ПРЕДИЗВИКАТЕЛСТВА НА BIG DATA И BIG DATA ANALYTICS

проф. д-р Даниела Орозова
Бургаски свободен университет

CHALLENGES OF BIG DATA AND BIG DATA ANALYTICS

Daniela Orozova
Burgas Free University

Abstract: *Ново изследователско и инженерно предизвикателство са въпросите свързани с анализ и управление на големи данни в разпределени хранилища и информационни центрове, осигуряване на качеството при извличането и обработката на такива данни, аспекти на сигурността, устойчивостта и запазване на данните. В статията се разглеждат процесите, свързани с обработка и анализ на големи данни, базирайки се на Map/Reduce парадигмата.*

Key words: *Big Data, Big Data Analytics, NoSQL, Map/Reduce, Data Science.*

1. Въведение в парадигмата Big data

Big data е ново и развиващо се понятие, което описва голям обем структурирани, полуструктурирани и неструктурирани данни. С понятието Big Data Analytics се свързва група от технологии и методи за обработка на големите данни. Предизвикателствата, пред които са изправени IT специалистите в областта, включват събиране, съхраняване, търсене, споделяне, прехвърляне, анализиране и визуализиране на данните. Характеристиките на големите данни се описват с помощта на **пет V** [1]:

- *Volume* (обем) – големи данни във физически смисъл, терминът се използва за големи информационни масиви от порядъка на петабайти и ексабайти данни.
- *Velocity* (скорост) – висока скорост на генериране на данни и анализи на потоци от данни.
- *Variety* (разнообразие) – данните постъпват в различни форми и от различни източници и трябва да се анализират и консолидират.
- Veracity (достоверност) – истинността и качеството на данните варира. При големия обем данни, тяхното качество и точност трудно се контролират.
- *Value* (стойност) – извлечените данни трябва да бъдат използваеми и полезни, съвместно с останалите данни.

Едно голямо **предизвикателство** пред Big Data и техния анализ е свързано с структурирането и съхранението на разнородните данни, който постъпват с голяма скорост. Източниците на данни обикновено са от различен тип, генерираните данни са подчинени на различни стандарти. Често данните постъпват в системата във вид неподходящ за директна обработка и интеграция, което налага допълнителни стъпки за привеждане на данните във формат, отговарящ на дефинираните правила в платформата, което е нетривиална задача и е свързана с използване на специализиран софтуер [2]. Данните се разделят в три класа според степента си на структурираност: структурирани, неструктурирани и полуструктурирани данни [3].

Пример за структурирани данни са таблици, бази от данни, доклади и др. Неструктурираните данни се генерират от: интернет на събитията; интернет на хората (социалните мрежи – Facebook, Tweeter, LinkedIN и др.); интернет на нещата; интернет на локацията (мобилни телефони, смартфони, таблети и др.) [4]. Понятието полуструкторирани данни се използва, когато данните се съхраняват във формат XML, JSON или друг. Различните типове данни изискват различен подход и софтуер за обработка.

2. Технологии за съхранение на данни

Поради необходимостта от обработка на големи масиви от данни, които се разширяват хоризонтално се използват хранилища, които предоставят механизъм за съхранение и възстановяване на данни със свободен модел. Важна цел е оптимизиране на производителността при обработка на големи обеми от данни, затова обикновено хранилищата имат разпределена архитектура.

В този случай важи теоремата CAP, известна още като теорема на Брюър, която гласи, че в една разпределена система трябва да се изберат две от следните правила:

- *Консистентност* (*Consistency* – C) – всички клиенти на базата от данни виждат една и съща информация, даже при конкурентно обновяване.
- *Наличност* (*Availability* – A) – всички клиенти на базата от данни могат да достъпват данните и да извличат информация.
- *Възможност за разделяне* (*Partition tolerance* - A) – базата от данни може да се разделя и локализира върху множество сървъри.

Едновременното осигуряване на трите не е възможно.

В повечето случаи системите за работа с нерелационни бази от данни избират наличност и възможност за разделяне; гарантирането на консистентност при разпределена база от данни изисква кворум между отделните сървъри, което води до забавяне във времето.

Различните хранилища използват различни технологии за съхранение на данните [10]:

- Технология за съхранение на данни *Key-Value*. Това е бързо, мащабируемо и високо достъпно хранилище за произволен достъп (четене/запис) до всякакви данни, асоциирани с ключ. При този подход базата от данни е съвкупност от наредени двойки от вида: <ключ, стойност>.
- Документни хранилища (*Document stores*). Това са хранилища от тип ключ-стойност, където стойността е документ, представен отново във вида ключ-стойност и служи за работа с документи в структуриран или неструктуриран формат. Те се разделят на два типа, като програмния код е оптимизиран за работа с определени външни формати:

- хранилища за съхранение и анализиране на неструктурирани данни в формат (PDF, MS Word, Excel и др.) Неструктурираните данни са данни, които се различават по състав и организация, както и по съдържание. Друга важна характеристика е високото ниво на свързаност между тях, например документ може да е свързан с непредвидим брой други документи, събития или лица.

- хранилища за съхранение и анализиране на структурирани данни (JSON, XML, YAML и други формати). Различните типове данни изискват различен подход и софтуер за обработка.

- Графови хранилища (*Graph stores*). Тези хранилища с данни, съхраняват и анализират графови структури [7]. Този тип хранилища се използват за записване на семантика на данни; записване на връзки и отношения между обекти; обработка на графови структури с цел търсене и вземане на решения и др. Пример за употреба във

Facebook – поддържане на графова структура за всеки потребител, като всеки профил е свързан с неговите приятели, харесвания на някой статуси, изпращани съобщения, коментирани постове и други много и разнородни данни. При това за работата на системата се поддържат милиони профили. Системи от този вид се използват за доставка на съдържание за уеб, управление и търсене на документи в предприятия, поддръжка на платформи за игри, медия и др.

Съвременните приложения трябва да съхраняват и обработват множество други данни – чертежи, карти, мултимедия, WWW и др. Те изискват обработка на различни структури. Тези изисквания се удовлетворяват до определена степен от обектно-ориентираните, обектно-релационните, пространствените и други типове системи за съхранение и управление на данни.

Друго **предизвикателство** пред Big Data и техния анализ е свързано с обема на големите данни. Обикновено те не могат да бъдат съхранявани на една машина, защото ще е необходимо много време за тяхната обработка. Новите инструменти за големите данни използват разпределени системи, така че данните могат да се съхраняват и анализират в разпределени бази от данни и да се обработват паралелно. В резултат на това могат да се решават мащабни изчислителни задачи като се обработват големи масиви входно/изходни данни и да се извършват голям брой изчисления.

3. Big Data Analytics and Data Mining

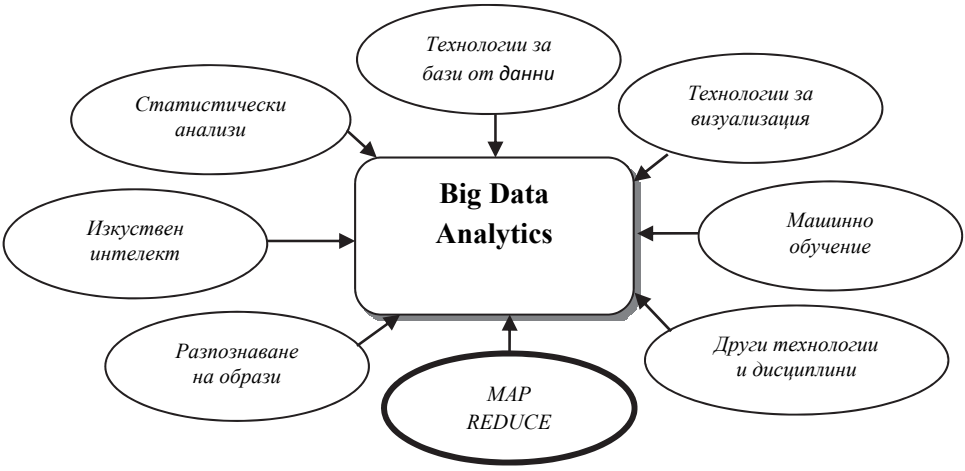
Днес с навлизането на Интернет на нещата в нашия живот, обвързването на съвременните технологии с Big data е от особено значение. Нараства нуждата от анализ на данните, за извличане на полезна информация. Например за по-добро разбиране и насочване към клиентите, компаниите могат да допълват своите бази от данни с нови данни от социални мрежи, браузъри, сензорни и др., за да получат по-пълна картина на своите клиенти. Банките и телекомуникационните компании могат по-добре да предвидят изгубването на клиенти; търговците могат да определят кои продукти ще се продадат заедно, а автомобилните застрахователни компании могат да проучат колко добре клиентите им шофират. Основната цел е да се създадат прогнозни модели, на базата на които да се взимат управленчески решения.

Термините Big Data, Big Data Analytics и Data Mining описват както самите данни, така и технологиите за събиране, обработка, управление на данните и методите за анализ.

Data Mining е процеса на търсене на скрити данни и закономерности, предварително неизвестни, нетревиялни и практически полезни, необходими за взимане на решения в различни сфери на човешките дейности. Тук акцентът е не само в извличане на нови факти, но и генериране на хипотези, които могат да бъдат проверявани. Традиционните инструменти на анализа се основават на математическата статистика – регресия, корелация, клъстеризация, анализ на времеви редове, дървета на решенията и др., а също и техники на изкуствения интелект като: машинно обучение, невронни мрежи, генетични алгоритми, размити логики и др.

Big Data Analytics се явява развитие на концепцията Data Mining. Също така е и развитие на решаваните задачи, сфери на приложение, източници на данни, методи и технологии на обработка. От появата на концепцията Data Mining до настъпване на ерата на BigData, се изменя обема на анализираните данни, появяват се високопроизводителни системи, нови технологии, в това число *Map/Reduce* и нейните многочислени програмни реализации.

Може да се каже, че Data Mining е „сплав“ от различни дисциплини и технологии. Но когато схемата е допълнена с технологията *MapReduce* и изискванията, произтичащи от 5-те V, тя отразява функционалните връзки на *Big Data Analytics* (фигура 1).



Фигура 1. Функционални връзки на Big Data Analytics

Независимо дали целта е да се открият интересни взаимовръзки, да се категоризират обекти в групи, да се оптимизира планирането на ресурси или да се определят тарифи за таксуване, основното разбиране на Data Mining техниките, може да помогне за извличане на полезни знания от големите обеми данни.

При задачата за *класификация* е зададено крайно множество от обекти, за които е известно към кои класове принадлежат, а класовата принадлежност на останалите обекти е неизвестна. Трябва да се построи алгоритъм, който класифицира произволен обект като укаже стойността на целевия атрибут. Когато възможните стойности на целевия атрибут са само две, то имаме „бинарен" класификационен проблем, в другия случай – „многокласов". Най-общо алгоритъмът работи като създава серия от случайни правила. Предварително данните са разделени на две групи с взаимно изключващи се елементи – тестови и тренировъчни групи данни. Генерират се правила и се избират тези, които да отговарят най-добре на данните. Процесът се повтаря определен брой пъти, докато се намери правило, което да удовлетворява (почти 100%) тренировъчните данни. След това правилата се проверяват чрез тестовите данни.

Регресионният анализ дава отговор на въпроса какви са причините. Той показва взаимните отношения между величините, които могат да бъдат интерпретирани като причинно-следствени. Това е статистически анализ, предназначен да дава количествен израз на ефектите на дадена група променливи X_1, X_2, ..., Xp , които условно се наричат „независими" върху друга променлива Y, която се нарича „зависима". Основната идея е търсене на естествена функционална връзка от вида: y = f (x1 , x2 , , xp). Уравнението се нарича уравнение на регресия.

Асоциативен анализ – изучава честотата на съвместно появяване на факти. Този анализ е свързан с откриване на „асоциативни правила", задаващи условия за стойностите на атрибутите, които се явяват често заедно в дадено множество от данни. Асоциативните правила имат вида: X => Y. Така правилото се състои от две части: X е условната (предшестваща) част, а Y е логическото следствие (резултантна част) [9].

При *клъстерния анализ* – целта е *n* на брой обекта да се групират в *k* на брой групи, наречени клъстери, като се използват *p* на брой признаци (променливи). Така клъстерът се формира от подобни обекти, независимо от техните класове. Целта е разкриване на евентуално скрита групировка на обектите. Едно важно деление на

клъстеризационните процедури е в зависимост от това, дали се задава предварително броя на клъстерите. Голямото разнообразие на процедурите се поражда от използваните правила за създаване на клъстерите [5]. По-известни са методите „на най-близкия съсед", „на най-отдалечения съсед", „на центроидите" и др.

Анализ на шумове – в данните могат да се съдържат обекти, които не поддържат основното поведение или модел на данните. Те се наричат отклонения (*Outliers*) и се определят от Data Mining системите като шумове. Понякога, обаче, тези данни може да са по-интересни от останалите случаи. Отклоненията са разликите между измерените стойности и съответните очаквания на базата на предишни или нормативни стойности. Една Data Mining система при откриване на множество от отклонения би могла да опише характеристиките на отклоненията, да се опита да обясни причината за това, да предложи действия за довеждане на стойностите обратно към техните очаквания.

Науката за данните (*Data science*) позволява да се съчетаят множество подходи като включва техники, свързани с анализ на данни от областта на статистиката, дейта майнинг и откриване на знания, машинно обучение, изкуствен интелект, програмиране, комуникация др. Науката за данните включва и процесите по изчистване и интеграция на данните, избор и трансформация на данни, извличане на знания, техния анализ, оценяване и представяне.

4. Парадигмата MapReduce за разпределени изчисления

MapReduce е основна парадигма за разпределени изчисления на големи масиви от данни. Идеята е разработена от Google през 2004 г за индексиране на уеб страници. Днес програмите, написани в MapReduce стил са с автоматична паралелизация и се изпълняват на големи клъстери от компютри с общо предназначение, като един клъстер може да се състои от стотици или хиляди машини. Run-time системата се грижи за детайлите, свързани с разделянето на данните, управлението на изпълнението на програмата върху множество машини, обработка на откази (сривове), и управление на необходимата комуникация между машините. MapReduce [13] е парадигма за изразяване на изчислителни задачи, която улеснява програмирането, осигурява мащабируемост, устойчивост на откази и ефективно натоварване и разделяне на данни с цел паралелната им обработка.

Работа на MapReduce алгоритъма се състои от две стъпки: *Map* и *Reduce*. Основната идея на тази парадигма е входните файлове да се разделят на *M* входни части. Отделните входни части се разпределят автоматично от управляваща програмата (master) между една или повече машини или други части на програмата (workers). Те получават входните данни и задачи за предварителна обработка. Резултатите от *Map*-стъпката се записват в междинни файлове.

На *Reduce* – стъпката данните от предварителната обработка се свиват. Главният възел получава отговорите от работните възли и генерира резултата – решението на проблема, който първоначално е формулиран.

Така потребителят задава функцията Map, която обработва входните данни и генерира набори от междинни двойки от вида *ключ* : *стойност*. Получените междинни списъци с ключове се разделят на R части и се обработват самостоятелно чрез Reduce процедурите. Кой ключ в коя група попада се определя чрез използване на разделяща функция, например *hash(key) mod R* – остатък при целочислено деление на хаш_кода_на_ключа и желания брой R групи. Броят на деленията (R) и функцията за разделяне се специфицират от потребителя, а функцията Reduce, обединява всички междинни стойности, свързани с един и същ ключ [13].

Съществува голямо разнообразие от програмни реализации на MapReduce технологията. Google [14] е реализирал MapReduce на C++ с интерфейс на езиците Python и Java. Apache Hadoop [13] е реализация с отворен код на езика Java. Phoenix е MapReduce реализация на езика C чрез използване на разделена памет. MongoDB позволява да се използва MapReduce за паралелна обработка на запитвания от няколко сървъра. Skynet е реализация с отворен код на езика Ruby. Disco е реализация, създадена от компанията Nokia, като нейното ядро е написано на езика Erlang, а приложения за нея могат да се създават на Python. Qizmt е реализация с отворен код от MySpace, написана на C#. Конкретният избор зависи от много фактори и от наличната среда за работа.

5. Заключение

Съвременните тенденции в компютърните системи са свързани с развитието на високопроизводителни изчисления (HPC) и работа с феномена големи данни. Посоката „BigData“ концентрира усилията в организирането, съхранението, обработката и анализа на огромни масиви от данни. Обработката и съхранението на големите данни изисква нов поглед и съвместно прилагане на редица утвърдени технологии. Отворени проблеми в направлението са свързани с оптимизиране на достъпа до големи обеми от данни, чрез прилагане на агенти за извличане на знания от данните и нови техники за анализ на данните. Важно условие за успешното развитие на икономиката на настоящия етап е способността да се улавят и анализират огромни масиви и информационни потоци [11]. Налага се мнението, че овладяването на ефективни методи за работа с Big Data е условие за индустриална революция.

Литература:

[1] Andrea De Mauro, Marco Greco and Michele Grimaldi. „What is Big Data? A Consensual Definition and a Review of Key Research Topics”. In „AIP Proceedings” 2014, „4th International Conference on Integrated Information”.
[3] Boyd D., Crawford K., Critical questions for Big Data, 2012 уеб ресурс: http://www.tandfonline.com/doi/pdf/10.1080/1369118X.2012.678878
[3] Bernard Marr., „Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance”. John Wiley & Sons Ltd, 2015.
[4] EMC Education Service, Data Science and Big Data Analytics, John Willey & Sons, 2015.
[5] Hornick, M., Marcade, E., Venkayala, S. (2006). Java data mining: strategy, standard, and practice, A practical Guide for Architecture, Design, and Implementation.
[6] Geoffrey Fox, Big Data HPC Convergence and a bunch of other things, 02/04/2016, http://www.slideshare.net/Foxsden/big-data-hpc-convergence-and-a-bunch-of-other-things.
[7] Ian Robinson, Jim Webber & Emil Eifrem, Graph Databases, Neo Technology, O’Reilly, second edition, 2015.
[8] Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman. „Mining of Massive Datasets”. Cambridge University Press. 2012.
[9] Patrick J. Wolfe. „Making sense of big data”. PNAS. November 5, 2013, vol. 110, no. 45, 18031–18032.
[10] Database guide. (2017). What is a Document Store Database?, Available at: http://database.guide/what-is-a-document-store-database/.
[11] Шваб, К. Четвъртата индустриална революция (превод от английски). Издателство „ХЕРМЕС”. Пловдив, 2016, 240, ISBN 978-954-26-1630-6.
[12] Рос, А. Индустриите на бъдещето (превод от английски). „НСМ Медия”, С., 2017, 216. ISBN 978-954-9913-61-3.
[13] Apache Hadoop. http://hadoop.apache.org
[14] Google. https://developers.google.com/maps/documentation/geocoding/