калібровки спектрів Cluster-based Peak Alignment (CluPA); “Peaks” надає функції для маніпуляцій із спектром, перенесений із ROOT/TSpectrum.

Ми використали дані комбінаційного розсіяння сполуки TeO_2, отримані за допомогою спектрографа ДФС-52, джерело – твердотільний лазер працював на довжині хвилі 532 нм. Ми скористались пакетом prospectr для згладжування даних та усунення шуму методом біжучого середнього, а також для побудови графіків.

Висновок: Пакет статистичних обчислень R має все необхідне для застосування до опрацювання даних спектроскопічних вимірювань, і ми вважаємо доцільним внести R разом із сучасними методами статистики до програми навчання фізиків-експериментаторів.

Джерела:

1. http://www.r-project.org/
2. GNU Project: http://www.gnu.org
3. Бібліотека CRAN: http://cran.r-project.org/
4. Journal of Statistical Software http://www.jstatsoft.org/ January 2007, Volume 18, Issue 1. An Introduction to the Special Volume “Spectroscopy and Chemometrics in R”

Бестіарій Великих Даних, або про екологію проектів навколо Apache Hadoop

Апуневич С.Є.

Астрономічна обсерваторія Львівського національного університету імені Івана Франка,
EPAM Systems,
вул. Кирила і Мефодія, 8, Львів 79005

This short report is an attempt of holistic analysis of Hadoop framework for Big Data computations in terms of ecology and complexity, to expose the beauty and ugliness of this system, imagine its future.

Бестіарій [1] – це особливий жанр старовинної літератури, щедро ілюстровані збірки описів та оповідей, певною мірою предтечі енциклопедій. В них описували тварин, комах, потвор, і навіть камені, часто безоглядно до реальності описуваного, бо автори здебільшого самі не бачили цих сутностей і покладалися на непевні джерела. Бестіарії були популярні у Середньовіччі, коли різниця між реальністю та вигадкою була нечіткою, і ніхто не ставив під сумнів правдивість відомостей викладених у книжці, а істинність потверджувалася моральним висновком, що супроводжував кожен допис. До чого тут Apache Hadoop? Я висуваю тезу, що така ж суб'єктивність часто панує у сфері сучасних високих інформаційних технологій.

Apache Hadoop [2] – це каркас, що надає змогу розподілено опрацьовувати великі набори даних на кластерах комп'ютерів за допомогою простих моделей програмування. Він спроектований із метою масштабування від кількох серверів до тисяч обчислювальних машин, використовуючи кожен вузол для зберігання даних та обчислень; також у системі закладено стійкість до несправностей, що дозволяє знизити вимоги до надійності кожного окремого вузла. Очевидно, що ці властивості досягнуто за рахунок компромісу, звуження функціональності таких систем у порівнянні із класичними реляційними базами даними.

Можна стверджувати, що за 10 років існування проект виконав свої головні обіцянки. Однак, на основі цього проекту як платформи виникло багато, навіть дуже багато супутніх проектів, які взаємодіють нетривіальним чином між собою та із платформою. Дедалі частіше ці проекти мають на увазі компенсувати компроміси закладені у Hadoop під час проектування, та наблизити операції над великими даними для користувача до оперативності, цілісності, надійності "класичних" систем опрацювання та зберігання даних. Чи це можливо і яку ціну доведеться за це заплатити?

Проекти Apache Hbase, Cassandra, Hive, Spark, ZooKeeper, Drill, Impala, Oozie та інші переслідуть різні цілі, часто конкурують між собою, деколи доповнюють одне одного. Кожен із них має дивну екзотичну назву та талісман часто у вигляді доволі дивних істот. Спроби оглядів всієї ніші продуктів для Big Data вже стають бестіаріями, оскільки жоден із авторів вже не може мати досвід роботи із всіма продуктами. Тобто, очевидна головна ціна, яку сплатили – це складність.

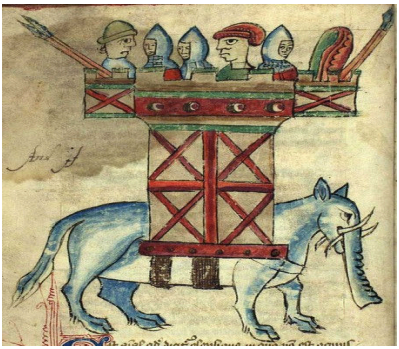
Який головний принцип інтеграції таких складних систем, як вони взагалі працюють? На згадку приходить "теорема" відомого науковця, винахідника ідеї підпрограми, Дейвіда Вілера,[3] яка звучить наступним чином:

"Всі проблеми у комп'ютерних науках можна розв'язати за допомогою іще одного рівня індирекції (непрямості) … окрім проблеми надто багатьох шарів індирекції.

Аналогічна думка закладена у Інтернет-меморандумі RFC 1925 [4]:

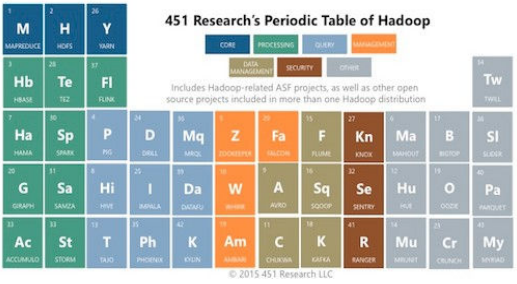
"…(6) Легше перемістити проблему деінде (наприклад, перемістити проблему в іншу частину загальної архітектури мережі), ніж її розв'язати.

(6а) (висновок). Завжди можна додати іще один рівень індирекції. ..."



Зображення слона із бестіарія, жовтий плюшевий слон є символом проекту Hadoop

Саме за таким принципом і надбудовується складна, багатошарова структура над низовою платформою Hadoop. Численні виклики програмних інтерфейсів (як через RPC, так і HTTP, або засоби серіалізації) пов'язують архітектуру докупи, хай і не надто елегантним чином.



Проекти екологічної ніші, укладені в подобу періодичної таблиці.

Інша аналогія, що напрошується для порівняння – це екологія в первинному значенні цього слова, не як щось добре чи здорове у стосунку до людини, а як наука про виживання та пристосування в складних біологічних системах. Не слід забувати, що відкрите/вільне ПЗ – це світ жорсткої конкуренції. Очевидно, що світ Великих Даних – відомий як джерело надприбутків, і проекти Apache Software Foundation зазнають сильного впливу комерціалізації, коли великі гравці на ринку намагаються диктувати власні правила, часто виходячи із маркетологічних, а не інженерних міркувань. Спостереження таке: екологічна ніша росте явно швидше, ніж заповнюється, отже боротьба за виживання гряде у майбутньому. Мають вижити тільки ті проекти, хто запропонує найбільші ефективні рішення. Швидке зростання проявляється вже зараз через брак інтеграції, технічні вимоги до продуктів змінюються, а складність стає проблемою, яку намагаються вирішити за допомогою ускладнення системи.

А поки ми всі чекатимемо невмолимої дії екологічних законів, вже зараз можна тішитися із того, що розподілені обчислення стали надзвичайно доступними для широких мас, а майбутнє їх спрощення тільки сприятиме поширенню високонавантажених обчислень.

Джерела:

1. https://uk.wikipedia.org/wiki/%D0%91%D0%B5%D1%81%D1%82%D1%96%D0%B0%D1%80%D1%96%D0%B9
2. http://hadoop.apache.org
3. https://en.wikipedia.org/wiki/David_Wheeler_(British_computer_scientist)
4. http://www.rfc-base.org/rfc-1925.html