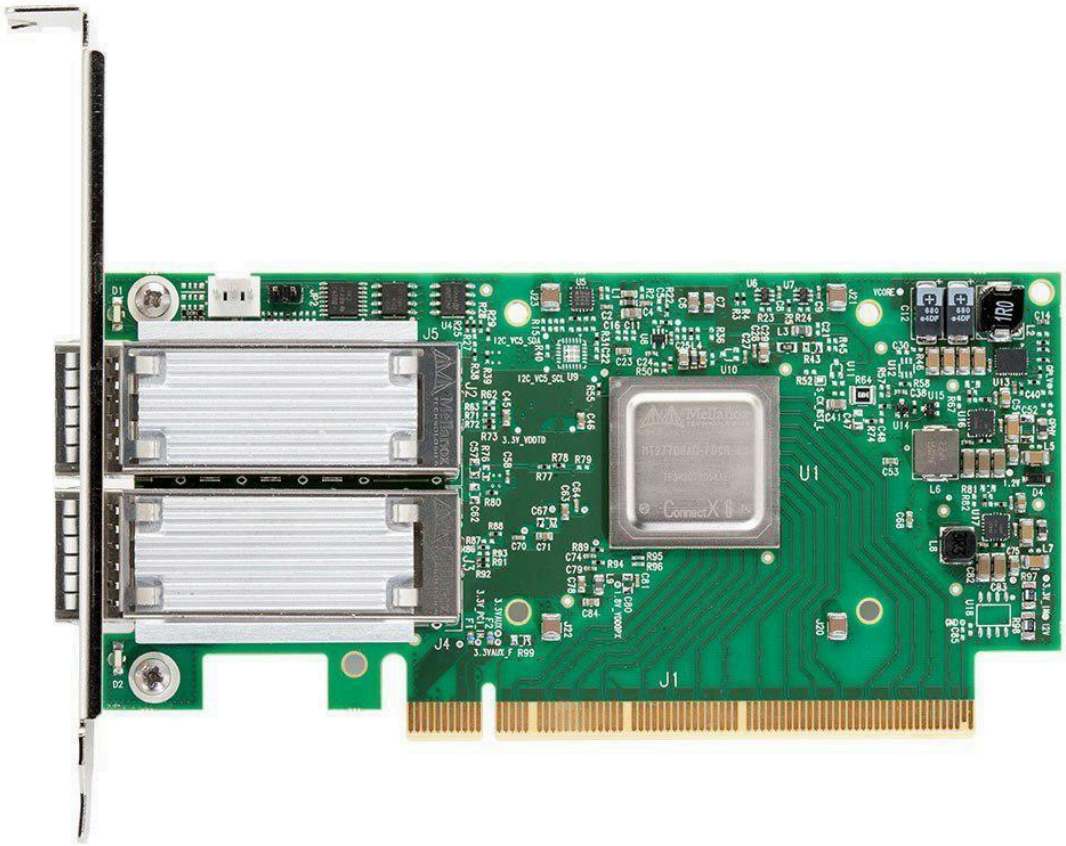

Рисунок 8 - ConnectX®-6 EN двухпортовой мережевий адаптер 200Gb / s Ethernet [22]

Питання для перевірки

1. Де монтується обладнання для обробки Великих даних?
2. З яких компонентів складається обладнання для обробки Великих даних?
3. Які параметри істотні для обчислювальних вузлів кластера?
4. Які параметри істотні для системи зберігання даних?
5. Які параметри істотні для мережевої інфраструктури?
6. Скільки максимально вузлів може мати кластер, який використовує архітектуру підключення вузлів гіперкуб, при наявності чотирьох швидкодіючих мережевих портів в кожному вузлі?

8. Центри обробки Великих даних

Великі дані обробляють в центрах обробки даних (ЦОД). У 2005 році асоціація телекомунікаційної індустрії (TIA) і інженерний комітет по телекомунікаційним кабельних систем (TR-42) випустили стандарт ANSI / TIA-942 на телекомунікаційну інфраструктуру центрів обробки даних (Telecommunications Infrastructure Standards for Data Centres), який регламентує вимоги до інфраструктури ЦОД.

Поточна версія стандарту TIA-942-A встановлює вимоги до мережевої інфраструктури, організації електропостачання, файловому зберігання,збереженню резервних копій і архівування, відмовостійкості систем,контролю доступу до мережі та мережевої безпеки, управління базами даних, веб-хостингу, хостингу аплікацій, розподілу контенту, управління кліматом, захист від зовнішніх фізичних руйнувань таких як пожежі, повені, погодних явищем, управління енергоживленням.При побудові датацентров також застосовуються стандарти ANSI / TIA / EIA-

568-B.1, ANSI / TIA / EIA-568-B.2, ANSI / TIA / EIA-568-B.3, ANSI / TIA-569-B, ANSI / TIA / EIA-606-A, ANSI / TIA / EIA-J-STD-607, ANSI / TIA-758-A, національні

правила щодо безпечного пристрою електроустановок NESC IEEE C2, національний електричний код NEC NFPA 70 (в РФ їх роль відіграють правила улаштування електроустановок 6 і 7 редакції, ПУЕ-6 і ПУЕ-7), захист IT-обладнання, стандарт NFPA 75, інженерні вимоги до універсальної телекомунікаційної стійці ANSI T1.336, рекомендовані правила заживлення і заземлення електронного устаткування IEEE 1100, рекомендовані правила для систем аварійного та резервного енергопостачання промислового та комерційного застосування IEEE 446, технічні вимоги Telcordia GR-63- CORE (NEBS) і GR-139-CORE та інші галузеві стандарти.

Проектування, будівництво, монтаж і введення в експлуатацію ЦОД здійснюється висококваліфікованими фахівцями, мають відповідне профільну освіту і забезпеченими спеціалізованим повіреним контрольно-вимірювальним обладнанням. стандарт регламентує чотири рівня ЦОД, від першого рівня, практично не забезпечує надійність, до четвертого.

Рівень 1 - базовий

- Коефіцієнт постійної готовності 99,671%.
- Схильний до порушень ходу роботи від планових або позапланових дій.
- має один канал розподілу електроживлення і охолодженнябез резервування.
- Може мати або не мати фальшпол, ДБЖ або генератор.
- Розгортається за 3 місяці.
- Річне час простою - 28,8 годин.
- повністю зупиняється для виконання робіт по плановопопереджувальному обслуговування і профілактичного ремонту.

Рівень 2 - з резервуванням

- Коефіцієнт постійної готовності 99,741%.
- Менш схильний до порушень роботи від планових і від позапланових дій.
- має один канал розподілу електроживлення і охолодження,але з резервованими компонентами (N + 1).
- Має фальшпол, ІБП і генератор.
- Час розгортання - від 3 до 6 місяців.
- Річне час простою - 22,0 годин.
- Технічне обслуговування та ремонт каналу електроживлення та інших елементів інфраструктури об'єкта вимагає зупинки процесу обробки даних.

Рівень 3 - з можливістю паралельного проведення ремонтних робіт

- Коефіцієнт постійної готовності 99,982%.
- планові дії НЕ порушує роботи, інцидент може привести до порушення.
- Має кілька каналів розподілу електроживлення та охолодження, але лише один з них активний; має резервовані компоненти (N + 1).
- Час розгортання - від 15 до 20 місяців.
- Річне час простою - 1,6 годин.
- Має достатньо потужності і розподільних можливостей для того, щоб при завантаженості одного каналу можна було обслуговувати або тестувати інший.

Рівень 4 - відмовостійкий дата-центр: коефіцієнт постійної готовності 99,995%

- Планові дії не порушують роботи, толерантний до одного інциденту найгіршого властивості без наслідків для критично важливою навантаження.
- Має кілька активних каналів розподілу навантаження й охолодження з резервними компонентами (2 (N + 1), тобто 2 ІБП з надмірністю N + 1

кожен).

- Час розгортання - від 15 до 20 місяців.
- Річне час простою - 0,4 години.

Рисунок 9 - Інженер обслуговує обладнання в центрі обробки даних [23]

Питання для перевірки

1. Яке обладнання потрібно для обробки Великих даних?
2. центр обробки даних якого рівня забезпечує максимальнунадійність?
3. Центр обробки даних якого рівня забезпечує резервування?
4. Центр обробки даних якого рівня дозволяє проводити обслуговування обладнання одночасно з обробкою даних?
5. Як багато часу необхідно для створення центру обробки даних

   "під ключ

Список джерел

1. Дивакар Майсор, Шрикант Кхупат, и Швета Джайн
   Архитектура и шаблоны больших данных. [Электронный ресурс] Режим доступа: https://www.ibm.com/developerworks/ru/library/bd-archpatterns1/index.html
2. Davy Cielen, Arno D. B. Meysman, and Mohamed Ali. Introducing Data Science. Big data, machine learning, and more, using Python tools https://www.manning.com/books/introducing-data-science
3. Hilbert, M. (2016). Big Data for Development: A Review of Promises and Challenges. Development Policy Review, 34(1), 135–174. http://doi.org/10.1111/dpr.12142
4. Environmental Protection Agency. Subpart A—General Provisions. https://www.gpo.gov/fdsys/pkg/CFR-2011-title40-vol1/pdf/CFR-2011-title40-vol1-part3-subpartA.pdf
5. Seth Gilbert and Nancy Lynch. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. https://doi.org/10.1145/564585.564601
6. Steven J. Plimpton and Karen D. Devine (2011), MapReduce in MPI for Large-scale Graph Algorithms, https://doi.org/10.1016/j.parco.2011.02.004
7. Big Data awesome list. https://github.com/onurakpolat/awesome-bigdata
8. Marr B. Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance. 2015. ISBN-10: 1118965833. https://www.amazon.com/Big-Data-Analytics-Decisions-Performance/dp/1118965833/
9. Hari Shreedharan, 2014, Using Flume: Flexible, Scalable, and Reliable Data Streaming
   ISBN-13: 978-1449368302
10. Neha Narkhede, Gwen Shapira, Todd Palino, 2017

    Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale
    ISBN-13: 978-1491936160
11. Д.И. Муромцев. Онтологический инжиниринг знаний в системе Protégé. –

    СПб: СПб ГУ ИТМО, 2007. – 62 с
12. Data is the new oil in the digital economy. https://www.wired.com/insights/2014/07/data-new-oil-digital-economy/
13. Data. Cambridge dictionary. https://dictionary.cambridge.org/dictionary/english/data

14. M.Rouse. Data Life Cycle. https://whatis.techtarget.com/definition/data-life-cycle
15. L. George. Getting Started With Big Data Architecture. http://blog.cloudera.com/blog/2014/09/getting-started-with-big-data-architecture/
16. Noll M.G. Running Hadoop on Ubuntu Linux (Multi-Node Cluster). http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster/
17. 7 phases for Data Life Cycle. https://www.bloomberg.com/professional/blog/7-phases-of-a-data-life-cycle/
18. Dresner Advisory Services, LLC. Big Data Analytics Market Study 2016. https://www.microstrategy.com/getmedia/cd052225-be60-49fd-ab1c-4984ebc3cde9/Dresner-Report-Big_Data_Analytic_Market_Study-WisdomofCrowdsSeries-2017
19. Bahga A., Madisetti V. Big Data Science & Analytics: A Hands-On Approach. 2016.ISBN-10: 0996025537. https://www.amazon.com/Big-Data-Science-Analytics-Hands/dp/0996025537/
20. Metadata Life Cycle. http://metadata.teldap.tw/design/lifecycle_eng.htm
21. Increasing Rate of Data Production Prompts Google to Rethink Data Center Storage. http://www.titanpower.com/blog/increasing-rate-of-data-production-prompts-google-to-rethink-data-center-storage/
22. Mellanox Scale-Out SN3000 Ethernet Switch Series. http://www.mellanox.com/page/products_dyn?product_family=280&mtag=sn3000_label
23. Take a 360-degree video tour of Google's Oregon data center. https://www.engadget.com/2016/03/24/google-360-video-tour-data-center/

Список інформаційних джерел для самостійної роботи

1. Blackwell M., Sen M. Large Datasets and You: http://www.mattblackwell.org/files/papers/bigdata.pdf
2. Ross N. FasteR! HigheR! StrongeR! – A Guide to Speeding Up R Code for Busy People: http://www.noamross.net/blog/2013/4/25/faster-talk.html
3. Ryan R. Rosario. Taking R to the Limit, Part II: Working with Large Datasets: http://www.bytemining.com/wp-content/uploads/2010/08/r_hpc_II.pdf
4. Big Data Specialization. https://www.coursera.org/specializations/big-data
5. Mining Massive Datasets. Online course.https://online.stanford.edu/course/mining-massive-datasets-self-paced

6. G. Press. 12 Big Data Definitions. What's yours? https://www.forbes.com/sites/gilpress/2014/09/03/12-big-data-definitions-whats-yours/
7. Big Data awesome list. https://github.com/onurakpolat/awesome-bigdata
8. Keuper F., Schmidt D., Schomann M. Smart Big Data Management. ISBN-10: 3832537686. 2014. https://www.amazon.com/Smart-Data-Management-Frank-Keuper/dp/3832537686
9. Mayer-Schönberger V. Big Data: A Revolution That Will Transform How We Live, Work, and Think. 2013. ISBN-10: 1848547927 https://www.amazon.com/Big-Data-Revolution-Transform-Think/dp/1848547927/
10. Smith M.D., Telang R. Streaming, Sharing, Stealing: Big Data and the Future of Entertainment (MIT Press). 2016. https://www.amazon.com/Streaming-Sharing-Stealing-Future-Entertainment/dp/0262034794/
11. Karimi H.A. Big Data: Techniques and Technologies in Geoinformatics. 2014. ISBN-10: 1138073199 https://www.amazon.com/Big-Data-Techniques-Technologies-Geoinformatics-ebook/dp/B00HZNQKMM/
12. Marz N., Warren J. Big Data: Principles and best practices of scalable realtime data systems. 2015. ISBN-10: 1617290343 https://www.amazon.com/Big-Data-Principles-practices-scalable/dp/1617290343/
13. Separation of storage and compute in BigQuery. https://cloud.google.com/blog/big-data/2017/11/separation-of-storage-and-compute-in-bigquery
14. Mayer-Schonberger V., Ramge T. Reinventing Capitalism in the Age of Big Data. 2018. ISBN-10: 046509368X
15. https://www.amazon.com/Reinventing-Capitalism-Age-Big-Data/dp/046509368X/
16. Bahga A., Madisetti V. Big Data Science & Analytics: A Hands-On Approach. 2016.ISBN-10: 0996025537. https://www.amazon.com/Big-Data-Science-Analytics-Hands/dp/0996025537/
17. Marr B. Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance. 2015. ISBN-10: 1118965833. https://www.amazon.com/Big-Data-Analytics-Decisions-Performance/dp/1118965833/
18. Marr B. Data Strategy: How to Profit from a World of Big Data, Analytics and the Internet of Things. 2017.ISBN-10: 074947985X. https://www.amazon.com/Data-Strategy-Profit-Analytics-Internet/dp/074947985X/
19. Jones H. Data Analytics: An Essential Beginner's Guide To Data Mining, Data Collection, Big Data Analytics For Business, And Business Intelligence Concepts. 2018. ISBN-10: 1985097974.

20. https://www.amazon.com/Data-Analytics-Essential-Collection-Intelligence/dp/1985097974/
21. Sawchik T. Big Data Baseball: Math, Miracles, and the End of a 20-Year Losing Streak. 2015. ISBN-10: 1250063507. https://www.amazon.com/Big-Data-Baseball-Miracles-20-Year/dp/1250063507/
22. Ferguson A.G. The Rise of Big Data Policing: Surveillance, Race, and the Future of Law Enforcement. 2017. ISBN-10: 1479892823. https://www.amazon.com/Rise-Big-Data-Policing-Surveillance/dp/1479892823/
23. O'Neil C. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy. 2016. ISBN-10: 0553418815. https://www.amazon.com/Weapons-Math-Destruction-Increases-Inequality/dp/0553418815/
24. Tenner E. The Efficiency Paradox: What Big Data Can't Do. 2018. ISBN-10: 1400041392. https://www.amazon.com/Efficiency-Paradox-What-Data-Cant/dp/1400041392/
25. Lei Zeng M., Qin J. Metadata. 2016. ISBN-10: 1555709656 https://www.amazon.com/Metadata-Second-Marcia-Lei-Zeng/dp/1555709656/
26. Gartner R. Metadata: Shaping Knowledge from Antiquity to the Semantic Web. 2016. ISBN-10: 3319408917
27. https://www.amazon.com/Metadata-Shaping-Knowledge-Antiquity-Semantic/dp/3319408917/
28. Baca M. Introduction to Metadata. 2016. ISBN-10: 1606064797 https://www.amazon.com/Introduction-Metadata-Third-Murtha-Baca/dp/1606064797/

Навчальне видання

КОНСПЕКТ ЛЕКЦІЙ
з дисципліни
«Методи обробки великих даних»

(для магістрів спеціальності 126 «Інформаційні системи і технології»

(Електронне видання)

Укладач:
Іванов Віталій Геннадійоич

Редактор *В.Г. Іванов*
Техн. редактор *В.Г. Іванов*
Оригінал - макет *В.Г. Іванов*

Підписано до друку _____
Формат $60\times84\ ^{1}/_{16}$. Папір типограф. Гарнитура *Times*.
Друк офсетний. Ум. друк. арк.___. Обл.-вид.арк._____.
Тираж ___ прим. Вид. №_____. Замовл. №_____. Ціна договірна.

Видавництво Східноукраїнського національного університету імені Володимира Даля

Свідоцтво про реєстрацію: серія ДК № 1620 від 18.12.03 р.
Адреса університета: просп. Центральний 59-А
м. Сєвєродонецьк, 93400, Україна
e-mail: vidavnictvoSNU.ua@gmail.com.