Univerzita Pardubice

Fakulta elektrotechniky a informatiky

BigData ecosystem

Tomáš Prudký

Bakalářská práce
2021

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Tomáš Prudký**
Osobní číslo:	**I17273**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Téma práce:	**BigData ecosystem**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování

Cílem bakalářské práce je představení technologií BigData a jejich možné využití v praxi. V teoretické části bude popsán úplný přehled technologií v oblasti BigData a NoSQL databází. Výstupem praktické části bude výběr jedné komplexní technologie (hlavní technologie i související), na které bude představena práce s BigDaty. Práce bude analyzovat jednotlivé technologie, kdy u každé technologie bude uveden praktický příklad použití, výhody a nevýhody, možnosti API, apod. včetně detailního popisu systémových prostředků nutných pro běh dané technologie.

Rozsah pracovní zprávy: **min. 30 stran**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK. *Big Data a NoSQL databáze*. Praha: Grada, 2015. Profesionál. ISBN 978-80-247-5466-6.
MAYER-SCHÖNBERGER, Viktor a Kenneth CUKIER. *Big Data*. Brno: Computer Press, 2014. ISBN 978-80-251-4119-9.
Big Data: Definitions and Concepts. Big Data : Concepts, Challenges and Solutions. Available from: http://bigdata-tech.blogspot.com/p/big-data-definitions-and-concepts.html
What is the difference between data science, data analysis, data mining, machine learning, AI, and big data? Quora. Available from: https://www.quora.com/What-is-the-difference-between-data-science-data-analysis-data-mining-machine-learning-AI-and-big-data

Vedoucí bakalářské práce: **Ing. Monika Borkovcová, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **31. října 2020**
Termín odevzdání bakalářské práce: **14. května 2021**

L.S.

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

Ing. Jan Panuš, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 26. února 2021

Prohlašuji:

Práci s názvem BigData ecosystem jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 27. 7. 2021

Tomáš Prudký

Poděkování

Mé poděkování patří Ing. Monice Borkovcové, Ph.D. za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnovala.

Anotace

Cílem této bakalářské práce je představit čtenářovi Big Data ekosystém. Práce se v teoretické části zaměřuje na představení samotných Big dat, NoSQL databází, relačních databází, technologií od společnosti Apache, seznámení s pojmem open data a s vizualizačními nástroji. Praktická část je věnována práci se sadou dat, na které jsou provedeny operace jako nahrání dat, jejich analýza a vizualizace. Celý Big Data ekosystém je velice rozsáhlý, z toho důvodu celá práce je zaměřena spíše pro začínající uživatele.

Klíčová slova

apache, big data, ekosystém, noSQL databáze, open data, vizualizační nástroje

Title

BigData ecosystem

Annotation

The aim of this bachelor thesis is to introduce the Big Data ecosystem to the reader. The theoretical part of the thesis focuses on the introduction of the Big data itself, as well as the NoSQL databases, relational databases, Apache technologies, the concept of open data and visualization tools. The practical part is devoted to working with a dataset on which operations such as data loading, data analysis and visualization are performed. The whole Big Data ecosystem is very large, so for this reason the whole work is rocising more on novice users.

Keywords

apache, big data, ecosystem, noSQL database, open data, visualization tools

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM ZKRATEK

ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
BASE	Basically Available, Soft State, Eventually Consistent
BSD	Berkeley Software Distribution
BSON	Binary JSON
CAP	Consistency, Availability, Partition tolerance
CMD	Command Prompt
CQL	Cypher Query Language
CSS	Cascading Style Sheets
CSV	Comma-separated values
ELK	Elasticsearch, Logstash, Kibana
EMS	Elastic Maps Service
GB	Gigabyte
GPS	Global Positioning Systém
HDFS	Hadoop Distributed File http
HTTP	Hypertext Transfer Protocol
IOT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
KQL	The Kibana Query Language
LDAP	Lightweight Directory Access Protocol
MB	Megabyte
MMK	Mezinárodní Masarykova konference
NDJSON	Newline Delimited JSON
NoSQL	Non-SQL
PNG	Portable Network Graphics
RAM	Random Access Memory
RHEL	Red Hat Enterprise Linux
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLS	Transport Layer Security

USD	United States dollar
XML	Extensible Markup Language

ÚVOD

Big Data ekosystém v sobě zahrnuje nové technologie, které se soustředí na zpracování velkých objemů dat různého charakteru, které uživatelé svou činností zanechávají. Pojem Big Data, který nemusí být širokou veřejností znám, si každý však dokáže představit odlišně. Většina lidí vidí jen velké množství dat nebo velké soubory, ale Big Data jsou mnohem více. Druhý pojem z názvu je ekosystém, ten je pro laika více známý, ale ne každý opravdu ví, co takový ekosystém znamená a jak vůbec pracuje. Velká data v tomto ekosystému pochází z různých průmyslových, sociálních a vědeckých oblastí mající základní vlastnosti jako objem, rychlost, rozmanitost, hodnotu a důvěrnost.

V dnešní době plné moderních technologií se všude kolem nás každou chvíli generují velké množství dat. Běžnou lidskou činností v 21. století se jedná například o platby v obchodech, informace o telefonních rozhovorech, zprávy na sociálních sítích. S příchodem novějších sofistikovanějších technologií bude těchto dat ještě více přibývat, proto je potřeba umět s těmito daty správně pracovat a využívat je. Právě proto vzniknul pojem Big Data ekosystém. Jedná se o ekosystém, který umožňuje využít a zpracovat data z různých zařízení a různých oborů.

Tato práce má za cíl představit technologie sloužící v oblasti Big dat a NoSQL databází. Hlavním cílem bude popsat nejenom jejich vlastnosti, výhody, nevýhody a popis souvisejících prostředků pro jejich zpracování, ale také jejich využití v praxi. Práce se zaměřuje na současné trendy využívané v řešené oblasti, analyzuje a popisuje komponenty, které řeší hlavní výzvy velkých dat. Práce podrobně popíše celý Big Data ekosystém včetně jeho využití, NoSQL databáze, které se vážou s velkými daty a jsou součástí Big Data ekosystému, technologie Big Data a provede jejich srovnání. S příchodem Big Data úzce souvisí i otevřené datové sady tzv. Open Data, která jsou v práci také zmíněny. Pro výstupy z analýzy Big Data je vhodné využívat správné vizualizační nástroje, které jsou pro zpracování taktových to dat běžně používané, tato část bude popsána v samostatné kapitole.

V praktické části práce cílí na vybrané nástroje, s kterými bude představena práce s Big Daty. Výstupem práce bude ukázka, podle které bude čtenář schopen pracovat se souborem Big Dat a bude schopen jej zavést do vybraného systému, zanalyzovat a provést jejich vizualizaci.

1 BIG DATA

V úvodu první kapitoly se práce zaměřuje na vysvětlení pojmu Big Data. Seznamuje s využitím Big Dat v reálném světě a představuje pojem Big Data Ecosystem.

1.1 Vysvětlení pojmu Big Data

Big Data mají velké zastoupení v oblastech marketingu, sociálních sítích, cestovního ruchu, státní správě a dalších odvětvích. Pro osoby nepohybující se ve světě informačních technologií může být tento pojem cizí. Jediné, co si představí je velký objem dat.

Mnoho autorů uvádí, že přesná definice pojmu Big Data se nedá přesně specifikovat. Pojem Big Data se začal objevovat s příchodem sociálních sítí, nových technologií, služeb, mobilních technologií a dalších informací vyskytujících se na internetu. Díky velkému počtu zařízení a uživatelů se produkuje každou vteřinou velké množství nových informací, které se zaznamenávají do databází (Holubová, 2015). Společnost Gartner (2001) definuje pojem Big Data jako: „*Data, jejichž velikost, rychlost nárůstu a různorodost neumožňují zpracování pomocí doposud známých a ověřených technologií v rozumném čase*“. Tyto tři vlastnosti se nazývají anglickými slovy volume, velocity a variety a jsou označovány zkratkou „3V“.

Volume značí objem dat, který je zpracován. Jedná se o velké množství dat, která nejsou strukturovaná. Sociální média zpracují denně miliardy zpráv. Systémy v průmyslu jsou schopny generovat nepředstavitelné množství dat, v některých případech se může jednat o desítky terabajtů nebo až o stovky petabajtů (Bahga, 2016).

Velocity představuje rychlost, s jakou jsou data generována. Některé systémy mohou pracovat v reálném čase, proto potřebují tyto velké objemy rychle analyzovat a vyhodnocovat. Jedná se například o obchodování na burze, bezpečnostní služby, chatovací aplikace a služby jim podobné (Bahga, 2016).

Variety označuje různorodost dat. Datové systémy musí být flexibilní, aby byly schopny rozmanitost dat bez problému uložit. Mohou to být dlouhé texty, hlasové zprávy, obrázky, videa, datové toky či data ze senzorů (Bahga, 2016).

Postupem času přibyly další dvě „V“, value a veracity. Value (hodnota) odpovídá užitečnosti dat. Tato hodnota má velkou váhu při analýze, kdy pracuje jen

s informacemi, které jsou užitečné. Veracity obecně znamená pravdivost. Pro data není důležitá jenom jejich kvalita, ale také o důvěryhodnost zdroje, typ a zpracování. Je potřeba odstranit vlastnosti jako zkreslení, abnormality, duplikace, nekonzistence, které mají vliv na přesnost dat (Oracle, 2021)

1.2 Využití Big Dat

Tato kapitola se soustředí na to, jak Big Data ovlivňují jednotlivé sektory. Aby data mohla být plnohodnotně využita je potřeba mít k dispozici aplikace, které jsou schopny s takovým množstvím dat pracovat.

1.2.1 Web

Internetové analýzy se zabývají sběrem a analýzou dat o uživatelích přistupujících k internetu. Může se jednat o konkrétní stránky nebo cloudové aplikace. Mezi první zaznamenané informace, které uživatel poskytuje je datum, čas, IP adresa a stavový HTTP kód. K dalšímu zaznamenání informací se nejčastěji používá JavaScript, který při každé návštěvě může zaznamenávat informace o lidské činnosti a poskytnutých datech. Díky těmto informacím, které vědomě či nevědomě uživatel poskytuje, mohou organizace provádět lepší a složitější analýzy. Jednou z nejznámějších společností poskytující tyto analýzy je Google, která ve své službě Google Analytics nabízí sledování v reálném čase, informace o uživatelích a spoustu dalších informací, které se dají změřit (Bahga, 2016).

Dalším typem využití big dat na webu je monitorování výkonu. Velké projekty jako sociální sítě, systémy pro banky, zdravotnictví, obchod nebo jiné si nemohou dovolit výpadek při zvýšené zátěži systému. Proto se musí tyto systémy otestovat při velkém zatížení, kterého lze dosáhnout jedině s big daty. Díky těmto testům lze dostatečně připravit systémy na využití v reálném světě a předejít výpadkům, které by mohly v budoucnu stát organizace velké peníze (Bahga, 2016).

Velmi důležitou roli pro podniky na internetu je cílení reklamy. Jakmile uživatel zadá klíčová slova, která odpovídají konkrétní reklamě, tak po příchodu na stránku obsahující reklamu je mu nabízena ta, která odpovídá nejvíce jeho profilu. Tyto reklamní sítě využívají právě systémy big dat k porovnávání a umísťování reklam. Tyto systémy jsou pak schopny poskytnout statistické zprávy o reklamě. Může se jednat o úspěšnost reklamy, zacílení na konkrétní skupinu nebo přímo seznam konkrétních slov, která vedla k zobrazení reklamy. Díky těmto informacím, může

firma nejenom lépe cílit reklamu, ale také optimalizovat finanční rozpočet na reklamní kampaň (Bahga, 2016).

Poslední kategorií, která se často na internetu vyskytuje, je doporučení obsahu. Tyto aplikace využívají Big Data pro doporučení nového obsahu uživatelům. Může se jednat například o videa, produkty, hudbu nebo jiné typy dat. Nový obsah je doručován podle historie prohlížení, preferencí uživatele, zájmů a podle podobnosti uživatelů, kteří produkty již hledali (Bahga, 2016).

1.2.2 Finance

Banky a finanční instituce používají Big Data pro modelování úvěrového rizika. Aplikace, které využívají dostupná data o klientovi, se snaží předpovědět, zda dokáže dluh splatit nebo nikoliv. Systémy pracují s tzv. kreditním skóre, které je tvořeno z úvěrové historie, zůstatků na účtu, transakcí, příjmů a výdajů (Bahga, 2016).

Dalším způsobem pro využití big dat je detekce podvodů. Využívají se analýzy v reálném čase, které zkoumají jednotlivé transakce. Nástroje používané pro odhalení těchto podvodů používají strojové učení, kdy vytváří různé modely, které se snaží odhalit anomálie a podezřelé vzorce naznačující podvod. Může se jednat o podvody s kreditními kartami, legalizování finančních zdrojů z nekalé činnosti nebo pojistné podvody (Bahga, 2016).

1.2.3 Zdravotní péče

Samotný ekosystém zdravotní péče je složen z mnoha subjektů. Zapojují se do něj lékaři, pacienti, nemocnice, specializovaní odborníci, vláda, soukromé sektory, pojišťovny, zaměstnavatelé a jiné lékařské společnosti. Data, která se vyskytují v lékařském sektoru, jsou často uložena různými způsoby a v mnoha formátech. Může se jednat o uložení do relačních databází a souborových serverů. Pro sběr a kompletní analýzu těchto dat, jsou využívány technologie big dat, které umožňují provádět operace s různými formáty dat z rozdílných zdrojů (Bahga, 2016).

Big Data se využívají pro epidemiologické kontrolní systémy, které mají za úkol kontrolovat zdravotní stavy a snažit se detekovat výskyt nemocí, předpovídat ohniska nemocí, mapovat zdraví veřejnosti a dohlížet na zdravotní úroveň populace (Bahga, 2016).

Na základě pacientových příznaků, jsou technologie big dat schopny porovnat příznaky s ostatními pacienty, kteří dříve měli ty stejné příznaky. Právě díky stále rostoucím datům se tyto expertní systémy stávají mnohem spolehlivější. To může pomoci lékařům k přesnějšímu odhalení nemoci. Nemusí se jednat jen o nemoci, ale může se jednat i o zjištění nežádoucích účinků léků (Bahga, 2016).

1.2.4 IoT

Internet věcí neboli „Internet of Things“. Jedná se o ekosystém, kdy spolu jednotlivá zařízení jsou schopna komunikovat nebo spolupracovat bez zásahu člověka. Nejčastěji se jedná o připojení zařízení k internetu a zavedení OS. Přidáním těchto vlastností se některá zařízení stávají mnohem užitečnější. V dnešní době jsou typickým příkladem chytré hodinky, které jsou schopny nás upozornit na příchozí zprávy, hovory, hlídat náš tep nebo nabízí jiné vlastnosti (Koďousková, 2021).

Big Data v IoT mohou najít uplatnění při detekci narušení. Může se jednat o systém, který má za úkol monitorovat objekty. Jakmile kamery nebo jiné senzory detekují nebezpečí, jsou schopny odeslat upozornění ve formě SMS nebo e-mailu. Propracovanější systémy jsou schopny odeslat podrobnější informace, jako jsou fotografie, videa a jiné detailní zprávy (Bahga, 2016).

Dalším příkladem je chytré parkování, které má pomáhat řidičům v hledání parkovacího místa. Systémy IoT se starají o správu parkovacích míst, kdy detekují prázdná místa. Tyto informace odesílají do systémů, které mají za úkol správu těchto chytrých parkování. Skrze technologie big dat jsou tyto informace analyzovány a předány řidičovi (Bahga, 2016).

Dalšími kategoriemi, které se objevují ve světě IoT jsou chytré silnice, systémy pro monitorování stavu konstrukce a inteligentní zavlažování (Bahga, 2016).

1.2.5 Životní prostřední

Systémy, které monitorují životní prostředí, generují data o velkém objemu. Zároveň je nutné, aby data byla rychle zpracována a analyzována. Informace a analýzy mohou pomoci při snaze pochopit aktuální stav nebo při předpovědi, jak se bude životní prostřední vyvíjet (Bahga, 2016).

Mezi největší systémy využívající Big Data, patří sledování počasí. Jedná se o nespočet zařízení, která mohou zaznamenávat různá data, jako teplotu, vlhkost nebo

rychlost větru. Tyto data jsou následně odesílána do různých aplikací, které provádí analýzy dat. Výsledná data mohou být vizualizována pro aktuální počasí nebo předpověď (Bahga, 2016).

Protnutí světa životního prostředí s big daty nalezneme také při monitorování kvality vody, ovzduší nebo monitorování hluku v okolí. Systémy big dat se dají použít i pro detekci povodní nebo lesních požárů (Bahga, 2016).

1.2.6 Logistika a přeprava

Další využití, se kterým se lze v souvislosti s big daty setkat, je sledování provozu v reálném čase. Skrz technologii GPS je možné kontrolovat aktuální polohy vozidel. Tyto data o poloze jsou odesílána do aplikací big dat, které provádí jejich analýzu. Z těchto analýz jsou schopny poskytnout upozornění o blížících se kolonách a jiných omezeních. Na základě těchto informací mohou naplánovat alternativní cestu a optimalizovat dodavatelský řetězec (Bahga, 2016).

Sledování zásilek při přepravě využívá také technologií big dat. Nejedná se jen o sledování, kde se aktuálně zásilka vyskytuje. Může to být sledování teploty, vlhkosti a sledování podmínek uvnitř kontejneru, které je pro převoz potravin klíčové. Díky těmto informacím, může být obsah přepraven na bližší místo, aby nedošlo k znehodnocení a potraviny mohly být v pořádku prodány. Pro předměty, které by mohly být poškozeny, jsou měřeny informace o vibracích, skladování a manipulování. Z těchto informací pak lze dohledat, kde došlo k poškození (Bahga, 2016).

Uplatnění lze najít i při vzdálené diagnostice vozidel. Tyto diagnostické systémy shromažďují data od jednotlivých senzorů ve vozidle a odesílají je do cloudových aplikací big dat. Ty provádí analýzu těchto dat a jsou schopny poskytnout informaci o vadě ve vozidle (Bahga, 2016).

Platformy, které jsou určeny k objednání a doručení zboží využívají restaurace, obchody s potravinami a jiné podniky. Tyto systémy umožňují vyřídit si objednávku pomocí webových nebo mobilních aplikací. Těchto služeb v dnešní době využívá velká část populace. Proto systémy musí být schopny zpracovat velké množství objednávek naráz. Při provádění analýzy pak musí systém brát v úvahu polohu zákazníka a objednávku přidělit podniku, který ji může vyřídit (Bahga, 2016).

1.3 Ekosystém

V praxi se lze setkat s výrazy jako „Big Data Ecosystem“, „Big Data Environment“ nebo „Big Data Landscape“. Všechny tyto výrazy se podle většiny definic dají chápat téměř stejně. Tyto pojmy se většinou shodují na tom, že jsou složeny z infrastruktury, analytiky a aplikací na sběr, uložení, zpracování a analýzu dat. Jedním z důvodů, proč se používá ekosystém je ten, že se předpokládá jeho vývoj v čase, jako u skutečného ekosystému. Ekosystém označuje ucelenou strukturu technologií, které jsou propojeny. Navzájem mezi sebou komunikují a předávají si informace, které se ovlivňují, vyvíjí a mění se v čase a prostoru (Lewis, 2018).

Pro různé organizace se může jednat o velice rozdílné ekosystémy. U menších se lze setkat pouze s potřebou ukládat data, ke kterým budou mít přístup a velice omezené možnosti analýz. Větší organizace mohou mít celý ekosystém velice složitý. Každým dnem se mohou jejich požadavky měnit, vzhledem k tomu, že data extrémně narůstají. Pro tyto firmy může být klíčové nejenom ukládání dat, ale i jejich vizualizace, složité analýzy, strojové učení a jiné technologie, které se mohou v ekosystému objevovat (Lewis, 2018).

1.3.1 Infrastruktura

Základem pro dobře vybudovaný ekosystém je pevná infrastruktura. Nejedná se jenom o hardware, ale i o software pro ukládání, shromažďování a organizaci dat. Infrastruktura obsahuje servery, vyhledávací jazyky, zabezpečení a hostitelské platformy. Slouží pro příjem a uložení tří typů dat, kterými jsou strukturovaná, nestrukturovaná a více strukturovaná data.

Strukturovaná data jsou typicky organizovaná a dobře srozumitelná. Tyto data se vyskytují většinou v relačních databázích. Může se jednat například o jména, příjmení, adresy a údaje o kreditních kartách. Prakticky jsou to všechna data, která se dají rozepsat do pevných polí a sloupců. Jazyk používaný pro správu těchto dat se nazývá SQL („Structured Query Language“), v překladu strukturovaný dotazovací jazyk (Pickell, 2018).

Nestrukturovaná data jsou často označována jako kvalitativní data. Nelze je zpracovat a analyzovat standartními metodami, jako strukturovaná data. Mezi nestrukturovaná data se řadí například text, video, zvukové soubory, mobilní aktivita, příspěvky na sociálních sítích, satelitní snímky a spoustu dalších typů dat. Tento typ dat, je jeden

z nejčastěji se vyskytujících ve světě Big Dat. Tyto data potřebují dobré analytické nástroje a technické znalosti pro jejich využití. Správné pochopení a využití dává uživatelům mocný nástroj, díky kterému mohou zlepšit své služby (Pickell, 2018).

Více strukturovaná data mohou být například kombinací textových a vizuálních obrázků nebo kombinací strukturovaných a nestrukturovaných dat. Jedná se o data, která mohou být odvozena z interakcí mezi člověkem a strojem. Tyto informace poté slouží pro zlepšení zkušeností se zákazníkem a mohou se v čase vyvíjet (Arthur, 2013).

1.3.2 Analytika

Již dříve, když ještě pojem Big Data neexistoval, používali lidé základní analytiku. Jednalo se o údaje, které měli k dispozici. Mohlo se jednat například o údaje z prodejů, na kterých byli schopni odhalit různě vyvíjející se trendy. Je všeobecně známo, že dnes mají podniky k dispozici velké množství dat. Data jim nabízí široké možnosti a firmy je tak často využívají ve svůj prospěch. Nástroje, které provádí analytiku, jsou schopny poskytnout okamžité informace, které dávají možnost okamžitě zareagovat a zůstat agilní. Tyto analýzy odhalují skryté vzorce, korelace a další poznatky, které nelze zjistit se surových dat.

Analýza Big Dat dodává organizacím nové příležitostí, kterých mohou využít. To má za následek chytřejší obchodní tahy, efektivnější operace, vyšší zisky a spokojenější zákazníky. Tom Davenport, ředitel výzkumu IIA „Institute of Internal Auditors“, do svého výzkumu zahrnul více než 50 firem, u kterých zjišťoval, jak využívají analýzu Big Dat. Zjistil, že většina firem získá výhody pro snížení nákladů, rychlejší a kvalitnější rozhodování při změnách a při vývoji nových produktů a služeb (SAS, © 2021).

1.3.3 Aplikace

Aby společnosti mohly provádět analýzu dat, potřebují na to aplikace, které jsou schopny tuto analýzu zvládnout. Aplikace musí být schopné zpracovat velké objemy dat a získat z nich co nejvíce informací, které jsou následně schopny zahrnout do činností organizace a stát se efektivnější. Vzhledem k tomu, že jsou tyto informace velice ceněné, tak jsou společnosti ochotny utratit velké objemy finančních prostředků právě za tyto aplikace (Sunagar, 2020). Podle společnosti Wikibon, byly celosvětové tržby z trhu s Big Daty v roce 2018 42 miliard USD a v roce 2027 se předpokládá vzestup až na 103 miliard USD.

Většina těchto aplikací nejčastěji najde uplatnění v různých oblastech, jako jsou finance, telekomunikace, e-komerce, vzdělávání, obchodování, cestovní průmysl, automobilový průmysl a průmyslu s médii a zábavou (DataFlair, © 2021). Některé využití je popsáno v předchozí kapitole.

2 NOSQL

V této kapitole jsou představeny základní principy NoSQL databází, kategorizace databází a nejpoužívanější NoSQL systémy.

Pro zkratku „NoSQL“ se objevují dva výrazy. Někdo ji chápe jako „Non SQL“ a jiní jako „Not only SQL“. NoSQL databáze poskytují ukládání dat jinak než relační databáze. Databáze však může podporovat podobné SQL dotazy. Na rozdíl od relačních databází není přesně známo, jaká data bude muset databáze uchovávat. Za poslední roky se požadavky na systémy pro správu dat velice změnily. To znamenalo vývoj nových technologií pro správu dat. To mělo dopad na vznik velkého množství nových technologií, kdy široký výběr umožňuje vybrat takovou, která bude pro potřeby organizace nejvíce vyhovovat (Ghavami, 2016), (What is NoSQL?, 2021).

2.1 Základní principy

Tyto principy jsou základním stavebním kamenem pro skoro každou NoSQL databázi. Jedná se o škálovatelnost, konzistenci a distribuci. Tyto tři vlastnosti budou představeny v následujících podkapitolách.

2.1.1 Škálovatelnost

Existují dva typy škálovatelnosti, vertikální a horizontální. Jedná se o schopnost systému reagovat na zvyšující se objemy dat nebo zátěž systému. U tradičních databází se lze setkat s vertikálním škálováním. To znamená, že škálovatelnost je zajištěna přidáním výkonnějšího hardwaru nebo změnou vlastností toho stávajícího (IT slovník, 2017). Hlavní nevýhodou u vertikálního škálování je cena. Dalším nedostatkem je lehce dosažitelná hranice maximálního výkonu. Dříve než začne implementace, je vhodné odhadnout objem dat a dle toho zvolit vhodný hardware (Holubová, 2015).

Na rozdíl od vertikálního škálování, je horizontální řešeno přidáním dalšího výpočetního uzlu do clusteru. Cluster je označení pro více jednotek, které jsou propojeny do jednoho logického celku a v tomto celku mohou spolupracovat mnohem efektivněji. Nově přidaný výpočetní uzel může být běžný počítač (IT slovník, 2017), (Holubová, 2015).

2.1.2 Konzistence

Konzistence znamená, že jakmile jsou data úspěšně zapsána do databáze, tak všechny dotazy, které budou následovat, mají přístup ke stejným datům. Nesmí se stát, že by

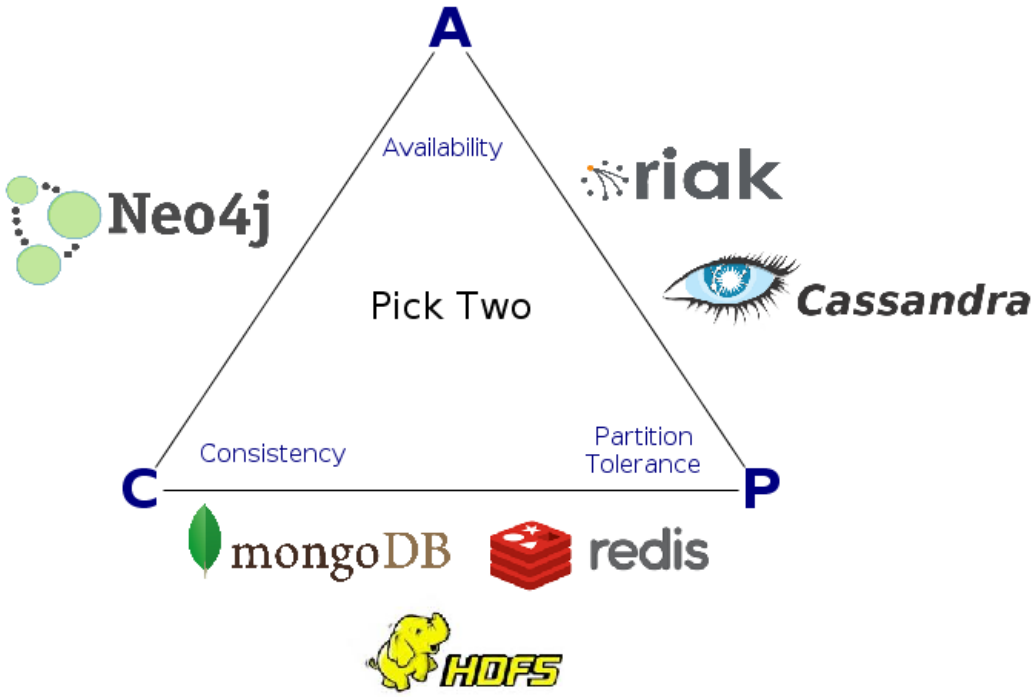
například došlo k odečtení peněz z jednoho účtu a nepřičetly se na druhý (Benčat, 2012). Pro zajištění konzistence databázové systémy pracují s transakcemi. V tradičních databázích se objevují vlastnosti ACID nebo BASE. U NoSQL databází se většinou s těmito vlastnostmi nesetkáme, existují tady pouze velmi omezené transakce. Lze se ale setkat s pojmem CAP teorém (Holubová, 2015).

ACID je zkratka pro anglické pojmy „Atomicity“, „Consistency“, „Isolation“, „Durability“. Atomicita zajišťuje, aby transakce proběhla celá nebo vůbec. Používají se operace ROLLBACK pro zajištění, aby transakce nebyla potvrzena a operace COMMIT, která transakci potvrzuje. Konzistence slouží pro zajištění přechodu mezi stavy konzistentní, nekonzistentní a zaručuje zápis pouze platných dat. Izolace skrývá operace, které se provádí uvnitř transakcí, aby nebyly viditelné ostatním transakcím před jejím dokončením. Trvalost se stará o to, aby po úspěšné transakci, byla data uložena do databáze (Aleksic, 2020).

BASE znamená „Basically Available“ (převážně dostupný), „Soft state“ (volný stav), „Eventual consistecy“ (převážně konzistentní). Převážná dostupnost znamená, že systém je většinu času dostupný. Mohou se zde vyskytovat výpadky, ale nedojde k výpadku celého systému. Volný stav znamená, že nemusí být vždy zaručena konzistence, protože se v systému neustále provádí změny. Odpovědnost, jak systém bude pracovat s konzistencí je předána na vývojáře a ten podle potřeb systému zajistí dostatečnou konzistenci pro daný systém. Občasná konzistence znamená, že nikdy nebude zaručena neustálá konzistence dat (Kopal, 2015).

Vlastnosti ACID nebo BASE by ve světě NoSQL databází byly užitečné, ale málokdy je lze použít. Nejčastěji je nelze využít kvůli replikaci, výpadkům sítě nebo distribuci. Zajištění těchto vlastností by bylo velmi obtížné a docházelo by k výraznému zpomalení (Holubová, 2015). Proto je používán CAP teorém, „Consistency“ (konzistence), „Availability“ (dostupnost), „Partition tolerance“ (odolnost vůči výpadkům sítě) označovaný jako Brewerův teorém. Tento pojem znamená, že pro NoSQL systémy nelze zajistit více než dvě ze tří garancí. Konzistence znamená, že všichni uživatelé vidí stejná data současně. Z toho důvodu musí být všechna data replikována do všech uzlů předtím, než bude zápis považován za úspěšný. Dostupnost dat, i kdyby některé uzly byly nedostupné, tak klient dostane správnou odpověď. Pod odolností vůči rozpadu sítě si lze představit, že systém bude funkční i za předpokladu,

že některé uzly v sítí přestanou pracovat. Systém musí navzdory těmto nefunkčním částem pracovat, jako by byl plně funkční a poskytovat odpovědi (IBM, 2019).



Obrázek 1, CAP Theorem (Olety, 2014)

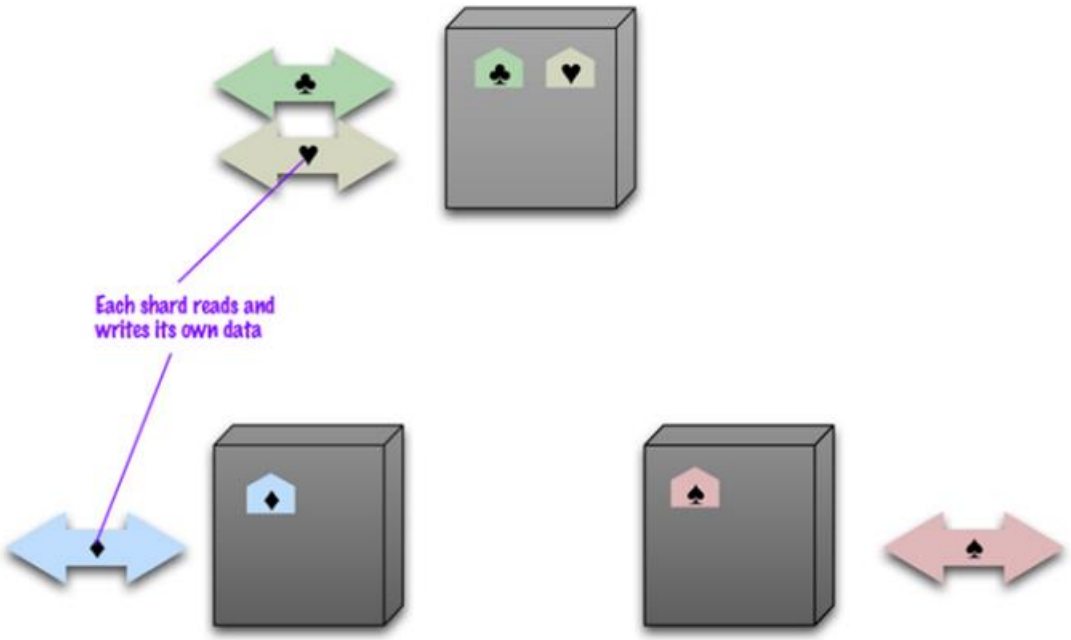
2.1.3 Distribuce

Tato podkapitole seznamuje čtenáře s různými typy distribucí dat. Pro účely NoSQL databází je většinou jeden uzel nedostatečný a je potřeba data distribuovat mezi více uzlů.

Nejjednodušší možností, se kterou se lze setkat je mít data uložena na jednom zařízení „Single server“. Ovšem tím pádem nedochází k distribuci, ale lze se tak vyhnout složitostem, které se musí řešit v ostatních případech, což ovšem je proti základnímu pojetí těchto databází. Vzhledem k tomu, že NoSQL databáze jsou navrženy s myšlenkou rozdělení provozu na clustry, tak je toto řešení spíše výjimečné. Toto řešení ale najde své využití u grafových databází, kdy právě tento typ funguje na jednom zařízení nejlépe. Grafové databáze se potýkají se složitostí distribuce, když je graf téměř plný. Pokud je to možné je snaha data distribuovat pouze minimálně nebo vůbec, aby byly minimalizovány problémy (Sadalage, © 2013).

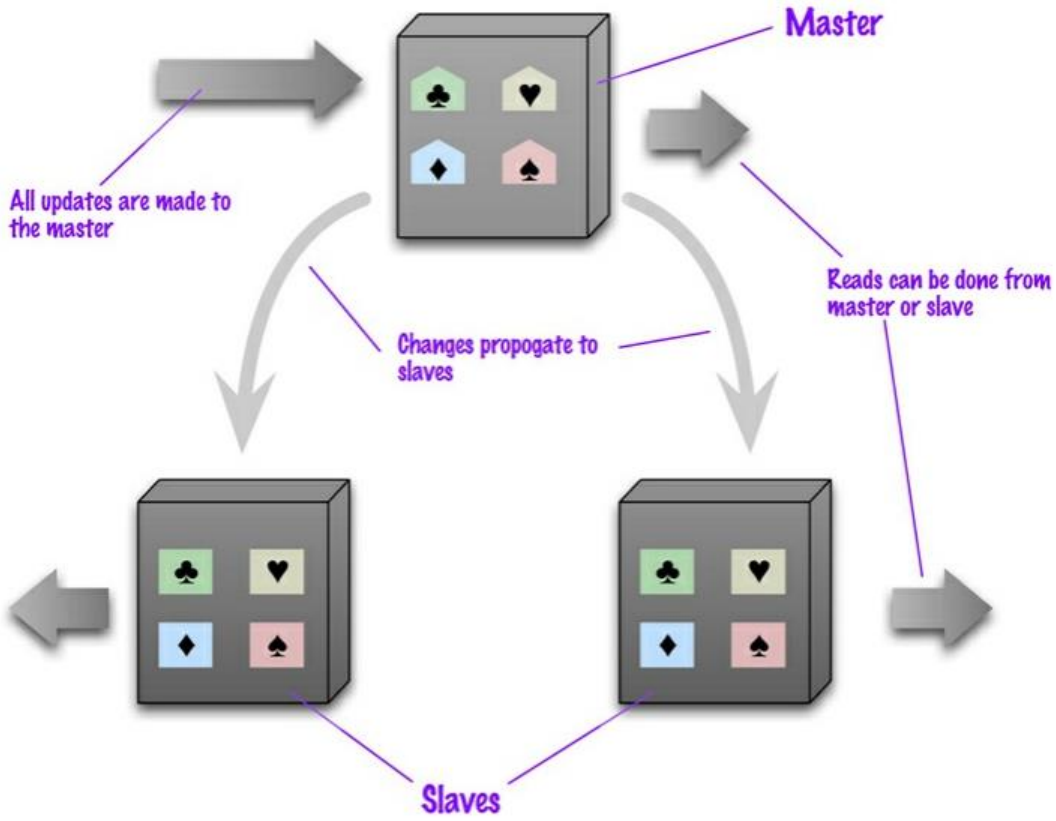
Rozdělení dat neboli „sharding“, označuje rozdělení dat na různé části. Jedna tato část je nazývána „shard“. Každá tato část je uložena na různých databázových serverech. Díky tomu, že jsou data rozmístěna na více serverů, nedochází k takovému zatížení na

jednom serveru a jednotlivé dotazy jsou zpracovány rychleji. Uživatel poté přistupuje pouze k serveru, který obsahuje data, o které žádá. Nevýhodou může být, že pokud vypadne jeden uzel nebo větší část, kde byla uložena stejná data a nejsou dostupné jejich zálohy, tak o ně správce databáze přijde. Při implementaci tohoto řešení je snaha, aby data byla rovnoměrně rozmístěna mezi všechny servery. Cílem je minimalizovat počet serverů, ke kterým musí uživatel přistupovat a co nejlépe je geograficky rozmístit (Sadalage, © 2013).



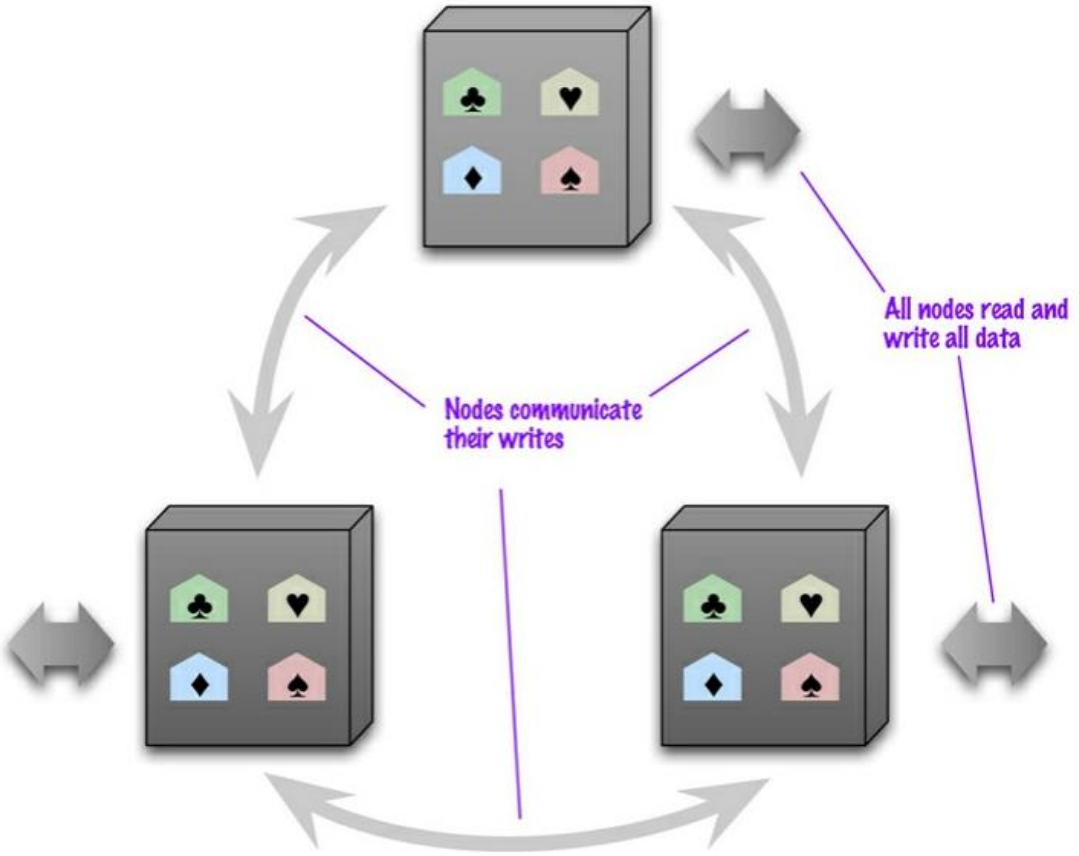
Obrázek 2, Distribuce typu sharding (Sadalage, © 2013)

„Master-slave“ replikace v překladu mistr-otrok, předpokládá existenci více uzlů, na kterých jsou uložena stejná data. V této distribuci existuje jeden „master“, který je určen jako hlavní server a libovolný počet „slaves“, kteří plní roli sekundárních serverů. Zápis dat je povolen pouze hlavnímu serveru, čtení však mohou provádět všechny servery. Tento způsob se vyplatí použít, když je očekáváno velké množství dotazů na data, ale menší potřebu zápisu dat. Vzhledem k tomu, že všechny servery mají stejná data, tak při výpadku hlavního serveru, stačí jmenovat otroka mistrem. Je tedy málo pravděpodobné, že by došlo ke ztrátě dat, pokud jsou servery dobře geograficky rozmístěny a nedojde ke globální katastrofě (Sadalage, © 2013).



Obrázek 3, Distribuce typu master-slave (Sadalage, © 2013)

„Peer-to-peer“ replikace překonává hlavní nedostatek předchozího typu „master-slave“. Všechny vyskytující se uzly jsou si rovny, to znamená, že mají právo pro čtení i zápis dat. Výpadek jakéhokoli serveru nezpůsobí téměř žádný problém, uživatel se bude pouze dotazovat jiného serveru. Hlavní nevýhodou tohoto řešení je zachování konzistence dat. Tato nevýhoda se většinou řeší tím, že se uzly při zápisu domluví, to má za následek zpomalení. Druhá možnost je nastavení volebního principu, což znamená, že k úspěšnému provedení je nutné, aby se operace provedla nad polovinou připojených zařízení (Sadalage, © 2013).



Obrázek 4, Distribuce typu peer-to-peer (Sadalage, © 2013)

2.2 Databáze typu klíč-hodnota

Databáze bývají jednodušší, lze si je představit jako asociativní pole. Jedná se o takovou datovou strukturu, která je složena z dvojice klíč-hodnota. Často bývá nazývaná jako „hash“ tabulka nebo mapa. Hodnoty jsou ukládány podle unikátního klíče. Tento typ databází většinou neposkytuje vyhledávání jinak než podle primárního klíče, takže nejsou potřeba žádné složité dotazy. Hlavním důvodem používání tohoto typu databází je, že mnoha aplikacím dostačuje vyhledávání podle primárního klíče. Úložiště, která se používají, bývají extrémně rychlá a dají se efektivně distribuovat (What is NoSQL?, 2021), (Holubová, 2015).

2.2.1 Redis

Redis je přední a typický zástupce pro databáze typu klíč-hodnota. V celém znění „Remote Dictionary Server“, v překladu vzdálený slovníkový server. Dříve se jednalo o soukromý projekt, který se stal postupem času open source pod licencí 3-Clause-BSD, což znamená, že se šíří s otevřeným kódem a jeho využití je zdarma. Byl napsán v jazyce C. Nejčastěji je využíván na posixových systémech jako například Linux,

BSD, MacOS a je také doporučeno provozovat ho pávě na Linuxu. Pro systémy Windows neexistuje oficiální podpora, ale existuje jeho experimentální verze. Dále jsou podporovány platformy jako RHEL, CentOS, Amazon Linux, Docker a Kubernetos (Redis, © 2021).

První vlastností, kterou se Redis liší od ostatních databází je, že drží svou databázi v mezipaměti (cache) a disk používá pouze pro trvalý zápis, pokud je to vyžadováno. Díky tomu rychle pracuje s daty a je velice výkonný, dokáže škálovat až stovky gigabajtů dat a miliony požadavků za vteřinu. Jeho další výhodou je podpora velkého množství datových typů, jako řetězce, což mohou být také celá čísla, desetinná nebo dokonce i binární data. Dále podporuje seznamy, hashe, sady, seřazené sady, bitmapy, hyperloglogy a geoprostorové indexy a streamy. Poslední důležitou vlastností je, že může data replikovat na libovolný počet „slaves“. Mezi další důležité vlastnosti, které stojí za zmínku, patří ukládání klíčů a hodnot do velikosti až 512 MB. Používá své hešovací funkce „Redis Hashing“ a jeho instalace a nastavení je velice jednoduché a snadné k používání (Patel, 2020).

Redis je databází, která by měla mít data uložena v paměti. Velkou nevýhodou je právě ona paměť. Pokud je požadováno ukládat větší objemy dat, tak zaplatíme vyšší částku. Vzhledem k tomu, že se jedná o databázi typu klíč-hodnota, tak zde nenajdeme ani žádné fulltextové vyhledávání. Nepodporuje agregované funkce jako součet, průměr a ani žádný dotazovací jazyk, a proto jsou potřeba všechny data zpracovávat až na straně klienta. Najdeme zde pouze základní možnosti zabezpečení. Neposkytuje žádnou kontrolu přístupu ani šifrovací mechanismus (Al-Saeedi, © 2016).

Redis je podporován velikým počtem programovacích jazyků. Podporují jej jazyky jako C, C++, C#, Java, PHP, Python, Node.js, Ruby, Perl, Go a mnoho dalších (Redis, © 2021).

Mezi nejznámější firmy, které používají Redis patří Twitter, GitHub, Pinterest, Snapchat, StackOverflow, Flickr, Instagram a mnoho dalších firem a vývojářů (Redis - Reviews, Pros & Cons | Companies using Redis, © 2021).

2.2.2 Riak

Dalším velice známým a používaným zástupcem databází typu klíč-hodnota je Riak. Byl napsán v jazyce Erlang. Riak nabízí dvě hlavní verze svého produktu, a to Riak KV (key-value) a Riak TS (time series). Tyto verze se poté ještě dělí na dalších pět

a to na „OSS“, „Developer“, „Pro“, „Enterprise“ a „Enterprise plus“. Riak TS by mohl být chápán jako nadstavba Riak KS. Je postaven na stejných základech s tím rozdílem, že je obohacen o některé funkce. Hlavním rozdílem je optimalizace pro IoT a časové řady. Jedná se o data seřazená v čase. Dokáže dobře spolupracovat s Redis, protože používá některé jeho funkce. Riak není podporován na operačních systémech Windows, ale podporuje Windows Azure, což je cloudová služba. Dále jsou podporovány operační systémy jako Mac OS X, Debian, Ubuntu, RHEL, CentOS, SmartOS, Solaris, SUSE a cloudová služba Amazon Web Services (Riak KV, 2008).

Výhodou Riaku je vysoká dostupnost dat, výkon, flexibilita a snadná správa. Riak je postaven na platformě, která má být velice odolná vůči selhání. Na rozdíl od Redisu již podporuje fulltextové vyhledávání. Při přechodu do režimu off-line stále funguje. V neposlední řadě Riak nabízí non-stop podporu k zakoupenému produktu (Riak - Reviews, Pros & Cons | Companies using Riak, © 2021).

Pokud Riak není provozován pod jednou ze dvou nejlepších verzí, tak podpora je pouze v pracovní hodiny a při využití Riak ve free verzi produktu, pak není nabídnuta žádná podpora. Některé zdroje si stěžují na nákladný provoz a také, že při změně indexování velice dlouho trvá (Riak - Reviews, Pros & Cons | Companies using Riak, © 2021).

Riak už nenabízí takovou podporu programovacích jazyků jako Redis, ale i přesto podporuje většinu nejpoužívanějších. Některé z nich jsou C, C#, C++, Java, JavaScript, Python, Ruby nebo Smalltalk (Riak KV vs. Riak TS Comparison, © 2021).

Mezi uživatele Riaku patří, například Uber, hazardní společnost bet365, videoherní spolčenost Riot Games, Yahoo!, McAfee, Virgin America a spoustu dalších (Riak, 2008).

2.3 Dokumentové databáze

Dalším typem jsou dokumentové databáze. Jsou o něco složitější než předchozí typ. Dokumenty, které slouží jako datová struktura, obsahují nejenom data, ale i metadata popisující jednotlivé části datové struktury. Často se zde objevují formáty jako JSON, BSON nebo XML. Tyto formáty představují stromovou strukturu, která obsahuje asociativní pole, seznamy a základní datové typy (What is NoSQL?, 2021), (Holubová, 2015).

2.3.1 MongoDB

Nejpoužívanější dokumentovou databázi je právě MongoDB a je zároveň jednou z nejvíce používaných NoSQL databází. Databáze byla napsána v programovacím jazyce C++ a je poskytována jako open source. Na výběr máme z produktu pro cloud „MongoDB Atlas“ nebo server, kdy je na výběr z verze „Comunity Server“ a „Enterprise Server“. Na stránkách si lze stáhnout i rozšiřující programy pro ulehčení práce. Na rozdíl od již zmíněných NoSQL databází, MongoDB už nabízí i podporu pro Windows. Dále poskytuje podporu pro Linux, MacOS, Docker (MongoDB System Properties, © 2021).

Kromě snadné instalace a nastavení, je k dispozici mnoho dalších výhod. Díky tomu, že jsou data ukládaná ve formě JSON (BSON), odpadá tak nutnost schématu, protože je definováno kódem. Tento formát dokáže dobře uchovávat nejenom samostatná data, ale celá pole nebo jiné dokumenty. Obsahuje svůj dotazovací jazyk „MongoDB Query Language“. Ten hraje významnou roli při dynamickém dotazování. MongoDB lze použít také jako souborový systém, který napomáhá vyrovnávat zátěž (Advantages of using MongoDB, © 2021).

Jednou z nevýhod je nedokonalé a pracné spojování dokumentů, které má za důsledek ovlivnění výkonu. Velikost jednoho dokumentu může být maximálně 16 MB. Dokumenty mohou mít maximálně 100 levelů zanoření. Pokud nebudou použity správné indexy, lze očekávat velice nízkého výkonu.

Opět i MongoDB podporuje velkou část programovacích jazyků jako například C, C#, C++, Go, Java, JavaScript, Perl, Python, Swift a jiné (MongoDB System Properties, © 2021).

Jak již bylo řečeno, tak MongoDB je jednou z nejpoužívanějších databází, a proto jej používají velice známé a velké firmy. Například Google, Cisco, eBay, Adobe, automobilka Toyota, vývojářská firma Sega, Electronic Arts. MongoDB nepoužívají pouze firmy, ale například i město Chicago. Jak uvádí MongoDB „Chicago používá MongoDB k vytvoření chytřejšího a bezpečnějšího města“. Systém zpracovává údaje od policie, o dopravě, požáru, zpoždění na silnicích, vyzvednutí odpadu, mimořádných událostí, veřejné tweety a spousty dalších údajů (Our Customers | MongoDB, © 2021).

2.3.2 Firebase Realtime Database

Další zajímavou databází je Firebase od společnosti Google. Jedná se o první databázi, která se označuje za komerční, přesto nabízí open-source verzi. Firebase je cloudová databáze, která ukládá data ve formě JSON. Data se synchronizují a ukládají v reálném čase s každým připojeným klientem. Každý klient má přístup k jedné instanci databáze a díky tomu dostávají všichni nejnovější aktualizovaná data v reálném čase. Platforma se používá především pro vývoj mobilních a webových aplikací (Firebase Realtime Database | Store and sync data in real time, 2021).

Jako své klíčové vlastnosti firma uvádí práci v reálném čase, pohotovost v off-line módu, přístup z klientského zařízení a škálovatelnost přes více databází. První vlastnost je pochopitelná z názvu, ale pro ujasnění se jedná o synchronizaci dat pokaždé, když dojde k nějaké změně. Každé připojené zařízení dostane v co nejkratším čase tuto aktualizace. Tento způsob nahrazuje HTTP požadavky. Když dojde k odpojení od internetu nebo k neočekávanému výpadku, tak databáze zapisuje data na disk a po opětovném připojení k síti, jsou data aktualizována. K databázi se lze připojit z mobilního zařízení nebo webového prohlížeče, není nutné se připojovat přímo ze serveru. Škálovatelnost přes více databází lze chápat, jako rozdělení Firebase databáze na více samostatných instancí. Každá tato instance může mít vlastní pravidla a při připojení klienta jsou kontrolovány jeho přístupová práva k dané instanci (Firebase Realtime Database, 2020).

Firebase distribuuje společnost Google, proto se nelze divit, že je podporován hlavně pro zařízení s Android systémy. Z toho důvodu má tedy omezenou podporu pro iOS. Další nevýhodou je cena. Lze si vybrat verzi, která je zdarma, ale pro větší projekty bude nedostačující. Jestliže roste velikost a úspěšnost projektu, roste také cena. Systémy pracující v reálném čase, mají tu nevýhodu, že od nich nelze dostat žádnou zprávu o nadcházející změně nebo může například docházet ke konfliktu mezi daty. Další problém je migrace dat, nabízí téměř nulové SQL funkce. Firebase nebude fungovat tam, kde není podporován Google. Ten není podporován třeba na Kubě, Krymu, Íránu, Severní Koreji nebo Sýrii (Isichko, 2020).

Podpora programovacích jazyků není příliš velká. Nabízí podporu například pro Switft, Objective-C, Java, Kontlin+KTX, JavaScript, C++ nebo herní engine Unity.

Firebase je využíván hlavně na systémech, kde je potřeba podpora v reálném čase. Například pro streamovací platformu Twitch.tv, platební společnost Square, pro společnosti obchodující s akciemi, jako například Bitpanda. Adidas používá Firebase pro svoji aplikaci Runtastic, která slouží k záznamu trasy běhu a poskytnutí dalších údajů (Firebase - Reviews, Pros & Cons, © 2021).

2.4 Sloupcové databáze

U sloupcových databází je základní jednotkou pro ukládání dat sloupec, který má název a hodnotu. Data z jednotlivých sloupců pak tvoří řádek, který lze jednoznačně určit klíčem. Sloupce zde tvoří takzvané rodiny sloupců. Tyto rodiny nemusejí mít pevně daný počet sloupců, to znamená, že počet sloupců se může u jednotlivých řádků lišit. Data jsou ukládána v denormalizované podobě, díky tomu lze získat všechny informace pomocí čtení jednoho řádku (Nayak, 2013).

2.4.1 Apache Cassandra

Databáze byla vytvořena jako nástroj pro vyhledávání v poště pro sociální síť Facebook. Postupem času přešla do správy společnosti Apache Software Foundation. Databáze byla napsána v programovacím jazyce Java a nabízí kompatibilitu s operačními systémy Windows, Linux, macOS a BSD. Apache Cassandra se pyšní tím, že se jedná o distribuovanou a decentralizovanou databázi nebo úložný systém, který je nabízen jako open-source. Je velice odolná proti chybám, a proto je vhodná pro ukládání kritických dat (Cassandra System Properties, © 2021).

Cassandra poskytuje dobře zpracovanou dokumentaci na oficiálních stránkách a spousty ostatních materiálu k dohledání na internetu, proto je výhodou s ní pracovat. Roste její komunita, takže lze lehce získat jakékoli tipy k jejímu použití nebo vyhledání pomoci. Jednou z klíčových vlastností je dostupnost. Databáze využívá datových center nebo takzvaných uzlů, které jsou rozmístěny na různých místech. To předchází nedostupnosti dat při nějakém nečekaném výpadku či přírodní katastrofě. Další výhoda se týká právě těchto uzlů, které jsou snadno nahraditelné při výpadku nebo selhání. Všechny uzly jsou si rovny a není tu žádný řídící uzel (Ferrando, © 2021).

Replikace dat může být i nevýhoda. Databáze nereplikuje pouze korektní data, ale i data, která obsahují chybu. Jako nevýhoda by se dala brát i skutečnost, že pokud nedojde k opravě poškozeného uzlu v určitém časovém okně, tak si uzly rozdělí data, která měla být na poškozeném uzlu. Tím může dojít ke změně uspořádání dat nebo

větší zátěži na ostatní uzly. Pokud se uzel obnoví v čas, tak jsou na něj data odeslána od ostatních uzlů, kde dočasně setrvávaly. Cassandra neumožňuje volání neočekávaných dotazů, protože musí mít nejdříve nastavené indexy (Ferrando, © 2021).

Mezi podporované programovací jazyky patří C#, C++, Go, Java, JavaScript, Perl, PHP, Python, Ruby a Scala (Cassandra System Properties, © 2021).

Databáze je velice odolná proti výpadkům, proto ji používají společnosti, pro které by výpadek mohl znamenat velké ztráty nebo jejich renomé. Jak již bylo řečeno Cassandra vznikla jako projekt pro Facebook, a tak se nelze divit, že ji používá stále. Dále databázi využívají projekty, jako Uber, Netflix, Instagram, Spotify, eBay, Booking.com, Starbucks a mnoho, mnoho dalších (Cassandra - Reviews, Pros & Cons | Companies using Cassandra, © 2021).

2.4.2 Apache HBase

Další často používanou NoSQL databází je HBase, kterou také zaštiťuje stejná společnost Apache jako v předchozím případě. Protože je Apache neziskovou společností, tak i HBase je open-source, který byl implementován v jazyce Java. HBase vznikl po vzoru databáze BigTable od společnosti Google a je označován za Hadoop databázi. Data jsou uchována ve velkých tabulkách, které jsou distribuovány napříč clustery komoditního hardwaru. HBase nabízí lineární a modulární škálovatelnost a spoustu dalších robustních funkcí, které slouží pro práci s tabulkami. Všechny tyto funkce jsou poskytovány „Hadoopem“ a HDFS „Hadoop Distributed File System“ (Apache HBase – Apache HBase™ Home, 2021).

Jelikož je HBase sloupcová databáze a velice podobá relačním databázím, tak tam kde jsou relační databáze nedostačující, je velice vhodné použít právě HBase, která dokáže uložit velká data na úložiště HDFS. Zároveň agreguje a analyzuje miliardy řádků ve svých tabulkách. Nabízí dobré zabezpečení pomocí Apache Atlas nebo pluginu Ranger. Jednoduché použití rozhraní Java API pro přístup klienta. Podpora převzetí služeb při selhání a sdílení zátěže. Neobsahuje žádnou integrovanou autentizaci a možnosti oprávnění (HBase Pros and Cons | Problems with HBase, © 2021).

Nevýhodou je, že v HBase neexistuje žádná podpora transakcí. Spojování tabulek se provádí ve vrstvě MapReduce. Nenabízí SQL dotazování. Indexovat a třídit lze pouze pomocí klíče (HBase Pros and Cons | Problems with HBase, © 2021).

Databázi lze provozovat na operačních systémech Linux a Unix. Na Windows je potřeba instalace programů Cygwin, které napodobují chování unixových systémů. HBase nabízí pouze podporu programovacích jazyků jako C, C#, C++, Groovy, Java, PHP, Python a Scala (HBase System Properties, © 2021).

HBase našla uplatnění v projektech jako Pinterest, Tumblr, Hubspot, Awin, Cask, GameDuell, HeadHunter, Chartbeat a jim podobných (HBase - Reviews, Pros & Cons | Companies using HBase, © 2021).

2.5 Grafové databáze

Jak bylo řečeno u výše uvedených typů, tak využívaly klíče pro identifikování dat. Tento typ používá jako datovou strukturu graf. Ten je tvořen uzly, které jsou propojeny pomocí hran. Jednotlivé uzly představují objekty a hrany představují vztahy mezi nimi. Každý objekt má své vlastnosti, ale ty mohou mít i hrany, jako například typ, dobu platnosti vztahu, podmínky platnosti vztahu a jiné. Grafové databáze jsou dobré pro procházení vztahů a hledání vzorů. Mají uplatnění u sociálních sítí a detekce podvodů (What is NoSQL?, 2021), (Holubová, 2015).

2.5.1 Neo4j

Jak uvádí Neo4j na svých oficiálních stránkách „Neo4j je nativní databáze grafů, vytvořena od základu k použití nejen dat, ale také vztahů mezi nimi. Neo4j propojuje data tak, jak jsou uložena, což umožňuje používat dotazy, které si nikdo nedokázal představit, a to rychlostí, která byla považována za nemožnou.“ Na rozdíl od klasických databází neukládá Neo4j data do řádků, sloupců či tabulek, ale má flexibilní strukturu definovanou vztahy mezi záznamy. Databáze byla vytvořena v programovacích jazycích Java a Scala a podporuje operační systémy Windows, Linux, macOS, Solaris (Neo4j System Properties, © 2021).

Obsahuje vlastní dotazovací jazyk Neo4j CQL, který je dobře čitelný a snadno pochopitelný. Je kompatibilní s vlastnostmi ACID a poskytuje plnou podporu transakcí. Neo4j má dobrou podporu, nabízí výukové materiály, které mohou pomoci s nasazením databáze, dále poskytuje velké množství knih od odborníků (Neo4J - Features & Advantages - Tutorialspoint, © 2021).

Mezi nevýhody Neo4j lze zařadit nemožnost sdílení, pokud jsou data uložena pouze na jednom serveru, je možné použít pouze vertikální škálování. Pokud není zprovozněna verze Enterprise, je databáze omezena o mnoho vylepšení, jako například

velikostí grafů, online zálohováním, řízením přístupu dle rolí a dalších výhod (Introduction - Operations Manual, © 2021).

Neo4j podporuje spoustu populární programovacích jazyků. Mezi nejznámější patří třeba Java, JavaScript, Python, Go, Ruby, PHP, Erlang, Perl, C, C++. Vývojářům je doporučeno používat vývojové prostředí Jetbrains IDE hlavně kvůli pluginu, který ulehčuje práci s vývojem a také protože s ním mají vývojáři velmi dobré zkušenosti.

Databáze má velké využití v mnoha odvětví. Mezi nejznámější firmy používající Neo4j patří například Allianz, Adobe, eBay, Microsoft, IBM (Neo4j Customers, 2021).

2.5.2 JanusGraph

Jedná se o databázi optimalizovanou pro ukládání a dotazování nad grafy, které obsahují až stovky miliard vrcholů a hran. Jedná se o projekt organizace The Linux Foundation, na kterém se účastní společnosti Expero, Google, GRAKN.AI, Hortonworks, IBM a Amazon. JanusGraph je vyvíjen v jazyce Java pro operační systémy Linux, MacOS, Unix a Windows a je poskytován jako open-source (JanusGraph, © 2021).

JanusGraph velice dobře ukládá velké grafy a velikost se dokáže zvětšovat s každým dalším připojeným zařízením. Databáze je schopna uložit 2^60 hran a až polovinu vrcholů. Podporuje vyhledávání dle číselného nebo geografického rozsahu a fulltextového vyhledávání hran a vrcholů. Obsahuje implementovanou podporu jazyka Gremlin, který lze použít jako dotazovací jazyk. V neposlední řadě JanusGraph nabízí optimalizaci reprezentace dat na disku, která má za následek efektivní ukládání dat a přístupovou rychlost k nim (The Benefits of JanusGraph, © 2021).

Velikost, kterou nabízí JanusGraph se může zdát dostatečná, ale ve světě Big Dat tomu tak nemusí být, protože může dojít k brzkému vyčerpání úložného prostoru. Databáze vyžaduje přesné definování datových typů nebo používání flexibilního Object.class. Pro dosažení nejlepšího výkonu a bezpečnosti se doporučuje používat skutečné datové typy, které bude databáze používat. Datové typy se nedají měnit, lze však přidat nové, aby se dalo reagovat na měnící se požadavky. Získání hrany není operace s konstantním časem, vyžaduje volání indexu na jednom ze sousedních vrcholů. Pokud je databáze rozsáhlejší, může to trvat déle. Za účelem zkrátit tuto dobu, se systém pokouší vybrat vrchol s menším stupněm. JanusGraph má problém s načtením velkého

počtu hran do jednoho vrcholu najednou nebo v krátké době. Při tomto načtení může dojít k selhání (Technical Limitations - JanusGraph, © 2021).

Projekt JanusGraph je poměrně nový, proto je jeho podpora programovacích jazyků prozatím menší než u předchozích databází. Na výběr jsou tři možnosti, kterými je Java, Python a Clojure (JanusGraph System Properties, © 2021).

JanusGraph využívají společnosti jako eBay, FiNC, Netflix, Red Hat, Uber, Celum nebo G Data (JanusGraph, © 2021).

2.6 Hybridní databáze

Poslední typ databází jsou databáze hybridní neboli multi-modelové. Tento typ kombinuje různé typy databázových modelů ať už relační, objektově orientované, klíč-hodnota, sloupcové, dokumentové nebo grafové. Díky velkému množství modelů lze ukládat skoro všechny typy dat včetně polostrukturovaných a nestrukturovaných. Tento benefit odstraňuje některé problémy, se kterými se setkávají databáze podporující pouze jeden datový model. Hybridní databáze většinou poskytují základní operace, jako ostatní databáze. Prodejci často obohacují databáze o funkce navíc (Imanuel, 2013).

2.6.1 ArangoDB

Společnost ArangoDB GmbH vyvinula nativní databázový systém s více modely. Databáze podporuje tři datové modely klíč-hodnota, dokumenty a grafy. ArangoDB má jednotný dotazovací jazyk AQL (ArangoDB Query Language), který umožňuje kombinovat různé vzory přístupu k datům v jednom dotazu a ke všem modelům (ArangoDB, © 2021). Díky spolupráci všech datových modelů, dokážou systémy využívající ArangoDB mnohem lepších výkonu než systémy, které využívají více NoSQL databázi. Systém je nabízen jako open-source a v komerční verzi. Byl napsán pomocí jazyka C++ a JavaScriptu. ArangoDB má oficiální podporu pro operační systémy Linux, macOS, Windows (ArangoDB System Properties, © 2021).

Jedná se o databázi, která umožňuje využívání více modelů. Jsou zde nižší náklady a větší flexibilita. Databáze se v čase vyvíjí, a pokud je potřeba snížit nebo zvýšit výkon, tak ArangoDB nabízí snadnou změnu systému. Dotazovací jazyk je velice intuitivní a nabízí bohaté možnosti využití. Nabízí velkou podporu API a poskytuje velmi dobře zpracovanou dokumentaci na webu (ArangoDB - Reviews, Pros & Cons | Companies using ArangoDB, © 2021).

Databáze nedosahuje takových rychlostí jako například Redis, který pracuje v paměti. Implementace ArangoDB není optimalizována pro velmi dlouhodobé nebo velmi objemné operace. Transakce byly navrženy pro konkrétní případy použití, a tak nemusí být vždy vhodné. Vnořené transakce nejsou podporovány a vyvolají chybu. Pokud je snaha během transakce vyvolat některou ze specifických operací, které nejsou dovoleny, dojde k vyvolání chyby a operace se neprovede. V případě, že během potvrzení transakce selže některý server, tak se na některých serverech mohou operace potvrdit a na jiných nikoliv. Operace se nevrátí do původního stavu a klient uvidí chybu (Limitations | Transactions | Manual | ArangoDB Documentation, © 2021).

ArangoDB podporuje programovací jazyky jako C#, C++, Clojure, Elixir, Go, Java, JavaScript, PHP, Python, R a Rust (ArangoDB System Properties, © 2021).

Databáze je například využívaná britskou mezinárodní bankovní a finanční společností Barclays, indickou technologickou společností Infosys, kanadským prodejcem softwaru OpenText, společnostmi Accenture, Agero, Altait a americkou firmou VMware (ArangoDB, © 2021).

2.6.2 Elasticsearch

V tomto případě se nejedná úplně o NoSQL databázi jako spíše o vyhledávací a analytický nástroj pro všechny typy dat, včetně textových, číselných, geoprostorových, strukturovaných a nestrukturovaných. Jako NoSQL databáze je označován hlavně pro svoji schopnost ukládat data ve formě dokumentů typu JSON. Dokáže data optimálně ukládat tak, aby byly dobře použitelná v reálném čase. Kromě vyhledávacího nástroje a NoSQL databáze je Elasticseach označován za „Spatial DBMS“, což znamená, že umožňuje zpracování geo dat. Databáze usnadňuje manipulaci s daty, definování, budování a manipulaci a aktualizování databáze (Kimpl, 2010). Vývoj probíhá v jazyce Java a systém je nabízen ve verzi open-source, která je značně omezena o mnoho užitečných funkcí. Další verze jsou Standart, Gold, Platinum a Enterprise, všechny tyto verze si lze zdarma vyzkoušet na jakémkoliv operačním systému obsahující Java virtual machine (Elasticsearch, © 2021)

Databáze nabízí koncept brány, který umožňuje vytvořit snadné zálohy. Obsahuje mnoho možností pro vyhledávání. Dokáže vyhledávat podle jednotlivých slov v textu, fulltextově, umí vyhledávat nezávisle na pravopisné chybě v textu. Všechna vyhledávání jsou prováděna v době, kdy uživatel zadává svůj požadavek. Pokouší se

předpovídat dle zatím zadaných parametrů, co se uživatel snaží vyhledat (Novoseltseva, 2018).

Jelikož se jedná spíše o vyhledávací nástroj, nenabízí takové možnosti uložení dat, jako jiné NoSQL databáze, které jsou zaměřeny především na uložení dat. Pokud je nucen zpracovávat například velké proudy dat, může docházet ke ztrátám těchto dat. Přestože se jedná o velice flexibilní a výkonný nástroj pro vyhledávání, není úplně jednoduchý k používání (JavaTpoint, © 2018).

Elasticsearch podporuje programovací jazyky jako .Net, Groovy, Java, JavaScript, Perl, PHP, Python, Ruby a spoustu dalších neoficiálně podporovaných jazyků jako například C++, Go, Haskell, Erlang a další (Elasticsearch System Properties, © 2021).

Systém ke svému provozu využívají společnosti jako je Adobe, Cisco, Shopify, eBay, Facebook, Uber, Docker, GitHub, SoundCloud, Orange a spoustu dalších firem (Elastic customer stories of all shapes and sizes, © 2021).

3 RELAČNÍ DATABÁZE

Tento typ databází byl vyvinut dříve než NoSQL databáze, které se využívají pro ukládání objemných dat. Databáze je založena na datovém schématu neboli modelu, který reprezentuje, jak budou data uložena. Tyto data jsou uložena ve formě tabulek, které mezi sebou mají různé vztahy. V relačních databázích má každý záznam v tabulce svoje identifikační číslo, pomocí kterého lze s daty manipulovat. Toto jednoznačné identifikační číslo se nazývá klíč. Každý řádek pak obsahuje hodnoty podle atributů tabulky, které mají definovaný datový typ (Oracle, © 2021).

Na počátku databází ukládaly aplikace data ve své vlastní jedinečné struktuře. Pokud chtěl někdo použít tyto data pro vývoj, musel se nejdříve seznámit s touto strukturou, aby věděl, jaká data vůbec obsahuje a předešel možným problémům. To bylo velice neefektivní, proto byly navrženy relační databáze, aby odstranili problémy, které jedinečné datové struktury způsobovaly. Relační databáze poskytovaly standardní způsob ukládání dat a dotazování na data, která se lépe vývojářům používala. Hlavní benefit tohoto modelu byly tabulky, které jsou dostatečně flexibilní a intuitivní na práci. Postupem času se začal používat dotazovací jazyk SQL, který umožňoval ještě efektivněji pracovat s daty (Oracle, © 2021).

Jednou z hlavních výhod relačních databází je fakt, že už tu jsou od 70. let. To z nich činí nejrozšířenější model pro ukládání dat. Jedná se o velice jednoduchý a výkonný model, který používají různé druhy organizací. Zajišťují integritu dat, to znamená, že data jsou konzistentní. Pro transakce využívá vlastnosti ACID. V dnešní době se lze setkat s relačními databázemi na většině internetových stránkách, ale nemusí se jednat pouze o internetové stránky (Oracle, © 2021).

3.1 Srovnání relačních databází s NoSQL databázemi

V této kapitole jsou představeny rozdíly mezi relačními databázemi a databázemi NoSQL. Holubová ve své knize uvádí srovnání, které je uvedeno níže. Rozdíly jsou zaměřeny na srovnání požadavků na zpracovávaná data. Tabulku lze chápat jako stručný seznam, který slouží jako obecný přehled. Tento přehled je pouze orientační a měl by být chápán s určitou rezervou, protože nemusí vždy platit.

Tabulka 1, Srovnání relačních databází s NoSQL databázemi (Holubová, 2015)

Relační databáze	**NoSQL databáze**
Integrita dat je zásadní.	Stačí, pokud je většina dat většinu času v pořádku.
Datový formát je konzistentní a dobře definovaný.	Datový formát nemusí být známý nebo konzistentní.
Předpokládáme dlouhodobé uložení dat.	Vzhledem k velkému množství dat často ukládáme pouze určité "časové okno" (např. poslední měsíc, poslední rok).
Aktualizace dat jsou časté.	"Write-once/read-many", tedy vložená data už typicky nejsou dále modifikována (nebo alespoň ne příliš často). Obvykle data neustále přibývají, aniž by byla modifikována. Již nepotřebné záznamy jsou pak smazány.
Předvídatelná (lineární) nárust velikosti dat.	Nepředvídatelný (exponenciální) nárůst dat.
Nástroje pro dotazování dat umožňují přístup i ne-programátorům.	Typicky pouze programátoři píší (implementují) zpracování dat.
Probíhají pravidelné zálohy dat.	Pro řešení výpadku je využívána replikace dat.
Přístup k datům zajišťuje jediný server.	Data jsou umístěna na více serverech, přistupujeme tedy ke clusteru uzlů.

4 TECHNOLOGIE BIGDATA

Tato kapitola obsahuje pohled na jedny z nejvíce používaných technologiích, které spolupracují s big daty. Budou zde popsány nástroje od společnosti Apache. Jedná se pouze o krátký výčet technologií z mnoha. Dále bude kapitola obsahovat srovnání vybraných technologií.

4.1 Apache Hadoop

Projekt, který vyvíjí společnost Apache, obsahuje sady nástrojů pro zpracování velkých objemu dat napříč clustery. Je navržen tak, aby byl schopen se rozšířit z jednoho serveru na tisíce dalších zařízení, které poskytnou svůj výpočetní výkon a úložný prostor. Hadoop se nespoléhá na vysokou dostupnost hardwaru, proto je navržen tak, aby detekoval a zpracovával chyby, které se objeví na aplikační vrstvě. To zapříčiňuje vysokou dostupnost služeb nad clusterem zařízení, kde každé může být náchylné k chybám (Apache Hadoop, © 2021).

4.1.1 HDFS

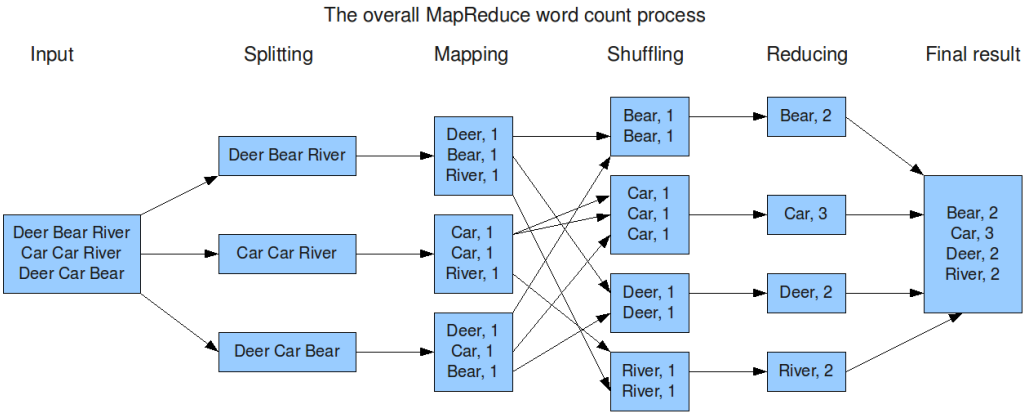
„The Hadoop Distributed File System“ (HDFS) je distribuovaný souborový systém, který byl navržen tak, aby z obyčejných levných zařízení byl schopen vybudovat dobře škálovatelný systém úložišť. Byl vyvinut pro velké úlohy, které potřebují škálovatelnost, flexibilitu a propustnost. U přijatých dat neřeší jejich formát ani velikost. Je schopen uchovat data o jakékoli velikosti. Nabízí integrovanou odolnost proti chybám. Ta je zajištěna automatickou replikací, která se stará, aby bylo dostupných více záloh našich dat. To znamená, že některé servery mohou selhat, ale naše data by měla být vždy dostupná (Cloudera, 2019).

4.1.2 MapReduce

Tato technologie byla představena poprvé firmou Google, která ji popsala jako „jednoduché a výkonné rozhraní, které umožňuje automatickou paralelizaci a distribuci výpočtů nad rozsáhlými daty, a k němu příslušející implementaci tohoto rozhraní, jež umožňuje dosáhnout vysoký výkon s využitím velkého clusteru běžně dostupných počítačů“. Velké množství projektů využívá právě řešení od Hadoopu, hlavně pro jejich rozsáhlou návaznost s ostatními projekty (Holubová, 2015).

Technologie je založena na funkcích „Map“ a „Reduce“. Přijatá vstupní data jsou zpracována na jednotlivé dvojice klíč a hodnota. Tento výsledný objekt je poté

redukován na základě již zmapovaných objektů (Holubová, 2015). MapReduce je odolný proti chybám, které mohou vznikat při zpracováním velkého množství dat (MapReduce Tutorial, 2020).



Obrázek 5, Chování funkce MapReduce (Lăpușan, © 2019)

4.1.3 YARN

YARN vznikl pro zlepšení celkového systému Hadoop. Na začátku se označoval jako „MapReduce 2“ nebo „NextGen MapReduce“. Starší verze příliš svazovala HDFS s MapReduce. V té době se musel MapReduce starat o plánování úloh, správu zdrojů a neumožňoval spouštění jiných aplikací než podporující MapReduce. Nová verze přinesla řešení, které odděluje správu a plánování od MapReduce, který se může plně věnovat zpracování dat. Ve zkratce je Hadoop YARN správce plánování a monitorování úloh. Objevují se zde pojmy jako „Resource Manager“ a „Node Manager“. První zmíněný pojem rozhoduje o přidělování jednotlivých prostředků mezi všemi aplikacemi v systému. „Node Manager“ se stará o monitorování samotného zařízení a využití jeho zdrojů, tyto data pak předává na „Resource Manager“ přesněji jemu plánovači. Plánovač se stará pouze o přidělování zdrojů. Úlohy přiděluje na základě své plánovací funkce, která přijímá data od jednotlivých zařízení. Odpovědnost za sledování správného provedení těchto úloh přebírá „ApplicationManager“, který je také součástí „Resource Manageru“. Má také za úkol monitorovat pokrok těchto úloh (Apache Hadoop YARN, 2020).

4.2 Apache Spark

Jedná se o analytický nástroj poskytovaný firmou Apache, která jej nabízí jako open-source, jak už má ve zvyku. Spark je velice rychlý analytický nástroj pro velké

objemy dat a strojové učení. Poskytuje kvalitní API rozhraní pro jazyky Java, Scala, Python, R (Databricks, 2013). Tato technologie je založena na technologie Hadoop MapReduce a rozšiřuje ji, aby byla schopna ji efektivně použít pro více výpočtů včetně interaktivních dotazů a zpracování streamů. Za účelem dosažení lepšího výkonu, využívá pro svoje výpočty clustery. Díky svému designu je schopen postarat se o velké množství úloh a snížit zátěž pro údržbu (Apache Spark - Introduction, © 2021).

Jedná se o jeden z největších otevřených projektů pro zpracování dat. Po jeho příchodu mnoho firem přešlo k jeho používání. Začali jej využívat firmy jako Netflix, Yahoo nebo eBay, kterým umožňuje zpracovat petabajty dat na clusterech o velikosti větší než 8000 uzlů. Mezi jeho vlastností patří snadnost použití. Obsahuje více než 100 operátorů pro transformaci dat a rozhraní, ty ulehčují manipulaci s polostrukturovanými daty. Dokáže pracovat až stokrát rychleji než Hadoop, díky využívání zpracování výpočtů v paměti a dalším optimalizacím. Je schopen pracovat ve velké rychlosti, i když jsou data uložena na disku a je držitel světového rekordu v rozsáhlém třídění na disku. Obsahuje knihovny, které podporují SQL dotazy, streamování dat, strojové učení a zpracování grafů. Tyto knihovny umožňují zvýšení efektivity při vývoji složitých pracovních postupů (What is Apache Spark?, © 2021).

4.2.1 Apache Spark Ecosystem

Tento ekosystém se skládá z „Spark SQL + DataFrames“, „Streaming“, „MLlib“ a „GraphX“. Tyto komponenty pohání „Apache Spark Core API“, které se skládá z jazyka R, SQL, Python, Scala a Java.

„Apache Spark SQL“ je komponenta podporující zpracování strukturovaných dat pomoci SQL dotazů. Díky těmto dotazům získá Spark více informací o struktuře dat a výpočtech. Na základě těchto informací lze provést další optimalizace. SQL hlavně slouží pro přístup k datům, a to jak strukturovaným či polostrukturovaným (Apache Spark Ecosystem – Complete Spark Components Guide, 2018).

Jednou pro nás z nejdůležitějších komponent je „Apache Spark Streaming“. Ta slouží hlavně pro zpracování a analýzu dávkových dat, ale i datových toků v reálném čase. Spark Streaming umožňuje běh interaktivním a analytickým aplikacím, napříč historickými a datovými proudy. I přes zásah dalších aplikací si zachovává snadnost použití a odolnost proti chybám. Nabízí snadnou integraci s daty z HDFS, Apache Kafka nebo Twitteru (Databricks, 2020).

Jeho provoz je rozdělen do tří fází. Shromáždění, zpracování a uložení. Shromažďovat data lze ze základních zdrojů, jako jsou souborové systémy, připojené sokety nebo z pokročilých zdrojů. Jedná se o již zmiňovaný Apache Kafka, HDFS nebo jiné zdroje, které jsou k dispozici skrze speciální třídy. Druhou fází je zpracování dat. U dat, která jsou přijata, dochází ke zpracování pomoci funkcí, které jsou založeny na složitých algoritmech. Mezi tyto funkce patří například mapování, filtrování, redukování, aktualizování a jiné transformační funkce (Apache Spark Streaming Transformation Operations, © 2021). Zpracovaná data jsou funkcemi vrácena v podobě „DStream“, což Spark označuje jako nepřetržitý proud dat. V poslední fázi jsou tyto zpracovaná data potřeba uložit, může se jednat o uložení do souborového systému, databáze nebo odeslání na řídící panely, které promítají výsledky v reálném čase (Apache Spark Ecosystem – Complete Spark Components Guide, 2018).

„Apache Spark MLlib“ neboli „Machine Learning Library“ v překladu knihovna strojového učení. Důvodem k zavedení této knihovny bylo učinit strojové učení škálovatelné a lehce použitelné. Knihovna obsahuje různé algoritmy strojového učení od složitých po primitivní. Toto strojové učení přidává systému možnost reagovat na nová data a podle toho se přizpůsobit. Může se jednat ať o zefektivnění činnosti, tak změnu podmínek vstupních parametrů (Canadata, 2020).

Poslední zmíněnou komponentou je „GraphX“. Jedná se o analytický nástroj, který zpracovává síťové grafy a je schopen je ukládat. Nabízí procházení grafů, vyhledávání cest, hledání cest a další možnosti. Obsahuje i vlastní optimalizace pro reprezentaci hran a vrcholů (Apache Spark Ecosystem – Complete Spark Components Guide, 2018).

4.3 Apache Flink

Flink je nástroj, který slouží pro zpracování omezených a neomezených streamů dat. Omezený stream má označený, kde je začátek a konec. Tento stream bývá nazýván jako dávkový. Neomezený stream dat, má definovaný začátek, ale nemá definovaný konec. U těchto dat je zapotřebí, aby docházelo ke zpracování průběžně. Některá data musejí být zpracována v určitém pořadí, takže je nutnost vyčkat na kompletní data. Tento nástroj se používá pro zpracování dat ve velkém měřítku a odesílání průběžných analýz na požadovanou streamovací aplikaci. Je navržen tak, aby byl schopný pracovat na všech zařízeních a prováděl výpočty rychlostí paměti. Nabízí komunikaci, odolnost

proti chybám a distribuci dat napříč clustery pro distribuované výpočty datových toků (What is Apache Flink?, © 2021). Jako jediný open-source nástroj poskytuje propustnost více než milionu událostí na obyčejných clusterech a jeho odezva je v řádu milisekund.

4.3.1 Apache Flink Ecosystem

Apache Flink Ecosystem se skládá z úložiště, vývoje neboli správy zdrojů, jádra, API a knihoven.

Flink je pouze nástroj pro výpočty, proto neobsahuje vlastní úložný systém. Data, která přijímá, jsou od jiných zdrojů a ta, která zapisuje, odesílá na jiné místo. Pro tyto operace může využívat například HDFS, HBase, MongoDB, relační databáze, lokální souborový systém nebo jiné systémy (Apache Flink Ecosystem Components Tutorial | Learn Flink, 2021).

Vývoj může probíhat ve třech variantách. Flink lze spustit na jednom zařízení jako JVM. Jedná se o virtuální stroj, který umožňuje spouštění programů Java a jiných programů, které jsou kompilovány do bajtkódu (JVM | Java Virtual Machine, © 2018). Další a nejtypičtější možností, která se používá pro velká data, je uložení na cluster. Zde využívá vlastního správce „Standalone“, která se stará o správu zdrojů. Populární možností je využití technologie Hadoop YARN nebo méně známý Apache Mesos. Poslední možností pro správu zdrojů je použití cloudových služeb od Googlu, Amazonu nebo jiných poskytovatelů (Apache Flink Ecosystem Components Tutorial | Learn Flink, 2021).

Třetí částí ekosystému je jádro. To se stará o distribuované zpracování úloh, spolehlivost, zajištění odolnosti proti chybám a jiné vlastnosti (Apache Flink Ecosystem Components Tutorial | Learn Flink, 2021).

Poslední částí celého ekosystému jsou API a knihovny, které rozšiřují možnosti samotného Apache Flink. Mezi tyto části patří „DataSet API“, která zpracovává data ze zdrojů, které nejsou typické jako zdroj dat. Další je „DataStream API“, která se stará o zpracování transformací pro proud dat. Jinými známými jsou například „Table API“, „Gelly“, „FlinkML – Machine Learning for Flink“, „FlinkCEP – Complex event processing for Flink“ (Apache Flink Ecosystem Components Tutorial | Learn Flink, 2021).

4.4 Srovnání technologií Hadoop, Spark a Flink

Pro shrnutí technologií big dat obsahuje tato kapitola přehled na obecné úrovni již zmíněných technologií Hadoop, Spark a Flink. Jedná se o technologie, které spravuje společnost Apache. Není proto divu, že jsou nabízeny jako open-source a jsou velice populární, což vede k tomu, že se jedná o jedny z nejvíce používaných technologií v oblasti big dat na dnešním trhu.

Tabulka 2, Srovnání technologií Hadoop, Spark a Flink, přepracováno podle (Hadoop vs Spark vs Flink Big Data Frameworks Comparison, © 2021)

	Hadoop	**Spark**	**Flink**
Zpracování dat	Dávkové zpracování.	Dávkové zpracování a proudy dat.	Dávkové zpracování a proudy dat.
Výkon	Pomalejší než Spark a Flink.	Pomalejší než Flink kvůli mikro-dávkovému zpracování.	Nejrychlejší
Správa paměti	Lze konfigurovat staticky nebo dynamicky.	Nejnovější verze přešla na automatizovanou správu paměti.	Automatická správa paměti, oddělena od "garbage collectoru" javy.
Iterativní zpracování	Nepodporuje	Iteruje data v dávkách. Iterace musí být předem naplánována a provedena samostatně.	Iteruje data pomocí streamovací architektury. Je schopen zpracovávat jen ty data, kde nastala změna.
Podporované jazyky	Java, C, C++, Ruby, Groovy, Perl, Python	Java, Scala, Python, R	Java, Scala, Python, R
Latence	Vyšší latence než Spark a Flink.	Díky ukládání dat do paměti, dosahuje menší latence než Hadoop.	Nízká latence a vysoká propustnost.
Vizualizace	Lze připojit nástroj zoomdata, který dobře spolupracuje.	Nabízí webové rozhraní pro vizualizaci.	Nabízí webové rozhraní pro vizualizaci.
Bezpečnost	Autentizační protokol Kerberos a LDAP	Pouze pomocí hesla, nebo lze zajistit bezpečnost podle platformy na které běží.	Kerberos a TLS/SSL
Náklady	Nízká cena, odpadá potřeba paměti RAM.	Se zvětšováním provozu, je potřeba více RAM. Vyšší cena.	Potřebuje pro provoz RAM. Cena se zvyšuje podle potřeby.

Obtížnost použití	Pro každou operaci je potřeba kód, to stěžuje použití.	Obsahuje operátory na vysoké úrovní, kteří velice ulehčují práci.	Obsahuje operátory na vysoké úrovní, kteří velice ulehčují práci.
Interaktivní režim	Neobsahuje interaktivní režim.	Interaktivní shell, který nás naučí vytěžit ze Sparku maximum.	Obsahuje interaktivní Scala Shell.
Analýzy v reálném čase	Nepodporuje, pouze dávkové zpracování.	Lze zpracovávat data v reálném čase.	Byl navržen hlavně pro analýzu dat v reálném čase.
Plánovač	Existují dva plánovače. Plánovač kapacity a spravedlivý plánovač.	Spark funguje podle vlastního plánovače.	Může se použít YARN plánovač, ale má i svůj vlastní.
SQL podpora	SQL spouštět lze pomocí Apache Hive.	Spark SQL	Table API. Podobný právě SQL.
Ukládání do mezipaměti	Nepodporuje	Podporuje	Podporuje
HW požadavky	Nejobyčejnější HW	Středně až vysoce kvalitní HW	Středně až vysoce kvalitní HW
Strojové učení	Je potřeba použít jiný nástroj, například Apache Mahout.	Obsahuje vlastní MLlib.	Obsahuje FlinkML.

4.5 Apache Kafka

Apache Kafka je nástroj, který má sloužit pro optimalizaci zápisu. Pojmenován byl po Franzu Kafkovi, jakožto tvůrcova oblíbeném spisovateli (Tovshteyn, 2018). Nástroj je poskytovaný jako open-source, který má za úkol zpracování streamů. Hlavní myšlenkou je, aby poskytoval vysokou průchodnost a nízké zpoždění datových toků v reálném čase. Je schopen zprostředkovat čtení a zápis stovek megabajtů dat pro tisíce klientů za vteřinu (Apache Project Information, © 2020).

Platforma ukládá zprávy, které jsou přijímány od různých producentů. Tyto data mohou být rozdělena do jednotlivých „partition“, na základě jejich typu. V každé „partition“ se data řadí podle pozice, indexují se a jsou uložena pod společným časovým razítkem. Poté přichází na řadu konzumenti, kteří mají o tyto data zájem. Může se jednat například o aplikaci, která pomocí API přistupuje k těmto datům. Tyto data může konzumovat a s jejich pomocí produkovat nová. Kafka rozděluje typy dat do dvou kategorií, kterými jsou běžné a kompaktní. Běžné mají nastavěný retenční čas nebo prostorové omezení. Retenční čas omezuje životnost dat na určitou dobu

a prostorové omezení uchovává data, dokud je místo v „partition“, a poté má právo je smazat. U kompaktních nelze nastavit retenční čas ani prostorové omezení. Využívají toho, že nové zprávy nahradí starší, pokud mají stejný klíč. Nikdy nedojde ke smazání nejnovější zprávy.

Tato technologie je odolná vůči chybám a obsahuje vlastnost, která umožňuje vypořádat se s chybou v rámci jednoho stroje nebo clusteru. Vzhledem k tomu, že jsou data ukládána do různých „partition“ může docházet k duplikaci dat. Kvůli tomu jsme nuceni pořídit větší úložný prostor (Apache Kafka, © 2021).

5 OPEN DATA

Tato kapitola se zabývá otevřenými daty tzv. open daty. Mnoho zdrojů se ztotožňuje s definicí, kterou uvádí „Open Knowledge Foundation“. Tato organizace shrnula pojem „Open definition“ jako „Znalosti jsou otevřené, pokud k nim může kdokoliv volně přistupovat, používat je, upravovat a sdílet je – za podmínek, které nejvíce zabezpečí zachování původu, tedy vlastnictví, a otevřenosti.“. Spojení open data lze chápat jako data, která může kdokoli volně používat a dále distribuovat, a to lze i pro komerční využití. Open data by měla splňovat tyto podmínky. Jsou zveřejněna na internetu. Snadno dostupná a přístupná, jsou k dispozici v plné formě jako celek a ve vhodném formátu k úpravě. V případě požadavku na další šíření dat, uživatelé, kteří by je mohli chtít použít, by měli mít stejná práva (Hrabinova, 2018).

5.1 Kde lze open data hledat

V dnešní době si na internetu lze vyhledat snad vše, ale ne vždy jsou tyto informace pravdivé nebo dostatečně důvěryhodné. Existují společnosti, jedinci nebo dokonce i státy, nabízející veřejně svá data, která mají dostupná. Může se jednat o data regionů či měst, které nabízejí data o dopravě, bydlení, architektuře nebo obyvatelstvu. Dále to mohou být data o veřejné správě, financích, vědě, technice, epidemii a spousty dalších dat, které mají k dispozici. Mezi známé země, které nabízejí svá data, patří například Austrálie, Brazílie, Česká republika, Indie, Itálie, Spojené státy americké, Velká Británie a spousta dalších států, které se snaží svá data poskytnout formou open dat. Mezi města pak patří také Buenos Aires, Chicago, Londýn, San Francisco, Vídeň nebo Vancouver. Open data nabízí instituce jako Světová banka, Světová zdravotnická instituce nebo Evropská unie.

Nejčastěji jsou data dohledatelná pod doménovým jménem „data“, které se nejčastěji nachází na třetím nebo čtvrtém řádu domény. Česká republika tyto data nabízí pod adresou data.gov.cz, město Buenos Aires pak pod data.buenosaires.gob.ar a Světová banka pod adresou data.worldbank.org. Existují, zpravidla, i jiné domény, které obsahují open data, nejčastěji pak obsahují v doméně některé spojení slov „open“ a „data“ (Open Data Essentials, 2020).

5.2 Využití open dat

Aby toto velké množství otevřených dat mělo nějaký užitek, musí je někdo využívat. Vývojáři je využívají pro nové systémy, služby nebo jiné produkty. Další možností je

zakomponovat tyto data do již existujících projektů. Tyto možnosti mají většinou za následek větší poptávku po otevřených datech, čímž dochází k jejich zkvalitnění a nárůstu takových datových sad. Díky tomuto cyklu různé instituce uvolňují větší množství dat, která mohou být zpřístupněna širší veřejnosti.

Data slouží pro zlepšení činnosti vlád, kdy jim pomáhají s řešením korupcí, zlepšování služeb pro obyvatele nebo lepšího rozdělování financí. Dále mají velký dopad pro občany měst nebo států, kde tito lidé mohou sledovat, jak se stát nebo město vyvíjí a případně vystoupit proti. Mají možnost sledovat také jiné informace, jako rozdělení pozemků, parkovišť, provoz, finance a mnoho dalších informací, které jsou poskytovány. Veliký vliv mají data při budování veřejných projektů.

Často jsou data využívaná na různých soutěžích nebo tzv. „hackathonech“. Jedná se o soutěže, které většinou mají vzdělávací účel, kde se pracuje na zadaném softwarovém projektu. Tyto soutěže už vyprodukovaly spousty projektů, které se zabývaly životním prostředí, financemi, zdravím, vzděláním a jinými odvětvími (Open Data Essentials, 2020).

5.3 Životní cyklus open dat

Často je těžké přesně určit, jak bude životní cyklus nejen open dat vypadat. Tento problém vzniká důsledkem rozdílných potřeb systémů, které s daty pracují. Sborník příspěvků z mezinárodní vědecké konference MMK 2015 uvádí, že by měl projít osmi fázemi. Nejdříve je nutné mít zdroj dat, se kterým je požadováno pracovat. Následuje získání těchto dat, jejich příprava, ukládání a archivace, zpracování a analýzy, vizualizace, zveřejnění, a nakonec rozhodovací procesy, které mají poskytnout informace o rozhodnutí nebo závěru podle původního záměru (Lněnička, 2015).

Obrázek 6, Životní cyklus open (big) dat (Lněnička, Komárková, 2015)

6 VIZUALIZAČNÍ NÁSTROJE

Další důležitou součástí ekosystému jsou vizualizační nástroje. V dnešní době již existují programy pro analýzu dat, které v sobě již obsahují zakomponované vizualizační nástroje. V této kapitole jsou některé tyto nástroje představeny a je vysvětleno, proč je dobré data zobrazovat tedy vizualizovat.

6.1 Vizualizace dat

Data, která se vykytují v databázích, jsou velice rozsáhlá a nemusí spolu všechna souviset. Jednou z nejsnadnějších a nejlepších možností, jak rychle porozumět velkému množství dat, je jejich vizualizace. Pod touto vizualizací si lze představit transformaci dat do podoby křivky, sloupcových grafů, koláčových grafů, síťových diagramů, teplotních map, stromových struktur a mnoho dalších druhů zobrazení. Tato vizualizace neslouží pouze k lepší představě, ale může také pomoci při analýze. Lze si například zobrazit, jak se data vyvíjela v čase nebo sledovat vztahy mezi nimi. Nejmodernější systémy dokáží vymodelovat vhodný typ vizualizace podle zadaných parametrů, které jsme do systému zadali (Holubová, 2015).

Grafické vizualizace dat, poskytují jiné úhly pohledu, a díky tomu lze v datech najít vzorce, které nelze získat ze surových dat. To je při analýze či rozhodování naprosto klíčové. Tyto vizualizace mohou odhalit vznikající trendy, kterých by si nikdo nevšiml. Velikou výhodou je, že pro pochopení těchto dat, nemusí být člověk žádný specializovaný odborník. Přesto je důležité dobře tyto data chápat a číst v nich, aby byla k využití (Co je vizualizace dat, 2014).

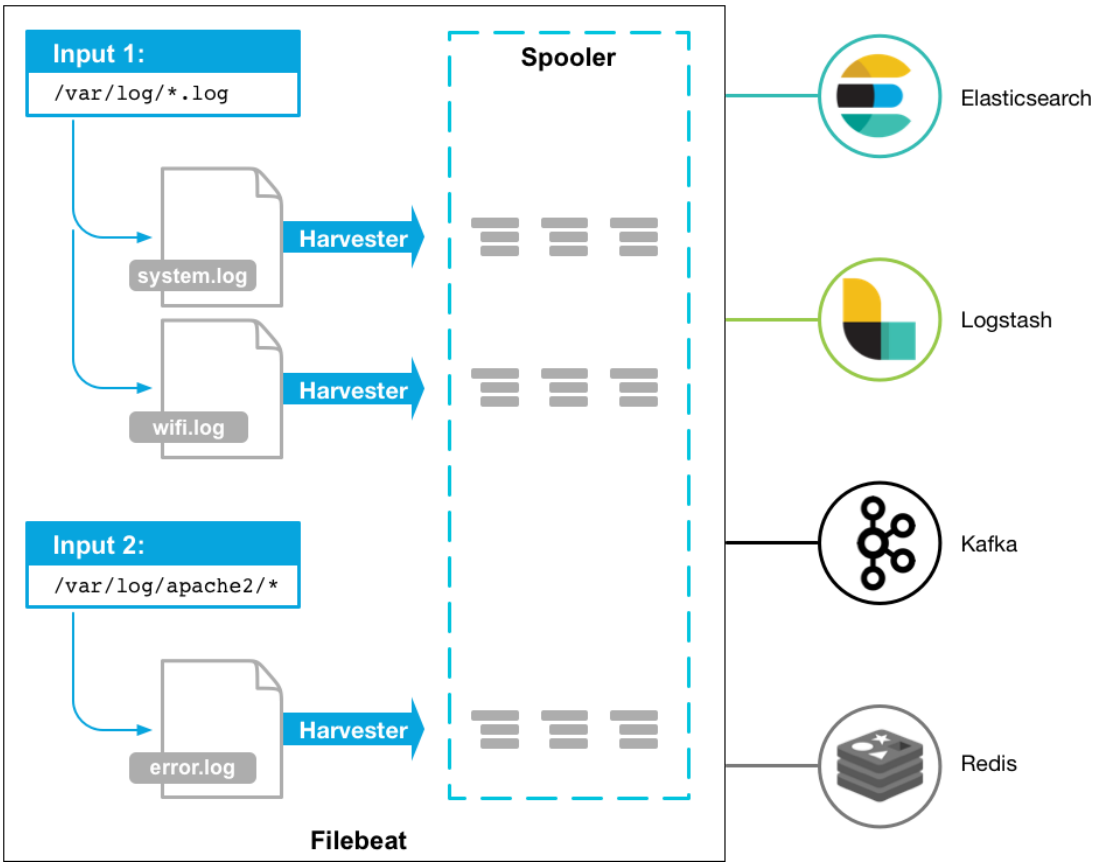
6.2 Elastic Stack

Projekt je od společnosti Elastic NV. Dříve pod názvem ELK Stack, kdy se jednalo o tři open-source produkty Elasticseach, Logstash a Kibana. Postupem času k nim byla přidána rodina produktů Beats. Tyto produkty slouží jako kompletní podpora pro vizualizaci a analýzu dat. Dokáží spolehlivě a bezpečně převzít data z jakéhokoli zdroje a formátu, která následně zpracují, analyzují a poskytnou jejich vizualizaci. Elasticsearch byl již zmíněn v kapitole NoSQL. Pro připomenutí se jedná o vyhledávací, analytický nástroj pro ukládání dat. Další produkty jsou vysvětleny v následujících podkapitolách včetně nástroje Filebeat, který patří do rodiny Beats (ELK Stack: Elasticsearch, Logstash, Kibana, © 2021).

6.3 Filebeat

Jak již bylo řečeno, nástroj Filebeat patří do rodiny Beats. Jednalo se o jeden z nejvíce používaných nástrojů ve spojení s ELK Stackem. Proto se produkty Beats staly součástí Elastic Stacku.

Filebeat je označován jako odlehčený přepravce dat. Na rozdíl od nástroje Logstash nezanechává takovou digitální stopu a nevyužívá tolik systémových prostředků. Slouží k odesílání logů neboli žurnálů. Log je soubor, který obsahuje záznamy činností a záznamy běhu zařízení. Filebeat je instalován jako agent na server. Monitoruje zde soubory s logy nebo místo, které jsme mu zadali pro sledování. Data, která získá, přeposílá na Logstash, Elasticsearch nebo je poskytuje jiným nástrojům (Filebeat: Lightweight Log Analysis & Elasticsearch, © 2021).



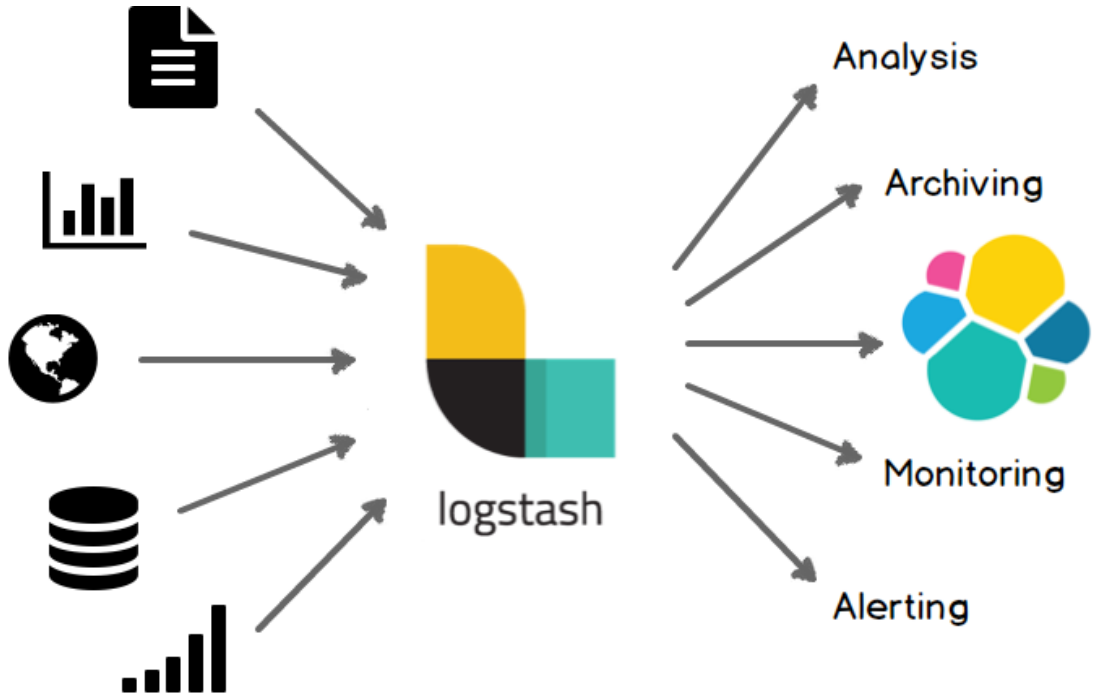
Obrázek 7, Činnost Filebeat (Filebeat overview, © 2021)

6.4 Logstash

Předtím než byl přidán Filebeat, používal se Logstash pro sběr logů. Na rozdíl od nástroje Filebeat nabízí Logstash větší podporu možností, využívá více systémových prostředků, zanechává větší digitální stopu a pracuje na straně serveru. Nabízí možnosti využití pipeline v reálném čase. To má za následek zvýšení výkonu, díky

zpracovávání více instrukcí najednou. Umožňuje data sjednotit z různých zdrojů a odeslat je jako celek na stanovené místo. Tyto data poskytne v takové formě, aby byly dobře použitelné pro naše vizualizace, analýzy, sledování nebo jiné možnosti.

Podporuje více než 200 pluginů a možnost vytvářet také naše vlastní. Zachycuje různé typy logů, jako například od Apache, log4j pro Javu, syslog, síťové logy, firewall logy a jiné. Je schopen pracovat i s metrikami. Dobře spolupracuje právě s nástrojem Filebeat, který mu logy předává (Logstash Introduction, © 2021).



Obrázek 7, Logstash distribuce a příjem dat (Logstash Introduction, © 2021)

6.5 Kibana

Na rozdíl od předchozích nástrojů Kibana poskytuje vizualizaci dat. Obsahuje nástroje pro procházení logů, analytiku časových řad a monitorování aplikací. Pracuje na straně klienta ve webovém prohlížeči. Uživatelům je nabízena spousta výkonných funkcí, které jsou lehce použitelné. Data umožňuje vykreslit v podobě grafů, histogramů, teplených map, spojnicových grafů a jiných grafů (What is Kibana?, © 2021).

Nerozlišuje, o jaký typ dat se jedná, může to být strukturovaný nebo nestrukturovaný text, číselná data, časové řady, geoprostorová data, protokoly, metriky, bezpečností události. Kibana je navržena tak, aby využívala Elasticseach jako datové úložiště (Kibana—your window into Elastic, © 2021).

Vizualizace a analýza je velice jednoduchá. Stačí použít data, která už máme dostupná nebo pomocí funkce „drag-and-drop“ vložit nová. Poté lze data prozkoumat pomocí funkce „Discover“. Tato funkce umožňuje hledat statistiky a vztahy. Podle vyfiltrovaných požadavků poskytne pouze data, která jsou požadovaná. Další krok je vizualizace těchto dat. Tyto data si lze vykreslit na komponentu „Canvas“ nebo si je například vyexportovat do formátu PDF. Data si lze přenést na mapu, kde lze sledovat, odkud data pochází a sledovat tak spojení mezi různými městy, zeměmi nebo jinými místy (Kibana—your window into Elastic, © 2021).

Kibana ze svého uživatelského rozhraní umožňuje spravovat také indexy a clustery. Máme možnost definovat životní cykly indexů, replikovat indexy ze vzdálených clusterů na lokální (Kibana—your window into Elastic, © 2021).

Dále nabízí také bohaté možnosti upozornění, například když dojde k posunu klíčových ukazatelů výkonnosti. Jedná se o změny systémových prostředků, může se jednat o nedostatek paměti místa na disku nebo nedostatek výkonu. Lze nastavit také upozornění na vysoký počet požadavků na službu nebo velký počet pokusů o přihlášení. Lze si nastavit triggery, což jsou spouštěče událostí, které například pošlou email, když se nějaké upozornění aktivuje. Kibana nabízí spousty dalších možností a rozšíření jako celý Elastic Stack (Kibana—your window into Elastic, © 2021).

7 PRAKTICKÁ ČÁST

V praktické části dojde k hlubšímu představení nástroje Kibana, který byl popsán v teoretické části na konkrétní ukázce pracující s reálnými daty. se bude zabývat funkcemi Discover, Maps, Visualize Library, Canvas a Dashboard. Výstupem praktické části bude vizualizace dat, která slouží pro lepší pochopení datové sady a seznámení se základními funkcemi Kibany. Programy, které jsou v praktické části použity, jsou Kibana, Elasticsearch, Logstash. Všechny tyto nástroje poběží lokálně přes příkazový řádek a jsou dostupné na webu https://www.elastic.co/.

7.1 Popis datové sady

Data, na kterých budou prováděny praktické ukázky se týkají pandemie COVID-19. Tato data jsou veřejně dostupná a jsou zpravována společností Our World in Data. Po dobu pandemie jsou denně aktualizována. Naše data jsou v rozsahu od 1. 1. 2020 do 26. 4. 2021. Jedná se o CSV soubor, tedy o tabulku, která obsahuje 84 296 řádků dat. Každý řádek těchto dat obsahuje informaci, například o který kontinent se jedná, jméno země, datum, celkový počet nakažených osob, nové případy, počet mrtvých, informace o počtu hospitalizovaných osob nebo o počtu testovaných. Celkový počet sloupců je 58. Jedná se tedy o soubor o rozsahu 58 x 84 296, který má velikost 20,9 MB.

Soubor obsahuje každý záznam dat v prvním sloupci. Na prvním řádku se vyskytuje název sloupců, kdy každý název je oddělen čárkou. Samotná data, jsou na dalších řádcích také oddělena čárkami. Pokud hodnota pro daný sloupec neexistuje, je psána další čárka jako ukončení sloupce.

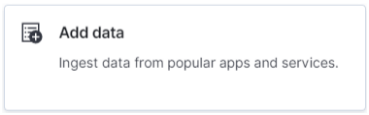
	A
1	iso_code,continent,location,date,total_cases,new_cases,new_cases_smoothed,total_deaths,new_deaths,new_deaths_smoothed,total_cases_per_million,new_cases_per_million,
2	AFG,Asia,Afghanistan,2020-02-24,1.0,1.0,,,,,0.026,0.026,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,8.33,38928341.0,54.422,18.6,2.581,1.337,1803.987,,597.029,9.59,,,37.746,0.5,64.83,0.511
3	AFG,Asia,Afghanistan,2020-02-25,1.0,0.0,,,,,0.026,0.0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,8.33,38928341.0,54.422,18.6,2.581,1.337,1803.987,,597.029,9.59,,,37.746,0.5,64.83,0.511
4	AFG,Asia,Afghanistan,2020-02-26,1.0,0.0,,,,,0.026,0.0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,8.33,38928341.0,54.422,18.6,2.581,1.337,1803.987,,597.029,9.59,,,37.746,0.5,64.83,0.511
5	AFG,Asia,Afghanistan,2020-02-27,1.0,0.0,,,,,0.026,0.0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,8.33,38928341.0,54.422,18.6,2.581,1.337,1803.987,,597.029,9.59,,,37.746,0.5,64.83,0.511
6	AFG,Asia,Afghanistan,2020-02-28,1.0,0.0,,,,,0.026,0.0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,8.33,38928341.0,54.422,18.6,2.581,1.337,1803.987,,597.029,9.59,,,37.746,0.5,64.83,0.511
7	AFG,Asia,Afghanistan,2020-02-29,1.0,0.0,0.143,,,0.0,0.026,0.0,0.004,,,0.0,,,,,,,,,,,,,,,,,,,,,,,,,,,8.33,38928341.0,54.422,18.6,2.581,1.337,1803.987,,597.029,9.59,,,37.746,0.5,64.83,0.511
8	AFG,Asia,Afghanistan,2020-03-01,1.0,0.0,0.143,,,0.0,0.026,0.0,0.004,,,0.0,,,,,,,,,,,,,,,,,,,,,,,,,,,27.78,38928341.0,54.422,18.6,2.581,1.337,1803.987,,597.029,9.59,,,37.746,0.5,64.83,0.511
9	AFG,Asia,Afghanistan,2020-03-02,1.0,0.0,0.0,,,0.0,0.026,0.0,0.0,,,0.0,,,,,,,,,,,,,,,,,,,,,,,,,,,27.78,38928341.0,54.422,18.6,2.581,1.337,1803.987,,597.029,9.59,,,37.746,0.5,64.83,0.511
10	AFG,Asia,Afghanistan,2020-03-03,2.0,1.0,0.143,,,0.0,0.051,0.026,0.004,,,0.0,,,,,,,,,,,,,,,,,,,,,,,,,,,27.78,38928341.0,54.422,18.6,2.581,1.337,1803.987,,597.029,9.59,,,37.746,0.5,64.83,0.511

Obrázek 8, Prvních deset řádků dat souboru covid-data.csv (vlastní zpracování)

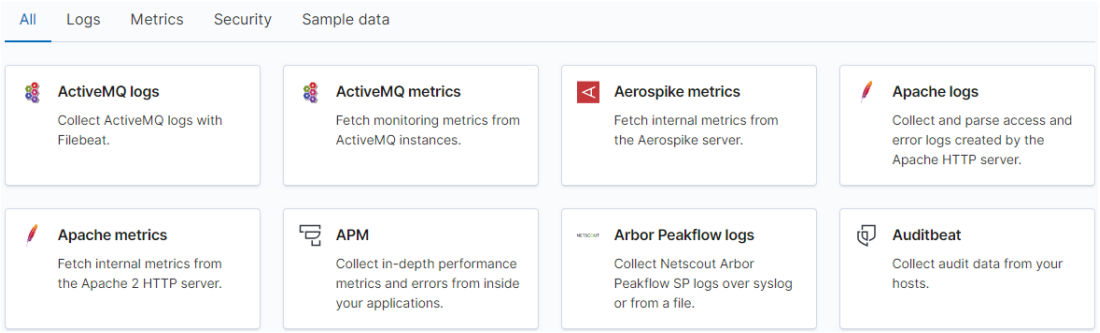
7.2 Import dat do Kibany

Po spuštění Kibany se objeví domovská stránka, kde se nachází možnost *Add data*. Lze zvolit, zda je požadována instalace dalších aplikací nebo služeb, která

zpracovávají generovaná data, jako jsou logy, metriky, data týkající se bezpečnosti nebo lze použít ukázková data.

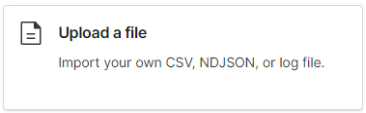


Obrázek 9, Tlačítko pro import dat skrze aplikaci nebo službu



Obrázek 10, Ukázka možných nástrojů pro import dat

Další možností je nahrát přímo soubor dat, jako CSV, NDJSON nebo log.



Obrázek 11, Tlačítko pro import dat z vlastního souboru (vlastní zpracování)

7.2.1 Logstash

První možností pro import dat je nahrání pomocí nástroje Logstash. Je zapotřebí mít data, která budou nahrána a konfigurační soubor, díky kterému lze nahrát data do indexu Elasticsearch a přistupovat k nim v Kibaně.

Pokud zařízení obsahuje všechny potřebné programy, tak je nutné přejít do složky, kde se Logstash nachází. Zde se vytvoří v podsložce *bin* konfigurační soubor ve tvaru nazev_souboru.conf. Pro tuto ukázku bude obsahovat tři hlavní moduly *input, filter* a *output*. Tyto moduly mohou obsahovat více bloků pro nastavení. Bloky se provádí v pořadí, v jakém jsou zapsány v souboru.

Input modul slouží k načtení specifického zdroje dat událostí. Může se jednat o data z mnoha nástrojů. Zde je zvolen modul *file*, který načítá proud událostí ze souboru. Tento modul obsahuje další položky jako *path, start_position* a *sincedb_path*. Kdy pro *path* se jedná o cestu k souboru s daty a *start_position* značí, odkud se mají data začít

číst. Třetí *sincedb_path* umožňuje zaznamenávat aktuální pozici do samostatného souboru, a díky tomu lze logstash ukončit a znovu spustit tam, kde skončil bez ztráty již přidaných řádků (File input plugin, 2021).

Filter zprostředkovává zpracování události. Tyto filtry často používají podmínky, které se aplikují při určité události. Ukázkový soubor obsahuje *plugin csv* a *mutate*. *Csv* slouží pro parsování dat, která jsou oddělena oddělovačem. Oddělovač je specifikován klíčovým slovem *separator*. Druhou hodnotou je možnost *columns*, do které je nutné zadat názvy jednotlivých sloupců (Csv filter plugin, 2021). Další plugin, který je použit se nazývá *mutate*, který umožňuje provádět změny polí. Ukázkový soubor obsahuje možnost *convert*, kdy obsahuje jméno sloupce a datový typ, který je požadován, aby mu byl přidělen. Tento převod se musí provést pro všechny sloupce, jinak se načtou jako datový typ string (Mutate filter plugin, 2021).

Output je poslední částí v procesu zpracování souboru. Odesílá data událostí do určitého cíle. Je použit *plugin Elasticsearch* a *stdout. Elasticseach*, který obsahuje hodnotu *hosts* odkazující na adresu, kde běží onen Elasticsearch a hodnotu *index*, díky které lze přistoupit k datům v Kibaně (Elasticsearch output plugin, 2021). Plugin *stdout* slouží pro vypsání dat na standartní výstup. V ukázkovém případě se vypíše v běžícím shellu. Hodnota *rubydebug* je defaultní hodnotou, kdy se použije knihovna *ruby awesome_print* (Stdout output plugin, 2021).

```
input{
    file{
        path => "D:/ELK Stack/covid-data.csv"
        start_position => "beginning"
        sincedb_path => "NULL"
    }
}
filter{
    csv{
        separator => ","
        columns => ["iso_code","continent","location","date","total_cases","new_cases",
    }

    mutate{
        convert => {
            "total_cases" => "float"
        }
    }
}
output {
    elasticsearch {
        hosts => ["localhost:9200"]
        index => "covidfloat"
    }
    stdout {
        codec => rubydebug
    }
}
```

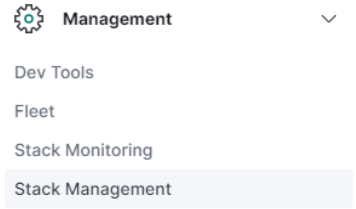
Obrázek 12, Ukázka souboru logstash.conf (vlastní zpracování)

Po vytvoření konfiguračního souboru lze spustit příkaz ve tvaru *„logstash -f název_konfiguračního_souboru.conf“*. Příkaz načte konfigurační soubor z určitého adresáře, který se postará o načtení dat. Pokud existuje více souborů pro načtení, tak budou načteny v lexikografickém pořadí.

```
D:\ELK Stack\logstash-7.12.1\bin>logstash -f logstash.conf
```

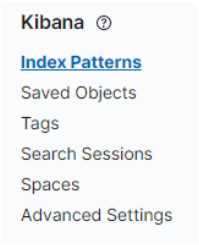
Obrázek 13, Příkaz pro načtení souboru do Elasticsearch (vlastní zpracování)

Jakmile v CMD skončí načtení všech dat lze přestoupit do Kibany. V levém horním rohu je nutné provést rozbalení menu. Ve spodní části je k nalezení položka *Management*. Po kliknutí se rozbalí menu, kde se zvolí *Stack Management*.



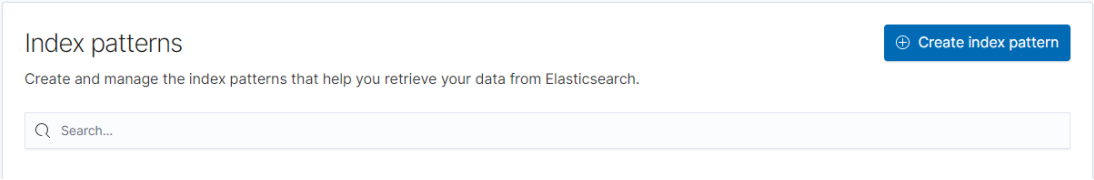
Obrázek 14, Kibana menu Management

Tím dojde k přesměrování na stránku, která zobrazí nastavení. Kompletní menu *Stack Management* je v levé části obrazovky. Pro účely ukázky nás zajímá *Kibana* a podpoložka *Index Patterns*.



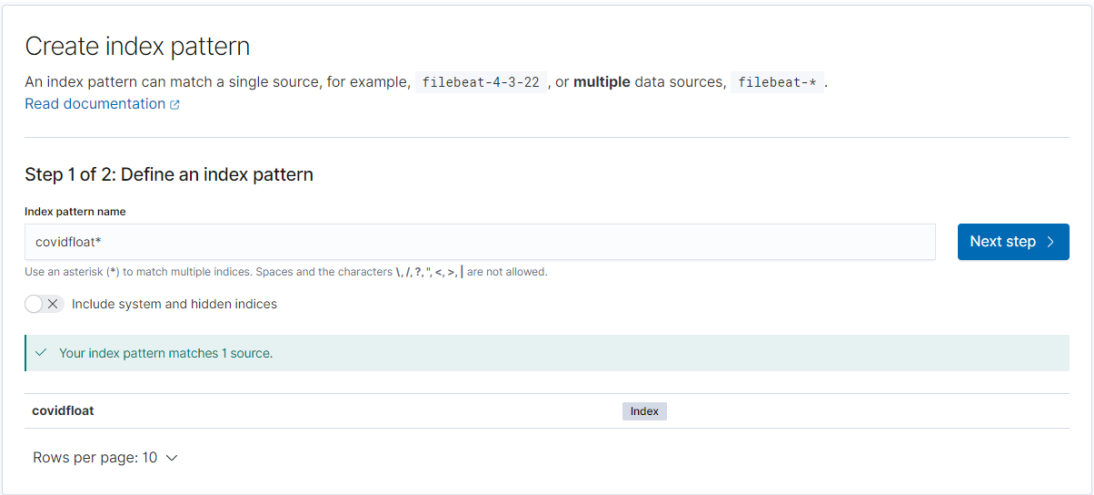
Obrázek 15, Management možnosti pro Kibanu (vlastní zpracování)

Při otevření *Index patterns* se zobrazí okno, kde lze přidat nový index. Pokud nějaký index již byl vytvořen, tak je k vidění v seznamu pod vyhledáváním. Dále je provedeno přidání nového indexu pomocí tlačítka *Create index pattern*.



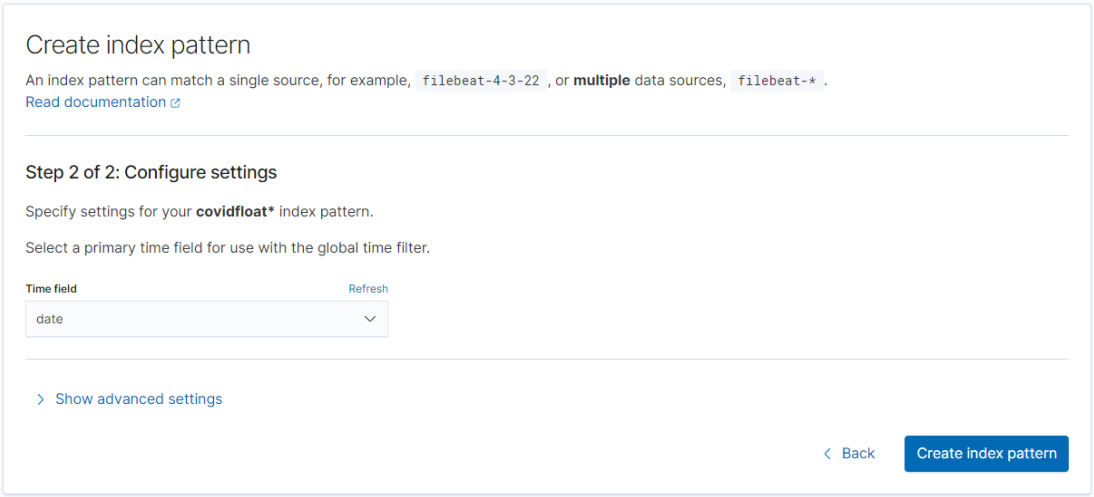
Obrázek 16, Vyhledání indexu nebo jeho tvorba (vlastní zpracování)

Do textového pole je zadáno jméno indexu, které je shodné se zadáním jména do konfiguračního souboru. Dále stačí pokračovat stisknutím na tlačítko Next step.



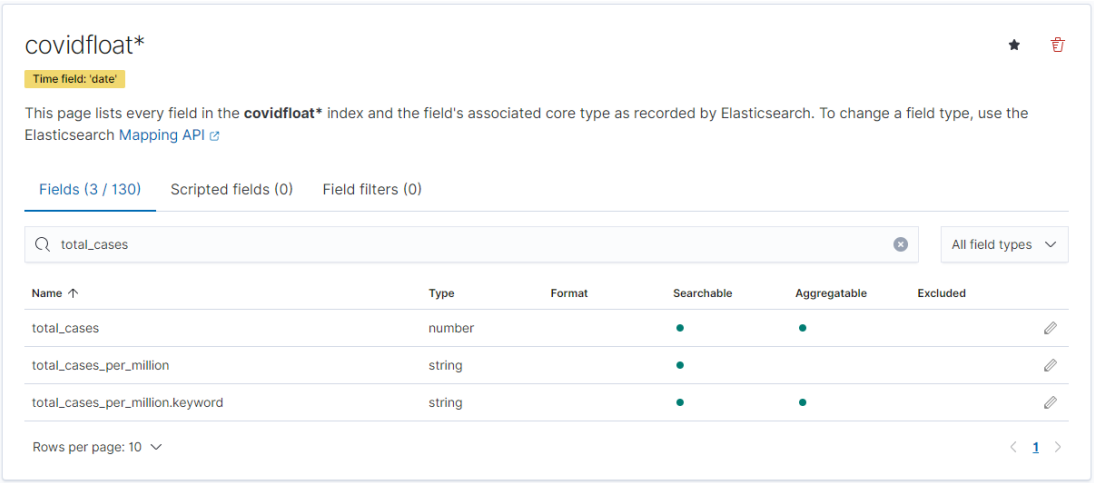
Obrázek 17, Tvorba indexu, definice jména indexu (vlastní zpracování)

Ve druhém kroku je provedeno zadání primárního časového pole. Lze si vybrat z hodnoty, kterou přidá Kibana nebo z vlastních dat. Vlastní data obsahují časový údaj date, který je použit.



Obrázek 18, Tvorba indexu, výběr primárního časového pole (vlastní zpracování)

Po vytvoření index patternu se zobrazí výpis všech dat, která se načetla a ta, která Kibana vkládá defaultně. V konfiguračním souboru byla provedena změna pouze datového typu pro total_cases. Po vyhledání polí obsahující řetězec total_cases lze vidět, že obsahuje datový typ number.

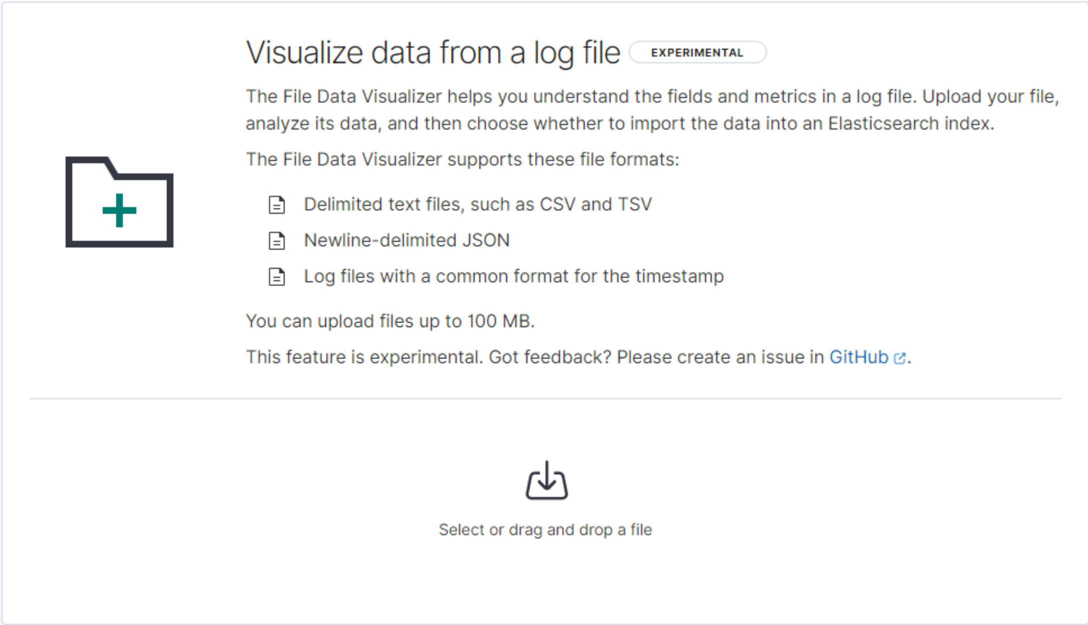


Obrázek 19, Index covidfloat a jeho pole obsahující slovo total_cases (vlastní zpracování)*

7.2.2 Machine Learning import dat

V dalším kroku je zvolena možnost *nahrát vlastní data*. Tím se zobrazí stránka, kde se nachází možnost nahrát data přetažením dat do spodní části nebo vyvoláním dialogového okna kliknutím na spodní ikonu. Data, která jsou nahrávána, by měla být ve formátu CSV, JSON nebo by se mělo jednat o soubor logu. Nahrát lze data o velikosti 100 MB až do 1 GB. Pokud je vyžadováno nahrání dat větších než 100 MB, je nutné provést změnu v pokročilém nastavení.

Tato metoda je pouze experimentální a nemusí se vyskytovat ve starších nebo novějších verzí. Doporučuje se používat pouze pro prvotní průzkum dat.



Obrázek 20, Vizualizér dat souboru pro nahrání dat (vlastní zpracování)

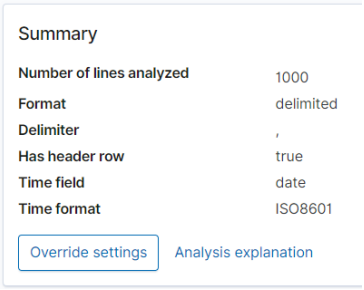
Po nahrání souboru proběhne analýza dat ze souboru. V dalším kroku Kibana zobrazí analýzu těchto dat.

První část zobrazí prvních 1000 řádků souboru, podle kterých proběhla analýza dat. Na těchto datech díky strojovému učení proběhla prvotní analýza dat. Ty jsou převedeny do jednotlivých sloupců a řádků, které jsou připraveny k uložení.

Obrázek 21, Okno zobrazující prvních 1000 řádků dat ze souboru (vlastní zpracování)

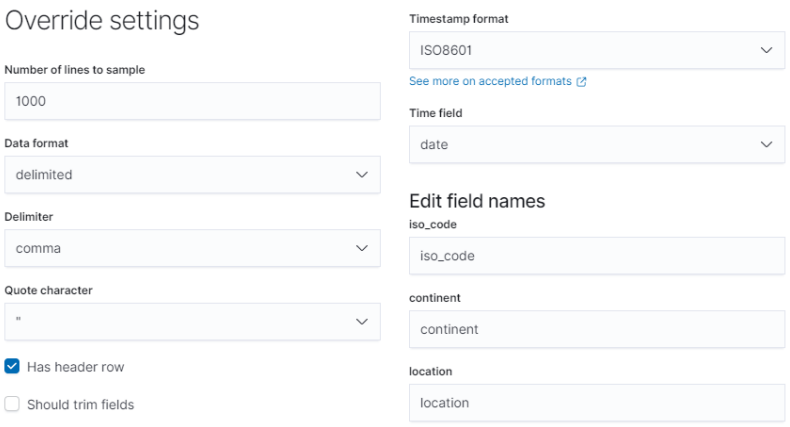
V dalším kroku jsou zobrazeny parametry, podle kterých byla data zpracována. Na snímku (obrázek 22) lze vidět, že bylo načteno opravdu prvních 1000 řádků. Dále obsahuje *Format delimited*, což znamená, že data jsou ve formátu s oddělovačem a jednalo se o oddělovač typu čárka. *Has header row* udává, že soubor obsahuje také

řádek s hlavičkou, v ukázce se jedná o názvy jednotlivých sloupců. Řádek, na kterém je možné vidět *Time field*, obsahuje hodnotu date, která se vyskytuje v souboru v hlavičce. Jedná se o časový údaj, který je v souboru. Poslední řádek značí formu času.



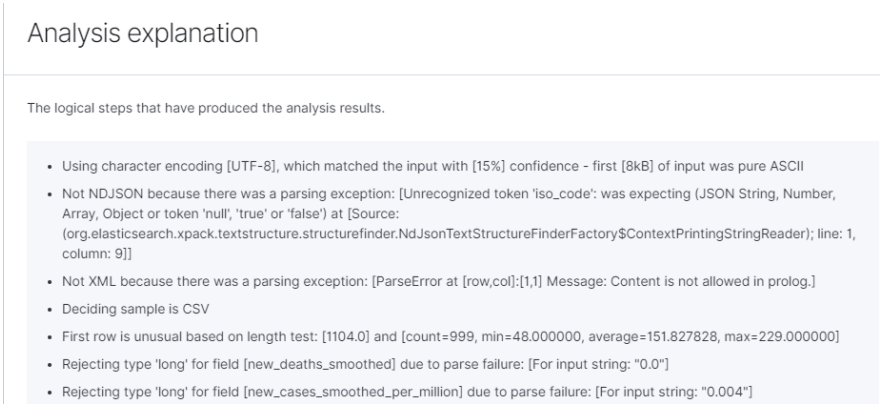
Obrázek 22, Souhrn informací z prvotní analýzy pro zpracování dat (vlastní zpracování)

Pokud Kibana neprovedla prvotní analýzu správně, existuje možnost po stisknutí tlačítka *Override settings* udělat potřebné změny. Lze změnit počet řádků, kdy na větším množství lze dosáhnout přesnějších výsledků, změnit oddělovač dat, časový formát nebo jednotlivé názvy řádků dat.



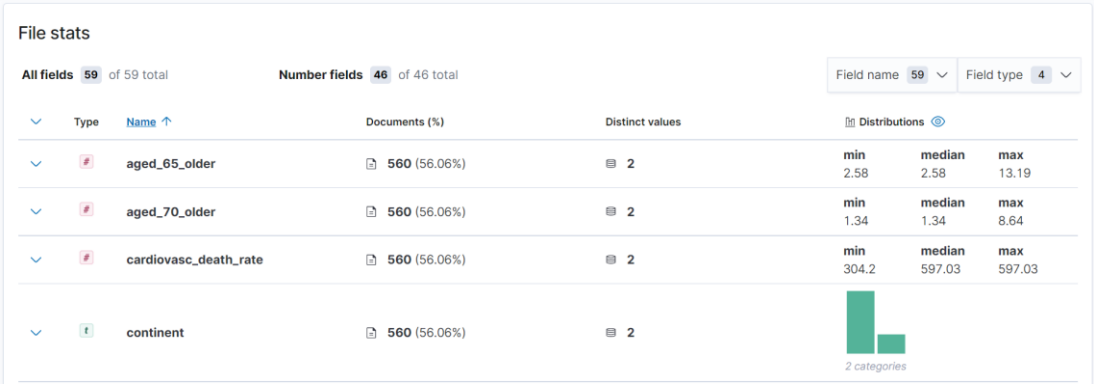
Obrázek 23, Rozšířené nastavení pro zpracování dat (vlastní zpracování)

Sousedící tlačítko *Analysis explanation* má pouze informační účel. Po rozkliknutí se zobrazí okno vysvětlující analýzu dat.



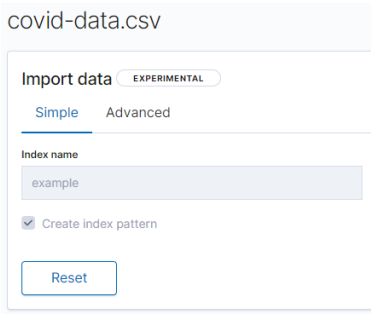
Obrázek 24, Ukázka vysvětlení analýzy dat (vlastní zpracování)

Poslední částí analýzy je statistika souboru. Ta zobrazuje statistiku dat provedenou na již zmíněných prvních tisících řádcích souborů. Zleva lze vidět jako první typ dat sloupce, jméno sloupce, počet odpovídajících dat hodnotou a v závorce v procentech. Další hodnota udává počet unikátních hodnot. Data v poslední sloupci poskytují pro každý datový typ trochu odlišná data. Může se jednat o kombinaci minimum, medián a maximum nebo o sloupcový graf hodnot. V některých případech nejsou k dispozici žádná data.



Obrázek 25, Statistiky souboru (vlastní zpracování)

Na konci stránky je tlačítko pro *import dat*. Po jeho stisknutí je vyžadováno vytvoření indexového jména. Zde je na výběr z možností *Simple* a *Advanced*. Pro ukázku postačí možnost *Simple*, kdy stačí zadat pouze jméno indexu. Název indexu smí obsahovat znaky malé abecedy, čísla a vybrané znaky.

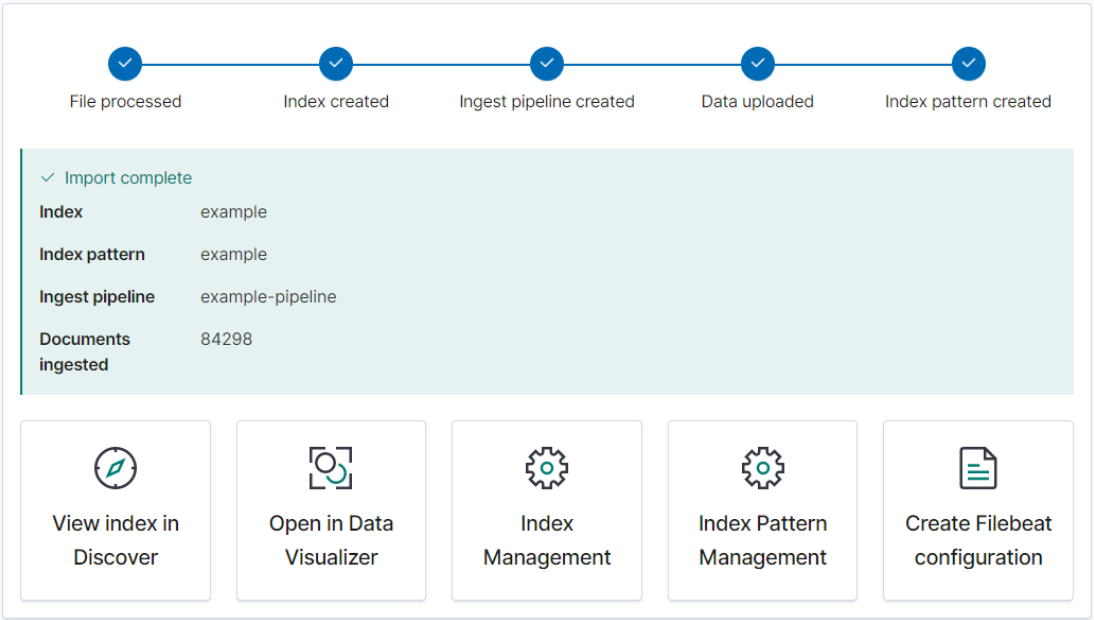


Obrázek 26, Okno pro vytvoření indexu (vlastní zpracování)

Tento index slouží pro přístup k datům, která jsou uložena v Elasticsearch. Aby Kibana mohla k těmto datům přistupovat, potřebuje znát index, který vybere pouze data, která by měla být dále zpracována. Index může odkazovat na data z jiného dne, datový tok nebo alias jiného indexu.

Po vytvoření indexu proběhne kompletní zpracování souboru, kdy se data převedou do formátu NDJSON, pro jejich lepší použití. Vytvoří se index a zároveň se provede mapování. Vytvoří se pipeline pro příjem dat. Nahrají se data do nového indexu v Elasticseach. Pokud bylo v předchozím kroku zaškrtnuto vytvoření vzoru indexu, tak se vytvoří.

Jakmile dojde k dokončení těchto akcí, zobrazí se shrnutí, kde lze vidět názvy jednotlivých indexů a počet přijatých dat. Pod touto rekapitulací máme možnost přejít ke zkoumání dat nebo správě indexu (Gowdy, 2019).

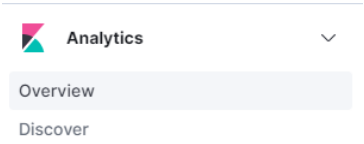


Obrázek 27, Rekapitulace importu a zobrazení nabídky pro pokračování v práci (vlastní zpracování)

7.3 Discover

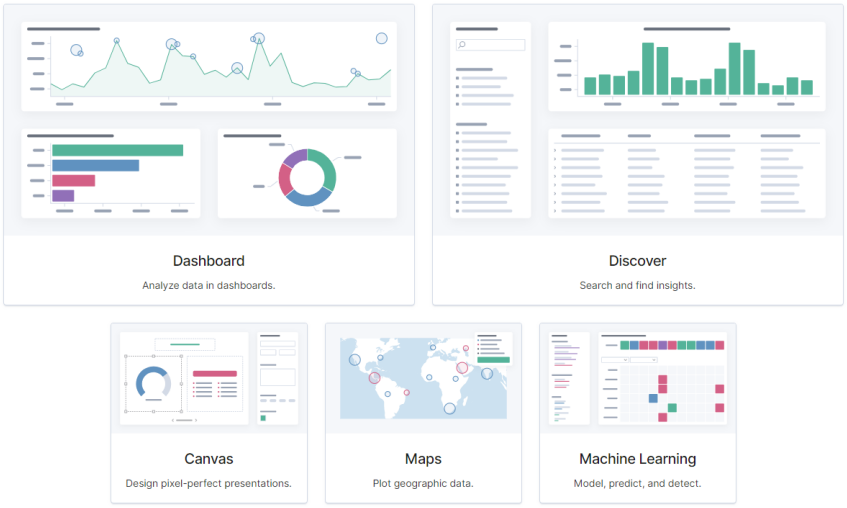
Funkce Discover umožňuje rychlé vyhledávání a filtrování dat. Díky tomuto rychlému prohledávání lze získat informace o struktuře dat. Vyhledané informace lze poté uložit a vyexportovat, jako soubor csv nebo link, který umožňuje přístup k vyfiltrovaným záznamům.

K Discover lze přistoupit dvěma možnostmi. První se nachází v rozklikávacím menu na levé straně, jako podnabídka *Analytics*. Druhá možnost přes položku *Overview*. Ta je k nalezení buď na domovské stránce po kliknutí na položku Kibana nebo také v levém menu.



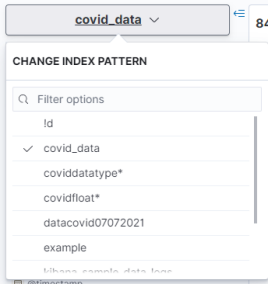
Obrázek 28, Položka menu Analytics obsahující Overview a Discover (vlastní zpracování)

Přes možnost *Overview* je možné přejít také do *Dashboard*, *Canvas*, *Maps* a *Machine Learning*.

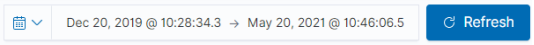


Obrázek 29, Overview (vlastní zpracování)

Po přechodu do *Discover* je nutné v levé části vybrat *index pattern*, který umožní pracovat s požadovanými daty. To se provede rozkliknutím combo boxu, kde lze použít fulltextové vyhledávání nebo kolečkem projet nabídku. Po výběru indexu je potřeba zvolit časové rozpětí, se kterým je požadováno pracovat. Aby se provedly změny, je potřeba použít tlačítko *Refresh*.



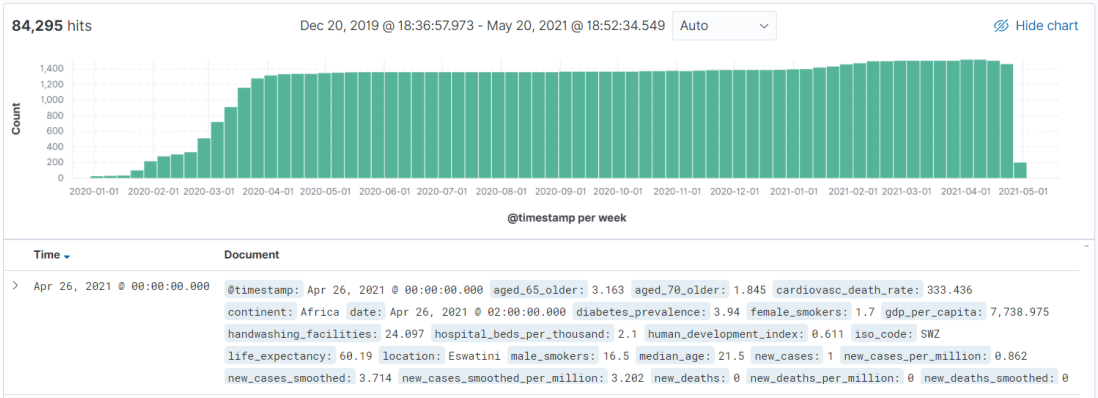
Obrázek 30, Výběr index patternu (vlastní zpracování)



Obrázek 31, Časové rozpětí dat, tlačítko Refresh (vlastní zpracování)

Jednou z hlavních částí panelu *Discover* je histogram spolu s jednotlivými záznamy. Tento histogram zobrazuje rozdělení dat v čase. Na ose Y je počet výskytů a na ose X časový údaj @timestamp. Jedná se o časový údaj, který si Kibana spravuje sama podle položky date, která se vyskytuje v našich datech. Nad histogramem je možné změnit časový úsek z automatického rozdělení na milisekundy, sekundy, minuty, hodiny, dny, týdny, měsíce a roky. Pokud histogram není zajímavý nebo je nevypovídající lze jej skrýt a pracovat čistě s daty.

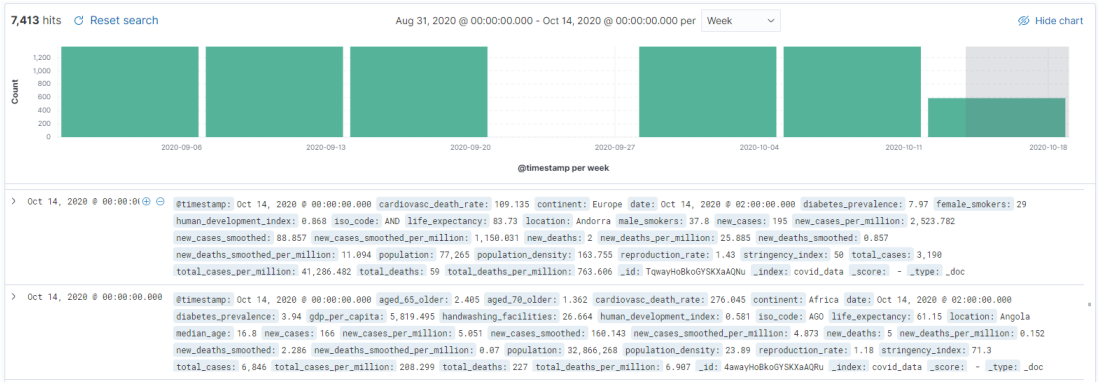
První možnost, jak pracovat s daty, je rozkliknutí jednoho dílku grafu nebo označení více částí. To poskytne pouze data, která se vyskytují v požadovaném rozmezí. Tento způsob se dá také nahradit časovým rozpětím, které bylo již ukázáno.



Obrázek 32, Discover histogram a seznam dat (vlastní zpracování)

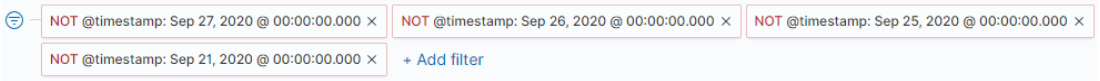
Další možností je vynechání konkrétních dat. Tato možnost se nachází, po najetí ukazatele myši na jednotlivé časové údaje, ve sloupci Time. Po najetí ukazatelem se zobrazí tlačítko plus a mínus. Plus přidává hodnoty, které se mají zobrazit a minus tyto hodnoty odstraňuje.

Na obrázku (obrázek 33) můžete vidět, jak vypadá graf, který je dělen dle týdnů. Pomocí tlačítka minus jsou vyjmuty hodnoty od 20-09-2020 do 27-09-2020.



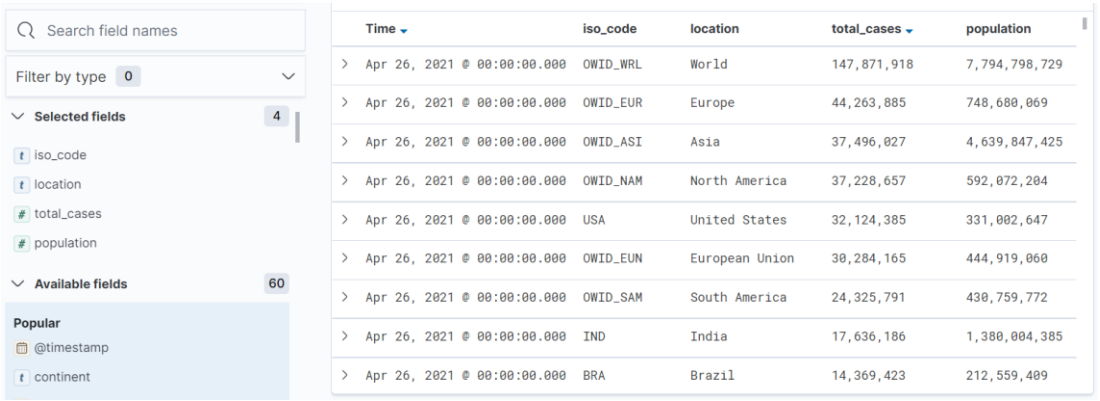
Obrázek 33, Discover histogram, výběr určitých dat (vlastní zpracování)

Hodnoty, které byly vyjmuty, lze vidět v horní části obrazovky. Pokud je požadováno kteroukoli z nich vrátit do výběru, je to možné pomocí křížku, který je u každé hodnoty.



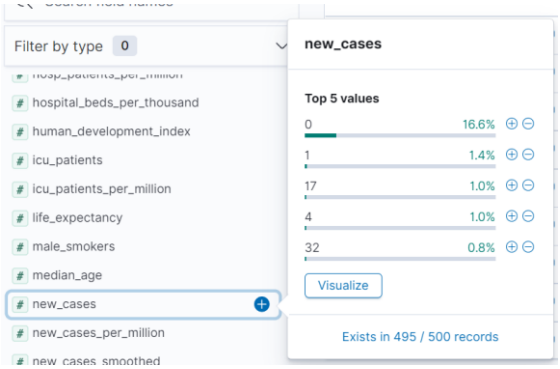
Obrázek 34, Vyjmuté hodnoty ve výběru (vlastní zpracování)

V levé části Discover pod položkou pro výběr index patternu se nachází seznam všech polí, která datová sada obsahuje. Tento panel nabízí spoustu zajímavých možností. Jednou z nich je výpis specifických dat přímo pod histogram. V této ukázce jsou vybrány položky iso_code, location, total_cases a population. Seznam je seřazen dle total_cases. Díky tomuto výběru lze vidět, celkový počet případů v daný den v každé lokalitě.



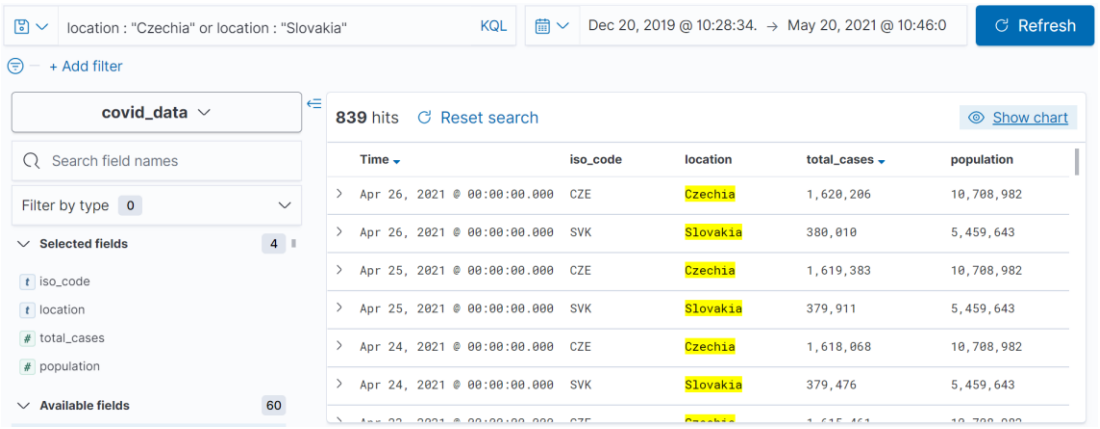
Obrázek 35, Discover, výběr určitých hodnot (vlastní zpracování)

Druhou možností, jak lze využít levý panel je rozkliknout jednu z položek. Pokud obsahuje dostatečný počet záznamů, zobrazí se pět položek, které mají největší procentuální zastoupení. Tyto položky lze opět pomocí tlačítek plus a minus vyřadit z výběru. Další možností je zvolit položku *Visualize*, která provede přepnutí z *Discover* do *Visualize Library*. Samotnému *Visualize Library* je věnována další kapitola.



Obrázek 36, Discover, detail hodnoty (vlastní zpracování)

Poslední důležitou funkcí Discover je vyhledávání pomocí *Kibana Query Language (KQL)*. V případě nepoužití KQL, lze využít jazyk Lucene od společnosti Apache, kdy je možné vyhledávat od nejjednodušších výrazů až po složitější. V tomto případě byly použity stejné hodnoty, jako v předchozím případě. Došlo k vyhledání *„location : ”Czechia” or location ”Slovakia”“*.



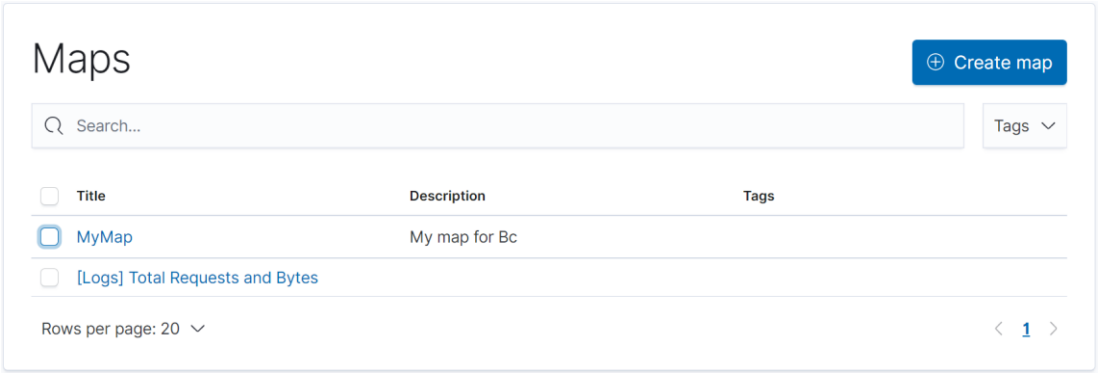
Obrázek 37, Discover, zobrazení dat pomocí vyhledávání (vlastní zpracování)

7.4 Maps

Další funkcí je tvorba map. Použitá datová sada obsahuje iso_code, continent a location. Kibana nabízí vytvořit mapy pomocí zeměpisných údajů, internetových odkazů, kódů zemí a jejich jména.

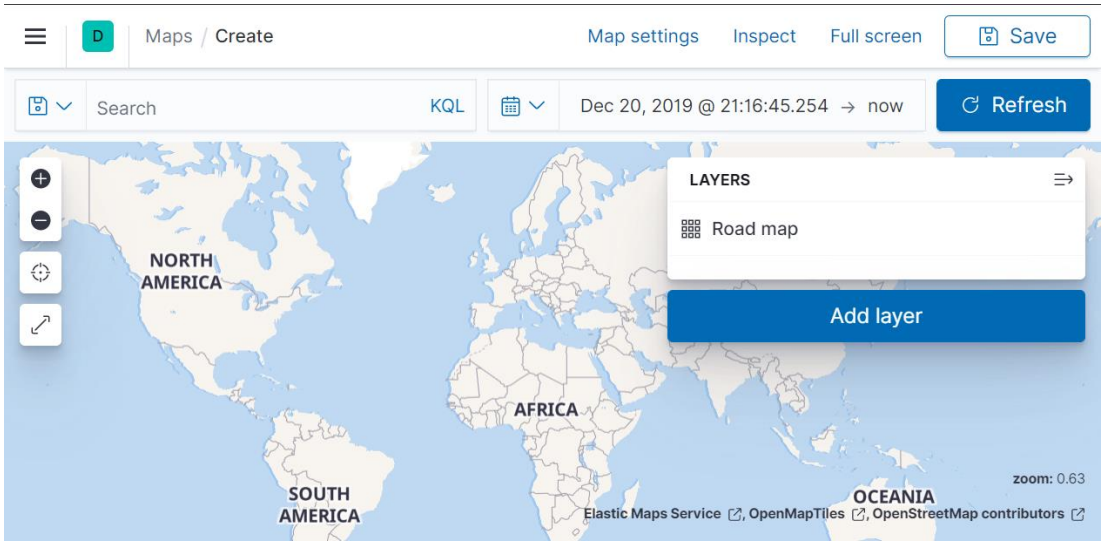
Aby bylo možné vytvořit mapu, je nutné v panelu na levé straně vybrat možnost *Maps*, která se nachází shodně jako položka *Analytics*. Další možnost, jak přejít k vytvoření map je přes *Overview*, které bylo zmíněno v předchozí části práce.

Po přechodu do *Maps* lze vidět již vytvořené mapy, které jsou již z předchozího kroku uložené. Dále bude využito tlačítko *Create map*.



Obrázek 38, Maps, hlavní menu (vlastní zpracování)

Na obrazovce, která se zobrazí lze vidět v horní části vyhledávání, které funguje na principu KQL, časové rozmezí pro data a tlačítko *Refresh*. Na levé straně jsou tlačítka na přiblížení, přechod na určitou pozici a tlačítko na přizpůsobení velikosti. Hlavní ovládací panel pro mapy se vyskytuje na pravé straně. Na tento panel se přidávají vrstvy map tlačítkem vyskytujícím se pod tímto panelem. Defaultně obsahuje vrstvu *Road map*, která zajišťuje zobrazení mapy světa.

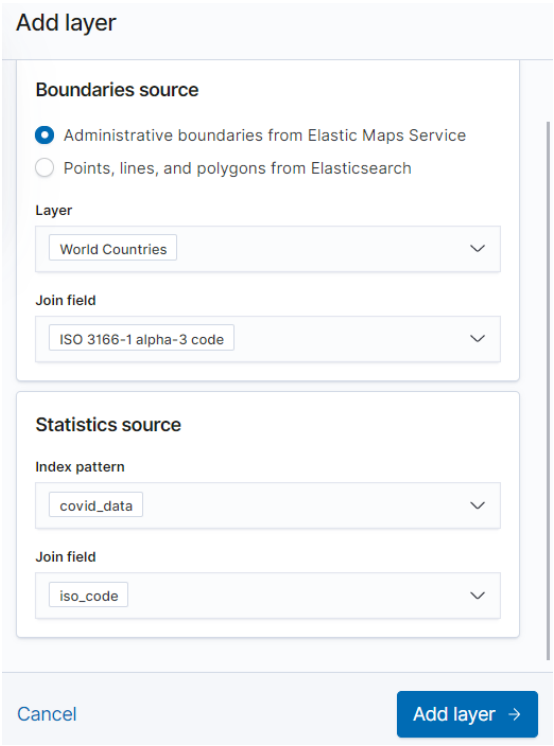


Obrázek 39, Maps create (vlastní zpracování)

Po rozkliknutí *Add layer* je možné přidat vrstvu, která se bude zobrazovat nad vrstvou *Road map*. Kibana nabízí možnosti *Upload GeoJSON*, *Documents*, *Choropleth*,

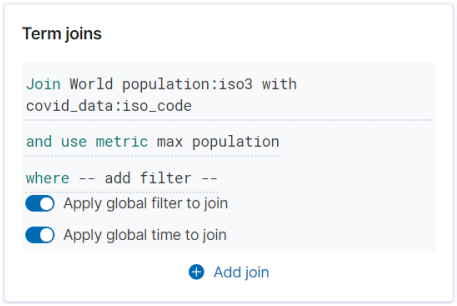
Clusters and *grids*, *Heat map*, *Tracks*, *Point to point*, *EMS Boundaries*, *EMS Basemaps*, *Tile Map Service*, *Web Map Service*, *Vector tiles*, *Observability* a *Security*.

Pro tuto ukázku je zvolena možnost *Choropleth*. V dalším kroku tak dojde k vybrání první možnosti *Administrative boundraies from Elastic Maps Service* a v políčku *Layer* možnost *World Countries*. Tento výběr zpřístupní další možnosti. První *Join field* se vybere formát, pomocí kterého se budou propojovat data *World Countries* s použitou datovou sadou. V tomto případě se bude jednat o možnost *ISO 3166-1 alpha-3 code*. Druhá možnost *Index pattern*, kdy se po výběru indexu zobrazí také *Join field* pro propojení s daty. Tady dojde k volbě možnost *iso_code*, a poté lze pomocí tlačítka *Add layer* přidat vrstvu.



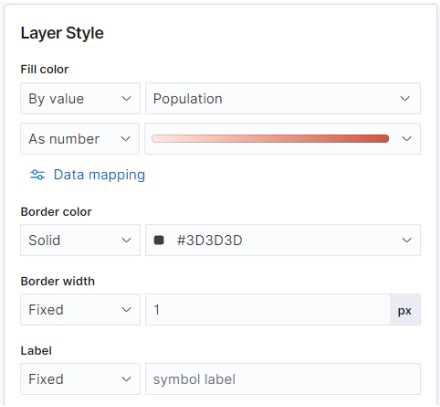
Obrázek 40, Panel Add layer (vlastní zpracování)

Po přidání vrstvy se v pravém panelu objeví možnosti *Layer settings*, kde je nutné nastavit jméno vrstvy například na *World population* a *opacity* na 100 %. Další možností je *Term joins*. Tento join se dělí na část, která byla nastavena v předchozím kroku a na část, kde je požadováno nastavení pro zobrazení. Po klinutí na druhou část se zobrazí okno, kde lze nastavit agregaci pro některé pole z použitých dat. V ukázce dojde k výběru možnosti *Max* pro pole population a nastavení *Custom label* na Population. Po dokončení těchto změn se mapa barevně změní.

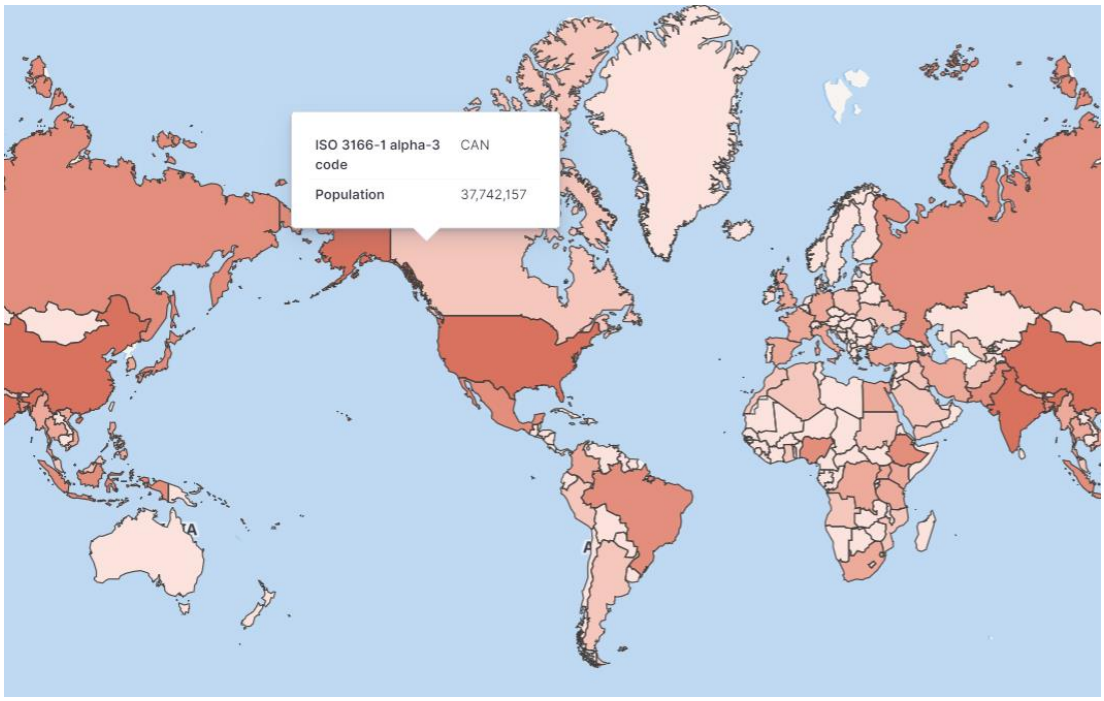


Obrázek 41, Term joins (vlastní zpracování)

Pro ovlivnění barvy jednotlivých států je zapotřebí přejít níže k možnosti *Layer Style*. Zde je nutné mít ve *Fill color By value* hodnotu Population a pro *As number* je nutné vybrat požadované barevné schéma. Při požadavku na zobrazení konkrétní hodnoty populace každého státu, je tato volba o něco níže v možnosti *Label*, kde je nutné nastavit změnu z možnosti *Fixed* na *By value*. Pro ukázku je ponechána možnost *Fixed* pro lepší přehlednost mapy. Pro zjištění populace některého ze státu, lze přejet myší na daný stát a hodnota se zobrazí.



Obrázek 42, Layer Style (vlastní zpracování)



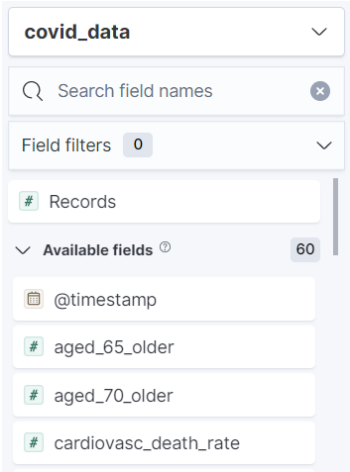
Obrázek 43, Mapa dle počtu obyvatel a zobrazení počtu obyvatel Kanady (vlastní zpracování)

7.5 Visualize Library

Visualize Library, jak z překladu názvu lze odvodit, představuje knihovnu, která nabízí nástroje k vizualizaci dat. Všechny tyto vizualizace se dají uložit a přidat na *Canvas* nebo *Dashboard*, které jsou rozebrány v následujících kapitolách. Knihovna obsahuje nástroje *Lens*, *TSVB*, *Custom Visualization*, *Aggregation based* a *Maps*, který již byl zmíněn v předchozí kapitole. Lze přidat také texty, obrázky nebo kontrolní prvky. Tato kapitola se zaměřuje na nástroj *Lens*, který poskytuje většinu možností, jako ostatní zmíněné nástroje.

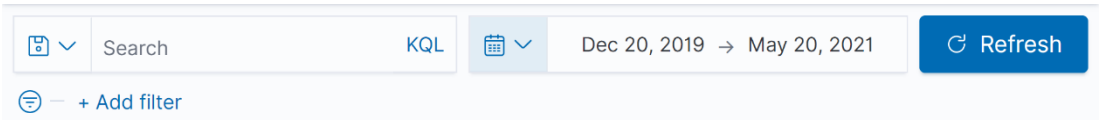
7.5.1 Lens

Nástroj Lens slouží k rychlé tvorbě grafů pomocí metody *drag and drop*, což v překladu znamená táhnout a pustit. Stránka obsahuje na levé straně *index pattern*, kde je nutné vybrat odpovídající index pro použitou datovou sadu. V ukázkovém případě se jedná o covid_data. Hned pod tímto index patternem se nachází různá pole, která lze použít pro tvorbu grafů.



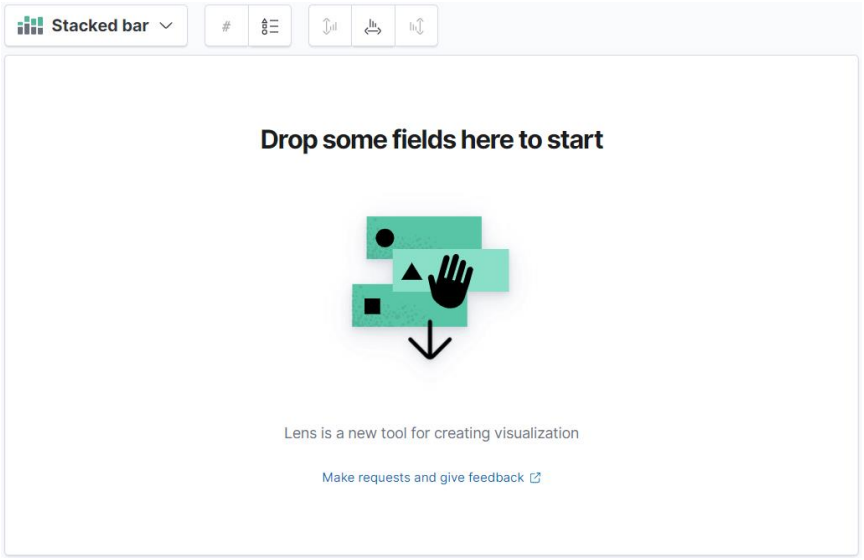
Obrázek 44, Lens, index pattern a dostupné pole (vlastní zpracování)

V horní části obrazovky se nachází vyhledávání pomocí KQL, díky kterému je možné specifikovat data, která se mají promítnout v grafu. Vpravo lze určit časové rozmezí a obnovit hodnoty pomocí tlačítka *Refresh*. Tlačítko *Add filter* slouží pro určení bližších hodnot. Lze hledat specifikou hodnotu, jednu z výčtu hodnot, zda hodnota existuje anebo jejich negace.

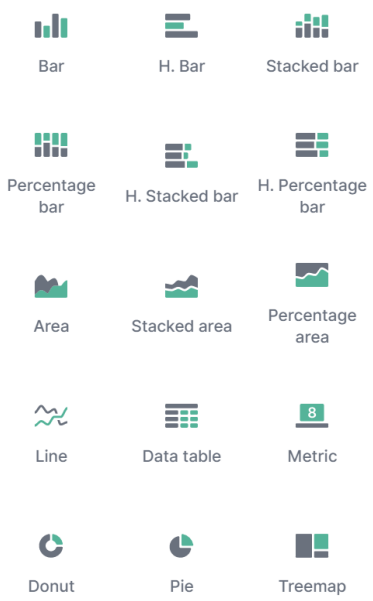


Obrázek 45, Lens, Vyhledávání pomocí KQL a určení času (vlastní zpracování)

Hlavní panel Lens slouží pro vizualizaci polí, které lze na něj jednoduše přetáhnout. Nad tímto panelem se nachází možnosti, kde lze měnit typ grafu nebo jednotlivé osy.

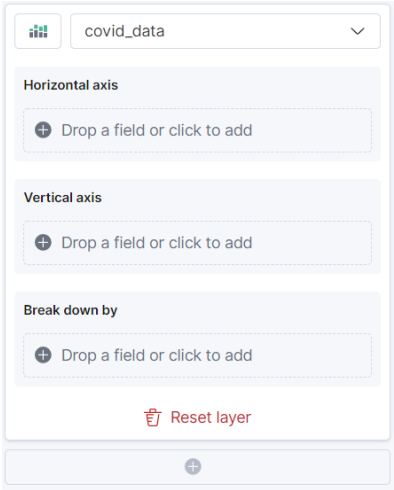


Obrázek 46, Lens, hlavní panel pro vizualizaci dat (vlastní zpracování)



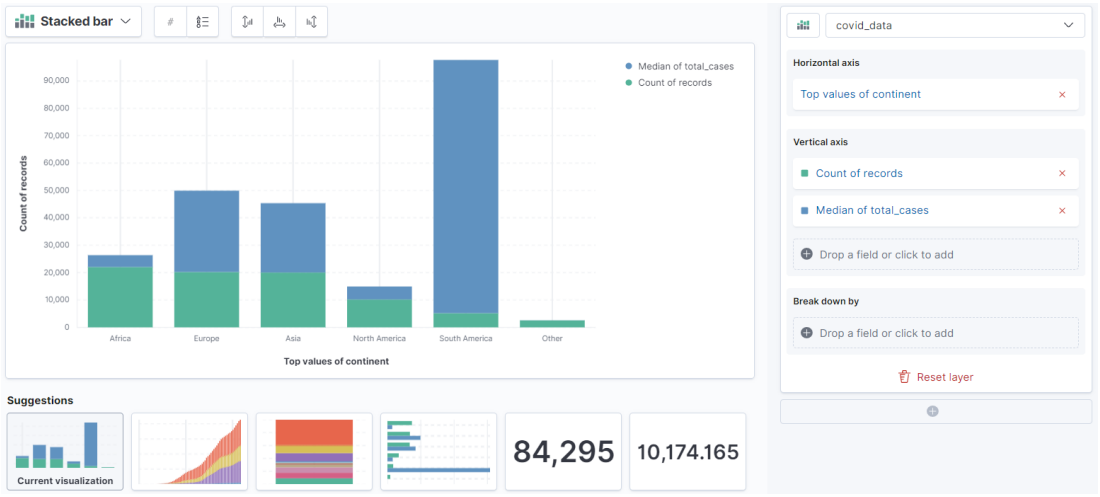
Obrázek 47, Typy grafů (vlastní zpracování)

Poslední panel, který se v Lens vyskytuje, je soustředěn na pravou stranu obrazovky. Má stejný účel, jako středový panel, jen na něm lze rovnou určit, jaké pole má být na jaké ose. Po přenesení polí na hlavní panel nebo na pravý, existuje možnost na pravém blíže pracovat s poli. Lze určit, kolik hodnot z polí se má zobrazovat, jejich pořadí zobrazení, barvu na grafu, číselný formát nebo zda se má hodnota zobrazovat pomocí průměru, sumy, počtu, minima, maxima nebo pomocí jiných funkcí.



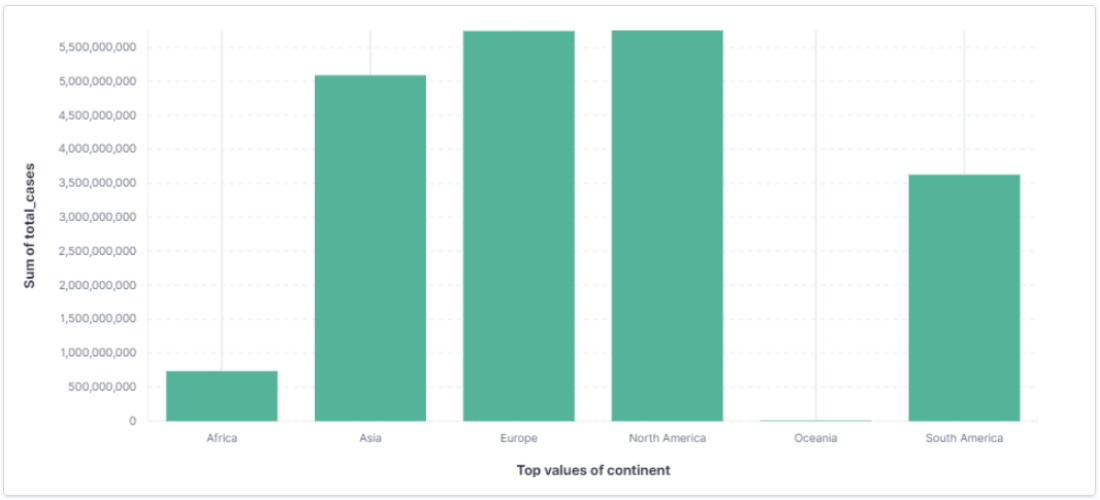
Obrázek 48, Lens, pravý panel pro umístění a práci s poli (vlastní zpracování)

Jako první dojde k přetažení na hlavní panel pole continent a pole total_cases. Zobrazí se graf podle typu, který byl přednastaven v horní části. Ve spodní části pod hlavním panelem lze vidět typy dalších grafů, které Kibana navrhuje. Lze však vybrat i jiné typy grafu, pokud to neohrozí ztrátu některých dat.



Obrázek 49, Lens, první graf continent a total_cases (vlastní zpracování)

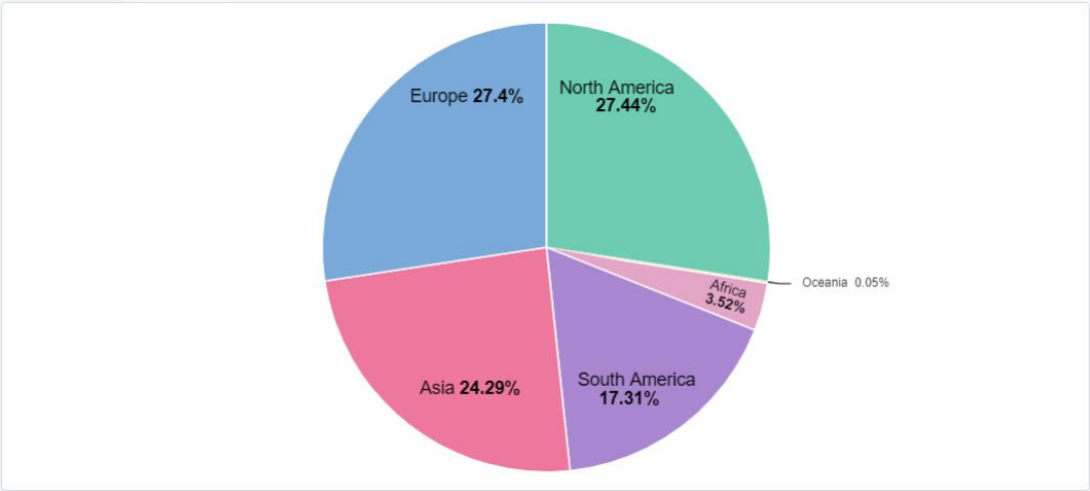
Kibana sestrojí graf podle sebe, proto použila navíc hodnotu *Count of records* a *Median of total_cases*. Na spodní ose jsou kontinenty správně, jen namísto hodnoty *Other* je vybrána South America. To se provede kliknutím na hodnotu *Top values of continent*, kde posuvníkem je možné změnit hodnotu z pěti na hodnotu šest nebo více. Dále je vyžadováno odebrat pomocí křížku *Count of records* a kliknutím na *Median of total_cases* změnit *Median* na *Sum*. Výsledkem je graf, na kterém lze vidět celkový počet případů pro jednotlivé kontinenty. V tuto chvíli je možné pracovat i se specifickými časy pomocí časového rozmezí v horní části obrazovky nebo například pomocí KQL oddělat celý kontinent. V ukázce není graf pozměněn.



Obrázek 50, Lens, druhý graf continent a total_cases (vlastní zpracování)

Pokud nevyhovuje tento typ grafu, lze v horní části překliknout například na koláčový graf. Kibana se postará o potřebné změny. Hodnoty v grafu jsou vyjádřeny

procentuálně. Pro zobrazení konkrétní hodnoty lze přejet myší nad požadovaný kontinent a lze tak uvidět hodnotu total_cases pro daný kontinent.



Obrázek 51, Lens, třetí graf continent a total_cases (vlastní zpracování)

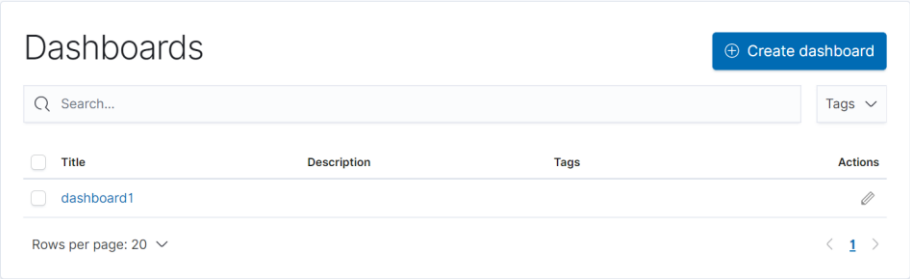
7.6 Canvas

Funkce Canvas je zde zmíněna pouze doplňkově, jelikož funguje v podstatě jako takový vestavěný prezenční program. Obsahuje možnosti jako přidávat text, tvary, vytvořené grafy, obrázky, použít filtry nebo ukazatele průběhu. Prezentaci lze upravovat také pomocí CSS kódu. Lze tak vytvořit plnohodnotnou prezentaci, kterou je možné v Kibaně přímo prezentovat nebo si ji vyexportovat. Zajímavějším řešením pro prezentování vytvořených grafů je použít *Dashboard*.

7.7 Dashboard

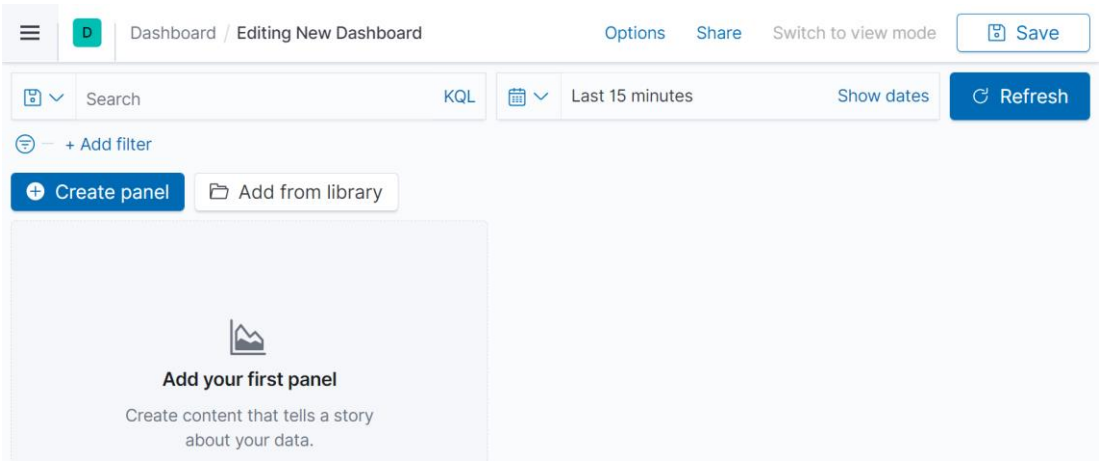
Právě ono lepší a zajímavější řešení pro představení výsledků získaných z dat je Dashboard. Lze jej najít v levém hlavním panelu nebo přes nabídku v *Overview*. Na tento panel lze přidávat všechny grafy, mapy, nástroje či text, které je možné ovlivnit pomocí KQL nebo časovým rozmezím.

Po přejití do *Dashboard* se zde nachází tlačítko na přidání nového dashboardu nebo lze použít již vytvořené desky.



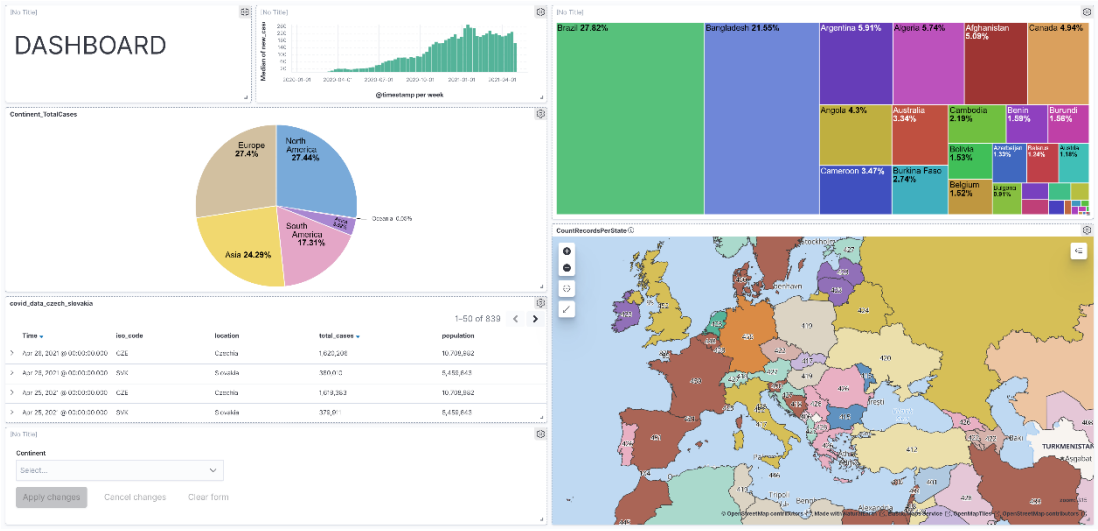
Obrázek 52, Dashboard, vytvoření nového dashboard a vyhledání existujících (vlastní zpracování)

Pokud je zapotřebí vytvořit novou desku dojde k přesměrování na její editační stránku. Na této stránce v horní části je opět vyhledávání pomocí KQL, časové rozmezí a tlačítko pro obnovení. Nad tlačítkem pro obnovení je tlačítko *Save*, které uloží dashboard a poté jej lze sdílet tlačítkem *Share*, jako kód pro vložení, odkaz, PDF nebo PNG.



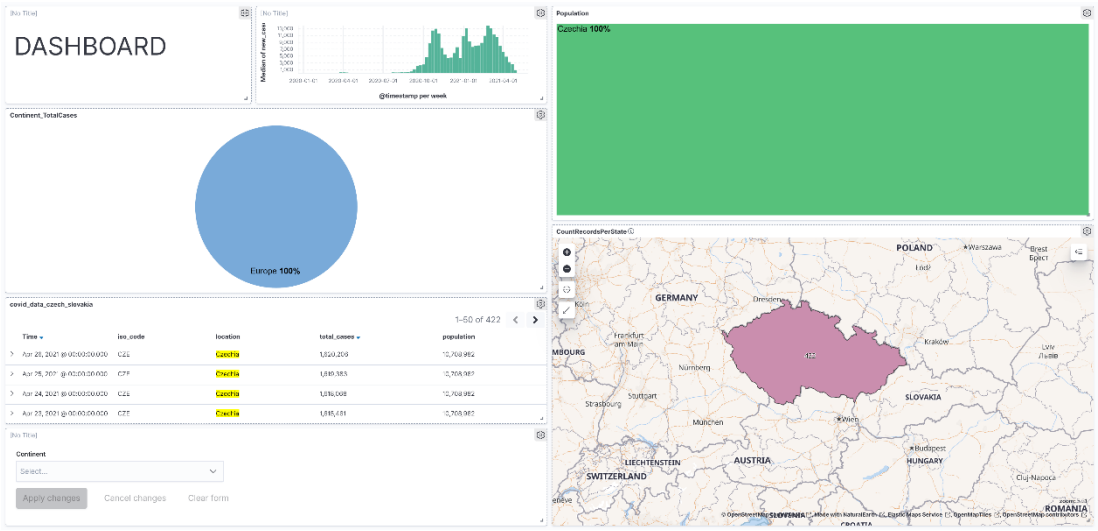
Obrázek 53, Dashboard, nově vytvořený dashboard (vlastní zpracování)

Tlačítkem *Create panel* je možné přidat vizualizace, a to způsoby, které byly již představeny. Je možné vytvořit grafy, mapy, texty nebo přidat kontrolní prvky. Při použití tlačítka *Add from library* lze najít již vytvořené vizualizace. Zobrazí se nabídka, ve které stačí najít již vytvořené vizualizace, které se po kliknutí přidají na dashboard. Poté je zapotřebí zadat časové rozmezí dat, jinak by vizualizace byly prázdné. Vizualizace lze různě zvětšovat, zmenšovat nebo přesouvat. Po přidání a uspořádání všech prvků lze provést *export Dashboard* jako PNG.



Obrázek 54, Dashboard po přidání prvků (vlastní zpracování)

Pro případ zobrazení stejných vizualizací pouze pro Česko, se lze proklikat skrze vizualizace nebo do KQL a zadat příkaz *„location : ”Czechia”“*. Jestli vizualizace pracuje s hodnotami, kdy lze použít jen lokaci, tak se vizualizace přizpůsobí hledání. Koláčový graf, ale pracuje s kontinenty, a tak zobrazí total_cases kontinentu, kterému Česko připadá tedy Evropu.



Obrázek 55, Dashboard pouze pro hodnoty z Česka (vlastní zpracování)

V praktické části bylo předvedeno, že i velký soubor dat, který by byl v některých nástrojích těžko zpracovatelný, se dá dobře zpracovat a jeho výsledky vizualizovat. K tomuto procesu byl využit nástroj Kibana, který umožnil jednodušeji extrahovat data ze souboru. Vzhledem k možnostem, kterými Kibana disponuje, byla data nahrána do Elasticsearch a pomocí vytvoření indexu v Kibaně, bylo možné k těmto datům přistupovat. Dále byly představeny a využity nástroje, které Kibana nabízí pro

analýzu a vizualizaci dat. Celková práce obsahuje materiály, které mohou být přínosem pro nově příchozí uživatele nahlížející do světa Big Data.

ZÁVĚR

Bakalářská práce se zabývala představením technologií Big dat a jejich možnému využití v praxi. Práce nebyla mířena jenom na samotná Big Data, ale na celý jejich ekosystém. V průběhu práce se čtenář dozvěděl, co to vlastně Big Data jsou a jaké mají uplatnění v dnešním světě. Seznámen byl také s pojmem ekosystém, který hraje důležitou roli ve světě Big dat. Práce obsahuje z velké části seznámení s NoSQL databázemi, které jsou nedílnou součástí pro ukládání Big dat. Dále došlo ke srovnání NoSQL databází s relačními, které ukládají data do tabulek. Představeny byly technologie Big dat, konkrétně od společnosti Apache. U těchto technologií se čtenář seznámil s hlavními funkcemi, díky kterým mají své využití v praxi. Tři vybrané technologie Hadoop, Spark a Flink byly mezi sebou porovnány. Nedílnou součástí ekosystému Big dat jsou i Open data a jejich využití, která byla také představena. Teoretickou část uzavírají vizualizační nástroje, které jsou použity v praktické části.

Praktická část měla za úkol vybrat jednu konkrétní technologii a tu kompletně představit. Pro ukázku byla vybrána technologie Kibana z rodiny ELK Stack, kdy byly také použity programy Elasticsearch a Logstash. Pro demonstrační ukázku byla zvolena otevřená data, konkrétně datová sada se záznamy ohledně pandemie Covid-19. Tyto data byla stručně představena a následně byl podrobně popsán proces jejich importu do Kibany pomocí programu Logstash a pomocí funkce Kibany Machine Learning. Další kapitoly se již věnovaly samotné analýze dat a postupům vedoucím k základním vizualizacím dat. Jednalo se o metody Discover, Maps, Canvas, Dashboard a Lens, která jsou součástí Visualize Library.

Celá práce slouží pro pochopení Big dat ekosystému. Čtenář by měl získat pochopení základních termínů a technologií z oblasti Big Data a porozumět rozdílům mezi jednotlivými typy NoSQL databází. Díky praktické části by měl být schopen dokázat importovat data do Kibany a následně provést základní vizualizace. Big Data ekosystém je velice rozsáhle téma, a proto se práce zaměřovala spíše na základní pochopení hojně využívaných technologií než na podrobný popis všech dostupných nástrojů. Práce by mohla být dále rozšířena o další technologie a ukázky, ale se zaměřením se na konkrétní téma či problém.

POUŽITÁ LITERATURA

ALEKSIC, Marko, 2020. ACID Vs. BASE: Comparison Of Database Transaction Models. *Phoenixnap* [online]. Arizona: Aleksic [cit. 2021- 4- 3]. Dostupné z: https://phoenixnap.com/kb/acid-vs-base

AL-SAEEDI, Bilal, © 2016. Factors Influencing the Adoption of a NoSQL Solution. *Factors Influencing NoSQL Adoption* [online]. [cit. 2021-4-4]. Dostupné z: http://alronz.github.io/Factors-Influencing-NoSQL-Adoption/site/

ARTHUR, Lisa, 2013. What Is Big Data? *Forbes* [online]. [cit. 2021-4-18]. Dostupné z: https://www.forbes.com/sites/lisaarthur/2013/08/15/what-is-big-data/?sh=8b48e585c85b

BAHGA, Arshdeep a Vijay MADISETTI, 2016. *Big Data Analytics: A Hands-On Approach*. ISBN 978-1-949978-00-1.

BENČAT, Marian, 2012. Výuka transakce - největší zlo, kterého se na vás učitelé dopustili. *Medium* [online]. San Francisco: Medium, 2018 [cit. 2021-4-3]. Dostupné z: https://medium.com/@marianbenat/v%C3%BDuka-transakce-nejv%C4%9Bt%C5%A1%C3%AD-zlo-kter%C3%BDch-se-na-v%C3%A1s-u%C4%8Ditel%C3%A9-dopustili-d7c032e7ad83

CANADATA, 2020. Ani konopný průmysl se bez AI a Big Data neobejde. *Canadata* [online]. [cit. 2021-4-18]. Dostupné z: https://cannadata.cz/clanky/ani-konopny-prumysl-se-bez-ai-a-big-data-neobejde

CLOUDERA, ed., 2019. HDFS, MapReduce, and YARN. *Cloudera* [online]. California [cit. 2021-4-18]. Dostupné z: https://www.cloudera.com/products/open-source/apache-hadoop/hdfs-mapreduce-yarn.html

DATABRICKS, 2020. Apache Spark™ - What is Spark. *Databricks* [online]. San Francisco [cit. 2021-4-18]. Dostupné z: https://databricks.com/spark/about

DATAFLAIR, ed., © 2021. Big Data Applications – A manifestation of the hottest buzzword. *DataFlair* [online]. [cit. 2021-4-18]. Dostupné z: ttps://data-flair.training/blogs/big-data-applications/

DEMCHENKO, Yuri a Peter MEMBREY, 2014. *Defining architecture components of the Big Data Ecosystem* [online]. Minneapolis, MN, USA: IEEE [cit. 2021-2-19]. ISBN 978-1-4799-5158-1. Dostupné z: https://ieeexplore.ieee.org/document/6867550

ELASTICSEARCH, ed., © 2021. Elasticsearch: The Official Distributed Search & Analytics Engine. *Elastic* [online]. [cit. 2021-4-18]. Dostupné z: https://www.elastic.co/elasticsearch/

FERRANDO, Antoni, © 2021. Apache Cassandra. *VIP trust* [online]. Praha [cit. 2021-4-6]. Dostupné z: https://viptrust.com/technologie/ostatni/apache-cassandra

GHAVAMI, Peter, 2016. *BIG DATA GOVERNANCE: Modern Data Management Principles for Hadoop, NoSQL & Big Data Analytics*. ISBN 1519559720.

GOWDY, James, 2019. Import CSV and Log Data into Elasticsearch from Kibana with File Data Visualizer. *Elastic* [online]. [cit. 2021-7-27]. Dostupné z: https://www.elastic.co/blog/importing-csv-and-log-data-into-elasticsearch-with-file-data-visualizer

HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK, 2015. *Big Data a NoSQL databáze*. Praha: Grada. Profesionál. ISBN 978-80-247-5466-6.

HRABINOVA, Svetlana, 2018. Otevřená definice -- definice otevřených znalostí. *Open Definition* [online]. [cit. 2021-4-18]. Dostupné z: https://opendefinition.org/od/2.0/cz/

CHIANG, Ray a Dennis DAWSON, 2015. Untangling Apache Hadoop YARN, Part 1: Cluster and YARN Basics. *Cloudera blog* [online]. California [cit. 2021-4-18]. Dostupné z: https://blog.cloudera.com/untangling-apache-hadoop-yarn-part-1-cluster-and-yarn-basics/

IBM, 2019. CAP Theorem. *IBM* [online]. New York: IBM [cit. 2021-4-4]. Dostupné z: https://www.ibm.com/cloud/learn/cap-theorem

IMANUEL, 2013. Top Multi-Model Databeses. In: *PAT RESEARCH* [online]. Toronto [cit. 2021-2-19]. Dostupné z: https://www.predictiveanalyticstoday.com/top-multi-model-databases/?fbclid=IwAR0vjh2Xf7qqtDl3obUAk76G6W6Q5uRzhIcfRd9caVByKCRcjL97to4CVfo

ISICHKO, Dmitry, 2020. Downsides of Firebase: limitations to be aware of. *Moqod* [online]. Amsterdam [cit. 2021-4-18]. Dostupné z: https://moqod.com/downsides-of-firebase-limitations-to-be-aware-of/

JAVATPOINT, ed., © 2018. Advantages and disadvantage of Elasticsearch. *Javatpoint* [online]. India [cit. 2021-4-18]. Dostupné z: https://www.javatpoint.com/advantages-and-disadvantages-of-elasticsearch

KIMPL, Libor, 2010. *Prostorové nadstavby nekomerčních databází - vstup a správa geoobjektů*. Olomouc. Bakalářská práce. Univerzita Palackého Katedra Geoinfomatiky. Vedoucí práce RNDr. Vilém Pechanec Ph.D.

KOĎOUSKOVÁ, Barbora, 2021. Internet věcí (IoT): definice, příklady využití, produkty. *Rascasone* [online]. 13. 4. 2021 [cit. 2021-4-19]. Dostupné z: https://www.rascasone.com/cs/blog/iot-internet-veci-definice-produkty-historie

KOPAL, Ondřej, 2015. Relační a nerelační datový model v kontextu business intelligence. *Webová integrace* [online]. Praha [cit. 2021-4-4]. Dostupné z: http://www.web-integration.info/cs/blog/relacni-a-nerelacni-datovy-model-v-kontextu-business-intelligence/

LĂPUȘAN, Tudor, © 2019. Hadoop MapReduce deep diving and tuning. In: *Today software* [online]. Romania [cit. 2021-4-19]. Dostupné z: https://todaysoftmag.com/article/1358/hadoop-mapreduce-deep-diving-and-tuning

LEWIS, Barbara, 2018. Building The Big Data Warehouse, Part 5: The Overall Data Landscape. *Digitalist Magazine* [online]. [cit. 2021-4-18]. Dostupné z: https://www.digitalistmag.com/cio-knowledge/2018/04/03/building-big-data-warehouse-part-5-overall-data-landscape-06039910/

LNĚNIČKA, Martin a Jitka KOMÁRKOVÁ, 2015. *Sborník příspěvků z mezinárodní vědecké konference: MEZINÁRODNÍ MASARYKOVA KONFERENCE PRO DOKTORANDY A MLADÉ VĚDECKÉ PRACOVNÍKY*. Hradec Králové.

NAYAK, Ameya, Anil PORIYA a Dikshay POOJARY, 2013. Type of NOSQL Databases and its Comparison with Relational Databases. *International Journal of Applied Information Systems* [online]. **5**(4), 4 [cit. 2021-2-17]. ISSN 2249-0868. Dostupné z: https://www.researchgate.net/profile/Dikshay_Poojary/publication/302557703_Article_Type_of_nosql_databases_and_its_comparison_with_relational_databases/links/5aeaa2b50f7e9b837d3c40e7/Article-Type-of-nosql-databases-and-its-comparison-with-relational-databases.pdf

NOVOSELTSEVA, Ekaterina, 2018. ElasticSearch: Advantages, Case Studies, and Stats. *DZone* [online]. New York: DZone [cit. 2021-4-18]. Dostupné z: https://dzone.com/articles/elastic-search-advantages-case-studies-amp-books

OLETY, Vijay, 2014. DynamoDB: An Inside Look Into NoSQL – Part 1. In: *Pristinecrap* [online]. [cit. 2021-4-19]. Dostupné z: https://pristinecrap.com/2014/06/04/dynamodb-inside-look-into-nosql-part-1/

ORACLE, ed., © 2021. What a Relational Database Is. *Oracle* [online]. [cit. 2021-4-18]. Dostupné z: https://www.oracle.com/database/what-is-a-relational-database/

PATEL, Harsh, 2020. Redis — What and Why? *Medium* [online]. San Francisco: Medium [cit. 2021-4-4]. Dostupné z: https://medium.com/weekly-webtips/redis-what-and-why-pros-cons-ae2f5bc750fd

PICKELL, Devin, 2018. Structured vs Unstructured Data – What's the Difference? *Learning Hub* [online]. Chicago, 2018 [cit. 2021-4-18]. Dostupné z: https://learn.g2.com/structured-vs-unstructured-data

RIAK, ed., 2008. RIAK CUSTOMERS. In: *Riak* [online]. [cit. 2021-4-18]. Dostupné z: https://riak.com/riak-users/

SADALAGE, Pramod J. a Martin FOWLER, © 2013. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Upper Saddle River: Addison-Wesley. ISBN 978-0-321-82662-6.

SAS, ed., © 2021. Big Data Analytics. *SAS* [online]. North Carolina [cit. 2021-4-18]. Dostupné z: https://www.sas.com/cs_cz/insights/analytics/big-data-analytics.html#todaysworld

STEDMAN, Craig a Jack VAUGHAN, 2020. Apache Hadoop YARN. *Search Data Management* [online]. [cit. 2021-4-18]. Dostupné z: https://searchdatamanagement.techtarget.com/definition/Apache-Hadoop-YARN-Yet-Another-Resource-Negotiator

STRONG, Colin, 2015. *Humanizing Big Data: Marketing at the Meeting of Data, Social Science and consumer insight* [online]. 2015: KoganPage [cit. 2021-2-13]. ISBN 978-0-7494-7212-2. Dostupné z: https://books.google.cz/books?hl=cs&lr=&id=1FPHBgAAQBAJ&oi=fnd&pg=PP1&dq=big%20data%20marketing&ots=WhYFQDYkwt&sig=uWEHM8NsW9GmzZo7eDDAN4AkQrY&redir_esc=y&fbclid=IwAR3N1Esl-W7-jH2XPVGPQ2Zdq08EgkwPeACnMcPZNWjWG00ZzDg-YWPKp6U#v=onepage&q&f=false

SUNAGAR, Pramod, 2020. Hybrid Computational Intelligence. BHATTACHARYYA, Siddhartha, Václav SNÁŠEL a Deepak GUPTA. *In Hybrid Computational Intelligence for Pattern Analysis and Understanding, Hybrid Computational Intelligence* [online]. Academic Press, s. 25-47 [cit. 2021-4-18]. ISBN 978-0-12-818699-2. Dostupné z: https://www.sciencedirect.com/book/9780128186992/hybrid-computational-intelligence#book-description

TOVSHTEYN, Chaim Yevgeniy, 2018. What is the relation between Kafka, the writer, and Apache Kafka, the distributed messaging system? *Quora* [online]. [cit. 2021-4-18]. Dostupné

z: https://www.quora.com/What-is-the-relation-between-Kafka-the-writer-and-Apache-Kafka-the-distributed-messaging-system

WITKOWSKI, Krzysztof, 2017. [online]. (182), 763-769 [cit. 2021-2-14]. ISSN 1877-7058. Dostupné z: doi:https://doi.org/10.1016/j.proeng.2017.03.197.

Redis - Reviews, Pros & Cons | Companies using Redis, © 2021. *StackShare* [online]. [cit. 2021-4-18]. Dostupné z: https://stackshare.io/redis

Riak KV, 2008. *Riak* [online]. [cit. 2021-4-18]. Dostupné z: https://docs.riak.com/riak/kv/2.2.3/index.html

MongoDB System Properties, © 2021. *DB-Engines* [online]. [cit. 2021-4-18]. Dostupné z: https://db-engines.com/en/system/MongoDB

Riak KV vs. Riak TS Comparison, © 2021. *DB-Engines* [online]. [cit. 2021-4-18]. Dostupné z: https://db-engines.com/en/system/Riak+KV%3BRiak+TS

ELK Stack: Elasticsearch, Logstash, Kibana, © 2021. *Elastic* [online]. [cit. 2021-4-18]. Dostupné z: https://www.elastic.co/what-is/elk-stack

Riak - Reviews, Pros & Cons | Companies using Riak, © 2021. *StackShare* [online]. [cit. 2021-4-18]. Dostupné z: https://stackshare.io/riak#description

Riak Documentation, 2020. *Riak* [online]. riak [cit. 2021-4-6]. Dostupné z: https://docs.riak.com/

Redis: Clients, © 2021. *Redis* [online]. San Jose: Redis [cit. 2021-4-4]. Dostupné z: https://redis.io/clients

Advantages of using MongoDB, © 2021. *Studytonight* [online]. [cit. 2021-4-18]. Dostupné z: https://www.studytonight.com/mongodb/advantages-of-mongodb

Elasticsearch System Properties, © 2021. *DB-Engines* [online]. [cit. 2021-4-18]. Dostupné z: https://db-engines.com/en/system/Elasticsearch

Elastic customer stories of all shapes and sizes, © 2021. *Elastic* [online]. [cit. 2021-4-18]. Dostupné z: https://www.elastic.co/customers/success-stories?usecase=enterprise-search

Our Customers | MongoDB, © 2021. *MongoDB* [online]. [cit. 2021-4-18]. Dostupné z: https://www.mongodb.com/who-uses-mongodb

Redis: Introduction to Redis, © 2021. *Redis* [online]. San Jose: Redis [cit. 2021-4-4]. Dostupné z: https://redis.io/topics/introduction

Firebase Realtime Database | Store and sync data in real time, 2021. *Firebase* [online]. [cit. 2021-4-18]. Dostupné z: https://firebase.google.com/products/realtime-database/

Firebase Realtime Database, 2020. *Firebase* [online]. [cit. 2021-4-18]. Dostupné z: https://firebase.google.com/docs/database

Distribution Models and Consistency, 2015. In: *Centro de Informática UFPE* [online]. Recife: Centro de Informática UFPE [cit. 2021-4-4]. Dostupné z: https://www.cin.ufpe.br/~if695/arquivos/leituras/NoSQL_Dist%20-%20Distribution%20Models%20and%20Consistency.pdf

Cassandra System Properties, © 2021. *DB-Engines* [online]. 2021 [cit. 2021-4-18]. Dostupné z: https://db-engines.com/en/system/Cassandra

Cassandra - Reviews, Pros & Cons | Companies using Cassandra, © 2021. *StackShare* [online]. [cit. 2021-4-18]. Dostupné z: https://stackshare.io/cassandra

Apache HBase – Apache HBase™ Home, 2021. *Apache HBase* [online]. [cit. 2021-4-18]. Dostupné z: http://hbase.apache.org/

HBase System Properties, © 2021. *DB-Engines* [online]. [cit. 2021-4-18]. Dostupné z: https://db-engines.com/en/system/HBase

HBase Pros and Cons | Problems with HBase, © 2021. *DataFlair* [online]. [cit. 2021-4-18]. Dostupné z: https://data-flair.training/blogs/hbase-pros-and-cons/

HBase - Reviews, Pros & Cons | Companies using HBase, © 2021. *StackShare* [online]. [cit. 2021-4-18]. Dostupné z: https://stackshare.io/hbase

Apache Hadoop, © 2021. *Apache Hadoop* [online]. [cit. 2021-4-18]. Dostupné z: http://hadoop.apache.org

Neo4j System Properties, © 2021. *DB-Engines* [online]. [cit. 2021-4-18]. Dostupné z: https://db-engines.com/en/system/Neo4j

IT slovník, 2017. *IT SLOVNÍK* [online]. [cit. 2021-4-3]. Dostupné z: https://it-slovnik.cz

Neo4J - Features & Advantages - Tutorialspoint, © 2021. *Tutorialspoint* [online]. [cit. 2021-4-18]. Dostupné z: https://www.tutorialspoint.com/neo4j/neo4j_features_advantages.htm

Introduction - Operations Manual, © 2021. *Neo4j* [online]. [cit. 2021-4-18]. Dostupné z: https://neo4j.com/docs/operations-manual/current/introduction/

MapReduce Tutorial, 2020. *Apache hadoop* [online]. [cit. 2021-4-18]. Dostupné z: https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Purpose

Neo4j Customers, 2021. *Neo4j* [online]. [cit. 2021-4-18]. Dostupné z: https://neo4j.com/customers/

[online]. In: . [cit. 2021-2-19].

Apache Hadoop YARN, 2020. *Apache hadoop* [online]. [cit. 2021-4-18]. Dostupné z: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

Apache Spark, 2013. *Databricks* [online]. San Francisco [cit. 2021-4-18]. Dostupné z: https://databricks.com/glossary/what-is-apache-spark

The Benefits of JanusGraph, © 2021. *JanusGraph* [online]. [cit. 2021-4-18]. Dostupné z: https://docs.janusgraph.org/

What is NoSQL?, 2021. *MongoDB* [online]. [cit. 2021-2-17]. Dostupné z: https://www.mongodb.com/nosql-explained

JanusGraph, © 2021. *JanusGraph* [online]. [cit. 2021-4-18]. Dostupné z: https://janusgraph.org/

Technical Limitations - JanusGraph, © 2021. *JanusGraph* [online]. [cit. 2021-4-18]. Dostupné z: https://docs.janusgraph.org/basics/technical-limitations/

JanusGraph System Properties, © 2021. *DB-Engines* [online]. [cit. 2021-4-18]. Dostupné z: https://db-engines.com/en/system/JanusGraph

ArangoDB, © 2021. *ArangoDB* [online]. [cit. 2021-4-18]. Dostupné z: https://www.arangodb.com/

ArangoDB System Properties, © 2021. *DB-Engines* [online]. [cit. 2021-4-18]. Dostupné z: https://db-engines.com/en/system/ArangoDB

ArangoDB - Reviews, Pros & Cons | Companies using ArangoDB, © 2021. *StackShare* [online]. [cit. 2021-4-18]. Dostupné z: https://stackshare.io/arangodb

What Is Big Data? | Oracle, 2021. *Oracle.com* [online]. USA [cit. 2021-2-13]. Dostupné z: https://www.oracle.com/big-data/what-is-big-data/#link5

Limitations | Transactions | Manual | ArangoDB Documentation, © 2021. *ArangoDB* [online]. [cit. 2021-4-18]. Dostupné z: https://www.arangodb.com/docs/stable/transactions-limitations.html

File input plugin, 2021. *Elastic* [online]. [cit. 2021-7-27]. Dostupné z: https://www.elastic.co/guide/en/logstash/current/plugins-inputs-file.html

Apache Spark - Introduction, © 2021. *Tutorialspoint* [online]. [cit. 2021-4-18]. Dostupné z: https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm

What is Apache Spark?, © 2021. *Databricks* [online]. San Francisco [cit. 2021-4-18]. Dostupné z: https://databricks.com/glossary/what-is-apache-spark

Apache Spark Ecosystem – Complete Spark Components Guide, 2018. *DataFlair* [online]. [cit. 2021-4-18]. Dostupné z: https://data-flair.training/blogs/apache-spark-ecosystem-components/

What is Apache Flink?, © 2021. *Cloudera* [online]. [cit. 2021-4-18]. Dostupné z: https://docs.cloudera.com/csa/1.2.0/flink-overview/topics/csa-flink-overview.html

JVM | Java Virtual Machine, © 2018. *Javatpoint* [online]. India [cit. 2021-4-18]. Dostupné z: https://www.javatpoint.com/jvm-java-virtual-machine

Apache Flink Ecosystem Components Tutorial | Learn Flink, 2021. *DataFlair* [online]. [cit. 2021-4-18]. Dostupné z: https://data-flair.training/blogs/apache-flink-ecosystem-components/

Mutate filter plugin, 2021. *Elastic* [online]. [cit. 2021-7-27]. Dostupné z: https://www.elastic.co/guide/en/logstash/current/plugins-filters-mutate.html

Apache Project Information, © 2020. *The Apache Software Foundation* [online]. [cit. 2021-4-18]. Dostupné z: https://projects.apache.org/project.html?kafka

Apache Kafka, © 2021. *VIP Trust* [online]. [cit. 2021-4-18]. Dostupné z: https://viptrust.com/technologie/ostatni/apache-kafka

Stdout output plugin, 2021. *Elastic* [online]. [cit. 2021-7-27]. Dostupné z: https://www.elastic.co/guide/en/logstash/current/plugins-outputs-stdout.html

Open Data Essentials, 2020. *World Bank Group* [online]. [cit. 2021-4-18]. Dostupné z: http://opendatatoolkit.worldbank.org/en/essentials.html

Co je vizualizace dat, 2014. *Oracle* [online]. [cit. 2021-4-18]. Dostupné z: https://www.oracle.com/cz/business-analytics/what-is-data-visualization/#link1)

Filebeat: Lightweight Log Analysis & Elasticsearch, © 2021. *Elastic* [online]. [cit. 2021-4-18]. Dostupné z: https://www.elastic.co/beats/filebeat

Logstash Introduction, © 2021. *Elastic* [online]. [cit. 2021-4-18]. Dostupné z: https://www.elastic.co/guide/en/logstash/current/introduction.html#power-of-logstash

What is Kibana?, © 2021. *Amazon Web Services* [online]. [cit. 2021-4-18]. Dostupné z: https://aws.amazon.com/elasticsearch-service/the-elk-stack/kibana/

Kibana—your window into Elastic, © 2021. *Elastic* [online]. [cit. 2021-4-18]. Dostupné z: https://www.elastic.co/guide/en/kibana/current/introduction.html

Csv filter plugin, 2021. *Elastic* [online]. [cit. 2021-7-27]. Dostupné z: https://www.elastic.co/guide/en/logstash/current/plugins-filters-csv.htm

Elasticsearch output plugin, 2021. *Elastic* [online]. [cit. 2021-7-27]. Dostupné z: https://www.elastic.co/guide/en/logstash/current/plugins-outputs-elasticsearch.html

Apache Spark Streaming Transformation Operations, © 2021. *DataFlair* [online]. [cit. 2021-7-27]. Dostupné z: https://data-flair.training/blogs/apache-spark-streaming-transformation-operations/

Firebase - Reviews, Pros & Cons, © 2021. *StackShare* [online]. [cit. 2021-7-27]. Dostupné z: https://stackshare.io/firebase

Hadoop vs Spark vs Flink – Big Data Frameworks Comparison, © 2021. *DataFlair* [online]. [cit. 2021-4-19]. Dostupné z: https://data-flair.training/blogs/hadoop-vs-spark-vs-flink/

Filebeat overview, © 2021. In: *Elastic* [online]. [cit. 2021-4-19]. Dostupné z: https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-overview.html