

Съвременни тенденции в развитието на базите от данни

Митко Рагоев*

Резюме: Базите от данни отгавна са се наложили като ефективна и надеждна технология за организация и съхранение на данни в почти всички сфери на човешката дейност. Освен масовото им приложение в оперативната дейност на практически всички предприятия и организации, те са и източници на данни за научните изследвания и анализи на икономическите процеси.

Базите от данни се развиват бурно през втората половина на 20-и век и като резултат се появяват релационните бази от данни, които са един изключително мощен инструмент за съхранение и достъп до данните. От началото на 21-ви век, с изключителното нарастване на обеми на съхраняваната и обработваната информация, както и на значителния дял на неструктурираната и полуструктурираната информация, се появяват и получават своето развитие нов тип бази от данни, каквито са NoSQL базите от данни.

В настоящата статия се прави опит да се дефинират характеристиките на различните видове бази от данни, като на тази база се анализират техните предимства и недостатъци и се очертават сферите им на приложение. Представени са и тенденциите в развитието на базите от данни от гледна точка на автора.

* Митко Рагоев е доктор, доцент в катедра „Информационни технологии и комуникации“ на УНСС.

Ключови думи: бази от данни, релационни бази от данни, NoSQL бази от данни.

JEL: C88, C89.

1. Въведение

През последните двадесет години настъпиха сериозни промени в теорията и практиката на базите от данни. Наред с продължаващата доминация на релационните бази от данни се появиха и получиха широко разпространение така наречените NoSQL бази от данни. Като отговор на тази тенденция се появи и категорията NewSQL бази от данни, като нов клас съвременни релационни бази данни, които да постигнат производителността на NoSQL базите от данни.

Големите производители на релационни системи за управление на бази от данни като Microsoft и Oracle от своя страна също въведоха редица нововъведения. Те разработиха NoSQL бази от данни като отделни продукти (например Microsoft Azure Cosmos DB, Oracle NoSQL Database), също така осигуриха използването на нерелационни структури в рамките на релационните системи за управление на бази от данни (например PolyBase и Big Data Clusters в Microsoft SQL Server; Oracle Big Data SQL на Oracle).

По-нататък в статията ще бъдат представени характеристиките и тенденциите в развитието на различните видове системи за управление на бази от данни.

2. Реляционни бази от данни

Първите бази от данни, които възникват, използват йерархични (дървовидни) или мрежови структури за съхранение на данните. След тях се появяват реляционните бази от данни, които бързо натрупват популярност и до днес все още са най-разпространените бази от данни.

Решителният напредък в развитието на базите от данни настъпва през 1970 г., когато Едгар Франк Код публикува знаменития си труд за реляционния модел на данните (Codd, 1970). Към края на 70-те и в началото на 80-те години на 20-и век вече има разработки на СУБД, основани на реляционния модел на данните.

Прилагането на реляционния модел в практиката води до създаването на изключително мощни и гъвкави инструменти, каквито са реляционните системи за управление на бази от данни, които бързо се налагат и получават изключително широко разпространение. Последващото развитие на тези системи води до разширяване на техните възможности, като например създаването на езика SQL, което още повече затвърждава тяхното доминиращо положение, което е факт и до днес.

2.1. Характеристики на реляционните бази от данни

Реляционните бази от данни имат редица характеристики. Тук е направен опит за систематизиране на най-съществените от тях.

- **Структуриране**

Данните в базата от данни са структурирани съгласно изискванията на използвания модел на данните. От друга страна, структурата на данните трябва в максимална степен да съответства на същностите в реалния свят и техните свойства.

- **Свързаност**

Освен информация за същностите от ре-

алния свят, в базата от данни се съхранява информация и за съществените връзки между тях. В йерархичния и мрежовия модел тези връзки се задават явно. В реляционния модел няма такова изискване, могат да се осъществяват връзки по всяка двойка атрибуту (или множества от атрибуту), имащи общ домейн. На практика, обаче, почти във всички реляционни бази от данни има някаква форма на задаване на съществените връзки, например чрез ограничения, налагани на външните ключове.

- **Метаданни**

Заедно със самите данни, в базата от данни се съхраняват и метаданни, т.е. данни за самите данни. Метаданните включват информация за структурирането на данните, както и ограничения за осигуряване на цялостност на данните.

- **Споделяне**

Базата от данни е общ ресурс в рамките на предприятията или организациите и се използва от повечето (в идеалния случай всички) приложни програми, обслужващи тяхната дейност.

- **Контрол на дублирането**

Базите от данни се опитват да елиминират дублирането на данните или да го сведат до минимум. В реляционния модел, например, е необходимо да се дублират стойностите на първичните ключове като външни ключове, за да се моделират връзките между същностите.

- **Цялостност на данните**

В базите от данни цялостността обикновено се осигурява чрез налагане на ограничения. Ограниченията могат да се прилагат както към същностите и техните свойства, така и към връзките между тях.

- **Сигурност на данните**

Сигурността в базите от данни е свързана най-вече с опазването на данните от неоторизиран достъп. Това се осъществява чрез различни механизми, например чрез потребителски имена и пароли. Достъпът на потребителите до данните може да бъде

Икономическо развитие

ограничаван и според вида на операцията (извличане, добавяне, корекция, изтриване).

- **Надеждност на данните**

Базите от данни гарантират надеждното съхранение на данните и разполагат с механизми за възстановяване, в случай че базата от данни е повредена по някакъв начин.

- **Потребителски изгледи към данните**

Базите от данни дават възможност за дефиниране на различни потребителски изгледи към едни и същи данни. Всеки потребител може да има специфичен изглед, представен във форма, която е позната за този потребител. Изгледът включва само тези същности, атрибути и връзки в „реалния свят“, от които потребителят е заинтересован.

- **Независимост на данните**

Базите от данни се основават на архитектурата на няколко нива, осигуряваща независимост на външната структура (потребителските изгледи) от логическата структура на данните, както и на логическата структура от физическото представяне на данните.

- **Единен механизъм за съхранение, актуализация и извличане на данни**

При базите от данни е осигурен на потребителите и приложните програми единен механизъм за съхраняване, актуализиране и извличане на данни от базата от данни, чрез системата за управление на бази от данни (СУБД).

- **Потребителски заявки**

Данните в базата от данни са пряко достъпни за крайните потребители. Много СУБД предоставят езици за заявки или генератори на отчети, които позволяват на потребителите да получават необходимата им информация, без да се пише програма за извличане на тази информация от базата от данни.

- **Поддръжка на транзакции**

В базите от данни е осигурен механизъм, който да гарантира изпълнението на транзакции, т.е. последователности от действия, които са свързани логически и следва да бъдат изпълнени като едно цяло. Изпълнение-

то на транзакциите трябва да отговаря на следните изисквания:

- ✓ Атомарност – транзакцията е едно цяло. Или всички действия се изпълняват успешно, или цялата транзакция се отменя;
- ✓ Консистентност – след изпълнението на транзакцията базата от данни трябва да е в консистентно (непротиворечиво) състояние;
- ✓ Изолация – изпълнението на една транзакция не трябва да влияе на изпълнението на останалите;
- ✓ Устойчивост – резултатът от транзакцията трябва да бъде съхранен надеждно.

3. NoSQL бази от данни

Какво представляват NoSQL базите от данни? Този термин се тълкува по различни начини. Буквално би трябвало да се отнася до базите от данни, които не използват езика SQL. Съществува и тълкуване, че означава не само (Not Only) SQL. В крайна сметка, става дума не само за неизползващи езика SQL бази от данни, а въобще за нерелационни бази от данни и поради това по-точният термин би бил нерелационни бази от данни (Non-Relational Databases). От друга страна, не става дума и за съществуващите преди релационните бази от данни йерархични и мрежови бази от данни, а за нови бази от данни, основани на различни модели на данните. В изложението ще бъде представена накратко историята на NoSQL базите от данни и ще бъдат разгледани по-подробно различните NoSQL бази от данни според моделите, на които са базирани.

3.1. Причини за възникването на NoSQL базите от данни

Това ново направление в базите от данни започва своето развитие през първите години на 21-ви век с развитието на интернет приложенията и най-вече приложенията на Google. Една от първите публикации по те-

мата е от 2003 г. и е свързана с Google File System (Sanjay Ghemawat, 2003). Интернет приложенията на Google са последвани от Yahoo, Amazon, а впоследствие от Facebook, Netflix, eBay и много други, които водят след себе си много нови разработки в сферата на NoSQL базите от данни.

Кои са основните причини за възникването и бурното развитие на това ново направление? Ето най-важните от тях:

- **Необходимост от съхраняване и обработка на огромни обеми от данни**

Всекидневно нараства обемът на съхраняваните от различните приложения снимки, видео информация, географска информация и т.н. Това създава сериозни проблеми пред базите от данни, като например необходимостта да се предвиди нарастването на обема им (Mihova, 2012).

- **Необходимост от достъп в реално време**

Съхраняваните огромни обеми от данни трябва да са достъпни в реално време от всяка една точка на света.

- **Необходимост от гъвкавост на структурите**

Възможност за лесна и бърза промяна в структурата на данните.

- **Необходимост от съхранение на неструктурирани данни**

Заедно със структурираните данни трябва да се съхраняват и неструктурирани или полуструктурирани данни. Това е предизвикателство за повечето реляционни бази от данни и изисква специален подход (Milev, 2015).

- **Необходимост от скалируемост**

Необходимо е лесно разширяване на мащаба на приложенията и съхраняваните данни.

Реляционните бази от данни трудно отговарят в необходимата степен на всички нови изисквания. Това отваря ниша за създаване и развитие на множество нови системи, базирани на различни модели.

3.2. Видове NoSQL бази от данни и техните характеристики

При NoSQL базите от данни могат да бъдат идентифицирани следните основни типове, в зависимост от използвания модел на данните (Vaish, 2013):

- **Key-value store**

Това е един от най-простите модели за съхранение на данните. Съхраняват се двойки ключ-стойност, като ключовете са уникални. Достъпът до данните става чрез търсене по стойностите на ключа. Подходящ е за съхранение на големи обеми от данни и осигурява бърз достъп по ключа. Съществуват различни разновидности според паметта, в която се съхраняват данните, подреждането по ключа и консистентността на данните. Най-известни представители са Redis, Memcached, Berkeley DB и Oracle NoSQL.

- **Document store**

Както може да се разбере от името им, те са създадени специално за съхранение на документи. Използваните формати са XML, JSON, BSON и др. Данните са полуструктурирани – съдържат двойки име на атрибут-стойност. Достъпът до данните става чрез търсене както по стойностите на ключа, така и по стойностите на атрибутите. Подходящи са за съхранение на текстови и XML документи и други полуструктурирани данни. Най-известни продукти от този тип са MongoDB, Amazon Dynamo DB, Couchbase, CouchDB.

- **Column-oriented**

За разлика от реляционните бази от данни, които са ориентирани по редове, при този тип структури данните са ориентирани по колони. Това дава възможност за лесно разширяване на структурата на данните, чрез добавяне на нови колони. Подходящи са за големи обеми разпределено съхранявани данни. Типични представители са Cassandra, HBase и Google Bigtable.

- **Graph**

Икономическо развитие

Данните се представят чрез подобна на граф структура, като възлите на графа представят обектите с техния набор от атрибути, а ръзките на графа – връзките между обектите. Подходящи са за съхранение на данни, връзките между които са особено важни за моделиране. Най-известните представители на този тип са Neo4j, Titan, Giraph.

Съществуват и продукти, базирани на гва или повече различни модели на данните. Такива са OrientDB, ArangoDB и гр.

Повечето NoSQL бази от данни се отказват от традиционната за релационните бази от данни поддръжка на транзакции (Lazarova, 2013), тъй като тя затруднява разпределянето на данните. Докато релационните бази данни са насочени към осигуряване на непротиворечивост на данните, NoSQL базите от данни в повечето случаи

дават приоритет на високата достъпност и разпределянето на данните. Тази тенденция води до създаването на системи, известни като BASE (Basically Available, Soft-state, Eventually consistent). Не може да не се отбележи, че тези системи, опитвайки се да решат някои проблеми, създават други, понякога по-сериозни проблеми.

3.3. Сравнение на NoSQL базите от данни с релационните бази от данни

Интересно е доколко NoSQL базите от данни притежават характеристиките на релационните бази от данни. Едно такова сравнение може да даде отговор на въпроса дали NoSQL базите от данни могат да заменят изцяло релационните бази от данни,

Таблица 1. Сравнение на характеристиките на релационните и NoSQL базите от данни

Характеристика	Key-value (Redis)	Document (MongoDB)	Column-oriented (Cassandra)	Graph (Neo4j)
Структуриране	не	частично	да	да
Свързаност	не	не	не	да
Метаданни	не	не	не	не
Споделяне	да	да	да	да
Контрол на дублирането	не	не	не	не
Цялостност на данните	не	не	не	частично
Сигурност на данните	не	да	да	да
Надеждност на данните	да	да	да	не
Потребителски изгледи към данните	не	не	не	не
Независимост на данните	не	не	не	не
Единен механизъм за съхранение, актуализация и извличане на данни	да	да	да	да
Потребителски заявки	не	не	не	не
Поддръжка на транзакции	да	не	частично	да
Общо	31%	35%	42%	50%

или те са подходящи за приложение само в ограничени области.

Конкретните продукти от всеки от типовете NoSQL бази от данни се различават в значителна степен по своите възможности. За нуждите на анализа, за всяка категория се вземат характеристиките на най-популярните продукти. Това са съответно Redis за Key-value stores, MongoDB за Document stores, Apache Cassandra за Column-oriented stores и Neo4j за Graph-oriented stores. При определяне на характеристиките на продуктите са използвани данни от техните уеб сайтове (Redis Labs, 2020), (MongoDB, 2020), (Apache Software Foundation, 2016), (Neo4j Inc., 2020), както и (A B M Moniruzzaman, 2013). Резултатите от сравнението са обобщени в таблица 1.

Резултатите от сравнението са красноречиви – никой от разглежданите продукти не покрива повече от 50% от характеристиките на релационните бази от данни. Същевременно, трябва да се има предвид следното:

- ✓ Не всички характеристики на базите от данни са еднакво важни за определяне на тяхната същност;
- ✓ Достъпната информация за разглежданите продукти не е достатъчна за да се определи категорично притежаването или непритежаването на определена характеристика;
- ✓ Граф-ориентирани системи се различават от останалите системи и имат потенциал да постигнат характеристиките на релационните бази от данни.

Тъй като нито един от разглежданите продукти не притежава над 50% от характеристиките на релационните бази от данни, използването на термина „база от данни“ по отношение на който и да е от тях не е напълно коректно.

Граф-ориентирани структури за съхранение най-плътно се приближават по

характеристики до релационните бази от данни и имат потенциал да се развият в пълноценни бази от данни.

Повечето NoSQL „бази от данни“ са по-скоро хранилища на данни, които са подходящи за създаване на системи, за които консистентността на данните не е абсолютно задължителна и които не изискват поддръжка на транзакции.

3.4. Развитие на NoSQL базите от данни

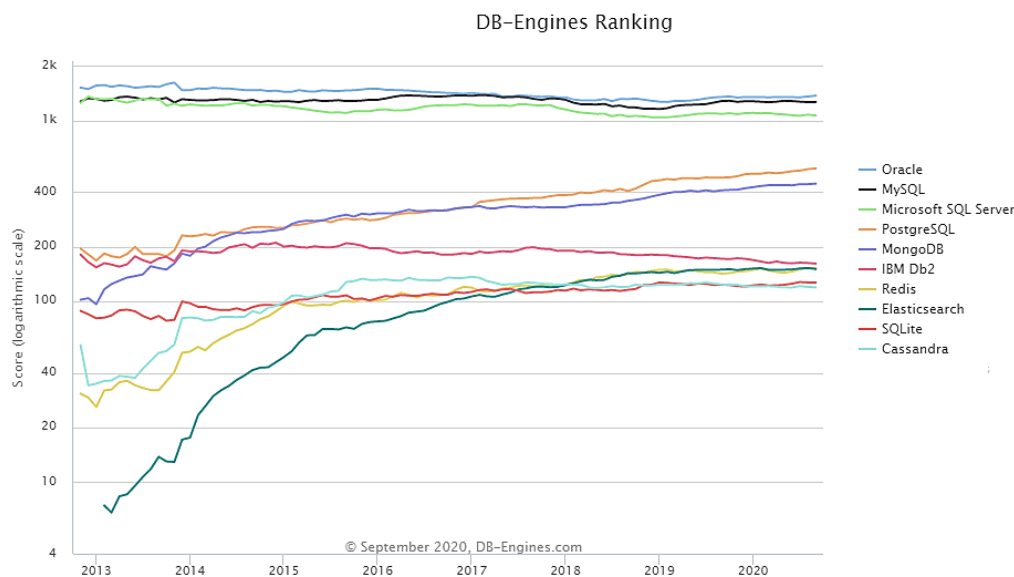
В повечето класации на системите за управление на бази от данни, представителите на NoSQL базите от данни отгавна са сред първите 10. Според сайта DB-engins (Solid IT GmbH, 2020), първите 10 СУБД по популярност за месец септември 2020 г. са:

1. Oracle
2. MySQL
3. Microsoft SQL Server
4. PostgreSQL
5. MongoDB
6. IBM DB2
7. Redis
8. Elasticsearch
9. SQLite
10. Cassandra

Според изследване на StackOverflow (Stack Overflow, 2020) сред разработчиците, най-популярните 10 СУБД за 2020 г. са:

1. MySQL
2. PostgreSQL
3. Microsoft SQL Server
4. SQLite
5. MongoDB
6. Redis
7. MariaDB
8. Oracle
9. Firebase
10. Elasticsearch

И в двете класации фигурират NoSQL СУБД като MongoDB, Redis и Elasticsearch, а също и Cassandra. Интересно е, обаче, че за разлика от периода преди 10-15 години, ко-



Фигура 1. Динамика на показателите по години на първите 10 СУБД (Solid IT GMBH, 2020)

гато имаше бурно навлизане в практиката на NoSQL базите от данни и техните представители се изкачваха бързо в класациите, в последните 5 години не се забелязва сериозно придвижване нагоре в класациите на техни представители, а те по-скоро запазват достигнатите позиции. Единствено MongoDB увеличава оценката си в последните 5 години, макар и недостатъчно, за да промени мястото си в класацията.

Динамиката на показателите на участващите в първите 10 СУБД може да бъде проследена на фигура 1.

Докато доскоро NoSQL базите от данни бяха нещо много модерно и нарастваше използването им не само при разработването на нови системи, но и като алтернатива на релационните бази от данни в някои съществуващи системи, сега като че ли положението се нормализира. Може би както разработчиците, така и потребителите вече са по-наясно с това какво избират и какви биха били последствията от техния избор. Доскоро потребителите очакваха да

получат всичко това, което са свикнали да получават от базите от данни, съчетано с предимствата, които имат NoSQL базите от данни. За съжаление, всяко нещо си има цена и всяко от предимствата на NoSQL базите от данни се заплаща със загубата на традиционните качества на базите от данни – контрол на гублирането на данни, осигуряване на цялостност на данните, поддръжка на транзакции и т.н. По ирония на съдбата, нещото, което най-много липсва на NoSQL базите от данни, е именно SQL (Tiwari, 2011).

4. Нови възможности в релационните бази от данни

Един от най-добрите примери за разширяване възможностите на релационните системи за управление на бази от данни в последните години е Microsoft SQL Server. Като типична релационна СУБД, Microsoft SQL Server не е проектиран да съхранява големи данни, нито да съхранява и обра-

ботва неструктурирани данни, като медийни файлове. За да разреши този проблем, Microsoft въвежда редица нововъведения в последните версии на Microsoft SQL Server.

4.1. PolyBase

В Microsoft SQL Server 2016 за първи път е включена възможността да се интегрират данни от външни източници, включително NoSQL и BigData чрез компонента PolyBase.

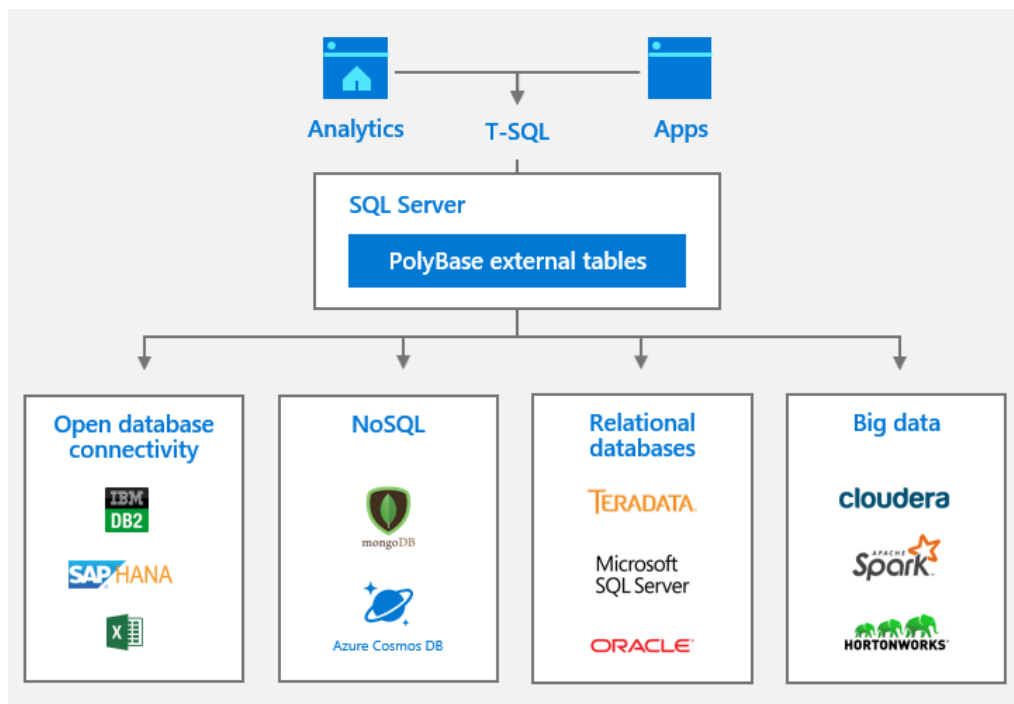
PolyBase позволява да се правят заявки върху данни, съхранявани в Hadoop, Cloudera и Azure Blob Storage. За целта се дефинира външна таблица в SQL Server, която след това може да се използва в заявките, както всяка друга таблица от релационната база от данни. Когато заявката се изпълни, данните от външния източник на данни се извличат и предоставят на потребителя без да се съхраняват постоянно в SQL Server.

Могат да се правят заявки към данни от външни таблици, свързани с таблици от релационна база от данни в SQL Server. Тази опция може да бъде полезна, когато трябва да се използват съвместно данни от различни системи, например да се интегрират данни от хетерогенни системи (Tsaneva, 2019), да се комбинират динамични данни с данни от релационна база данни, както и в много други случаи (Lazarova, 2015).

SQL Server 2019 разширява възможностите на PolyBase с нови конектори за създаване на външни таблици, включително Oracle, Teradata, MongoDB, Azure CosmosDB или други източници на данни през ODBC. Различните източници на данни могат да се видят на фигура 2.

Освен за директни заявки, PolyBase може да се използва и за:

- ✓ Импортиране на данни от Hadoop, Azure Blob Storage или Azure Data Lake Store в



Фигура 2. Източници на данни, интегрирани от PolyBase (Microsoft Corporation, 2018)

Икономическо развитие

- реляционни таблици.
- ✓ Експортиране на данни в Hadoop, Azure Blob Storage или Azure Data Lake Store.
- ✓ Интегриране с BI инструменти.

4.2. Big Data Cluster

В Microsoft SQL Server 2019 се разширява платформата за данни, за да обхващат големи и неструктурирани данни, като се интегрира Apache Spark и HDFS в кълстера за големи данни – Big Data Cluster (Microsoft Corporation, 2018).

Big Data Cluster използва мащабируем слой за съхранение, който интегрира SQL Server и HDFS. Също така, интеграцията на SQL Server с Apache Spark позволява използването на библиотеки за обработка на данни с отворен код, както и обработка и анализ на големи обеми данни в разпределен изчислителен слой в паметта.

На фигура 3 са показани компонентите на Big Data Cluster в SQL Server.

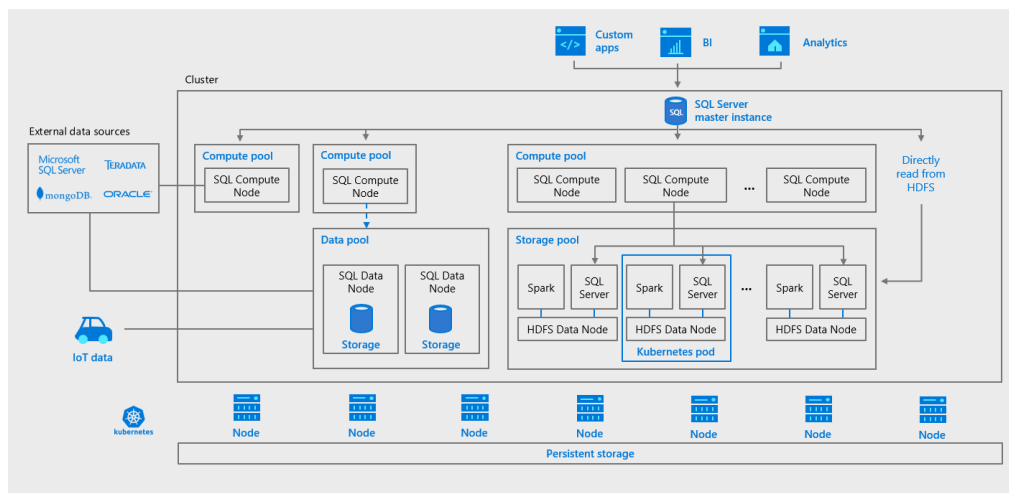
данни. Данните могат да се гоставят от SQL заявки или задания (jobs) на Spark.

Пултът за съхранение (Storage pool) се състои от множество групи, включващи SQL Server, HDFS възли за данни и Spark контейнери. Всички възли за съхранение в Big Data Cluster са членове на HDFS кълстер. Това осигурява мащабируемо ниво на съхранение.

Пултът за изчисления (Compute pool) гостава изчислителни ресурси на кълстера. Изчислителният пул съдържа няколко групи от един или повече контейнери. Те могат да използват данни от SQL Server или външни източници – Oracle, HDFS или групи.

Контролерът (Controller) е отговорен за управлението и сигурността на кълстера. Включва услугата за контрол и други услуги като Kibana, Elastic Search и InfluxDB.

Big Data Cluster има значителни възможности за съхраняване и обработка на големи данни. Има различни опции за използване на данни от множество различни източници. Данните могат да се използват за



Фигура 3. Компоненти на Big Data Cluster (Microsoft Corporation, 2018)

Пултът за данни (Data pool) се използва или за кеширане на данни от външен източник, или за съхраняване на входящи поточни

анализ, машинно обучение и изкуствен интелект.

Кълстерът за големи данни на SQL

Икономическо развитие

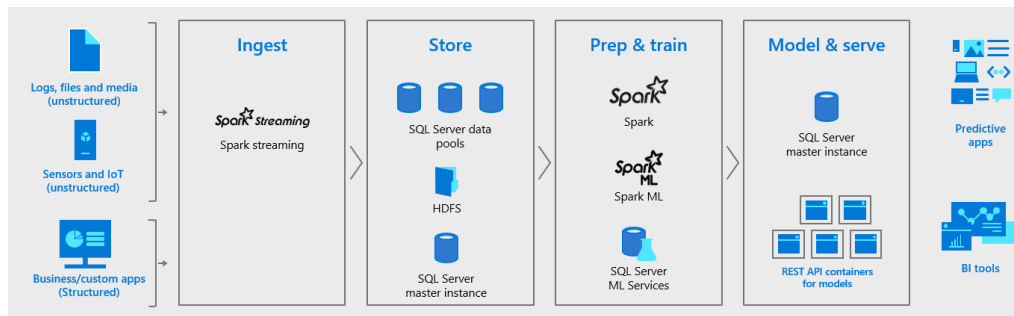
Server 2019 предоставя цялостна платформа за машинно обучение и изкуствен интелект с всички инструменти и услуги, необходими за зареждане, съхранение, подготовка и анализ на данни. Може да се използва Spark, както и вградени инструменти за изкуствен интелект в SQL Server с помощта на R, Python, Scala или Java.

Пълната платформа за изкуствен интелект и машинно обучение в Big Data Cluster е представена на фигура 4.

Бази от данни

ните и поддръжка на транзакциите.

- Тенденциите в развитието на NoSQL базите от данни показват, че в последните години те запазват своя дял, без да го увеличават. Това е индикация, че те са намерили своята сфера на приложение и вече потребителите не ги смятат за по-добра алтернатива на релационните бази от данни във всички случаи.
- Релационните бази от данни не са изгубили своето значение – напротив, все



Фигура 4. Платформа за изкуствен интелект в Big Data Cluster (Microsoft Corporation, 2018)

5. Заключение

В резултат на анализа на характеристиките и съвременните тенденции в развитието на базите от данни могат да се направят следните изводи:

- NoSQL базите от данни са подходящи за съхраняване на големи обеми данни, в т.ч. неструктурирани, полуструктурирани или данни с чести промени в структурата. Те са по-гъвкави, осигуряват по-лесна скалируемост на данните и по-добро бързодействие.
- NoSQL базите от данни не са подходящи в случаите, когато моделирането на връзките е от особено значение (с изключение на граф-ориентирани системи), а също и когато трябва да се осигури висока степен на надеждност, сигурност и непротиворечивост на да-

още има много области на приложение, в които те са незаменими. Повечето OLTP (Online transaction processing) системи използват релационни бази. Такива са, например, банковите системи, както и голяма част от бизнес приложенията.

- Нововъведенията, добавени в релационни бази от данни, комбинират техните възможности с достъпа и обработката на данни от външни източници. Така те стават подходящ инструмент и за съхранение и обработка на големи данни, както и за изкуствен интелект и машинно обучение.

Познаването на характеристиките на съвременните бази от данни е от особена важност за вземане на адекватни решения при проектирането и разработването на информационни системи.

Цитирани източници:

- A B M Moniruzzaman, S. A. H., 2013. NoSQL Database: New Era of Databases for Big data Analytics – Classification, Characteristics and Comparison. *International Journal of Database Theory and Application*, 6(4), pp. 1-14.
- Apache Software Foundation, 2016. Apache Cassandra Documentation v4.0-beta3. [Online] Available at: <https://cassandra.apache.org/doc/latest/>
- Codd, E. F., 1970. A relational model of data for large shared data banks. *Communications of the ACM*, June, 13(6), pp. 377 – 387.
- Lazarova, V., 2013. Principles of Transactional Approach in the Classical Web-based Systems and the Cloud Computing Systems – Comparative Analysis. *Economic Alternatives*, Issue 1, pp. 98-106.
- Lazarova, V., 2015. The Expansion of the E-business Information Systems in the Cloud-Problems and Solutions. *Ikonomicheski i sotsialni alternativni*, December, Issue 4, pp. 34-47.
- Microsoft Corporation, 2018. Microsoft SQL Server 2019 Big Data Clusters Technical white paper. [Online] Available at: <https://info.microsoft.com/ww-landing-SQLDB-Microsoft-SQL-Server-WhitePaper.html>
- Mihova, V., 2012. Monitoring the future state of a database. UNITE 2nd Doctoral Symposium R&D in Future Internet and Enterprise Interoperability.
- Milev, P., 2015. Opportunities for Extracting Web Content in a Structured Way. Proceedings of International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE).
- MongoDB, 2020. The MongoDB 4.4 Manual. [Online] Available at: <https://docs.mongodb.com/manual/>
- Neo4j Inc., 2020. The Neo4j Operations Manual 4.1. [Online] Available at: <https://neo4j.com/docs/pdf/neo4j-operations-manual-4.1.pdf>
- Redis Labs, 2020. Redis. [Online] Available at: <https://redis.io/documentation>
- Sanjay Ghemawat, H. G. S.-T. L., 2003. The Google File System. Lake George, *ACM SIGOPS Operating Systems Review*.
- Solid IT GMBH, 2020. DB-Engines Ranking. [Online] Available at: <https://db-engines.com/en/ranking>
- Stack Overflow, 2020. Stack Overflow Developer Survey. [Online] Available at: <https://insights.stackoverflow.com/survey/2020>
- Tiwari, S., 2011. Professional NoSQL. Indianapolis: John Wiley & Sons.
- Tsaneva, M., 2019. A Practical Approach For Integrating Heterogeneous Systems. *Business management*, Issue 2, p. 11.
- Vaish, G., 2013. Getting Started with NoSQL. Birmingham: Packt Publishing.