EURO²

EuroHPC
Joint Undertaking

2023

Ръководство за проектиране на бизнес референтни архитектури насочени към използване на NoSQL бази от данни

УНСС | София

Въведение

В настоящия момент, когато промените и дигиталната трансформация са необходимост за бизнеса, за да остане конкурентноспособен, компании, които не успяват да се справят с промяната, рискуват да останат на заден план или да прекратят своето съществуване. Технологиите, развиващи се с по-бързи темпове от обикновени телефони през сензори, събиращи данни и използващи се почти навсякъде в ежедневието ни до компютри, роботика, Изкуствен интелект, облачни технологии, Големи данни, Интернет на нещата и др. се считат за основата на дигиталната трансформация в различни сфери.

Дигитализацията и дигиталната трансформация обхващат различни видове процеси в живота ни, като прогресират с изключително бързи темпове. Адаптирането на нови технологии помага за повишаването на ефикасността и продуктивността при дигитализацията на ръчните процеси. Има изключително много ползи от дигиталната трансформация, но един от най-важните фактори е, че помага на бизнеса да стане по-ефективен и организиран. Този процес има потенциала да направи организациите по-гъвкави, ефективни и клиентско-ориентирани.

В последното десетилетие управлението на експоненциално нарастващите количества данни става все по-предизвикателна задача. С развитието на технологиите, дигитализацията на процесите и дигиталната трансформация на бизнесите води до генерирането на огромни количества данни, които трябва да бъдат съхранявани, управлявани и анализирани, чрез подходящи методи. Развитието на технологиите и генерирането на данните представляват първоначалната и най-лесна стъпка. Предизвикателствата започват със съхраняването, обработката, анализирането и извличането на резултати от събраните данни.

В миналото, когато е поставено начало на развитието на информационните технологии, основният тип събирани данни са били структурираните. С дигитализацията на все повече и повече бизнеси и дигитално трансформиране, започна и генерирането на различни типове данни – полуструктуриране и неструктурирани. За разлика от структурираните данни, другите два типа представляват изключително голямо предизвикателство за съхранение и обработка. Обикновените релационни бази от данни, използвани за съхранението на структурирани данни не са подходящи за полуструктурираните и неструктурираните, което води до необходимостта от създаването и развитието на нови начини за съхранение на различните видове данни.

Едно от ключовите изисквания за съхранението на Големи данни е, че трябва да се справя с огромни количества данни, като възможностите за съхранение продължават да се увеличават без да бъде прекъсван работния процес.

През последните години, финансовият сектор се развива и се дигитализира все повече, което води до нуждата от промени. За да могат финансовите институции да отговорят на изискванията, нуждите и очакванията на своите клиенти е от изключителна важност да се правят промени и да се адаптират бързо към развиващите се технологии. В допълнение на това, този тип институции трябва да управляват изключително големи обеми от данни докато правят промени на техните системи, без да причинят загубата им или проблеми за клиенти си. Финансовите услуги предлагани от банки, кредитни дружества, счетоводни компании, застрахователни такива, инвестиционни фондове, стокови борси и други, се нуждаят от бази данни, които да могат да се адаптират към нуждите за автоматизиране, на които нерелационните бази могат да отговорят. NoSQL базите могат да се справят със съхранението и обработката на големи количества от данни, със своята скалируемост и предоставянето на по-добра производителност при обработката на данни.

Бизнес референтни архитектури насочени към използване на NoSQL бази от данни

Референтната архитектура е документ или съвкупност от документи, които предоставят препоръчана структура и интеграции на ИТ продукти, които да формират решение. РА обединява в себе си най-добрите практики в индустрията, като обикновено предлага оптималния метод за специфични технологии. Референтната архитектура предлага най-добрите ИТ практики в лесно разбираем формат, който направлява прилагането и използването на сложни технологични решения.

Референтните архитектури добавят стойност към компаниите по следните начини. Премахват възможните объркванията чрез стандартизация, правят разрешаването на проблеми по-лесно чрез прилагането на точни и ясни насоки. Снабдяват с източници за дизайна на ИТ архитектури, екипи и системи. Референтните архитектури спестяват време, усилия и пари, използвайки вече наличните ресурси, както и оптимизират разрешаването на проблеми чрез прилагането на стандартни добри практики и поддържат съвместимост и повторно използване на компонентите.

Използването на добри практики при създаването на референтни архитектури подобрява ефикасността, отговаря на нормативните изисквания и намалява шанса за грешки. Референтната архитектура помага при вземането на решения за избор на най-добрия модел и начин за създаване на софтуерна архитектура, за да се срещнат бизнес целите.

Подход за изграждане на референтни архитектури

Създаването на референтна архитектура е изключително предизвикателна задача при липсата на предварително зададени стъпки или процес за създаването ѝ. Има пет основни стъпки, които трябва да се изпълнят: идентифициране на целта, формулиране на принципи, залагане на технически правила и стандарти, построяване на правила и стандарти, и прилагане на контекст.

- При **идентифициране на целта** трябва да се определи домейнът и обхватът на референтността, кои са заинтересованите страни, как ще се използва архитектурата, какви са ограниченията, предположенията и средата, с които е асоциирано.
- **Формулирането на принципи** се случва след като успешно се идентифицира целта, трябва да се включат компонентите необходими за архитектурата. Елементите трябва да са съгласувани с основополагащите изявления на култура и ценностите на компанията.
- **Залагане на технически правила и стандарти** се случва след формулирането на принципите. Следващата стъпка е да се решат рамките и моделите, които ще бъде необходимо да се следват в компанията. В тази част се залагат всички необходими правила, за да се постигнат принципите формулирани по-рано.
- Трябва да се заложат **правила и стандарти** в различните отдели на компанията. Трябва да се направи списък с всички възможни предизвикателства и да се подберат правилните стандарти за всеки възможен сценарий.
- Към правилата и стандартите се включва и **контекста** на всяка ситуация. Повечето компании включват логически процес, което прави лесно и възможно включването на добри практики.

Когато се използват добрите практики, за да се създаде корпоративна референтна архитектура, следва да се подсигури това, че да може компанията и клиентите да се възползват от нейните предимства. Може да се подобри ефективността, да се отговори

на нормативните изисквания и да се намали възможността от грешки. Част от добрите практики, които могат да бъдат приложени са:

- конкурентно предимство;
- сравнителни показатели;
- стандарти за съответствие;
- повторно използване на компоненти и управление

Проектиране на бизнес референтна архитектура за финансови услуги с NoSQL бази от данни

Всяка една финансова институция занимаваща се с различни видове финансови услуги генерира, събира, обработва и анализира данни от различни типове – структурирани, полуструктурирани и неструктурирани в големи обеми, което от своя страна води до различни нужди на институциите. В зависимост от типа финансова институция зависи вида данни, който бива генериран, което пък от своя страна води до различни изисквания за място за съхранение, методи за обработка и инструменти за анализи.

С внедряването на нерелационни бази от данни във финансовите институции и адаптирайки техните исторически системи към новите бази, води до развиването и нуждата на услугите в сферата на финансите да запазят своето конкурентно предимство, както и да излязат по-напред от техните конкуренти в сферата.

Целта на създаването на референтна архитектура, независимо дали за използване само в една организация или за универсалното и адаптиране в множество финансови институции, е да подсигури последователността и приложимостта на използването на дадени технологии в конкретна организация.

Продължавайки примерът с персоналните финанси и по-конкретно банкирането или услугите пряко свързани с клиента, могат да бъдат разгледани следните елементи, които са базирани върху различните данни събирани за клиентите:

- ❖ Всяка финансова институция има пакетни обработки, които не изискват изпълнение в реално време или почти реално време, но след надхвърлянето на определен обем, обработката им продължава в работното време на институцията, когато трябва дейностите да се случват бързо, а в случая това забавя основната работа и обслужването на клиенти.

Това може да доведе до момент, в който тези обработки на данни няма да са приключили преди да е необходимо да започнат отново. За много институции обработката на памет е най-ефективния начин за постигане на необходимите нива на изпълнение и скалируемост. В този тип обработка могат да се съхраняват големи обеми от данни на памет и да се използва масивната паралелна обработка за предоставянето на до 1000 пъти по-бърза производителност за приложения създадени върху диск-базирани бази. Когато данните се съхранявани и обработвани в паметта движението на данни в мрежата намалява или е елиминирано напълно. Традиционните релационни бази от данни не са създадени за обработки на огромни количества данни в реално време, като по тази причина трябва да се избере кой обем от работа да бъде оптимизиран. Могат да се справят или само с оперативната натовареност, което се отнася до всекидневните бизнес транзакции или до аналитичната натовареност, която се отнася до бизнес интелигентните системи и анализи. Невъзможна е реализацията на 2-та вида натоварване едновременно, тъй като релационните бази от данни не могат да се справят със смесването им. SQL базите от данни са създадени да бъдат специализирани на цената на гъвкавостта. От друга страна при нерелационните бази от данни, може да се подобри представянето и изпълнението на операциите в базата като четене и запис, извличане на данни и анализ на такива. Това се случва, защото обработката на памет позволява по-бърз достъп до данните, намалявайки латентността свързана с традиционно диск-базирани места за съхранение. Това позволява много по-бърз достъп на данни и време за обработка, правейки го полезно във финансовите услуги, които са чувствителни от към време за обработка, например като търговия на ценни книжа, управление на риска и засичане на опити за измама. Обработката на памет води до бърза обработка на големи количества данни в реално време, което може да подобри вземането на решения, повиши ефикасността и да намали разходите при финансовите услуги.

- Финансовите институции поддържат клиентски досиета съставени от класифицирани по определени признаци блобове (полуструктурирани и неструктурирани данни - документите свързани с йерархията клиент-продукти). Характерна особеност обаче е, че блобовете имат много

контекст защото възникват и се съхраняват във връзка с точно определен клиент и банкова/финансова операция регистрирана в компютърна система . Върху тях не се правят обработки, които изискват анализ на тяхното съдържание, а само търсене с цел верификация на подписи и визуализация/печат по време на обслужване при поискване

На база на изискванията, които могат да произлязат от финансовите услуги, заедно с направеното теоретично проучване на литература, както и на практическото такова на вече съществуващи решения на финансови услуги върху различни нерелационни бази от данни, бе създаден модел на универсална компонентна референтна архитектура за финансови услуги.

➢ Концептуален модел

Концептуалният модел на референтната архитектура за финансови услуги представен на Фигура 1 представлява генералната структура и елементи, от които се състои тя, както и необходимите данни и системи, за да се постигнат бизнес изискванията, което означава поддръжката на фирмени процеси, записване на бизнес събития и проследяването на представянето. Дефиниран е подход за изграждането на референтна архитектура, който се състои от пет основни стъпки. Създаденият концептуален модел на бизнес референтната архитектура за финансови услуги е в резултат от първите две стъпки – идентифицирането на целта и формулирането на целта. Целта на бизнес референтната архитектура за финансови услуги е да адаптира модерните начини на съхранение на Големи данни за различни типове – полуструктурирани и неструктурирани данни към финансовите институции, които продължават своето бързо развитие и дигитална трансформация.

Референтната архитектура се състои от 3 слоя: Система за съхранение на данни, Интеграционен слой и Сървър за съхранение на данни, като те са свързани със системи за обработка на финансови услуги.

❖ Система за съхранение на данни

Системата за съхранение на данни е системата, която съхранява данните след като са постъпили от системите за обработка на финансови услуги.

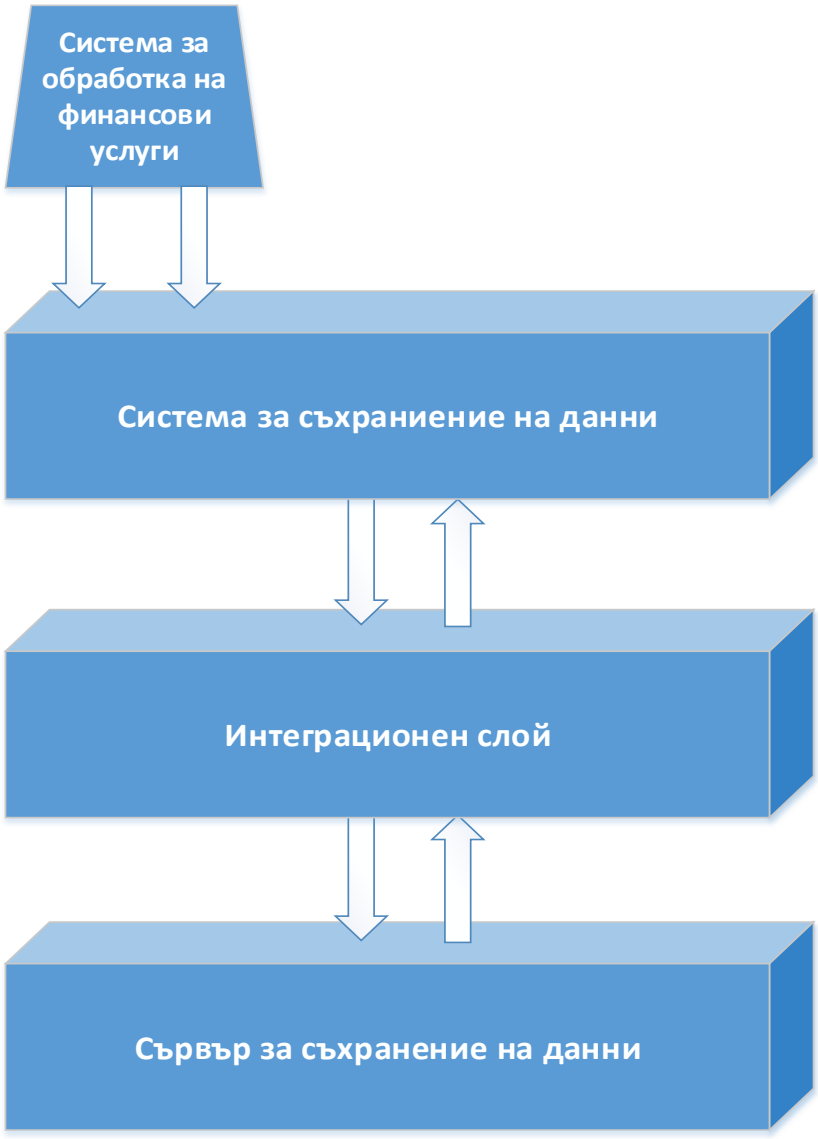
❖ Интеграционен слой

Интеграционният слой в референтната архитектура се отнася до компонентите отговорни за интегрирането на системите за съхранение на данни

и сървърите за съхранението им. Този слой предоставя централизирана платформа за управлението, движението на данни и комуникациите между различните системи. Интеграционният слой се състои от конектори за данни между системата за съхранение и сървъра.

- **Сървър за съхранение на данни**

Сървърът за съхранение на данни е мястото, на което се съхраняват данните, които достигат, чрез интеграционния слой и конекторите за данни от системите за съхранение на данни, които влизат към системата за съхранение на данни от системите за обработка на финансови услуги.



Фигура 1 Концептуален модел

➢ Логически модел

В логическият модел са заложени техническите правила и стандарти за създаването на бизнес референтната архитектура, като при него се залагат правилата, за да могат да се постигнат заложените принципи. През системата за обработка на финансови услуги влизат данните от финансовите институции, които в последствие са съхранявани, обработвани и анализирани в останалите слоеве на референтната архитектура. На Фигура 2 е представен логическият модел на бизнес референтната архитектура за финансови услуги, като по-долу е разгледан подробно всеки един от нейните слоеве.

❖ Система за съхранение на данни

В първият слой на референтната архитектура, т.нар. система за съхранение на данни могат да бъдат разположени 1 или повече четирите типа нерелационни бази от данни – документно-ориентирани, ключ-стойност, широко-колонни и граф бази. Типовете NoSQL бази от данни бяха разгледани на теоретично ниво, както какви са техните възможност в Глава I, а в настоящата глава бяха разгледани на практическо ниво – какви приложения свързани с финансовите услуги са създадени върху конкретни нерелационни бази от данни, което да спомогне изборът на конкретна база.

Изборът на такава се реализира базирано върху следните елементи, които трябва да бъдат разгледани внимателно преди пристъпването към конкретна NoSQL база:

- Налична система/системи на финансовата институция, която ще трябва да бъде мигрирана към новата база
- Тип финансови услуги, с които се занимава финансовата институция
- Типове данни, които генерира финансовата институция на база финансовите услуги, които обслужва – структурирани, полуструктурирани и неструктурирани данни

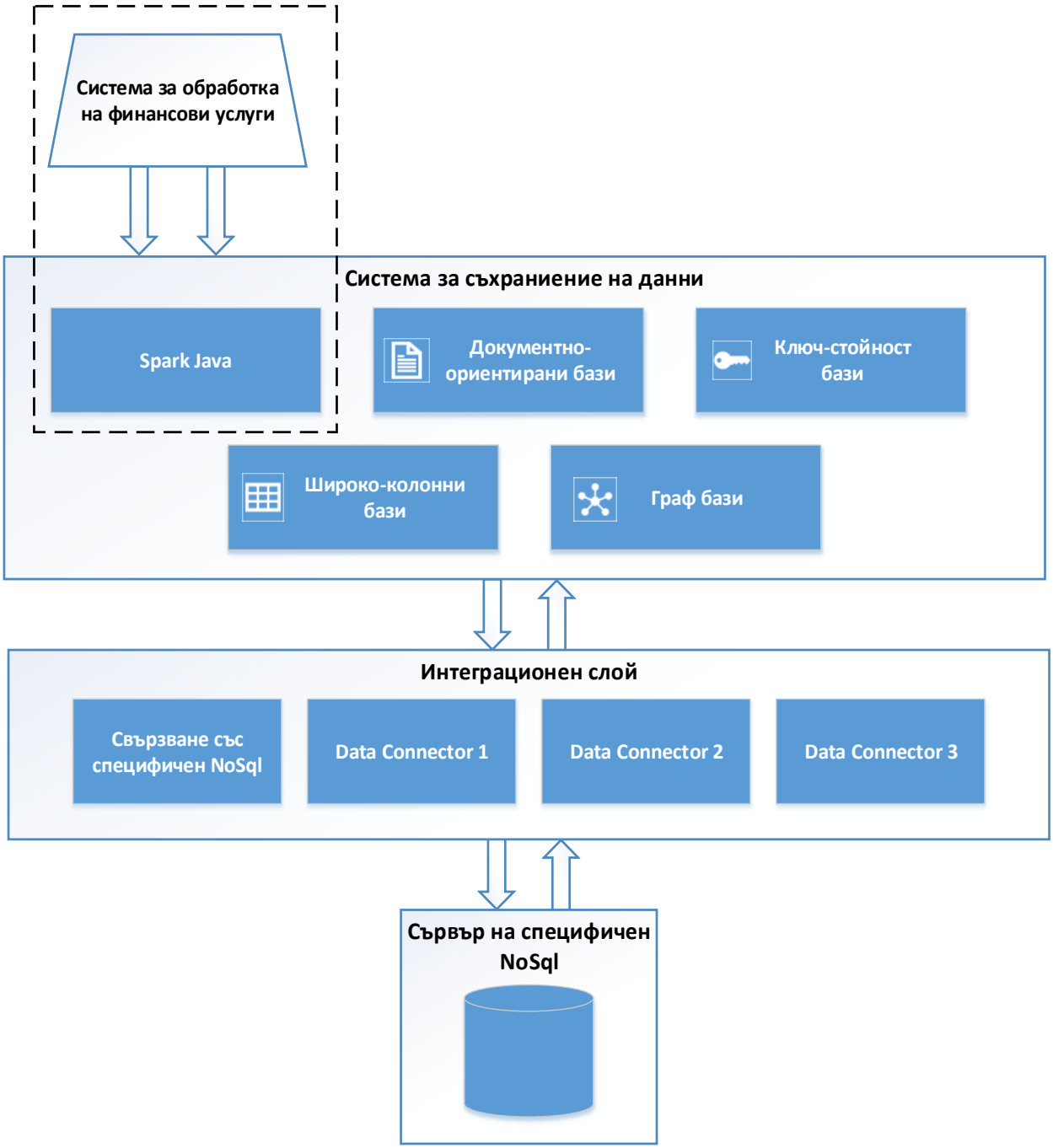
❖ Интеграционен слой

На база на нерелационната база, която е избрана в системата за съхранение на данни на база на разгледаните елементи, се избира начинът за свързване със

специфичен NoSQL, който се позиционира в интеграционния слой на референтната архитектура.

- **Сървър на специфична NoSQL база от данни**

В зависимост от избраната нерелационна база от данни в системата за съхранение на данни, последвалия избор на начин за свързване на конкретна NoSQL база от данни, се достига до сървър за специфичен NoSQL, в който се съхраняват данните на финансовата институция.



Фигура 2 Логически модел

➢ Физически модел

Създаденият физически модел на Фигура 3 е реализиран с конкретни примери на нерелационни бази от данни и конектори за тях към Hadoop. Всеки един от тези елементи може да бъде заменен според нуждите на финансовата институция, която иска да адаптира тази референтна архитектура с NoSQL бази от данни за собствените си нужди.

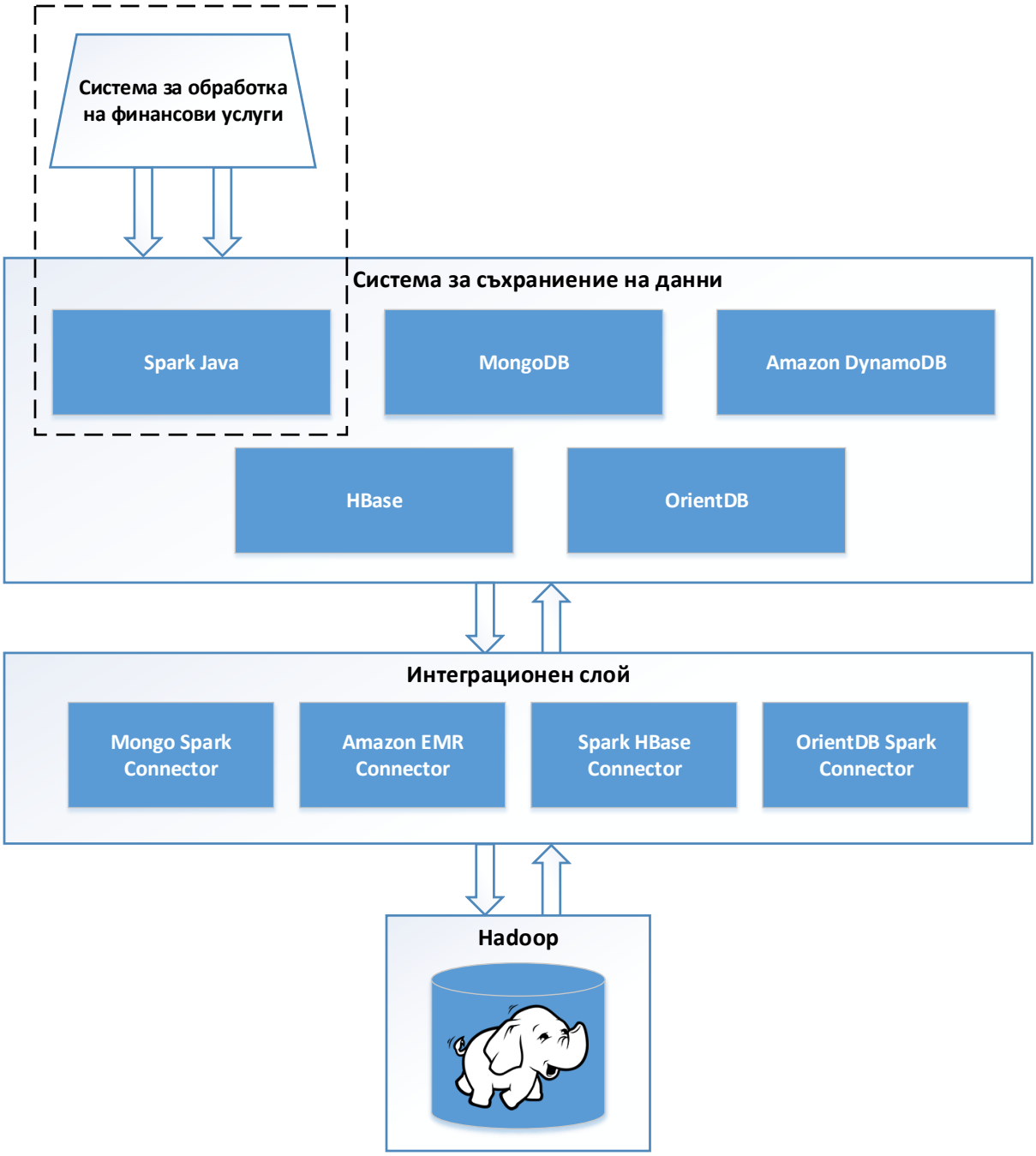
В първия слой, който е системата за съхранение на данни, могат да се разположат четирите типа нерелационни бази от данни – документно-ориентирани, ключ-стойност, широко-колонни и граф бази, като в този слой се разполага конкретната база, която е избрана чрез метода за оценка и избор на NoSQL бази от данни за финансови услуги. Във физическия модел са разположени бази от всяка една от видовете:

- Документно-ориентирана база: **MongoDB**
- Ключ-стойност база: **Amazon DynamoDB**
- Широко-колонна база: **HBase**
- Граф база: **Orient DB**

Във втория слой, т.нар. интеграционен слой се намират свързващите елементи с различните NoSQL бази от данни, напр. NiFi, Spark Connector и други видове Data Connectors подходящи за другите типове нерелационни бази от данни. Конкретните конектори на данни от системата за съхранение на данни към сървъра на специфичната NoSQL база от данни са следните:

- За документно-ориентираната база **MongoDB** може да бъде използван **Mongo Spark Connector**
- За ключ-стойност базата **Amazon DynamoDB** може да бъде използван **Amazon EMR Connector**
- За широко-колонна база **HBase** може да бъде използван **Spark HBase Connector**
- За граф базата **Orient DB** може да бъде използван **Orient DB Spark Connector**

Третият слой е сървър на специфичната NoSQL база, като за реализацията на физическия модел на референтната архитектура за финансови услуги е избран Hadoop.



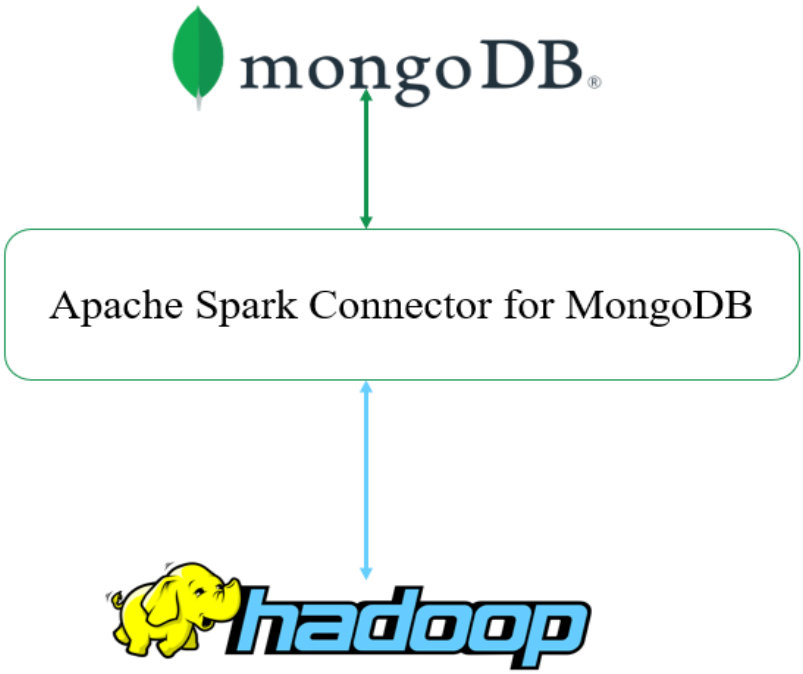
Фигура 3 Физически модел

Подход за използването на проектираната референтна архитектура водеща към създаването на ICT архитектура

ICT Архитектурата (Информационни и комуникационни технологии) осигурява концептуален модел, специфицирайки на основно ниво елементите на ICT архитектурата (приложение, бази от данни, технологични ICT елементи), както и връзките помежду им. На базата на референтната архитектура за финансови услуги с NoSQL бази от данни, която бе създадена в предишната точка и е демонстрирана на Фигура 3, ще бъде

създадена конкретна ICT архитектура, която ще се реализира и тества в следващите точки.

Разработеният физически модел за референтна архитектура съдържа 3 слоя – система за съхранение на данни, интеграционен слой и слой за специфичен NoSQL, като в предишната точка разгледахме подробно различните типова за нерелационни бази от данни, с различни видове конектори за данни, които могат да бъда свързани със слоя за специфичен NoSQL. При създаването на ICT архитектурата за система за съхранение на данни(първият слой), ще бъде използвана документно-ориентираната нерелационна база от данни MongoDB, тъй като тя ба избрана чрез създадения метод в предишната точка и отговаря на изискванията и има вече създадена финансова услуга, която използва типа данни, с който ще бъде тествана архитектурата. Вторият интеграционен слой ще използва Apache Spark Connector за MongoDB. Поради факта, че поддръжката на директния конектор на MongoDB към Hadoop е спряна, ще използваме този през Apache Spark. Третият слой остава за системата големи данни Hadoop, която ще използваме за съхранение – Фигура 4.



Фигура 4 Конкретна ICT архитектура

Приложение на метод за проектиране на референтна архитектура за финансови услуги

С все по-голямата автоматизация и дигитализация на процесите, и навлизането на по-големи обеми от данни от различни типове – най-вече от полуструктурирани и неструктурирани, бизнесите се изправят пред огромното предизвикателство за събирането, съхранението и анализа на неструктурирани данни. Финансовите институции и услугите, които предлагат са от бързо развиващите се бизнеси, като по тази причина те трябва да се адаптират бързо и навременно към разгръщащите се технологии и към увеличаващите се изисквания на клиентите, както и на постоянната конкуренция.

На база направеният детайлен анализ на научна литература свързана с нерелационните бази от данни и използването им за реализиране на системи за финансови услуги, както и на база разработените бизнес решения, бе разработена бизнес референтна архитектура, състояща се от система за съхранение на данни, интеграционен слой и сървър на специфичен NoSQL, която бе развита и достигна до реализирането на физически модел. В този модел могат да бъдат разположени всеки един от четирите типа нерелационни бази от данни, както в комбинация заедно, така и по отделно.

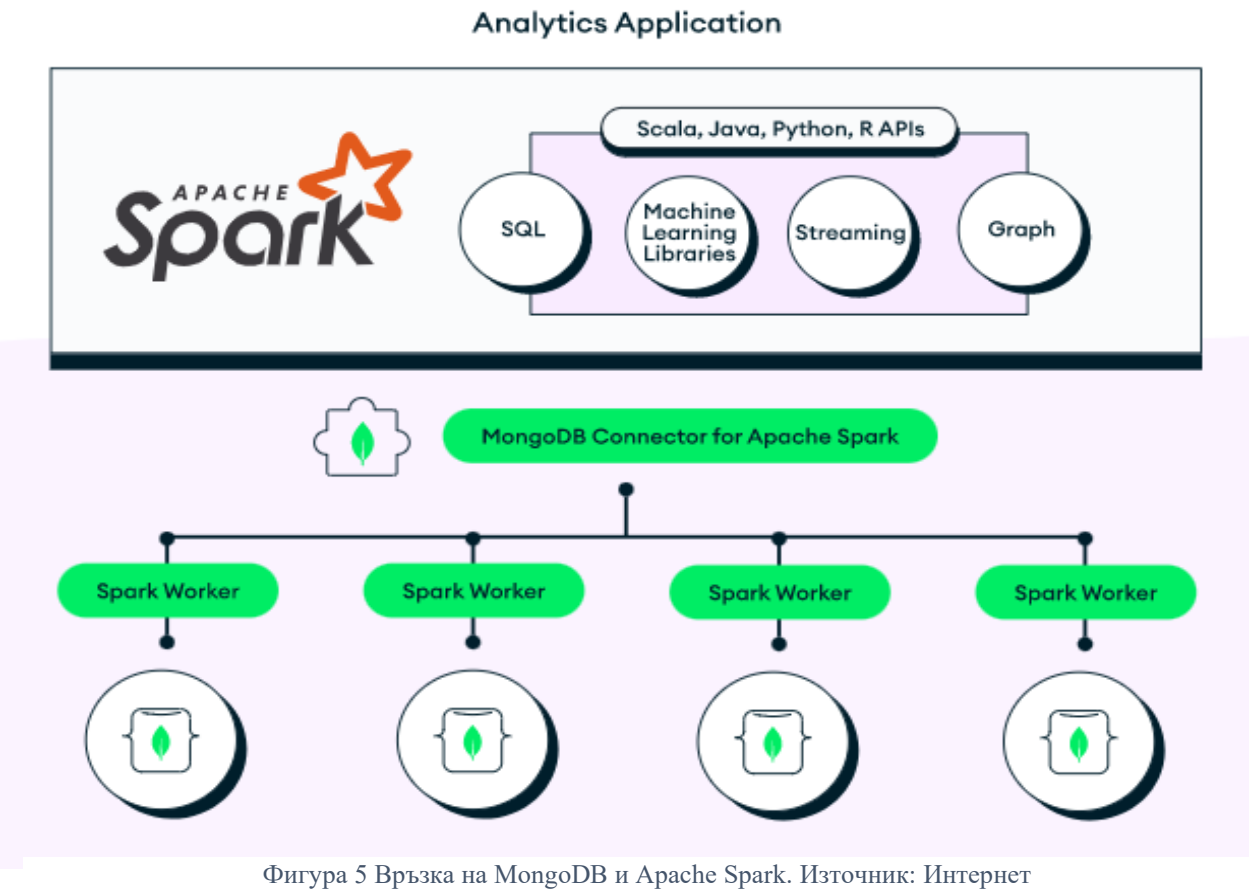
ICT архитектурата, която е тествана практически се състои от система за съхранение на данни – MongoDB, интеграционен слой – Apache Spark Connector за MongoDB и сървър на специфичен NoSQL, като в случая е използвана разпределената файлова система на Hadoop.

Hadoop е технология създадена за съхранение на големи обеми от данни разпределени в различни клъстъри. MongoDB от своя страна е изключително мощна документно-ориентирана NoSQL база от данни. Комбинирани заедно, Hadoop и MongoDB, могат да създадат завършено приложение за работа и анализ с Големи данни. Hadoop разполага с данните, които се съхраняват в MongoDB, като се обединява с данните, които разполага и по този начин генерира анализи, както и модели за машинно обучение. Резултатите се зареждат обратно в MongoDB, като по този начин се използват за създаването на по-добри клиентски оферти, за по-добра идентификация и засичане на опити за измама, по-добро предсказване на промени в борсите и други.

MongoDB конекторът за Spark осигурява интегрирането между MongoDB и Apache Spark, а от там след това се извличат данни от Hadoop. Чрез конекторът,

потребителят има достъп до всички Apache Spark библиотеки, които могат да бъдат използвани с данните, които се съхраняват в NoSQL базите от данни.

На Фигура 5 е демонстрирано как се извършва връзката чрез MongoDB конекторът за Apache Spark, както и всички библиотеки, до които има достъп.



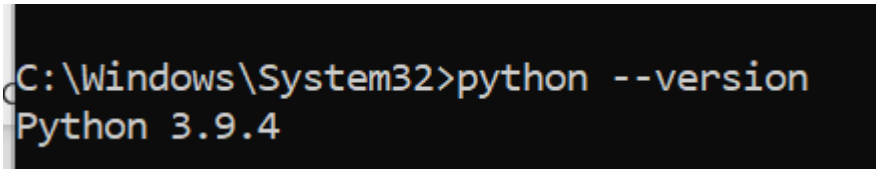
Фигура 5 Връзка на MongoDB и Apache Spark. Източник: Интернет

Конекторът за данни на MongoDB през Spark позволява интеграцията на документно-ориентираната база MongoDB и Apache Spark, което позволява на потребителите да реализират сложни анализи с големи сетове от данни. В допълнение на това, че конекторът позволява операции за четене, също дава възможност за запис на данни обратно към MongoDB през Spark конектора. Това дава изключителна възможност за записване на резултати от извършените обработки в Spark да бъдат записани обратно в MongoDB и след това да бъдат използвани за допълнителен анализ.

За реализацията на физическия модел на бизнес референтната архитектура са необходими следните компоненти:

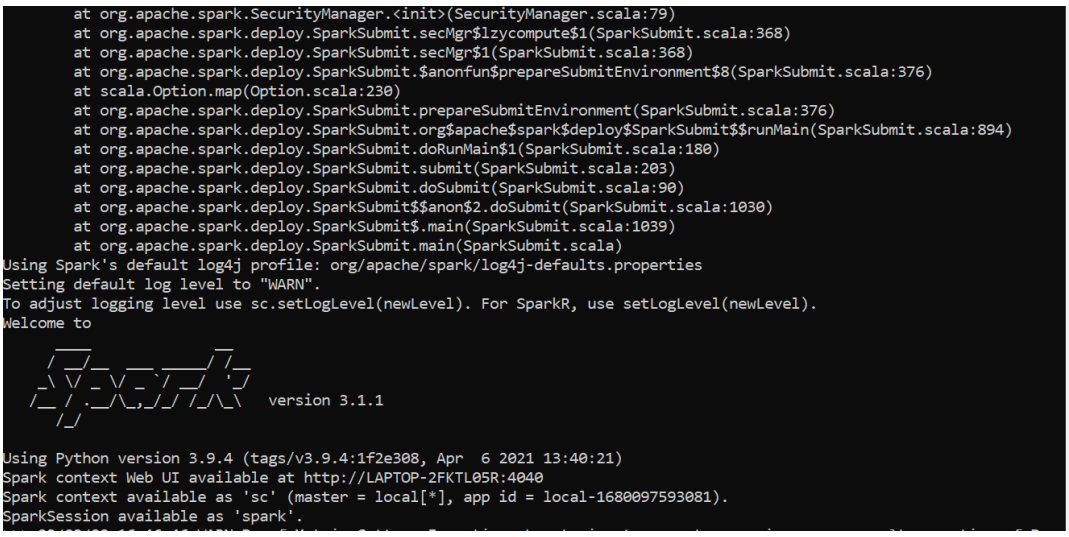
- Hadoop
- Apache Spark конектор за MongoDB
- Python

Към момента ICT архитектурата е реализирана на локално ниво - връзка на MongoDB чрез Apache Spark конекторът към Hadoop. Започва се с локална инсталация на Python, тъй като ще бъде използван PySpark. За да се инсталира и конфигурира Python се свалят необходимите файлове, които се стартират на локалната машина, чрез стандартен инсталационен софтуер, с който разполага. След завършването и, се проверява наличната версия на Python на машината, за да е сигурно, че процесът е завършил правилно и работата по следващите стъпки може да бъде продължена. В Command Prompt се изписва следната команда python –version, като ако инсталационния процес е преминал успешно трябва да се появи следния резултат – Фигура 6.



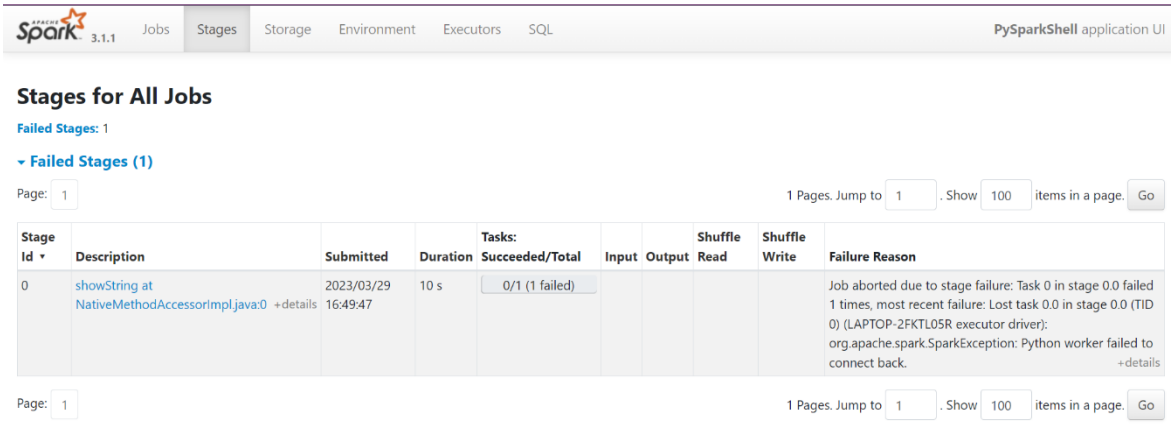
Фигура 6 Резултат при правилна инсталация на Python

Инсталацията на Spark на локално ниво се извършва след избор на конфигурациите и сваляне на необходимите файлове на машината. При правилна инсталация на Spark и въвеждане на следната команда в Command Prompt: C:\Spark\spark-3.1.1-bin-hadoop2.7\bin\spark-shell, Apache Spark се стартира и се визуализира екрана по-долу – Фигура 7.



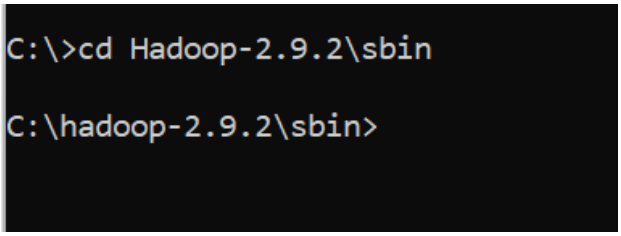
Фигура 7 Резултат при правилна инсталация на Apache Spark

При правилната инсталация и стартиране на Apache Spark, освен съобщението в Command prompt, ни се предоставя потребителски интерфейс през браузъра, който може да бъде използван за наблюдението на различните обработки, които се извършват – Фигура 8, на следния адрес: http://localhost:4040/.



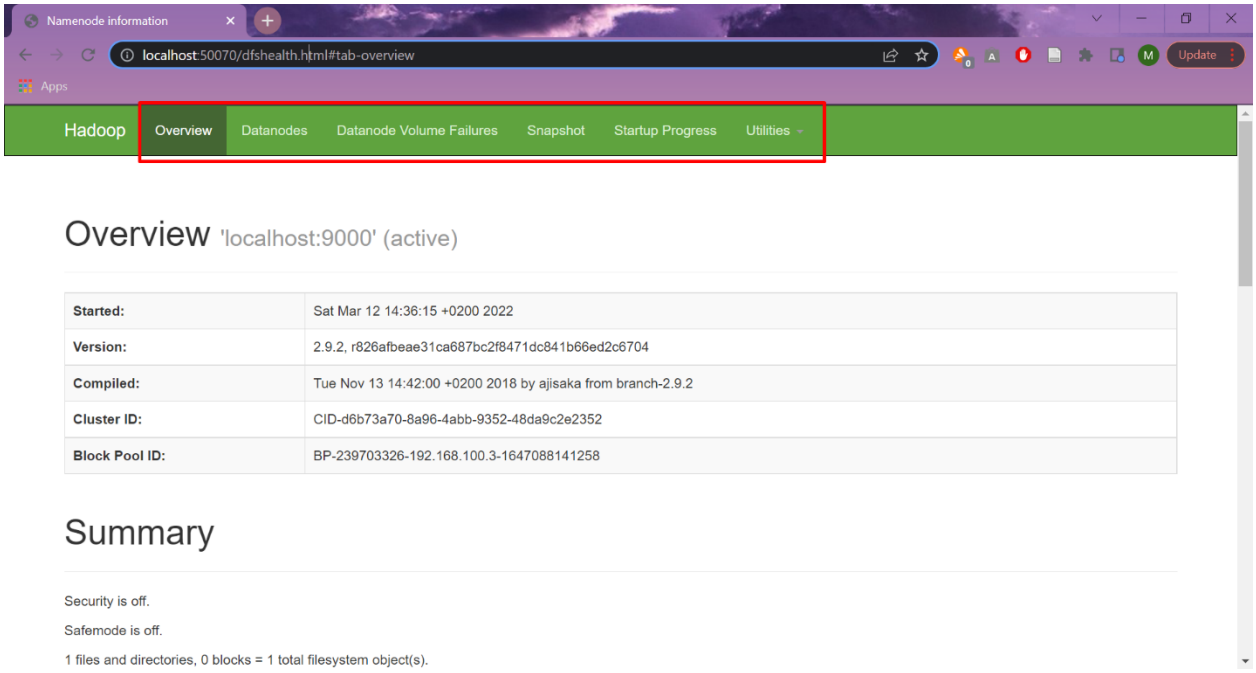
Фигура 8 Интерфейс на локално ниво в браузъра за наблюдение на изпълнение на задачи на Apache Spark

Следващата стъпка за реализацията на физическият модел на бизнес референтната архитектура, която трябва да се изпълни е инсталацията на Hadoop. Той се сваля, правят се необходимите конфигурации на машината, както се добавят и допълнителни конфигурации към някои от файловете на Hadoop, които могат да бъдат открити в приложенията на настоящия дисертационен труд. За проверка на това дали инсталацията е била успешна в Command prompt на машината се изписва следната команда: cd Hadoop-2.9.2\sbin , като трябва да се появи следното нещо на Фигура 9.



Фигура 9 Успешна инсталация на Hadoop

В браузърът на следния адрес: http://localhost:50070/ се отваря потребителският интерфейс, чрез който при правилно стартиране на Hadoop може да се проследят клъстърите с данни и други.



Фигура 10 Интерфейс на Hadoop

Последната стъпка за реализирането на физическата архитектура е инсталацията на Apache Spark конектора за MongoDB, което се реализира чрез Pyspark и следните команди --packages, като след това от предоставените възможности се избира mongo-spark-connector, който се инсталира на машината.

Стартирането на връзките се случва, чрез следния код:

```
./bin/pyspark --conf "spark.mongodb.read.connection.uri=mongodb://localhost:27017/FinancialData.Stocks?readPreference=primaryPreferred"
        --conf "spark.mongodb.write.connection.uri=mongodb://localhost:27017/FinancialData.Stocks"
        --packages org.mongodb.spark:mongo-spark-connector_2.12:10.1.1
```

- **Приложение на типични неструктурирани данни при финансови услуги**

Всяка една финансова услуга работи с различни типове данни – структурирани, полуструктурирани и неструктурирани, събирани от различни източници в различни

формати. Създадената референтна архитектура за финансови услуги с документно-ориентираната база MongoDB свързана чрез конектор на Apache Spark с Hadoop, работи с данни в документен вариант.

Този тип неструктурирани данни са типични за много услуги в основно банкиране, е-плащания, акции, облигации, персонализиране, е-портфейл, застраховки, клиринг на клиенти, управление на клиенти и други. По тази причина създадената референтна архитектура е тествана с данни, като договори, фишове, имейли, репорти, данни за стокови акции и облигации и други. Част от тези данни са макетни, генерирани допълнително, поради факта, че данните, с които работят финансовите услуги са изключително чувствителни и не могат да се открият такива със свободен достъп.

- **Тестване на референтна архитектура с финансови данни**
  - **Източник на финансови данни – акции**

Данните за и свързани с клиенти, които са генерирани, използвани и съхранявани във финансовите институции са изключително чувствителни и не бива да бъдат разпространявани свободно, като също трябва да бъдат съхранявани според изискванията за защита на личните данни. Финансови данни могат да бъдат открити много трудно свободно в интернет пространството, като за целите на тестването са използвани свободно достъпни данни за акции от Forbes, Nasdaq, Nyse и SP500 в json формат.

Данните Forbes, Nasdaq, Nyse и SP500 съдържат следните полета :

Полета	Стойности
currency	Съответната валута
symbol	Символ
exchangeName	Обменно име
instrumentType	Тип инструмент
firstTradeDate	Първа дата на продаване
regularMarketTime	Регулярно време на пазара
gmtoffset": -18000	
timezone	Времева зона
exchangeTimezoneName	Времева зона на обмен
regularMarketPrice	Редовна цена на пазара
chartPreviousClose	
priceHint	Подсказка за цената

currentTradingPeriod	Настоящ период за търгуване
pre	
timezone	Часова зона на предварителната продажба
start	Стартова дата
end	Крайна дата
gmtoffset	Разлика от времето по Гринуич
regular	
timezone": "EST",	Часова зона на регулярна продажба
"start": 1670855400,	Стартова дата
"end": 1670878800,	Крайна дата
gmtoffset	Разлика от времето по Гринуич
post	
timezone	Часова зона след продажба
start	
end	Крайна дата
gmtoffset	Разлика от времето по Гринуич
dataGranularity	Детайлност на данните
range	Диапазон
validRanges	Валидни времеви диапазони

Таблица 1 Полета с данни от Forbes, Nasdaq, Nyse и SP500

Данните за акции от различните източници - Forbes, Nasdaq, Nyse и SP500 са над 10ГБ, които ще бъдат заредени в архитектурата, която се реализира в настоящата глава.

- **Зареждане на финансови данни**

Създадената архитектура може да обменя данни от MongoDB през MongoDB конектора към Apache Spark и обратно. Зареждането на данни от Hadoop към Mongo се реализира, чрез PySpark.

За целта, първо е необходимо да се добавят необходимите библиотеки и да се създаде сесия на Spark със следния код:

```
spark = SparkSession.builder.FinancialTest("HadoopToMongoDB")
.config("spark.mongodb.output.uri","mongodb://localhost:27017/FinancialData.Stocks ")
.getOrCreate()
```

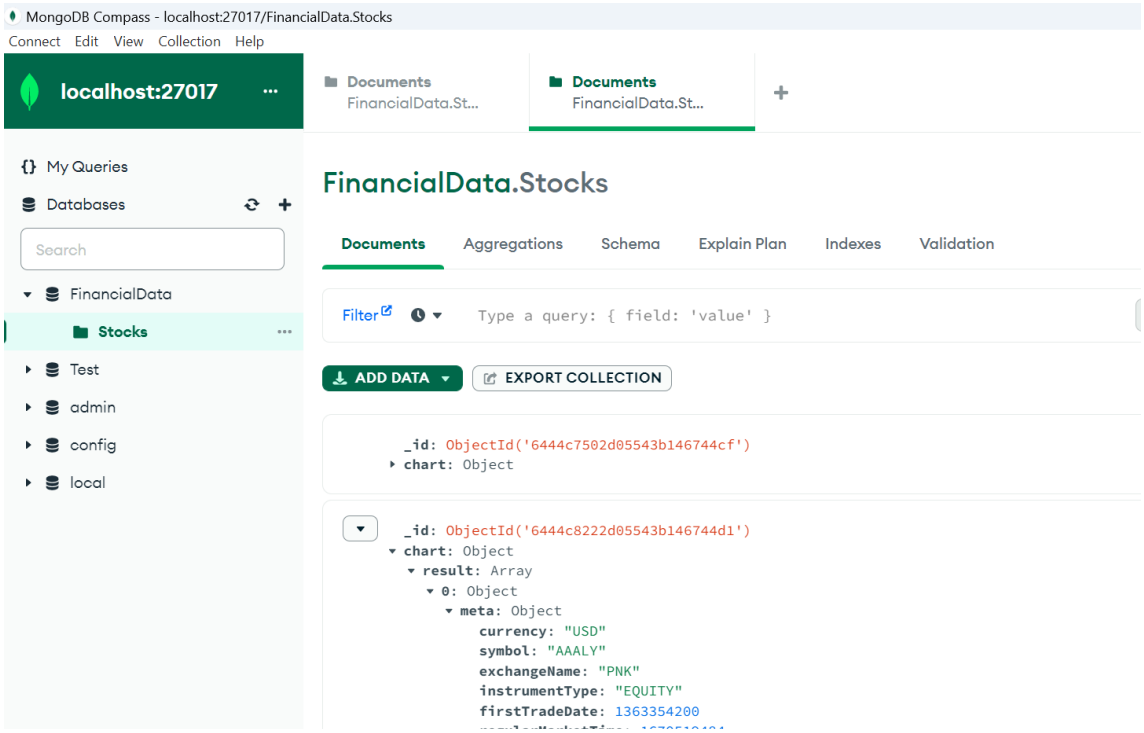
С кода по-долу зареждаме данните от Hadoop в PySpark Frame:

```
df = spark.read.format("com.mongodb.spark.sql.DefaultSource").option("uri", "mongodb://http://localhost:50070/FinancialData.Stocks.load()
```

Като накрая записваме данните от Hadoop в MongoDB, чрез следния код:

```
df.write.format("com.mongodb.spark.sql.DefaultSource").mode("append").option("uri", "mongodb://localhost:27017/FinancialData.Stocks") .save()
```

На Фигура 11 са визуализирани данните заредени през Apache Spark конектора на MongoDB от Hadoop.



Фигура 11 Заредени данни от Hadoop към MongoDB

```
client = MongoClient()

db = client['Financial_data']

collection = db['stocks']



spark = SparkSession.builder

    .appName("Financial Stock")

    .getOrCreate()
```

```
df = spark.read.format("com.mongodb.spark.sql.DefaultSource")
    .option("database", "Financial_data")
    .option("collection", "stocks")
    .load()



df = df.filter((col("symbol") == "ACGL") & (col("firstTradeDate") >= "2017-05-02") & (col("firstTradeDate") <= "2017-10-10"))



result = df.groupBy(year("date").alias("year"), month("date").alias("month"))
    .agg({"close": "avg"})  .orderBy("year", "month")


result.write.format("com.mongodb.spark.sql.DefaultSource").option("database", "Financial_data") .option("collection", "stock_analysis") .mode("overwrite") .save()


client.close()
spark.stop()
```

Създадената програма на PySpark използва данните от акциите, които бяха заредени в началото на експеримента с референтната архитектура от Hadoop към MongoDB. Първата стъпка е, че се прави връзка с MongoDB, а след това се отваря сесията със Spark. Данните се филтрират по наименованието на акциите, а в последствие се търсят само онези, чиято първа дата на продажба е между зададения интервал. След това програмата групира резултата по дата и извежда усреднената стойност на цената на конкретната акция за целия период в интервала. Данните се записват в MongoDB базата и сесията се затваря.

Източници

Kate Blumberg, “7 Big Data Use Cases in Financial Services and Benefits of Data Science,” SAFEGRAPH, Nov. 04, 2021. https://www.safegraph.com/blog/top-big-data-use-cases-financial-services (отворен на Фев 24, 2022)

“Big Data in Finance - Your Guide to Financial Data Analysis,” Talend. https://www.talend.com/resources/big-data-finance/ (отворен на Юни 15, 2022)

Mark Smallcombe, “Structured vs Unstructured Data: 5 Key Differences,” Integrate.io, Feb. 16, 2023. https://www.integrate.io/blog/structured-vs-unstructured-data-key-differences/ (отворен на Юни 18, 2022)

“What is Apache Hadoop?,” IBM. https://www.ibm.com/analytics/hadoop (отворен на Авг 29, 2022)

“Advantages of Hadoop | Disadvantages of Hadoop,” RF Wireless World. https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Hadoop.html

“Apache Hadoop Architecture Explained,” Phoenixnap, May 25, 2020. https://phoenixnap.com/kb/apache-hadoop-architecture-explained (отворен на Септ 16, 2022)

“Banking industry architecture,” IBM. https://www.ibm.com/cloud/architecture/architectures/banking/reference-architecture (отворен на Окт 2, 2022)

“MongoDB for Financial Services,” MongoDB. https://www.mongodb.com/industries/financial-services (отворен на Окт 2, 2022)

“Hadoop and MongoDB,” MongoDB. https://www.mongodb.com/hadoop-and-mongodb (отворен на Фев 11, 2023)

“DATA PLATFORMS IN FINANCIAL SERVICES.” Medici. [Online]. Available: https://content.dataversity.net/rs/656-WMW-918/images/Data-Platforms-in-Financial-Services-NoSQL-Edge-Whitepaper.pdf (отворен на Фев 27, 2023)

“MongoDB Connector for Apache Spark,” MongoDB. https://www.mongodb.com/products/spark-connector (отворен на Март 11, 2023)

MEDICI, “Why Financial Services Should Look to NoSQL.” [Online]. Available: http://pages.aerospike.com/rs/229-XUE-318/images/Aerospike_Wp_Why_Financial_Services_Should_Look_to_NoSQL.pdf (отворен на Март 11, 2023)