ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Slavíček** Jméno: **Ondřej** Osobní číslo: **420304**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Studijní obor: **Softwarové inženýrství**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Big Data ETL pro bankovní data

Název diplomové práce anglicky:

Big Data ETL for banking data

Pokyny pro vypracování:

Cílem diplomové práce je porovnání vybraných nástrojů pro Big Data ETL v oblasti zpracování bankovních dat. Autor se nejprve seznámí s existujícími technologiemi pro ETL nad klasickými daty i Big Data, provede jejich srovnání, identifikuje jejich vlastnosti, výhody a nevýhody. Nad několika vybranými nástroji bude implementováno ETL zpracování dat o produktech a transakcích za účelem vytváření vybraných sumárních reportů. Navržené řešení bude otestováno v rámci zkušebního provozu nad uměle vygenerovanými daty. Na základě výsledků provedených experimentů dojde ke srovnání jednotlivých ETL technologií, a to minimálně s ohledem na jejich škálování či další vlastnosti. Diskutován také bude vztah pracnosti implementace a dosaženého výkonu.

Seznam doporučené literatury:

[1] Zomaya Albert Y., Sakr Sherif, Handbook of Big Data Technologies. ISBN: 978-3-319-49339-8, DOI: 10.1007/978-3-319-49340-4, Springer International Publishing AG, 2017
[2] Wiese, Lena, Advanced Data Management: For SQL, NoSQL, Cloud and Distributed Databases. ISBN: 978-3-11-044140-6, DOI: 10.1515/9783110441413, Walter de Gruyter GmbH, 2015
[3] Holubová Irena, Kosek Jiří, Minařík Karel, Novák David, Big Data a NoSQL databáze. ISBN: 978-80-247-5466-6. Grada Publishing, a.s., 2015.
[4] Gartner, Magic Quadrant for Data Integration Tools, 2017. <https://www.gartner.com/home>
[5] Noel Yuhanna, The Forrester Wave?: Big Data Fabric, Q4 2016, A Critical Platform For Enterprises To Succeed With Big Data Initiatives, 2016. <https://www.forrester.com/report/The+Forrester+Wave+Big+Data+Fabric+Q4+2016/-/E-RES132141>

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Martin Bém, Adastra, s.r.o.

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **17.01.2018** Termín odevzdání diplomové práce: _____

Platnost zadání diplomové práce: **30.09.2019**

Ing. Martin Bém podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

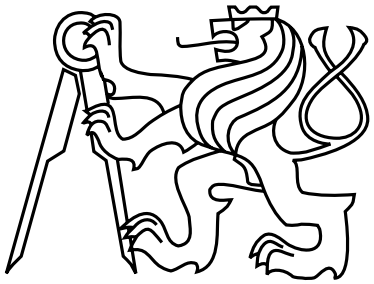
III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Diplomová práce

Big Data ETL pro bankovní data

Bc. Ondřej Slavíček

Vedoucí práce: Ing. Martin Bém

Studijní program: Otevřená informatika, Magisterský

Obor: Softwarové inženýrství

21. května 2018

Poděkování

Rád bych poděkoval vedoucímu mé diplomové práce Ing. Martinu Bémovi za technické a inspirativní rady, jeho ochotu a čas, který mi věnoval. Dále bych chtěl poděkovat svým blízkým za pomoc a trpělivost při mém studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 24. 5. 2018 .....

Abstract

The new data have grown in last few years. The new coming data is unstructured, has a large volume and is generated very quickly. Based on this fact, a new concept Big Data has been created. It offers capabilities to process new generated data. The diploma thesis aims to describe Big Data and Big Data processing with ETL. The solutions for ETL processing Big Data are compared. There are native Big Data processing based on MapReduce framework, a specialized ETL tool for Big Data processing by Talend and standard ETL tool supports Big Data by Hitachi Vantara. ETL tools are compared based on scalability for the increasing volume of processed data and man-days needed to implement the solution. The results show that native ETL processing offers better performance than other solutions. On the other hand, implementing a native solution requires more effort.

Keywords: Big Data, ETL, Hadoop, MapReduce, Talend, Pentaho

Abstrakt

V posledních letech narostlo množství nově vznikajících dat. Vznikající data jsou v zásadě nestrukturovaná, mají velký objem a jsou vytvářena velmi rychle. Na základě toho vznikl nový koncept Big Data, který nabízí možnosti zpracování těchto dat. Cílem této práce je popsat koncept Big Data a způsob zpracování Big Data datovou pumpou ETL. V práci jsou porovnány dostupná řešení ETL zpracování. Porovnáváno je nativní zpracování Big Data pomocí MapReduce, specializovaný nástroj na zpracování Big Data formou ETL od Talendu a standardní ETL nástroj s podporou Big Data od Hitachi Vantara. Nástroje jsou porovnány na základě škálovatelnosti vůči zvětšujícímu se objemu zpracovávaných dat, následně je diskutována pracnost řešení vůči dosaženému výkonu. Bylo zjištěno, že nativní ETL zpracování nabízí mnohem větší výkon než ostatní řešení. Na druhou stranu implementace nativního řešení vyžaduje větší pracnost.

Klíčová slova: Big Data, ETL, Hadoop, MapReduce, Talend, Pentaho

Obsah

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

Dnešní svět závisí na datech. Na datech, která jsou nestrukturovaná, nestálá, jejichž objem je obrovský a rychlost s jakou jsou generována se neustále zvětšuje. V důsledku tohoto rozvoje bylo zapotřebí vymezit novou oblast, která se zabývá zpracováním těchto dat. Z tohoto důvodu vznikla Big Data. Díky novým formátům a datům, která jsou standardně nezpracovatelná analytickými nástroji je zapotřebí se věnovat převodu těchto dat do srozumitelné podoby. Jedním z řešení tohoto problému je zpracování Big Data ETL.

Důvodem výběru tématu diplomové práce je jeho aktuálnost. Do budoucna se data budou neustále rozvíjet, budou více složitá, problematická. Díky novému přístupu lze získat z dat novou přidanou hodnotu. Tato přidaná hodnota pak může ovlivňovat trh financí, práce atd., ale i každodenní život uživatele aplikací.

Cílem práce je porovnat dostupné nástroje pro zpracování Big Data formou ETL. Tohoto porovnání je dosaženo pomocí měření rychlosti zpracování dat na základě škálovatelnosti vůči zvětšujícímu se objemu dat. Dále jsou nástroje porovnány na základě pracnosti nutné k implementaci řešení vůči dosaženému výkonu zpracování dat.

Struktura práce je systematicky rozdělena do jednotlivých kapitol. Úvodní kapitoly jsou věnovány teoretické části, které přecházejí v praktickou část.

V rámci teorie je vymezen termín Big Data, jejich základní definice a specifikace vlastností. Jsou rozebrány jednotlivé oblasti využití Big Data a jejich možné přínosy. Rovněž je v práci popsána metodika a principy zpracování Big Data, architektura zpracování. Dále je definována datová pumpa ETL a popsány aktuální nástroje pro zpracování Big Data formou ETL.

V praktických kapitolách jsou popsány jednotlivé nástroje, které jsou porovnávány. Definována metodika porovnání a specifikace testovacích dat. Následuje samotné porovnání ETL nástrojů a srovnání naměřených hodnot při faktickém měření.

Závěr práce obsahuje vyhodnocení porovnání Big Data ETL nástrojů.

Kapitola 2

Big Data

V dnešní době jsme obklopeni velkým množstvím digitálních dat a jejich zdrojů. Produkce dat ve světě každým dnem roste. Ze studie organizace IDC „The Digital Universe of Opportunities“ je zřejmé, že produkce nových dat každoročně naroste o 40 %. Na základě tohoto předpokladu lze v roce 2020 počítat s objemem dat až 44 zettabyte dat celosvětově (1 ZB = 10^{21} byte). [11]

Se zvyšujícím se počtem uživatelů sociálních sítí, internetových a mobilních aplikací, rozvojem nových technologií, a z nich vycházejících služeb, vznikají nové datové zdroje, které je zapotřebí efektivně zpracovat a uložit. [22] Nárůst celkového počtu vytvářených dat není žádnou novinkou, problémem je rychlost růstu jejich objemu, který je až exponenciální. [10]

Nová data vznikají na základě činnosti lidí, kteří je tvoří vědomě. Data vytváří také chytrá zařízení, která jsou připojena k internetu. Dalšími tvůrci dat jsou uživatelé aplikací, webových stránek, služeb atd. Pro tento zdroj dat je možné měřit různé statistiky, např. prokliky na webových stránkách nebo počet odeslaných e-mailů. Samotný internet představuje jeden z největších zdrojů dat. Dle online statistik „Internet live stats“, bylo dne 6.2.2018 průměrně během jedné náhodné sekundy vystaveno 7 924 tweetů na sociální síti Twitter, nebo například odesláno více než 2 600 000 e-mailů a provedeno 65 157 vyhledání na Googlu. [65]

Zmíněné zdroje dat poskytují nové možnosti získávání cenných informací. Problém nastává ve zpracování dat jako takových, které musí být vzhledem k jejich objemu rychlé. Dalším problémem je různorodost těchto dat. Tradiční databázová řešení zpracovávají strukturovaná data. V této nové oblasti mluvíme o datech částečně strukturovaných (např. XML, JSON, textové dokumenty) nebo nestrukturovaných (např. audio, video). [22] Tyto nestrukturovaná data budou v roce 2020 tvořit odhadem 90 % všech dat. Díky nestrukturovanosti nově vznikajících dat je zapotřebí, aby vznikala další data, tzv. metadata. Během tvorby dat vznikají i data nežádoucí, tzv. šum, který nemá žádné využití. [11]

Problémy se zpracováním obecně nestrukturovaných dat o velkém obejmu vedly k zavedení nového termínu Big Data.

2.1 Definice Big Data

Jak poznat, co už jsou Big Data a co ne? Jaké jsou formální specifikace pojmu Big Data a jaké technologie jsou využívány pro jejich zpracování? Formální definici pro Big Data nelze jednoduše vymezit [22]. Každý si může tuto problematiku vyložit dle své aktuální situace, projektových potřeb a objemu zpracovávaných dat. Díky tomuto faktu lze vymezit mnoho definic, neboť každý autor si vykládá problematiku Big Data jiným způsobem.

Jak již bylo zmíněno obecnou definici Big Data nenajdeme. Jednou z možností, jak určit co jsou Big Data, je přijmout definice výzkumných poradenských společností nebo společností, které mají pro zpracování Big Data hotová funkční řešení.

Významná výzkumná a poradenská společnost v oblasti IS/ICT technologií Gartner mluví ve svých publikacích o Big Data následovně:

„Termín Big Data jsou všechna aktiva společnosti v podobě získaných informací. Tyto informace mají rozličnou datovou strukturu, obrovský objem a je zapotřebí je rychle zpracovávat. Pro zpracování těchto informací je zapotřebí vytvořit nové formy zpracování dat, které jsou schopny podpořit rychlé zpracování, lepší rozhodování, objevování hodnot v datech a optimalizaci procesů.“ [49]

„Za Big Data lze považovat soubory dat takové, které svou velikostí překonávají možnost je zachytit, spravovat a zpracovat běžně používanými softwarovými nástroji v rozumném čase.“ [4]

V těchto specifikacích lze narazit na určité nedostatky, jako je například „rozumný čas“, který pro některé aplikace může být v řádech desítek minut, pro jiné v řádech vteřin.

Big Data ale nejsou pouze o velikosti dat, informacích a nových technických problémech, které je zapotřebí vyřešit. Big Data jsou také hlavně o nových možnostech využití dat a získání nové přidané hodnoty ze získaných dat.

„Big Data nabízejí společnostem využívat informace novými způsoby, čímž mohou produkovat nové užitečné poznatky, zboží nebo služby s velikou potencionální hodnotou. Big Data nově poskytují možnost provádět operace ve velkém měřítku, které dříve v malém nešly. Za příklad lze vzít extrakci nových poznatků nebo generování přidané hodnoty způsoby, které změní trh, samotnou společnost, vztahy mezi občany a úřady atd.“ [10]

Všeobecně uznávaná je definice na základě klíčových vlastností Big Data 3Vs z anglického: Volume (objem), Variety (rozmanitost), Velocity (rychlost). S touto definicí přišla přední světová společnost v oboru informačních technologií IBM. Definice je založená na zkušenostech z praxe. [50]

K této definici se přiklání většina dalších společností a autorů publikací o Big Data. Definice je dále rozšiřována o další V, někteří hovoří až o 10 klíčových vlastnostech. [19]

Oracle, jeden z velkých hráčů na trhu pro zpracování dat, využívá pro definici Big Data 3Vs model a rozšiřuje jej o další vlastnost Value (hodnota dat pro společnost) [68]. Společnost SAP uvedla na svém blogu Digitalist Magazine rozšíření základní definice 3Vs o vlastnosti Value a Veracity (věrohodnost dat) [15]. Internetový blog insideBIGDATA, psaný odborníky z praxe, kteří se zabývají problematikou Big Data, Cloudu atp., definuje Big Data pomocí 6V, a to: Volume, Variety, Velocity, Veracity, Validity (doba platnosti), Volatility (doba uložení) [38].

Obrázek 2.1: Klíčové vlastnosti Big Data. [19]

Samozřejmě existují i definice, které jsou založené na jiných klíčových vlastnostech. Jednou z těchto definic je definice na základě 3C z anglického: Cardinality (kardinalita), Continuity (kontinuita), Complexity (složitost) [51]. Obě verze zmíněných definicí se zaměřují pouze na samotná data. Nicméně mnohem důležitější je, že termín Big Data se váže i na technologie a architektury, se kterými pracují. [9]

2.1.1 Charakteristika Big Data dle 3Vs a dalších vlastností

V této části se zaměřím na popis klíčových vlastností modelu 3Vs, který je považován za nejznámější a nejuznávanější v oblasti Big Data. Rozvedeny budou také další vlastnosti rozšiřující tento model.

2.1.1.1 Objem – Volume

Objemem je myšlena celková velikost datového souboru nebo množství aktuálně dostupných dat, jejichž počet narůstá exponenciálně. [22]

Big Data obsahují obrovské objemy dat. V dnešní době jsou data generována stroji, sítěmi a lidskou interakcí na systémech, jako jsou sociální média atd. [38] Big Data vyžadují zpracování velkých objemů dat, které mohou být nestrukturovaná, tj. neznámé hodnoty, toky prokliků na webové stránce nebo v mobilní aplikaci, síťová komunikace, snímače zachycující data a mnoho dalších. Úlohou Big Data je přeměnit takové údaje na cenné informace. [22] Limitním objemem, který lze považovat za Big Data, je tak velká datová sada, kterou nelze smysluplně zpracovat tradičními technologiemi. [22]

Na druhou stranu je celkový objem dat v Big Data je relativní. Nelze přesně definovat, jak velký objem musí být. [8] Pro některé organizace to mohou být desítky terabytů, pro jiné

až tísíce petabytů [68]. Přesná velikost objemu dat není jasně určená a s vývojem nových technologií se hranice jeho velikosti posouvá [40]. Proto tedy není možné konstatovat fakt, že to, co je požadováno za Big Data dnes, bude za Big Data považováno i v následujících letech [17]. Kvůli nejasnostem týkajících se velikosti objemu dat je termín Big Data často považován za nesprávný a zavádějící označení. Více než na velikosti samotných dat záleží na jejich složitosti a dalších charakteristických vlastnostech. [76]

Pro jednodušší představu, o jak velký objem dat se jedná, ho lze přirovnat k objemu, který nelze uložit na jeden databázový server, ale pro jehož uložení je zapotřebí několik desítek nebo stovek databázových serverů. [22]

Možnost zpracování velkého objemu dat znamená ve většině případů výhodu pro budoucí analýzu. Standartní přístup k analýze dat zahrnuje vybrání určité množiny vzorků, na kterých se analýza provede. Na rozdíl od tohoto donedávna standardního přístupu, Big Data zpracovává všechna data, která jsou k dispozici bez ohledu na jejich množství. [33] Díky tomuto faktu bude výsledek analýzy nejaktuálnější a výsledek lze brát jako nejvíce prokazatelný, jelikož je k dispozici mnohem větší počet vzorků dat. [36]

Problematice práce s velkým objemem dat se v publikaci *„3D Data Management: Controlling Data Volume, Velocity and Variety“* věnuje společnost META Group (nynější Gartner). Dle dané publikace je při práci s velkými objemy dat zapotřebí věnovat se těmto segmentům [32]:

- **Data** – výběr dat, která jsou získávána
- **Datové zdroje** – přizpůsobení datových zdrojů k extrakci
- **Datové toky** – monitoring datových toků

2.1.1.2 Rychlost – Velocity

Rychlostí je myšlena dynamika, s jakou jsou nová data přijímána, jak rychle vznikají a jak rychle nastává jejich změna [50]. Big Data jsou závislá zejména na rychlosti, kterou přichází datové toky ze zdrojů [68]. Tok dat je masivní a kontinuální [38].

S rostoucím objemem dat roste i rychlost, kterou jsou data generována a přijímána ze zdrojových systémů. Pro zpracování a analýzu těchto dat je tedy zapotřebí mít nástroje, které dokáží rychle plynoucí data (streamovaná data) využít k nalezení nových obchodních příležitostí, vytěžit z dat maximální možnou užitečnou hodnotu. Možnost zpracovávat tato data je jednou z obrovských výhod celé technologie Big Data. [36]

Množství dat narůstá velmi rychle, rychlost nárůstu může být až exponenciální. Je tedy nutné data zpracovávat velmi rychle [22]. Některé aplikace vyžadují zpracování v reálném čase, je zapotřebí rozlišovat, zda data zapisovat do paměti nebo na disk [68].

V minulosti bylo běžně využíváno dávkové zpracování pomocí statických kroků, např. byly databáze aktualizovány každou noc. Zpracování dat a aktualizace databází zabírala mnoho času. V poslední době se začal přikládat velký důraz k rychlosti zpracování dat, který bude s vývojem nových technologií ještě větší. V dnešní době s mnoha novými možnostmi zdrojů dat vznikají data v reálném čase nebo téměř reálném čase, proto je zapotřebí je také v reálném čase zpracovávat. Bude výzvou pro každou společnost zda data vytvářená obrovskou rychlostí dokáže zpracovat. [34] [17]

Správné pochopení Big Data a získání jejich přidané hodnoty je považováno za schopnost, která přináší velkou konkurenční výhodu. Schopnost reagovat agilně na změny v datech a vývoj nových událostí je jednoznačné plusem pro každou společnost. [50]

Rychlostí není myšlena pouze rychlost nárůstu počtu dat a průchodu celým systémem, ale i to, jak rychle jsou data zpracována a analyzována. Rychlost zpracování dat lze rozdělit na tyto segmenty [10]:

- **Real-time** – Zpracování dat v reálném čase. Data, která přicházejí jsou neustále zpracovávána a analyzována v reálném čase.
- **Stream** – Data, která přicházejí jsou zpracována okamžitě po přijetí. Podobné zpracování jako real-time.
- **Near Real-time** – Zpracování dat, které přicházejí velmi malou chvíli po tom, co byla obdržena. Dochází k tzv. skoro real-time zpracování.
- **Batch** – Data jsou zpracována v určitém nastaveném časovém intervalu po jejich přijetí.

Dříve si nebylo možné představit způsob, jak analyzovat data o velikosti několika petabajtů. Vývojáři technických řešení přemýšleli, jak pomocí dostupného hardwaru tato data zpracovat. Z tohoto důvodu vznikla Big Data. Pokud se zaobíráme pouze rychlostí vzniku dat, lze mluvit o Fast Datech, podskupinou Big Data. [23]

Fast Data jsou generována v neuvěřitelných rychlostech, streamovaná data, finanční data, agregace záznamů nebo údaje ze senzorů. Data vznikají tisíckrát až desetitisíckrát za vteřinu. [23] Díky této vlastnosti je zapotřebí se na základě dat rozhodovat během několika milisekund, jelikož data v této situaci nejsou měřeny na objem terabajtů a petabajtů, ale na objem z hlediska času: megabajty za vteřinu, gigabajty za hodiny. [23] [71]

Samotná Big Data mohou být v zásadě klidná a zpracovávaná dávkově ve velkém objemu. Na rozdíl tomu Fast Data je zapotřebí zpracovat okamžitě, proto lze tuto skupinu vyčlenit. [23]

Hodnota Fast Dat je ztracena, pokud nejsou data zpracovány okamžitě. Pro potřebu zpracování těchto velice rychle vznikajících dat vznikly nové technologie. Základním kamenem pro zpracování Fast Data jsou streamovací technologie, které dokáží data rychle přenést. Dnes se využívá hlavně Apache Storm a Apache Kafka. Další nutnou technologií je uložiště, které dokáže obdržený záznam okamžitě zpracovat. [23]

2.1.1.3 Různorodost – Variety

Při sběru dat je nutné si uvědomit fakt, že ne všechna data mohou být ve vhodném formátu pro následné zpracování a provedení analýzy. Za poznávací znak Big Data lze považovat rozdílné zdroje s odlišnými datovými strukturami. [36]

Třetí základní vlastností je různorodost dat. Ta popisuje heterogenitu dat s ohledem na jejich typ, reprezentaci a sémantickou interpretaci. [2]

Dříve jsme ve standardních relačních databázových systémech zpracovávaly pouze strukturovaná data, což je např. jasně definovaná tabulka. V oblasti Big Data se zabýváme zpracováním dat, která jsou nestrukturovaná případně částečně strukturovaná. [22] Rozmanitost dat a jejich struktury odpovídá množství různorodých zdrojů [38]. Jedná se o nové nestrukturované a částečně strukturované datové typy. Přesto pro pochopení obsahu je zapotřebí, aby měly i nestrukturované záznamy některé shodné atributy, jako je tomu u strukturovaných dat, např. shrnutí, počet řádků, auditní atributy. [68]

Nestrukturovaná data definuje společnost Gartner jako:

„Nestrukturovaný obsah je takový, který není ukládán v souladu s předem definovaným datovým modelem popisující strukturu. Tento obsah není primárně určen pro ukládání do databázových tabulek a je vysoce orientován na lidi, kteří ho generují.“ [4]

Nestrukturovaný obsah může mít mnoho podob, jako je například e-mailová komunikace, obchodní dokumenty, webový obsah, obrazové nebo zvukové záznamy, příspěvky ze sociálních sítí, záznamy o GPS poloze, prokliky na webových stránkách atd. Tento obsah má jednu společnou vlastnost, není omezený pevnou strukturou. Záznamy tohoto typu většinou obsahují velké množství textu, který ale nemusí být ve srozumitelné podobě. Big Data mají za úkol z těchto dat vytěžit maximum, uspořádat data do vhodné podoby pro následné zpracování a analýzu, oddělit šum (nepoužitelné, poškozené nebo zbytečné údaje) a následně data zpracovat pomocí vhodných nástrojů. [2]

Podíl strukturovaných a nestrukturovaných dat je v poměru přibližně 1:80. Nestrukturovaných dat je naprostá většina - 80 až 90 %. Ve své surové podobě nejsou užitečná, cílem je získat z nich informace pro další použití. [21]

Různorodost dat lze rozlišit na základě datové struktury. Strukturovanost dat dělíme následovně [22]:

- **Strukturovaná** – Nejjednodušší forma dat. Strukturovaná data obsahují čísla a písmena. Záznamy mají pevně stanovený formát a musí dodržovat jistou strukturu. Díky dodržování pevné formy jsou data efektivně spravovatelná relačními databázovými systémy. Vhodné pro okamžitou analýzu.

- **Nestrukturovaná** – Formáty dat, které nelze jednoduše analyzovat a skladovat pomocí standardních databázových nástrojů. Nejsou vhodné pro okamžitou analýzu, je zapotřebí data zpracovat jinými způsoby a až následně analyzovat. Jedná se o videa, fotografie, e-maily, data z IoT, data ze sociálních sítí atp. V současné době tato forma dat převažuje. [21]

- **Semi-strukturovaná** – Datové formáty, které jsou částečně strukturované. Některé části dat mohou mít pevně určenou strukturu, větší část je ale nestrukturovaná např. text. Dobrým příkladem semi-strukturovaných záznamů jsou logy ze zařízení. Log hardwarového zařízení má přesnou definici – záznam události je na novém řádku a ukončen středníkem. Každý záznam začne identifikací zařízení pomocí pěti prvních znaků, po identifikaci následuje výpis dat a kódu prováděné instrukce. Tato část je přesně definována, zbytek logu obsahuje nestrukturovaný výpis z aplikace, která hardwarové zařízení obsluhuje. [2] Přestože jsou data částečně strukturované, nelze je zpracovat klasickými databázovými nástroji, jelikož nemají strukturu organizovanou na základě relačního modelu. Jedná se o formáty XML, JSON, textové dokumenty atp.

- **Kombinovaná** – Kombinace výše zmíněných datových forem. Kombinací formátů dochází ke zvýšení požadavků na systém, který má data zpracovávat.

V již zmíněné publikaci společnosti META Group (nynější Gartner) je věnována pozornost i struktuře dat, jejich zdrojům a problémům s jejich zpracováním. Při práci s daty bylo doporučeno věnovat se těmto oblastem [32]:

- **Profilování dat** – Zpracování dat automaticky za účelem optimalizace a zvýšení datové kvality. [67]
- **Využívání univerzálních formátů** – JSON, XML atd.
- **Přístup k datové vrstvě** – Úprava přístupu k datové vrstvě pomocí mezivrstvy (např. Middleware) pro zjednodušení práce.
- **Distribuované dotazy** – Použití softwarových nástrojů, které podporují distribuované dotazy.
- **Metadata** – Řízení vzniku metadat. Metadata jsou data, která uchovávají informaci o datech. Jde o formu popisu struktury a obsahu. Slouží k jednoduššímu pochopení dat pro jejich následnou analýzu a interpretaci výsledků. V metadatech je uložena i informace o prováděných transformacích zdrojových dat při ukládání do databáze. Metadata podporují kontrolu kvality dat, je možná kontrola hodnot na vstupu. [67]
- **Enterprise Application Integration** – Integrace softwarových a hardwarových aplikací, integrace webových služeb atd. Integrace technologií v rámci celé společnosti za účelem jednoduššího řešení problému a definování doménového přístupu. [48]

2.1.1.4 Věrohodnost - Veracity

Věrohodnost dat se vztahuje k důvěře, zda jsou data čistá, zda nevznikají v datech nějaké abnormality. Při dodržování velké rychlosti zpracování velkého objemu dat je zapotřebí vymezit datovou strategii, která dokáže data udržet dostatečně čistá pro zpracování. [38] Zároveň je zapotřebí se zabývat konzistencí, úplností a přesností dat [22].

Termín věrohodnost na sebe váže informaci o tom, že analyzovaná data mohou obsahovat zkreslená, neúplná či jinak nedostatečná data. Věrohodnost je ovlivněna zdrojem či formátem dat, proto i kontrola dat před analýzou má různou úroveň a výsledky analýzy mohou být zkresleny kvalitou vstupních dat. [34]

Bezcennost dat znamená, že přicházející data jsou nesprávná. V oblasti Big Data je zapotřebí počítat s možností abnormalit a zvláštností v datech. V rámci sbíraných dat se nevyskytují pouze data, která jsou smysluplná a dávají prokazatelnou hodnotu určité analýze, ale také data, která s problematikou nesouvisejí nebo souvisejí pouze okrajově. Tato data pak mohou mít za následek špatné výsledky analýzy. Proto je důležité při zpracování dat brát ohled na jejich věrohodnost, zaměřit se na kvalitu a čištění dat, aby nedocházelo k hromadění „špinavých“ dat v systému. Cílem je shromažďovat a analyzovat pouze věrohodná data. [54] [17]

Společnost IBM uvádí informaci o tom, že každý třetí manager ne vždy důvěřuje informacím, na základě kterých dělá svá rozhodnutí. Například data ze sociálních sítí poskytují velké množství informací, některé z nich ovšem nemusí být prokazatelné. Při sémantické analýze textu nelze jednoduše rozpoznat sarkasmus nebo ironii. Věrohodnost tedy neoznačuje pouze důvěryhodnost dat, ale také jejich spolehlivost, přesnost a srozumitelnost. [8]

V klasických databázových systémech se věnuje velká pozornost předzpracování, čištění a filtrování dat. Přestože nejsou tyto procesy vždy zcela bezchybné, lze považovat výsledná data za konzistentní, úplná a čistá. V oblasti Big Data je standardem zpracování velkého množství dat z různých zdrojů, často v reálném čase. Z toho důvodu není prostor na jejich čištění a filtrování. V některých procesech je filtrování a čištění dat dokonce nežádoucí, jelikož snižuje jejich hodnotu. Některé systémy dopředu neví, jak data budou využívat, proto je ukládají v jejich surové formě, aby nepřišly o žádné informace. [22]

Pro dosažení dostatečné úrovně věrohodnosti je často zapotřebí použít optimalizační techniky a přístupy, které mohou být velmi náročné. Je tedy potřeba vzít v potaz, že data mohou být nekvalitní a nepřesná. Rozhodnutí, zda datům věřit, a na jejich základě rozhodovat, musí učinit samy společnosti, které data zpracovávají. V důsledku toho se objevují názory, jenž zpochybňují, zda má vůbec cenu Big Data zpracovávat, případně zda nejprve nevybrat, jaká data zpracovat a jaká ne. [2]

2.1.1.5 Hodnota dat – Value

Hodnota znamená pro společnosti nejdůležitější položku. Samotná data nemají téměř žádnou hodnotu. Hodnotu z dat je potřeba vytěžit a přeměnit na cennou informaci. Cílem každé analýzy je získat přidanou hodnotu, která je důležitá pro zvýšení efektivity firemních procesů, nebo je dále využívána v dalších procesech. Shromažďování velkého množství dat z různých zdrojů v různých formátech nabízí možnost získání velmi hodnotných informací, které ze standardních dat nelze získat. [13]

Hodnota zpracovávaných dat v oblasti Big Data je důležitá pouze pro společnost, která je zpracovává [15]. Zpracovávané datové toky mají určitou vnitřní hodnotu. Tato hodnota musí být v datech nalezena. Hodnota dat je zcela individuální a každá společnost může využívat jiné informace. Pro nalezení hodnoty dat existuje řada analytických postupů, které ji dokážou odvodit. Za hodnotu dat lze považovat například spotřebitelské preference. Díky Big Data je možné analyzovat data kontinuálně, neboť existuje více vzorků, což umožňuje mnohem přesnější identifikaci cenných informací. [68]

Společnosti by se měly naučit shromažďovat a využívat Big Data. Big Data mohou přinést přidanou hodnotu ve velkém počtu oblastí. Například [54]:

- **Optimalizace procesů** – Zvýšení efektivity procesů, předpověď poptávky, změna ceny výrobků.
- **Preference zákazníků** – Poskytování doporučení zákazníkům na základě zjištěných preferencí.
- **Sport** – Chytrá sportovní zařízení, GPS.
- **Zdravotní péče** – Předpověď incidence chorob.

2.1.1.6 Limitovaná doba platnosti dat – Validity

Limitovaná doba platnosti dat udává, po jakou dobu jsou data platná pro svůj účel. Čistá a aktualizovaná data jsou základem úspěchu dobré analýzy dat. [38]

Doba platnosti poukazuje na fakt, že je důležité se zaobírat otázkou, zda jsou data časově vhodná pro zamýšlenou analýzu. [24]

Doba platnosti znamená časové období, po které jsou data platná a zůstávají uložena. Data jsou většinou přijímána v reálném čase. Je tedy potřeba určit, zda jsou data pro analýzu relevantní. [54]

2.1.1.7 Doba nutného uložení dat – Volatility

Dobou nutného uložení dat se rozumí, jak dlouho je nutné mít data uložena. Tato doba je úzce spojena s limitovanou dobou platnosti dat. Při rychlém zpracování dat v reálném čase je zapotřebí stanovit, zda jsou data pro danou analýzu ještě platná či nikoli. Je nutné definovat, jak dlouho mají být data uložena. Pokud jsou data pro analýzu nevalidní, nejsou již zapotřebí. [38]

Problematika Big Data se nezaměřuje pouze na sběr a ukládání dat, ale nastává zde problém s kapacitou uložiště, kterou není možné neustále navyšovat. Je potřeba ukládat pouze data, která jsou validní pro určitou problematiku či analýzu. Proto je nutné stanovit časovou dobu, po kterou mají být data ukládána, čímž eliminujeme narůstající objem dat, která jsou mazána a nahrazována novými. Nejsou tedy archivována pro pozdější využití. [24]

2.2 Zdroje dat pro Big Data

Zdroje pro Big Data jsou velice různorodé a specifické svými vlastnostmi. Uvedený fakt je potřeba zohlednit při začlenění dat ze zdroje do určité datové kolekce. Kombinací různých datových zdrojů lze získat novou přidanou hodnotu. Data, která se mají zpracovat, mohou nejprve vypadat bezcenně a až po kombinaci s jinou sadou dat získají hodnotu, a je možné z nich vytěžit přínosné informace. [5]

Některé datové zdroje pro Big Data již byly zmíněny, nové datové zdroje stále přibývají a je potřeba je aktualizovat. Zde uvádím výčet základních zdrojů [5]:

- **Multimédia** – Multimediální obsah v podobě fotografií, obrázků, videí, audio nahrávek atp.
- **Dokumenty** – Dokumenty formátů XML, JSON, XLS, CSV, PDF, DOC atp.
- **Sociální sítě** – Instagram, Facebook, LinkedIn atp.
- **Web** – Veřejně dostupný web, počasí, dopravní informace, finance, zdravotnické služby, úřady, světová banka atp.
- **Datová uložiště a sklady** – Relační databáze, souborové systémy, NoSQL databáze.
- **Archivy** – Archivované dokumenty, naskenované dokumenty, lékařské záznamy, korespondence, prohlášení atp.

- **Podnikové systémy** – CRM, ERP, intranet, automatizace, projektový management atp.
- **IoT data** – Senzorická data naměřená chytrými zařízeními (zařízeními připojenými do sítě), automobilové senzory, satelity, zdravotnická zařízení atd.
- **Strojová data** – Logy z aplikací, procesů, data na serverech atd.

Obrázek 2.2: Datové zdroje Big Data. [35]

2.3 Oblasti využití Big Data

Big Data jsou zatím rozšířena ve velkých společnostech nebo společnostech, které se potýkají s problémy ohledně zpracování velkého množství generovaných dat. Zjištěnou skutečnost dokazuje i výzkum provedený společností Accenture Analitics. [1]

Big Data lze využít v mnoha oblastech, které je zapotřebí rozlišovat charakteristikou dat. Data v různých odvětvích/oborech se značně liší v objemu, různorodosti, rychlosti, kterou vznikají, a dalších vlastnostech. Například data ve zdravotnictví se vyznačují velkou rychlostí, jsou velice různorodá (nestrukturovaný text, digitální obraz), ale mají v porovnaní s dalšími oblastmi malý objem. Data velkého objemu, vznikající velkou rychlostí, se vyskytují hlavně v oblasti bankovnictví, tato data jsou prakticky strukturovaná. Za idální Big Data lze považovat i multimediální data a data z telekomunikačních kanálů. Jejich objem, rychlost vzniku i různorodost je mnohem vetší než v ostatních oblastech. [36]

Následující tabulka poměřuje vlastnosti modelu 3Vs v jednotlivých oblastech a určuje potenciální přínos Big Data v dané oblasti.

Oblast	Objem	Rychlost	Různorodost	Přínos BD
Bankovnictví	Vysoký	Vysoká	Nízká	Vysoký
Komunikace	Vysoký	Vysoká	Vysoká	Vysoký
Vláda	Vysoký	Střední	Vysoká	Vysoký
Zdravotnictví	Střední	Vysoká	Střední	Vysoký
Výroba	Vysoký	Vysoká	Vysoká	Vysoký
Maloobchod	Vysoký	Vysoká	Vysoká	Vysoký
Vzdělání	Velmi nízký	Velmi nízká	Velmi nízká	Střední
Chemické zdroje	Vysoký	Vysoká	Vysoký	Střední
Pojišťovny	Střední	Střední	Střední	Střední
Doprava	Střední	Střední	Střední	Střední
Energitika	Střední	Střední	Střední	Střední

Tabulka 2.1: Oblasti využití Big Data a porovnání potencionálního přínosu Big Data na základě 3Vs modelu

Lze očekávat, že možnosti uplatnění a využití Big Data se budou rozšiřovat do všech oblastí lidské činnosti. Můžeme hovořit o revoluci v možnosti zpracování dat, jejich využití a nalézání nových přidaných hodnot, z čehož budou finančně získávat společnosti zpracovávající Big Data, ale i fyzické osoby, pro které budou k dispozici různé nové služby. Za příklad lze vzít společnosti, jenž Big Data již naplno využívají. V následujících odstavcích popíši některé příklady využití Big Data.

2.3.1 Finance a bankovnictví

Kapitálové trhy, akciové trhy či bankovní transakce generují obrovské množství dat, která se velmi podrobně analyzuje na základě různých technik dle typu dat. Analyzovaná data se dají využít k detekci podvodů, pro maximalizaci výdělku z obchodní činnosti, monitoring obchodů, řízení rizik nebo také pro segmentaci zákazníků do skupin a nabízení individuálních služeb. [36]

2.3.2 Multimedia a telekomunikace

Streamovaná hudba, videa a další data, které jsou v současné době velice rozšířená je potřeba analyzovat z důvodu zjištění preferencí uživatelů těchto služeb. Na základě analýzy lze uživatelům nabízet individuální služby. [36]

Příkladem může být velice zajímavý projekt personalizovaného radia Pandora, které na základě sesbíraných dat tvoří seznam skladeb podle dostupných preferencích daného uživatele. Hraje tak, aby se to uživateli líbilo. [20]

2.3.3 Sociální sítě

Sociální sítě jsou velmi populární, ale zároveň se stávají oblastí, ve které probíhá analytická činnost, jejímž výsledkem je cílená reklama, poskytování produktů a služeb uživatelům.

Za nejznámější a nejvíce propagovanou sociální síť, využívající Big Data, lze považovat Facebook, který používá sběr dat za účelem sledování chování a zájmů svých uživatelů. Na základě toho jsou zpracovávány odhady s doporučeními pro uživatele, například do jakých zájmových skupin se mají na základě svých zájmů přidat atd. [37]

Dalším příkladem může být pracovní sociální síť LinkedIn, která poskytuje online životopisy uživatelů, jejich vzájemné propojení a vazby. Zde jsou Big Data využívána jako zdroj pro propojení uchazečů o zaměstnání a nabídky pracovních příležitostí, a tím pomáhá personalistům ve vyhledání vhodných kandidátů na danou pozici. [39]

2.3.4 Zdravotnictví

Zdravotnictví a zdravotnické instituce mají k dispozici velké množství záznamů, které ve většině případů nejsou sdíleny. Tyto záznamy mohou být využity k hledání skryté přidané hodnoty, pomocí analýzy zdravotních záznamů lze nalézt pro pacienta optimální léčbu. [47] Zdravotnická zařízení, využívající senzory pro měření životních funkcí, mohou být využity k predikování zástavy srdce a dalších skutečností, které ohrožují pacienta na životě. Z širšího hlediska lze říci, že analýza Big Data pomůže zlepšit jak prevenci, tak i samotnou včasnou diagnózu a léčbu různých onemocnění. Analýzy lze použít i k predikování průběhu epidemií a jejich šíření např. pomocí sledování mechanického pohybu obyvatelstva. [10]

2.3.5 Věda a výzkum

Největším společností v této oblasti je CERN – Evropská organizace pro jaderný výzkum. Datový tok zde prováděných experimentů lze považovat za velice praktickou ukázku využití Big Data technologií ve vědě. Prováděné experimenty produkují až 25 GB/s dat, z kterých je ukládáno pouze 0,01 %. Pro analytickou činnost se zde používá např. Hadoop, Oracle DB. [6]

2.3.6 Stavebnictví

Tato oblast využití Big Data je úzce svázána s IoT (Internet věcí), jelikož jsou v chytrých stavbách zpracovávána data z měřících senzorů, na základě kterých jsou vyhodnocovány některé skutečnosti. Jako příklad lze uvést světelný a tepelný senzor, který upozorní systém na vysokou intenzitu slunečního svitu na západní straně domu, přičemž systém zareaguje zavřením rolet na oknech. Dalším příkladem může být monitoring volných parkovacích míst využívaný např. v nákupním centru Chodov Praha.

Most St. Anthonyho ve Spojených státech amerických obsahuje více než 200 sensorů zabudovaných na strategických místech stavby, které měří změny chování stavby v důsledku změn teplot. [75]

2.3.7 Vývoj techniky

V oblasti vývoji techniky se do popředí dostává oblast automobilizmu. Nové automobily jsou plné elektrických senzorů, které pomáhají řidičům, chrání posádku nebo kontrolují funkčnost celého vozu.

Pomyslnou špičkou ledovce v automobilismu jsou vozy Formule 1, které se skládají přibližně z 25 000 součástí, každá z nich pak představuje nějaké riziko poškození. Z tohoto důvodu jsou vozy podrobeny vysokovýkonostním testům, při kterých obsahující senzory na součástkách měří jízdní vlastnosti, vlastnosti motoru apod. Během Velké ceny USA v roce 2014 bylo shromážděno 243 TB dat. Komponenty pro vozy Formule 1 jsou vyráběny na základě datových analýz. [53]

Dalším zdrojem obrovského množství dat v oblasti techniky jsou dopravní letadla, která během jednoho letu mohou vygenerovat až 5 TB dat. [74]

2.3.8 Marketing

Big Data hrají velkou roli v marketingu, ať už je využívána metoda omni-kanálového nebo multi-kanálového marketingového přístupu. Díky získaným datům se lze jednoznačně zaměřit na uživatele a předložit mu nabídku na míru. [55]

Kapitola 3

Zpracování Big Data

Analytické nástroje pro zpracování dat kladou větší nároky na hardware, zejména se zvyšujícím se počtem zpracovávaných záznamů nebo složitostí výpočtů. Konkrétně se jedná o větší nároky na RAM paměť, výpočetní výkon procesoru nebo propustnost sítě.

Řešením tohoto problému je možnost škálování navýšením výkonu dostupných strojů pomocí nového hardwaru nebo zřízení clusteru, který distribuuje problémy na více uzlů, čímž docílíme toho, že výpočty budou probíhat paralelně. Tím se sníží nároky kladené na hardware. Další možností je využití cloudových služeb, které nabízí konfiguraci výpočetních jednotek a clusterů dle požadavků dané aplikace. Služby jsou zpoplatněny na základě objemu přenesených dat nebo využití procesorového času. Výhodou cloudových služeb je, že jsou data zálohována v datovém centru, uživatel tak nemusí řešit výpadky nebo poruchy hardwaru. [14]

Nejznámějšími produkty cloudového řešení jsou Microsoft Azure, Google Cloud Platform, Amazon Web Services, Oracle Cloud nebo IBM Cloud. [16]

Zpracovat Big Data není prakticky možné jiným způsobem než využitím distribuovaného přístupu. V této oblasti se pracuje s tak velkým množstvím dat, že běžné softwarové nástroje nejsou schopny tak velké množství dat pojmout a zpracovat v rozumném čase. [14]

3.1 Metodika zpracování Big Data

Základní přístup pro zpracování Big Data se odvíjí od distribuce problému na cluster propojených uzlů. Velikost clusteru se přizpůsobuje potřebám daného řešení problému. Uzly v clusteru mohou tvořit i běžné počítače, což sníží náklady na vytvoření clusteru, a přesto bude cluster mnohem výkonnější než jeden supervýkonný server. Distribuce problému do clusteru má ovšem svá omezení a problémy, například problémy s výpadky sítě, distribucí a konzistencí dat. [22]

3.1.1 Škálovatelnost

Škálování je z pohledu systému pro zpracování dat schopnost aktivně reagovat na změny požadavků na systém. V případě zpracování Big Data se jedná o celkovou zátěž systému a objem zpracovávaných dat. [22]

Standardní databázové systémy využívají k navýšení výkonu vertikální škálování. Dochází k navýšení výpočetního výkonu a upgradu hardwaru na úrovni příslušného serveru. Toto zvýšení výkonu vystačí mnoha aplikacím, ale nedochází k tak velkému nárůstu výkonu, aby bylo možné zpracovat Big Data. Přestože vertikální škálování vypadá jako jednoduchá možnost získání vyššího výkonu, a to pouze upgradem serverů, má i svá úskalí, kterými jsou [22]:

- **Vendor lock-in** – Výkonné servery jsou vyráběny malým množstvím specializovaných firem. Upgrade je nutné provádět u stejné firmy.
- **Náklady** – Výkonné servery jsou mnohem dražší než stanice s běžným hardwarem.
- **Omezení výkonu** – Každý server má i po značných upgradech určitá omezení, výkon tedy není možné neomezeně navyšovat.
- **Implementace** – Během implementace je nutné brát v potaz výkon serveru, použitého hardwaru a s ním i maximální možnou datovou velikost a propustnost dat.

Opakem vertikálního škálování je škálování horizontální, které distribuuje problém na více uzlů, čímž lze eliminovat hlavní nevýhody vertikálního škálování. V systému více uzlů (clusterů) lze pracovat s běžným levnějším hardwarem. Problém je počítán paralelně, čímž dochází k navýšení výkonu. Velikost clusteru nemá své omezení, cluster může obsahovat různé množství uzlů a pracuje vždy stejně. Velikost dat ke zpracování není nijak omezená. Nicméně ani toto řešení není zcela dokonalé. Horizontální škálování lze považovat za optimální řešení pouze, pokud síť clusteru splňuje následující podmínky [73]:

- **Spolehlivá síť** – 100% spolehlivost, bez výpadků
- **Nulové zpoždění na síti**
- **Neomezená šířka pásma**
- **Zabezpečená komunikace na síti**
- **Neměnná topologie sítě**
- **Administrátor sítě je pouze jeden**
- **Homogenní síť**
- **Nulové náklady na přenos dat**

Většiny těchto podmínek lze dosáhnout pouze za speciálních situací nebo jich dosáhnout nelze. Distribuované zpracování dat se snaží těmto podmínkám alespoň přiblížit. [22]

3.1.2 Konzistence

Pro efektivní a korektní zpracování dat je potřeba zajistit jejich konzistenci, tedy správnost a aktuálnost dat [22]. V závislosti na typu aplikace není vždy nutné považovat konzistenci za nejdůležitější vlastnost a lze se spokojit pouze s konzistencí občasnou. Pro velké množství aplikací je mnohem důležitější pracovat s daty rychle než dodržet jejich striktní konzistenci. Vynucený konzistentní stav dat v databázi zpomaluje práci s daty. [41]

Standardní databázové systémy pracují s integritními omezeními, která určují podmínky, jak mají data vypadat na základě požadavků dané aplikace. K dodržení integritních omezení a aktuálnosti dat jsou využívány transakce. Transakce jsou definovány jako sekvence logicky navazujících operací, které převádí data z jednoho konzistentního stavu do druhého. Během transakce mohou být data v nekonzistentním stavu. Důležité však je, aby byla opět konzistentní po jejím dokončení. Pro zachování konzistence dat je třeba, aby transakce splňovali určité vlastnosti, které jsou označovány na základě jejich počátečních písmen jako ACID [22]:

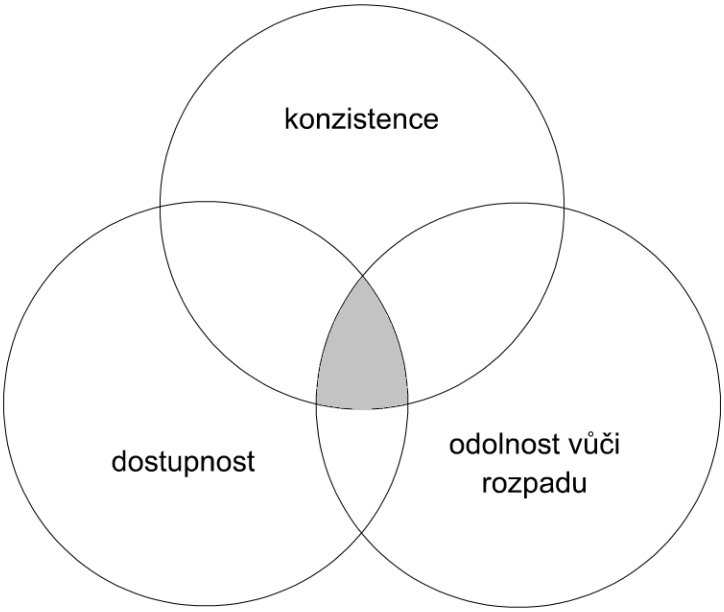
- **Atomicita (atomicity) transakce** – Transakce není dělitelná. Transakce proběhne celá nebo neproběhne vůbec.
- **Konzistence (consistency) dat** – Transakce zajistí přechod dat z jednoho konzistentního stavu do druhého.
- **Izolace (isolation) transakcí** – Transakce se vzájemně neovlivňují. Operace probíhající v jedné transakci jsou skryté před ostatními běžícími transakcemi.
- **Trvalost (durability) transakce** – Změny provedené transakcí se po jejím úspěšném dokončení uloží v databázi.

Díky ACID vlastnostem transakce nikdy nenaruší konzistenci dat v databázi. Aby byly tyto vlastnosti zachovány, je nutné řešit souběh a izolaci transakcí, jelikož nad databází pracuje více uživatelů najednou. [22]

Transakční zpracování s ACID vlastnostmi nelze využít v distribuovaném databázovém systému, jelikož by zpracování velice zpomaloval. Z důvodu distribuce dat, replikace a výpadků sítě by bylo dosažení konzistentního stavu velice náročné. Pro zpracování dat v distribuovaném prostředí se využívá přístup zvaný CAP teorém [7]:

- **Konzistence (consistency) dat** – V databázi je uložena pouze jedna aktuální verze dat.
- **Dostupnost (availability) dat** – Systém je vždy dostupný, zpracuje veškeré požadavky na čtení/zápis do systému.
- **Odolnost (partition tolerance) sítě vůči rozpadu** – Systém je funkční i po rozpadu na několik individuálních částí z důvodu výpadku sítě.

Jak je vidět na obrázku 3.1, ideálního stavu by bylo dosaženo průnikem všech tří vlastností. Dle Erica Brewera lze v distribuovaném prostředí však dosáhnout pouze dvou vlastností zároveň. [7]



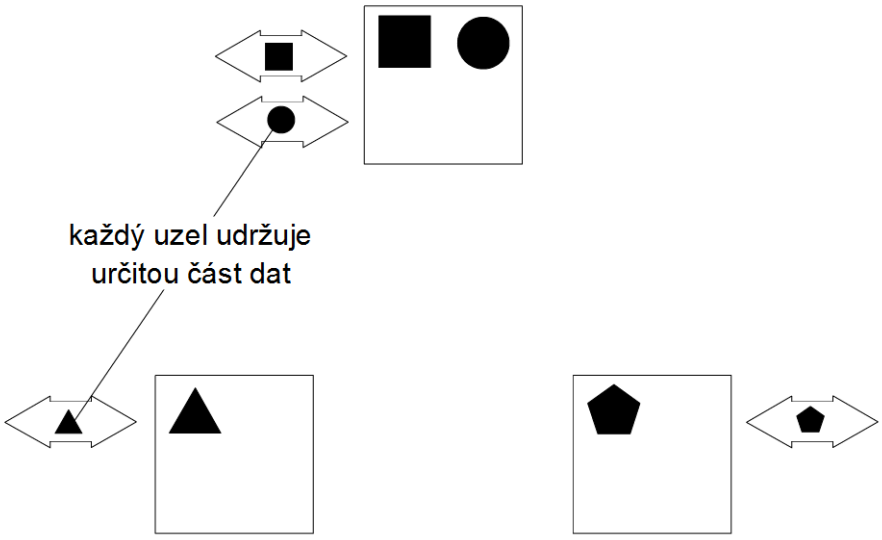
Obrázek 3.1: CAP teorém. [22]

Ve skutečnosti ale nelze distribuované prostředí prakticky využívat bez ošetření odolnosti vůči rozpadu sítě. CAP teorém spíše poukazuje na to, že při práci s distribuovaným systémem je potřeba snížit požadované nároky. Kvůli nepřesnosti CAP teorému je potřeba se zabývat občasnou konzistencí dat. Občasná konzistence dat je alternativou modelu ACID u standardního transakčního přístupu. Lze ji popsat podle modelu BASE – distribuovaný systém je po celou dobu užívání převážně dostupný. Systém je nedeterministický, dynamický a dochází v něm k neustálým změnám. V systému není zaručena neustálá konzistence. Díky těmto vlastnostem lze dosáhnout vysoké škálovatelnosti, a tím i navýšení výkonu systému, ovšem na úkor nižší konzistence dat. [22]

3.1.3 Distribuce

Jak již bylo zmíněno, zpracování Big Data je distribuované na více uzlů v clusteru, s čímž úzce souvisí i distribuce dat na uzly. Pro optimální distribuci dat jsou využívány dvě techniky nebo jejich kombinace. Jedná se o rozdělení dat (tzv. sharding) nebo replikaci dat. [45]

Sharding umožňuje rozdělit data na množiny, tzv. shards, a uložit je na jednotlivých uzlech clusteru (viz 3.2), což podporuje horizontální škálování, neboť uživatel přistupuje pouze na uzly, obsahující pro něj potřebná data. [45]

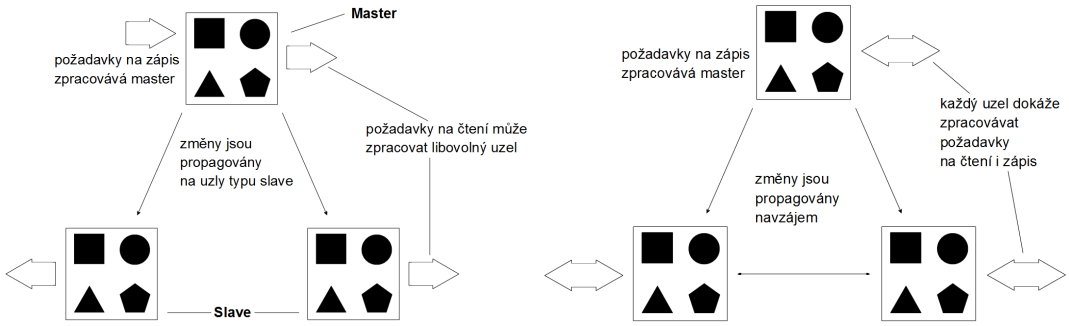


Obrázek 3.2: Vizualizace shardingu. [22]

Je patrné, že strategie rozmístění a uložení dat je velice důležitá pro dobrou efektivitu celého systému. Zpravidla je snaha rozmístit data tak, aby byla uložena mezi uzly rovnoměrně. Dále je potřeba minimalizovat počet uzlů, na které se při dotazech musí přistupovat, neboť související data se ukládají společně, a optimalizovat uložení dat na základě geografické příslušnosti k městu, zemi, firmě apod. Sharding nepočítá s možností výpadku sítě. Po výpadku některých uzlů jsou na nich uložená data nedostupná. Z tohoto důvodu se sharding často kombinuje s replikací. [45]

Druhou alternativou distribuce dat je replikace. Replikace slouží k uložení shodných dat na více uzlech, čímž se předchází výpadkům sítě. Pro replikaci lze využít jeden z osvědčených způsobů řízení práce s uzly, a to master-slave nebo peer-to-peer. Obě techniky replikace mají svá omezení. Například pokud dojde k souběhu transakcí, mohou se data na uzlech stát nekonzistentními. [45]

Master-slave replikace má jeden primární uzel a několik sekundárních uzlů. Primární uzel slouží k obsluze požadavků na zápis. Sekundární uzly obsluhují požadavky pro čtení. Nevýhodou tohoto přístupu je možnost vysokého zatížení primárního uzlu, v důsledku mnoha požadavků na zápis, a vznik tzv. bottlenecku, kdy je propustnost zápisu do systému stanovena výkonem primárního uzlu. Proto se tento přístup replikace hodí více pro systémy, v rámci kterých dochází spíše ke čtení, zatímco zápisy jsou prováděny minimálně. Jelikož všechny uzly obsahují shodná data, lze při výpadku primárního uzlu nahradit tento uzel jedním ze sekundárních uzlů, zatímco systém bude stále stabilně fungovat. [45]



Obrázek 3.3: Vizualizace replikace Master-slave a Peer-to-peer. [22]

Peer-to-peer replikace má všechny uzly na stejné úrovni. Všechny uzly tedy zpracovávají jak požadavky na čtení, tak na zápis. Tato technika odstraňuje problém zahlcení primárního uzlu, která hrozí u replikace master-slave. Při zápisech je nutno propagovat změnu dat mezi uzly, což zvyšuje nároky na komunikaci. Zároveň dochází ke zvýšení rizika možných konfliktů při změně stejných dat více uživateli nebo při čtení během nekonzistentního stavu. [45]

Poslední možností distribuce dat je kombinace shardingu a replikace dat. Data je nejprve potřeba rozdělit do množin dle strategických podmínek, následně jsou replikovány na více uzlů dle vybraného druhu replikace. [45]

3.2 Architektura řešení Big Data

V oblasti Big Data se zabýváme zpracováním dat dávkově a zpracováním dat v reálném čase. Nejdůležitějším je si uvědomit, že data jsou neomezená a stále v pohybu. Je zapotřebí si určit jaká data jsou důležitá pro zpracování v reálném čase a jaká data „stačí“ zpracovat dávkově. Zpracování Big Data přináší rozmanitost v objemu, rychlosti a struktuře dat. Díky tomu je vyžadována vysoká škálovatelnost, odolnost vůči chybám a předvídatelnost jaká data zpracovat jakým způsobem. [56]

Dvě nejznámější architektury jsou Kappa a Lambda architektura. [56]

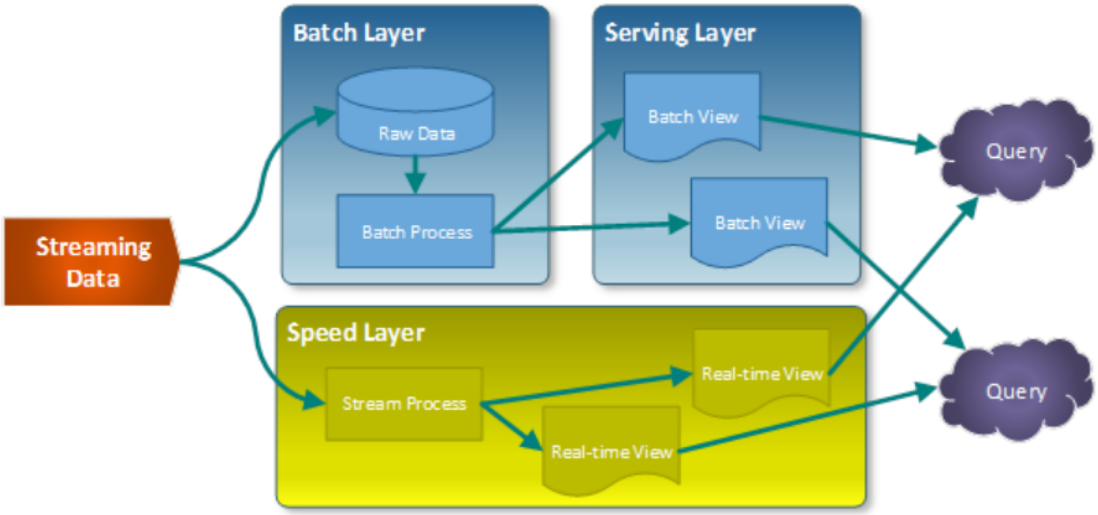
3.2.1 Kappa Architektura

Dnes jedna z nejběžnějších architektur pro zpracování dat v reálném čase navržená Nathanem Marzem. Navržena tak, aby odolávala chybám, měla nízkou odezvu a vysokou škálovatelnost. Kappa architekturu lze rozdělit na dvě vrstvy – dávkové zpracování, streamové zpracování. [56]

Vrstva pro dávkové zpracování ukládá surová data a následně je zpracuje tak, aby byla vhodná pro následnou konzumaci/analýzu, data jsou poskytována servisní vrstvou. Rozsah dat zpracovaných dávkou může být v řádu několika hodin až let. Streamová vrstva zpracovává příchozí data v reálném čase. [56]

Výsledný dotaz na data může využít informace z obou vrstev. Dávkové zpracování poskytuje informace, které jsou více komplexní a mají větší datovou kvalitu. Zatímco streamové

zpracování poskytuje aktuální data. Pokud data zpracovávané streamem pozbydou svou platnost, jsou nahrazeny daty z dávkové vrstvy. [56]



Obrázek 3.4: Kappa architektura. [56]

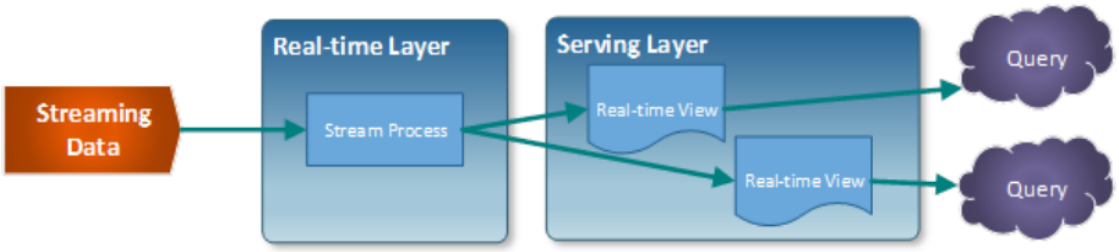
Největší výhodou této architektury je, že dokáže pracovat nad dávkově zpracovanými daty i při změně implementace a není omezena pouze na streamovaná data. Nevýhodou tohoto přístupu je, že je zapotřebí spravovat dvě vrstvy pro dávkové a streamované zpracování. [56]

3.2.2 Lambda Architektura

Poněkud jednodušší přístup volí Lambda architektura, která cílí pouze na zpracování streamovaných dat. Tato architektura byla navržena Jayem Krepsem. Data jsou zpracovávána jako jeden stream. Pokud dojde ke změně implementace, jsou původní data přehrána nově získanými. [56]

Tato architektura se pokouší zjednodušit Kappa architekturu tím, že udržuje pouze jednu vrstvu, dotazy jsou pak směrovány pouze na jedno uložiště dat. Za nevýhodu lze považovat samotné zpracování pouze streamovaných dat, které nejsou vhodné pro všechny dotazy. Například vícenásobné události, navazující události, údržba objednávek, které je jednodušší zpracovávat dávkou. [56]

Pro většinu řešení, která zpracovávají data v reálném čase je Lambda architektura vhodnějším řešením. Zejména pokud jsou analytické výsledky dávkového a streamového zpracování identické. Některé situace, kdy jsou výsledky analýzy dat zcela odlišné, vyžadují využití Kappa architektury. [56]

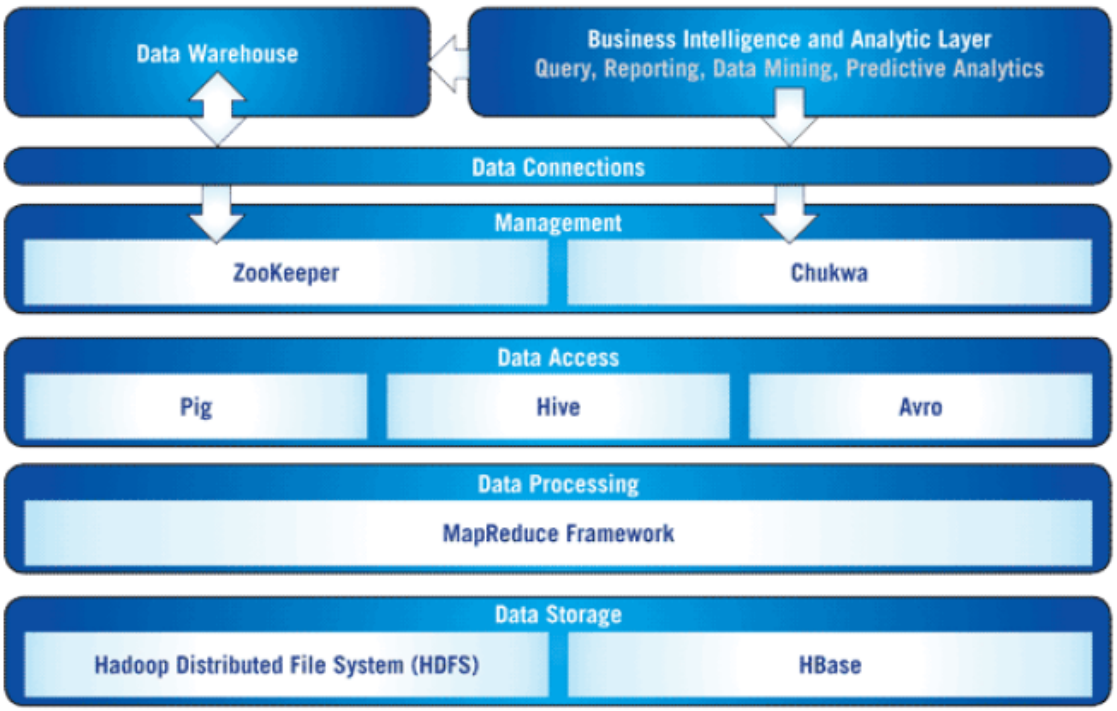


Obrázek 3.5: Lambda architektura. [56]

3.2.3 Architektura Apache Hadoop

Architektura samotného řešení Apache Hadoop je velice podobná tradičnímu řešení relačních databází, uložených v datových skladech. Lze ovšem nalézt i rozdíly, které jsou dané charakteristikou zpracovávaných dat. V oblasti Big Data je zapotřebí věnovat mnohem větší pozornost transformaci dat, aby bylo možné získat požadovaná data. [43]

Následující obrázek 3.6 zobrazuje architekturu systému Hadoop, kterou lze rozdělit do několika vrstev [46]:



Obrázek 3.6: Hadoop Big Data system. [46]

- **Datové uložiště** – Nestrukturovaná data

- **Zpracování dat** – Transformace
- **Datové uložiště určené pro přístup (datový sklad)) sítě vůči rozpadu** – Data vhodná k analýze
- **Správa přístupu**
- **Datová připojení**

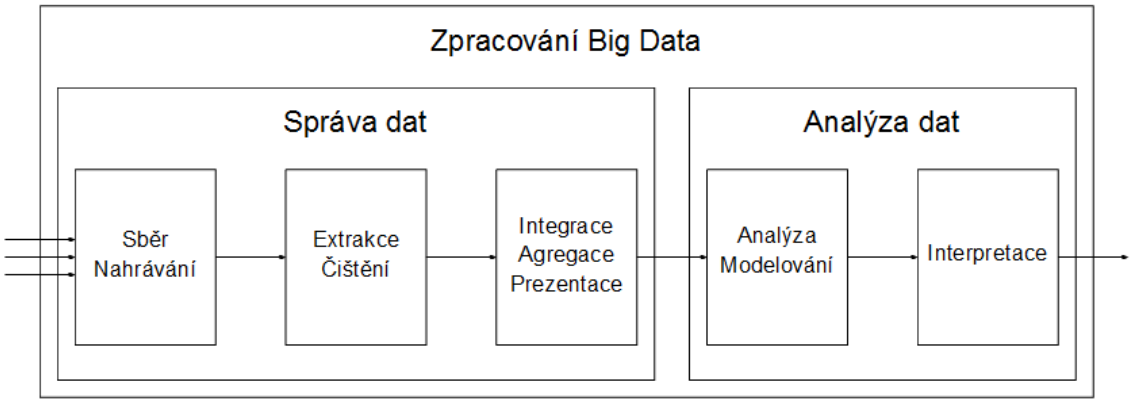
Architektura obsahuje uložiště podobné datovému skladu, který ukládá již zpracovaná data, ty jsou následně poskytnuta vyšším vrstvám za účelem analýzy dat. Mnohem důležitější jsou ale vrstvy pod datovým skladem, které umožňují provádět datové operace nad velkým množstvím nestrukturovaných dat. [43]

Princip řešení Big Data je u všech výrobců podobný schématu na obrázku 7. Všechna řešení jsou spojena komponentou MapReduce, pomocí které dochází k distribuovanému zpracování dat. Bezpochyby nejznámějším a nejvýznamnějším řešením je výše zmíněný systém Hadoop a z něj odvozené nástroje. Hadoop je využíván většinou současných open-source i komerčních řešení a stal se standardem v oblasti Big Data. [43]

Apache Hadoop je systém tvořen velikým množstvím nástrojů pro zpracování Big Data (viz 3.4), díky těmto nástrojům lze v Apache Hadoop implementovat Kappa i Lambda architekturu, např. použitím Spark a Kafka, Hive.

3.3 Postup zpracování Big Data

Samotné zpracování dat probíhá na základě modelu, vlastností samotných dat a použitých technologií. Obecně lze proces zpracování rozdělit do několika fází a podprocesů, kterými jsou správa dat a analytické zpracování. Správa dat se zabývá samotným získáním dat, jejich zpracováním a přípravou pro prezentaci či analytické zpracování. Analytický proces zahrnuje vytěžování dat, statistickou analýzu dat, matematické modely atd. [12]



Obrázek 3.7: Schéma zpracování Big Data. [12]

3.3.1 Sběr dat a nahrání dat do systému

V první fázi je nutno se věnovat tomu, z jakých zdrojů data získat, jaká data a jakým způsobem budou do systému nahrána. V této fázi nejsou data nijak zpracovávána. [12]

Je nutné se soustředit pouze na relevantní data, čímž zredukujeme celkovou velikost datové sady. Relevantní data lze vybrat dle vyhledaných informací nebo na základě filtrů, které se dle nastavených pravidel snaží odfiltrovat nechtěná data tak, aby nedocházelo ke ztrátě relevantních dat. Během této fáze je potřeba se věnovat i generování metadat. [12]

3.3.2 Extrakce informací a čištění dat

Data nahraná do systému nebývají většinou ve formě, která by byla vhodná k následnému zpracování a analýze. Proto je data potřeba převést do vhodné podoby. Kvůli různým formátům, malé datové kvalitě a různé komplexnosti dat, je tato fáze velice časově náročná. V této fázi jsou často využívány ETL nástroje pro převod dat do strukturované podoby. [12]

3.3.3 Datová integrace, agregace a prezentace

Poté, co jsou data převedena do strukturované podoby, je potřeba je i vhodně uložit, tak aby byla struktura počítačově srozumitelná. Je velice důležité se zaměřit na návrh databáze a způsob uložení dat. Nevhodný způsob uložení dat může velice ztížit následnou analýzu. [12]

3.3.4 Analýza a modelování dotazů

Fáze analýzy a modelování dotazů zahrnuje metody pro zpracování dotazů, vytěžování dat a řešení analytické úlohy. Výhodou Big Data je, že poskytují dostatečně velký vzorek dat pro analýzu. Nevýhodou je, že data jsou plná šumu, jsou vnitřně provázaná, dynamická a nelze jim ve všech případech důvěřovat. Statisticky získané informace lze i přes dané nevýhody brát jako prokazatelné, neboť individuální výkyvy v datech jsou kompenzovány počtem vzorků. V této fázi je většinou využívána jedna nebo více analytických metod, které se mohou používat iterativně. Dále lze kombinovat různé techniky vytěžování dat se statistickou analýzou nebo matematickými modely k určení závislostí proměnných. [12]

3.3.5 Interpretace dat

Závěrečná fáze zahrnuje práci s nástroji, které dokážou interpretovat výsledky analýz v co nejsrozumitelnější podobě. Díky komplexnosti Big Data není tato fáze jednoduchá, interpretace pouhých výsledků často není dostačující a je potřeba poskytnout informace i o analytických procesech a zdrojích dat. Proto je velice důležité mít vytvořená kvalitní metadata. Existují systémy, které nabízí vizualizační nástroje. Vizualizace je jednou z nejsrozumitelnější forem interpretace – informace jsou zobrazovány v abstraktní, schématické formě. Vizualizační nástroje nabízí možnost dohledat původ dat nebo přehrát analýzu krok po kroku, čímž dokáží poskytnout komplexní informace o výsledcích. [12]

3.4 Apache Hadoop

Jak již bylo zmíněno, standardem v oblasti zpracování Big Data se stal systém Hadoop, který se skládá z několika komponent pro správu Big Data. Hadoop především poskytuje výpočetní model MapReduce pro distribuované zpracování velkého množství dat, zároveň dokáže data uložit na distribuovaném uložišti.

Apache Hadoop je open-source framework společnosti Apache Software Foundation. Cílem frameworku je poskytnout paralelní zpracování, analýza a uložení velkých datových objemů v počítačovém clusteru, který je tvořen běžně dostupným hardwarem. [63]

Poprvé podobnou technologii použila společnost Google pro potřeby vyhledávače. Byl vytvořen distribuovaný souborový systém Google File System s podporou paralelního zpracování dat. Hlavní myšlenkou bylo zpracování dat pomocí modelu MapReduce, který umožňuje velice rychlé zpracování paralelně uložených dat. Na základě tohoto konceptu byl vytvořen nástroj Hadoop. [18]

Hadoop je zaměřen na získávání informací, které by byly běžnými prostředky nedosažitelné. Velkou výhodou je, že dokáže pracovat s mnoha dostupnými formáty a typy souborů. Pro dosažení vysoké škálovatelnosti má Hadoop specifický model pro přístup k výpočtům. Výpočetní funkce jsou přiřazovány k datům, namísto standardního přístupu přidělení dat k výpočetní funkci. Hadoop je provozován na několika vzájemně propojených serverech, které mezi sebou spolupracují. Díky tomu je dosaženo vysoké odolnosti vůči chybám. V případě nedostupnosti některé kopie dat, je dotaz přesunut na jiný server v clusteru. Stejný přístup má Hadoop i k výpočetním úlohám. Pokud nějaká úloha selže, přesune se celá úloha na jiný paralelní server, na kterém se spustí. [3]

Hadoop se skládá ze čtyř klíčových komponent [63]:

- **Hadoop Common** – Správa knihoven pro Hadoop moduly.
- **Hadoop Distributed File System (HDFS)** – Distribuované souborové uložiště.
- **Hadoop YARN** – Nástroj pro správu úloh a clusteru.
- **Hadoop MapReduce** – Nástroj pro paralelní zpracování velkých datových objemů.

Dnes pod Hadoop spadá několik projektů, které se zabývají zpracováním Big Data. Dohromady tvoří komplexní systém pro správu dat. Všechny níže uvedené projekty jsou podporovány prostřednictvím Apache Software Foundation [63]:

- **Ambari** – Webový nástroj pro vytváření, správu a sledování clusteru Apache Hadoop, který zahrnuje podporu Hadoop HDFS, Hadoop MapReduce, HCatalog, HBase, ZooKeeper, Oozie, Pig a Sqoop.
- **Avro** – Nástroj pro serializaci dat.
- **Cassandra** – Vysoce škálovatelná databáze s vysokou dostupností.
- **Chukwa** – Distribuovaný systém pro analýzu dat.
- **HBase** – Distribuovaná sloupcová databáze.

- **Impala** – Nativní analitická databáze pro Apache Hadoop [64].
- **Kudu** – Vrstva nad Apache Hadoop, která umožňuje rychlou analýzu Fast data [66].
- **Hive** – Distribuovaný datový sklad.
- **Mahout** – Knihovna pro strojové učení a vytěžování dat.
- **Pig** – Jazyk pro analýzy rozsáhlých datových celků.
- **Spark** – Programovací model pro zpracování streamovaných dat.
- **Tez** – Framework pro práci s acyklickými grafy.
- **ZooKeeper** – Služba pro koordinaci distribuovaného zpracování.

3.4.1 Hadoop Distributed File System

HDFS je virtuální distribuované uložiště. Souborový systém zprostředkovává distribuci dat na jednotlivé uzly Hadoop clusteru. Metadata, která popisují uložená data, jsou postavena mimo celý cluster na jeden uzel. HDFS řeší výkonnost clusteru, optimální uložení dat a také odolnost clusteru vůči výpadkům. Souborový systém je pouze virtualizovaný, není tedy zapotřebí řešit, kde jsou data fyzicky uložena. Stačí pouze nalézt vhodný uzel pro uložení dat a zajistit následný přístup k datům. Díky dávkovému přístupu pomocí MapReduce je přístup na data, jejich čtení a zápis, sekvenční. Tudíž je čtení i zápis velice rychlé, nejdéle trvá nalezení dat. [63]

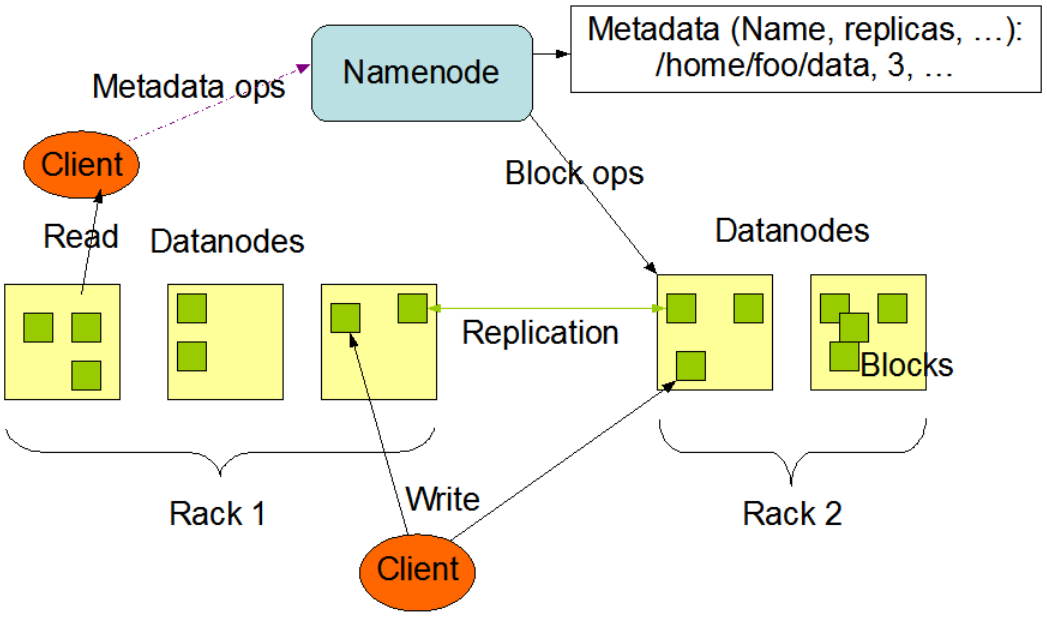
Samotná data jsou uložena do bloků o fixní velikosti. Bloky jsou uloženy ve více kopiích v celém clusteru, čímž je zajištěna dostatečná redundance při výpadku. Na fyzické vrstvě je jeden uložený blok rozdělen na několik podbloků na souborovém systému uzlu. Oproti lokálnímu souborovému systému jsou data ukládána jiným způsobem. Pokud není na HDFS obsazen celý blok, není považován za obsazený, jako je tomu u lokálního souborového systému. Souborový systém HDFS vychází z architektury master-slave, kde NameNode je master uzel a DataNode jsou uzly typu slave. [63]

3.4.2 NameNode

NameNode je master uzel, který je jediný svého druhu v celém clusteru. Úkolem masteru je spravovat metadata, samotná data nejsou na tomto uzlu uložena. Z důvodu jedinečnosti je tento uzel umístěn na výkonném a spolehlivém uzlu. Master nevykonává žádné výpočetní operace. Při čtení dat klient pošle požadavek na NameNode, který obsahuje informace o tom, kde jsou data uložena. Nevýhodou tohoto řešení je, že NameNode není nahraditelný. Pokud nastane výpadek na tomto uzlu, je tím ovlivněn celý cluster. Běžně je instalován sekundární NameNode, ale ani toto řešení není 100 % spolehlivé. [63]

3.4.3 DataNode

DataNode reprezentuje zástupce ze skupiny uzlů typu slave. DataNode obsahuje bloky dat uložených v HDFS. Pří zápisu do HDFS je soubor rozdělen do několika bloků a NameNode určí, kam mají být data uložena. Následně už probíhá komunikace klienta s DataNody. Po uložení bloků probíhá komunikace mezi DataNody a replikace dat na další uzly. Pokud bude nějaký uzel nedostupný, je jeho funkci schopen zastat jiný uzel, který má uložená shodná data, soubory jsou vždy dostupné. [63]



Obrázek 3.8: Architektura HDFS. [63]

Základní vlastnosti HDFS [63]:

- **Optimalizace pro velké soubory** – Systém je typicky používán pro soubory od několika gigabytů až po terabyty. HDFS je optimalizován pro práci s takto velkými soubory a poskytuje dostatečnou propustnost dat.
- **Odolnost proti chybám** – Neustále probíhá monitoring uzlů celého systému. Při výpadku je zajištěna obnova z této chyby. Jelikož je celý systém složen z velkého množství uzlů, je velká pravděpodobnost selhání. Prakticky neustále je některá část HDFS nefunkční.
- **Čtení/Zápis** – Již architektura Master-slave napovídá, že systém je orientován převážně na operaci čtení. Aplikace postavené na HDFS využívají write-once/read-many model. Soubor, který je vytvořen, již není modifikován, čímž se zjednoduší problémy s koherencí dat a zvyšuje se propustnost.

- **Důraz na propustnost** – HDFS je navrženo spíše pro dávkové zpracování místo interaktivního přístupu. Proto jsou pro práci s HDFS vhodné aplikace založené na MapReduce modelu.
- **Výpočet u zdroje dat** – Výpočet požadovaný aplikací je mnohem efektivnější, pokud je proveden v blízkosti dat, na kterých pracuje. To platí zejména tehdy, když je velikost souboru dat obrovská. Tím se minimalizuje přetížení sítě a zvyšuje se celková propustnost systému.

Kapitola 4

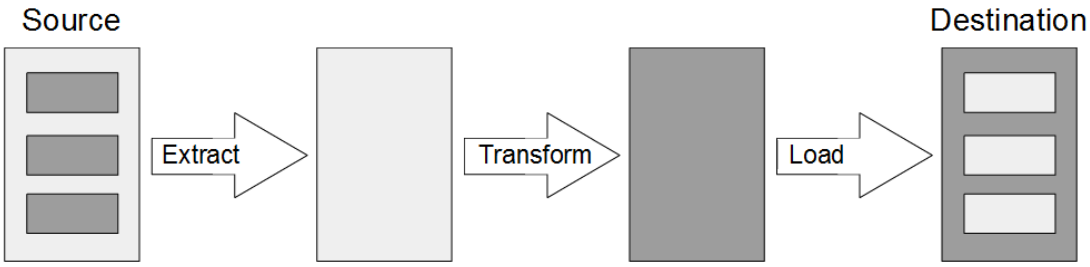
Datová pumpa - ETL

ETL z anglického Extract, Transform, Load je proces, který je zodpovědný za extrakci, transformaci a načtení dat z jednoho nebo více zdrojových systémů do cílového systému [47]. Jedná se o tzv. datovou pumpu [57].

ETL může být použito mnoha způsoby. Například pro získání dočasné podmnožiny dat pro reporty nebo jiné účely, získání trvalé sady dat pro účely uskladnění v datovém skladu nebo datovém tržišti. ETL lze použít i ke konverzi databáze jednoho typu na typ druhý nebo při migraci databáze či platforem na jinou. [44]

ETL je často porovnáváno s druhým přístupem implementace datových pump – ELT (Extract, Load, Transform), který zdrojová data nejprve přenese do cílového uložiště, a následně jsou nad požadovanými daty prováděny transformace, která data připraví pro použití. [44]

ELT je flexibilní s ohledem na dostupnost dat, ale mnohem náročnější na objem zpracovávaných dat. ETL je v porovnání s ELT mnohem více zaměřeno na přesné zacílení na požadovaná data. [57]



Obrázek 4.1: Schéma datové pumpy - ETL. [52]

4.1 Extrakce

Extrakce je operace, v rámci které jsou získány data ze zdrojového systému pro pozdější použití v cílovém systému. Vytvoření a návrh procesu pro získání dat je jedním z nejvíce

časově náročných úkonů v ETL. Systémy, které obsahují zdrojová data, mohou být velice složité a nemusí mít dobrou dokumentaci. Proto je velice obtížné určit, jaká data extrahovat a jaká ne. Výběr zdrojových dat obnáší časově náročnou analýzu zdrojového systému. [25]

4.2 Transformace

Fáze transformace aplikuje na extrahovaná data soubor pravidel za účelem jejich načtení do cílového systému. Transformace dat není pouze jednoduché mapování jednotlivých sloupců a tabulek na určitá místa v cílovém systému. Často je zapotřebí data změnit a upravit tak, aby odpovídala cílovému systému a omezené integritě dat na cílovém systému. [47]

Proces transformace může obsahovat tyto složky [61]:

- **Proces čištění dat** – Nahrazení NULL hodnot za nuly, nahrazení „Male“ za „M“ atp.
- **Proces selekce** – Filtrování pouze určitých hodnot, např. sloupců.
- **Aplikace byznysových pravidel** – Výpočet nových hodnot.
- **Spojování dat z více zdrojů**
- **Rozdělení dat na více částí**

4.3 Načtení

Po dokončení veškerých úkonů v transformační fázi jsou data připravena k načtení do cílového datového uložiště. Fáze načtení zajišťuje tento přesun dat. [47]

Kapitola 5

Big Data ETL

Důležité je získat co největší objem dat, ze kterých lze čerpat nové informace. V dnešní době potřebují společnosti mít přístup k datům různé velikosti a formy – videa, sociální média, IoT, protokoly ze serverových strojů, prostorová data atp. Data však nemusí mít formu, která je vhodná pro následnou analýzu. Z toho důvodu dodavatelé ETL nástrojů přišli s novým řešením, které podporuje integraci Big Data pro možnost zpracování těchto dat pro následnou analýzu. [69]

Charakteristika Big Data klade na celkovou platformu velké finanční nároky a rovněž nároky na efektivitu ukládání dat. Tradiční ETL nástroje, pracující se standardními datovými sklady, uváděné nároky nezvládají, proto bylo potřeba, aby pro zpracování Big Data vzniklo nové řešení.

Big Data ETL nástroje byly vyvinuty tak, aby podpořily integraci více řešení než tradiční datový sklad. Pokročilé nástroje ETL umožňují načíst a převést strukturovaná i nestrukturovaná data velkého objemu do prostředí Hadoop. Tyto nástroje provádí čtení a zápis více souborů paralelně, a zjednodušují tak sloučení dat do běžného procesu transformace. Některá řešení již obsahují předdefinované transformace ETL pro transakční nebo interakční data. ETL také podporuje integraci mezi transakčními systémy, datovými uložišti, platformami pro BI a cloudy. [69]

Dnešní trh nabízí mnoho nástrojů pro ETL zpracování dat. Zpracování dat formou ETL v oblasti Big Data nabízí více možností než jen standardizované nástroje:

- **Nativní řešení ETL v prostředí Hadoop** – MapReduce, Pig, Hive, Spark
- **Standardní ETL nástroje s podporou pro Big Data** – Hitachi Vantara (Pentaho) PDI, Microsoft SQL Server Integration Services
- **Big Data ETL nástroje** – Talend Open Studio for Big Data, Oracle ODI for Big Data, Apache Nifi

5.1 Nativní řešení ETL v prostředí Hadoop

Prostředí Hadoopu nabízí několik možností, jak data zpracovat pomocí ETL datové pumpy. Na nejnižší úrovni se nachází programovací model MapReduce, který pracuje přímo

s daty HDFS. Naopak komplexní ETL nástroje pro Big Data většinou pro analýzy využívají platformu Hive, která je nejbližší standartnímu zpracování dat pomocí SQL.

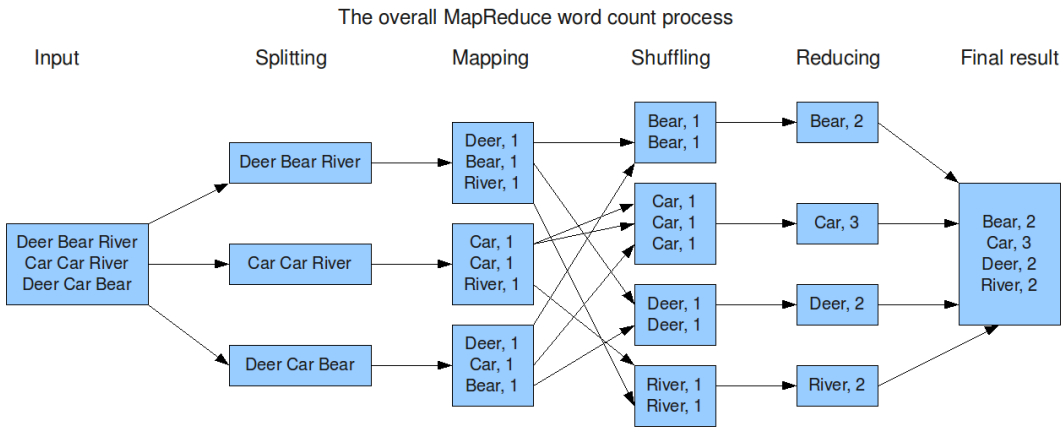
5.1.1 MapReduce

MapReduce je programovací model sloužící pro paralelní zpracování dat. Implementace MapReduce programu je možná v mnoha programovacích jazycích, např. Java, C++, Python. Cílem modelu MapReduce je efektivní analýza dat velkého objemu.

Vstupní data jsou rozdělena na malé části, většinou ve velikosti bloku HDFS, odděleně zpracována, zkombinována a následně je vypočítán celkový výsledek. Je potřeba ošetřit možnou chybu procesu pro zpracování. Pokud zpracování selže, je výpočet přesunut na jiný uzel v clusteru. Bez tohoto ošetření by výpočty byly nesprávně. [72]

MapReduce model řídí celý výpočet, řeší selhání procesů a rozděluje vstupní data na dílčí množiny. Samotná analýza pomocí MapReduce modelu probíhá ve třech fázích [63]:

- **Map** – Zpracování dílčích částí vstupní množiny dat.
- **Shuffle** – Kombinace dílčích výsledků.
- **Reduce** – Výpočet výsledku.



Obrázek 5.1: Schéma průběhu MapReduce programu - počet slov. [60]

Analýza MapReduce modelem se nazývá job, který se dále dělí na map a reduce tasky. Map tasky probíhají na uzlech, kde jsou uložena data (část množiny, která je zpracovávána) a dodržují princip datové lokality. [66] Cílem map tasku je zpracovat dílčí část vstupní množiny tak, aby bylo následné zpracování reduce taskem co nejjednodušší. Výstupem map tasku je dvojice klíč-hodnota. Po ukončení všech map tasků jsou všechny výsledky setříděny dle klíčů a následně přesunuty na jeden uzel, který obsluhuje reduce task. Reduce task počítá celkový

výsledek MapReduce jobu a zapíše jej na HDFS. Reduce tasků může být i více. V takovém případě jsou výsledky map tasku rozděleny do oddílů podle klíčů. Oddíly jsou pak zpracovány pomocí reduce tasků. [72]

Výhody a nevýhody MapReduce

MapReduce je velice efektivní způsob analýzy dat, který je velmi dobře konfigurovatelný a lze ho optimalizovat. Psaní MapReduce programů je složitější a vyžaduje určitou znalost patternu a programovacího jazyka, proto není vhodný pro jednorázové analýzy. MapReduce je ve většině případech využíván analytickými nástroji, které nevyžadují samotné psaní MapReduce programů (např. Hive).

5.1.2 Hive

Hive je open-source softwarová platforma pro datový sklad, která umožňuje správu velkého objemu dat uložených v distribuovaném souborovém systému HDFS. Tato platforma byla původně vyvinuta společností Facebook, aby spravovala a využívala mnoho dat, která jsou denně produkována na sociální síti. Jako uložiště byl zvolen Hadoop, zejména kvůli jeho efektivnímu ukládání a dobré škálovatelnosti. Hive byl vytvořen tak, aby umožnil analytikům, kteří mají velmi dobrou znalost SQL, spouštět dotazy nad velkou sadou dat uložených v HDFS. V dnešní době je Hive úspěšným projektem Apache a je využíván mnoha společnostmi jako obecná a dobře škálovatelná platforma pro zpracování Big Data. [72]

Hive umožňuje strukturování dat uložených v distribuovaném uložišti. Struktura je poté podobná klasickému relačnímu modelu. [22] Pro vytváření dotazů nad uloženými daty je využíván jazyk HiveQL, syntakticky velice podobný SQL. HiveQL umožňuje programátorům implementovat vlastní MapReduce funkce pro lepší analýzy, kterých by nebylo možné dosáhnout s klasickým přístupem dotazovacího jazyka. Samotné HiveQL querry jsou po spuštění převedeny na MapReduce úlohy. [47]

Hive obsahuje dvě komponenty, a to HCatalog a WebHCat. HCatalog je vrstva Hivu, která umožňuje správu uložiště Hadoopu, zajišťuje práci s nástroji pro zpracování dat (Pig, MapReduce) a usnadňuje zápis a čtení dat. WebHCat je služba, která zprostředkovává spouštění úloh MapReduce, Pig nebo Hive, a také povoluje práci s metadaty pomocí HTTP rozhraní. [26]

Výhody a nevýhody Hive

Hive umožňuje využívat Hadoop uživatelům, kteří jsou zvyklí na standardní přístup pomocí dotazovacího jazyka SQL. Díky tomu uživatelům velice zjednodušší práci bez nutnosti učit se přístup k zcela nové technologii. Přestože je Hive velice uživatelsky přívětivý, nemusí být řešení pomocí HiveQL querry optimální. Analýza pomocí uživatelských MapReduce programů je většinou rychlejší.

5.2 Big Data ETL nástroj – Talend Open Studio for Big Data

Talend Open Studio je volně stažitelný open-source projekt, který se zaměřuje na integraci Big Data. TOS obsahuje grafické vývojové prostředí, které poskytuje velké zjednodušení integrace dat, profilování dat, integrace aplikací bez nutností se učit zcela nové technologie. TOS pokrývá oblasti Big Data, datové integrace, ETL/ELT, datové kvality, Enterprise Service Bus a data managementu. [70]

Vývojové prostředí nabízí širokou škálu knihoven, hotových komponent a konektorů. Vývoj probíhá pomocí drag-and-drop rozhraní, bez nutnosti programování. Nativní kód je generován na pozadí a uživatel do něj může nahlédnout. Rozsáhlá škála komponent a konektorů je rozšiřována uživateli, vyvinuté komponenty jsou pak dostupné na Talend Exchange. Při používání nové i staré technologie zde s největší pravděpodobností zde naleznete hotové řešení. [70]

Talend zjednodušuje integraci Big Data díky využívání nástrojů jako je Apache Hadoop, Apache Spark, Apache Kafka, Apache Hive, NoSQL databází a dalších. Díky obsaženým komponentám agregačních a mapovacích funkcí je velice zjednodušena práce s ETL a ELT datovými pumpami. Během několika minut lze vytvořit kompletní vrstvy pro extrakci, transformaci a nahrání dat. Výhodou TOS je velice rychlé zpracování dat, jelikož na pozadí generuje MapReduce nebo Spark kódy, které vynikají vysokou rychlostí zpracování dat. [70]

Talend nabízí možnost vyzkoušet si práci s Big Daty díky připravenému SandBoxu, který stačí pouze nainstalovat jako virtuální stroj v Oracle VM VirtualBoxu nebo VMware Playeru. K dispozici jsou Hadoop distribuce Cloudera a Hortonworks. [42]

Výhody a nevýhody TOS

Velkou výhodou řešení od Talendu je široká škála předpřipravených komponent a možnost vytvářet řešení bez speciálních znalostí procesů na pozadí. Další výhodou je velice rychlé zpracování dat. Za nevýhodu lze považovat zpracování dat mimo prostředí Hadoopu. Dochází k vzdálenému čtení a zápisu, které může být pomalé.

5.3 Standardní nástroj pro ETL s podporou Big Data – Hitachi Vantara (Pentaho) PDI

Hitachi Vantara (Pentaho) PDI je ETL nástroj, který umožňuje přistupovat k datům, analyzovat data a získávat užitečné informace z tradičních dat a z Big Data. Cílem PDI je zjednodušit správu velkých objemů dat, bez ohledu na typ dat a počet zdrojů. PDI obsahuje grafické prostředí, které poskytuje uživateli možnost implementace datových pump bez nutnosti programovat SQL a MapReduce programy. [27]

PDI pracuje ve dvou režimech – transformace dat a řízení úloh. Řízení úloh je sekvenční sada položek, které zapouzdřují danou akci. Příkladem může být zkopírovat soubory do HDFS. Transformace dat se skládá z množiny kroků, které běží paralelně a pracují nad daty. Příkladem může být načtení dat ze systému, vypočtení nového sloupce a následné zapsání. [28]

PDI obsahuje plugin pro Big Data, který zajišťuje konektivitu pro transformace a připojení do Hadoop, Cassandra, MongoDB. PDI lze propojit prakticky se všemi známými Hadoop distribucemi. [28]

PDI je jedinečné v tom, že dokáže pracovat mimo i uvnitř Hadoop clusteru. Při práci mimo cluster lze např. načítat data z HDFS nebo Hive. Při spuštění uvnitř clusteru lze PDI transformace převést na MapReduce úlohy. PDI lze použít jako vývojový nástroj pro MapReduce úlohy bez nutnosti programování kódu. [28]

Vytvoření ETL nebo ELT transformací pomocí PDI v kombinaci s připojením do Hive databáze nabízí možnost práce s daty pomocí standardních SQL dotazů. Integrace je velice podobná tomu, co je známé při zpracování dat z relačních databází.

Výhody a nevýhody PDI

Výhodou PDI je, že je to zavedený nástroj, který byl a stále je využíván uživateli pro správu a integraci standardních dat. To přináší obrovskou výhodu, jelikož se vývojář nemusí seznamovat s novým nástrojem. Nevýhodou PDI je, že nástroj nebyl pro práci s Big Daty přímo vyvinut a využívá pouze plugin, díky kterému umožňuje práci s Hadoop clusterem.

Kapitola 6

Seznam sotwarových prostředků pro Big Data ETL

V diplomové práci jsem porovnával zpracování Big Data formou ETL pomocí nativního nástroje v prostředí Hadoop – MapReduce, Big Data ETL nástroje – Talend Open Studio for Big Data, standardního ETL nástroje s podporou Big Data – Hitachi Vantara PDI. Za pomocí těchto nástrojů bylo implementováno ETL zpracování bankovních transakcí produktů s cílem vytvořit sumární denní report. Nástroje byly vybrány za účelem porovnání různých přístupů, které jsou v dnešní době pro ETL zpracování Big Data v nabídce.

Nástroje byly podrobeny měření škálovatelnosti na základě zvyšujícího se objemu zpracovávaných dat. Také byla provedena analýza pracnosti pro dané řešení s diskuzí vůči dosaženému výkonu.

6.1 Použité nástroje

Pro implementaci v diplomové práci byly využity bezplatné verze nástrojů pro ETL zpracování Big Data.

Jako distribuce Hadoopu byl zvolen balíček QuickStart VM od společnosti Cloudera. Jedná se o kompletní virtualizovanou single node distribuci Hadoop clusteru určenou pro testování, vytváření demo aplikací a výuku. Distribuce Hadoopu je nainstalována na operačním systému CentOS 6.7 a je zabalená do balíčku, umožňující instalaci na Oracle VM VirtualBox, VMware Worksation Player nebo Docker.

Výčet použitých nástrojů včetně verzí:

- **Cloudera QuickStart VM - CDH 5.12**
- **VMware Workstation 14 Player**
- **Talend Open Studio for Big Data 6.5.1**
- **Pentaho Data Integrations Community Edition 8.0**

Kapitola 7

Příprava pro porovnání Big Data ETL nástrojů

Implementace ETL zpracování dat byly testovány nad uměle vygenerovanými daty. Cílem ETL zpracování bylo vytvořit denní report z transakcí pomocí agregačních funkcí a připojení číselníkových hodnot. Zpracování transakcí a výpočet denního reportu je čistě dávkové zpracování nad strukturovanými daty. Proto byla uvažována implementace nad HDFS a databází Hive, pro kterou je navržen datový model a generovaná data.

7.1 Specifikace datové domény

Jako testovací data byla zvolena bankovní data – transakce produktů. Byl navržen jednoduchý model bankovní transakce, který má vazbu na číselníky typ produktu a měna.

7.1.1 Datový model

Datový model pro testovací data byl navržen pomocí Oracle SQL Data Modeleru 18.1, který nepodporuje datové typy databáze Hive. V datovém modelu je uvažováno, že datový typ VARCHAR2 odpovídá datovému typu String.

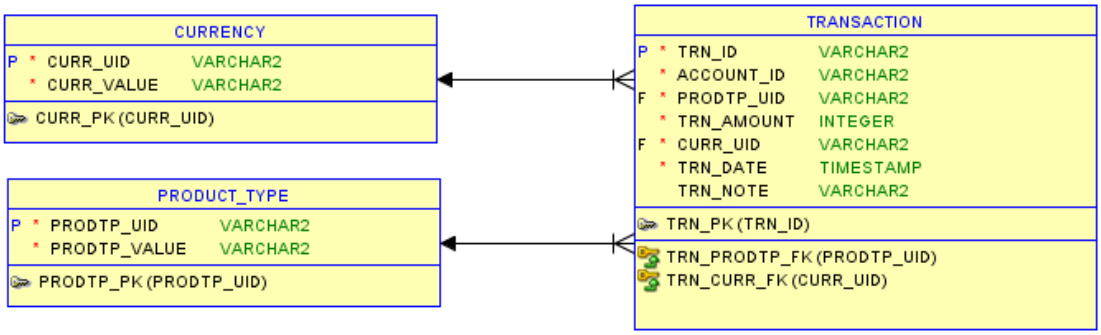
Zdrojem pro ETL transformaci jsou tři tabulky – TRANSACTION, PRODUCT_TYPE, CURRENCY.

Testovací bankovní transakce, tabulka TRANSACTION, obsahuje identifikátor transakce, identifikátor účtu, identifikátor typu produktu, transakční částku, identifikátor měny, datum transakce a poznámku k transakci.

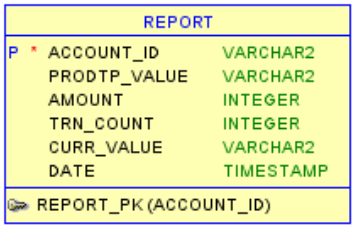
Tabulky PRODUCT_TYPE a *CURRENCY* jsou tabulky číselníkových hodnot a obsahují vždy identifikátor a hodnotu.

Cílem pro ETL transformaci je denní report – tabulka REPORT.

Výsledný report obsahuje identifikátor účtu, typ produktu, sumární částku transakcí za den, počet provedených transakcí, měnu a datum.



Obrázek 7.1: Datový model zdrojových tabulek.



Obrázek 7.2: Datový model cílové tabulky.

7.1.2 Generování testovacích dat

Pro generování testovacích dat jsem vytvořil jednoduchou Java aplikaci, která generuje data do textového souboru. Hodnoty ve sloupcích jsou odděleny tabulátorem, každý zapsaný záznam je ukončen novým řádkem. Oddělení tabulátorem jsem zvolil z důvodu, že se jedná o jeden z nativních oddělovačů pro Hive.

Java aplikace je vyvinuta pomocí nástroje pro správu a sestavování aplikací Maven. Pro generování náhodných čísel a slov je využita knihovna org.fluttercode.datafactory.

Data jsou generována ve smyčce na základě počtu požadovaných záznamů. Transakční identifikátor odpovídá číslu iterace smyčky. Identifikátor účtu je generován funkcí getNumberBetween(), náhodně jsou vybrána čísla od 100 do 700100. Následně jsou každému číslu účtu přiřazeny identifikátory typu produktu a měny. Pro přiřazení jsem zvolil rozdělení na základě dělitelnosti čísel, viz C. Částka transakce je generována opět funkcí getNumberBetween(), náhodně jsou vybrána čísla od -10000 do 100000. Datum transakce je náhodně vybrán mezi 3.4.2018 a 5.4.2018, formát datumu je YYYY-MM-DD. Následně je datum doplněn o nuly na formát YYYY-MM-DD HH:mm:ss, aby splňoval specifikaci pro datový typ TIMESTAMP (TOS neumí standardně pracovat s Hive datovým typem DATE). Poznámka transakce je generována funkcí getRandomWord(), kde jsou náhodně vybrána slova délky od 4 do 100 znaků.

Číselníky PRODUCT_TYPE a CURRENCY

V následujících tabulkách jsou zaznamenány klíče a hodnoty číselníkových tabulek PRODUCT_TYPE a CURRENCY.

Data	
CURR_UID	**CURR_VALUE**
CZK	Česká koruna
EUR	Euro
USD	Americký dolar
GBP	Britská libra

Tabulka 7.1: Data tabulky CURRENCY.

Data	
PRODTP_UID	**PRODTP_VALUE**
BU-FO	Běžný účet FO
BU-FOP	Běžný účet FOP
BU-PO	Běžný účet PO
PK-ST	Podnikatelské konto STANDARD
PK-EX	Podnikatelské konto EXCLUSIVE
SP	Spoření
AH	Americká hypotéka
STU	Stavební úvěr
HU	Hypoteční úvěr
SU	Spotřebitelský úvěr

Tabulka 7.2: Data tabulky PRODUCT_TYPE

7.2 Specifikace ETL transformace

Cílem ETL transformace je vytvořit denní sumární report, viz 7.1.1.

ETL transformace obsahuje agregační funkci, která na základě čísla účtu, datumu, typu produktu a měny sjednotí dané záznamy se shodnými atributy a vypočítá nové sloupce AMOUNT a TRN_COUNT. Sloupec AMOUNT je vypočítán jako suma všech částek transakcí za den. TRN_COUNT je počet proběhlých transakcí za den.

Po dokončení výpočtu jsou k výsledku připojeny číselníkové tabulky PRODUCT_TYPE a CURRENCY. Do výsledného reportu jsou doplněny číselníkové hodnoty typu produktu a měny.

7.3 Instalace nástrojů

7.3.1 Hardwarové prostředky

Pro účely diplomové práce jsem použil notebook Dell Latitude E6540, hardwarové konfigurace:

- **Procesor** – Intel Core i7, 4810MQ
- **RAM** – 16 GB
- **Diskové uložiště** – Samsung SSD PM871, 256 GB
- **Operační systém** – Windows 7, Service Pack 1

7.3.2 Cloudera Hadoop Cluster

Pro účely diplomové práce byla zvolena distribuce Hadoopu od společnosti Cloudera v balíčku virtualizovaného clusteru QuickStartVM, určeném pro instalaci na virtuální stroj. Tento balíček je dostupný na webových stránkách Cloudery v položce QuickStartVMs po vyplnění osobních údajů a účelu využívání distribuce. [58]

Pro instalaci jsem využil VMware Workstation 14 Player. Dle doporučených nastavení jsem virtuálnímu stroji přidělil 8 GB RAM paměti, dvě procesorová jádra a 64 GB diskového prostoru. Síťový adaptér byl nastaven na NAT – sdílení IP adresy s hostujícím operačním systémem.

Ihned po spuštění je cluster připraven k používání. Pro připojení z lokálního prostředí je zapotřebí mít nainstalovaný JDBC driver na databázi, do které se chcete připojit, znát IP adresu prvku v clusteru a port. IP adresu je možné zjistit v terminálu pomocí příkazu ifcofig nebo na úvodní stránce http://quickstart.cloudera. Dále je nutné znát porty, na kterých jsou dostupné funkcionality Hadoop Clusteru. Pro mé účely jsem využil připojení do HDFS na portu 8020 a Hive na portu 10000.

7.3.3 Talend Open Studio for Big Data

TOS je dostupné jako spustitelná aplikace v zip balíčku na stránkách Talendu. [70]

Stažený balíček stačí rozbalit a spustit TOS_BD-win-x86_64.exe nebo TOS_BD-linux-gtk-x86.sh soubor. Před prvním spuštěním je zapotřebí upravit parametry paměti pro JVM v souboru *.ini na základě platformy. Defaultní nastavení: -XX:MaxPermSize=512m -Xms64m -Xmx768m, doporučené nastavení pro můj systém: -XX:MaxPermSize=512m -Xms1024m -Xmx4096m . [30]

7.3.4 Pentaho Data Integration – Community edition

PDI Communiti edition je dostupné na stránkách komunity Hitachi Vantara. [59]

PDI je aplikace zabalená v zip balíčku. Po rozbalení lze spustit pomocí souboru Spoon.bat nebo Spoon.sh na základě platformy. Před startem aplikace byly opět upraveny parametry paměti JVM v souboru Spoon.bat na shodné parametry s TOS.

7.4 Měřené veličiny

Pro porovnání nástrojů pro Big Data ETL byla zvolena metrika škálovatelnosti na základě zvětšujícího se objemu zpracovávaných dat. Za běhu navržených řešeních byl měřen čas, v sekundách, za který jsou data zpracována.

Během implementace byl zaznamenáván čas (pracnost), jenž byl spotřebován pro vývoj, zprovoznění vývojového prostředí a další činnosti, které provázejí vývoj řešení. Čas spotřebovaný k implementaci byl zaznamenáván v MD, kdy 1 MD je 8 hodin. Na základě naměřených časů zpracování a nutného času pro implementaci bylo diskutováno, jaké řešení je nejvýhodnější z pohledu času versus dosaženého výkonu.

Všechna měření probíhala pomocí uživatelsky dostupných nástrojů. Doba zpracování MapReduce byla zaznamenávána pomocí nástroje Cloudera HUE. Big Data ETL nástroje (Talend, Pentaho) obsahují nástroje pro měření doby zpracování.

7.5 Implementace Big Data ETL zpracování

Cílem implementace bylo vytvoření denního reportu bankovních transakcí (viz 7.1), Specifikace ETL transformace. Pro implementaci byly využity již výše zmíněné nástroje (viz 6.1).

7.5.1 MapReduce

Nativní implementace Big Data ETL zpracování je implementována pomocí programovacího modelu MapReduce (viz 5.1). Implementaci předcházela instalace Cloudera Hadoop Clusteru (viz 7.3.2) a nahrání zdrojových dat do HDFS.

Implementace programovacího modelu MapReduce byla realizována programovacím jazykem Java ve verzi 1.8. Pro implementaci byly použity knihovny hadoop-common.jar a hadoop-mapreduce-client.jar, které jsou součástí CDH v adresářích /usr/lib/hadoop a /usr /lib/hadoop-mapreduce.

K nahrání dat do HDFS a spuštění MapReduce jobu byl využíván terminál - script startETLjob.sh na CD (viz A). Job lze také spouštět pomocí Cloudera HUE.

Map funkce

Vstupními hodnotami Map funkce byly datové typy Object, jako prázdný klíč, a Text, jako vstupní hodnoty ze souboru.

V mapovací funkci byly všechny načtené hodnoty ze vstupního souboru rozděleny na základě oddělovače – tabulátor. Následně byly setřízeny podle výstupního klíče ACCOUNT_ID| PRODTP_UID|CURR_UID|TRN_DATE, výstupní klíč byl typu Text. Výstupní hodnotou mapovací funkce byla položka TRN_AMOUNT, typu IntWritable.

Standardně lze mapovací funkci využít ke spojení více zdrojových souborů na základě shodného klíče, tzv. JOIN. Pro připojení číselníkových hodnot není toto řešení vhodné, jelikož se jedná o přímý překlad ID na hodnotu.

Reduce funkce

Vstupními hodnotami pro Reduce funkci byly datové typy Text a IntWritable. Při prvním volání Reduce funkce je zavolána metoda setup, která byla využita pro načtení číselníků z HDFS.

V redukovací funkci byla pro každý vstupní klíč sečtena hodnota transakce (AMOUNT) a vytvořena nová proměnná počtu transakcí (TRN_COUNT) pro daný klíč. Následně byly nahrazeny číselníkové hodnoty a zapsán výstup. Výstupním klíčem byl ACCOUNT_ID, výstupními hodnotami byly PRODTP_VALUE, AMOUNT, TRN_COUNT, CURR_VALUE, TRN_DATE. Výstupní data byla zapsána do cílového souboru report.txt.

7.5.2 Talend Open Studio for Big Data

Nástroj pro zpracování Big Data formou ETL/ELT nabízí vývoj zpracování dat bez nutnosti psaní vlastního kódu pomocí předpřipravených komponent. Samotné implementaci předcházela instalace nástroje (viz 7.3.3) a nahrání zdrojových dat do databáze Hive - scripty loadDataHive.sh a HiveQL.sql na CD (viz A).

Pro implementaci byly využity komponenty:

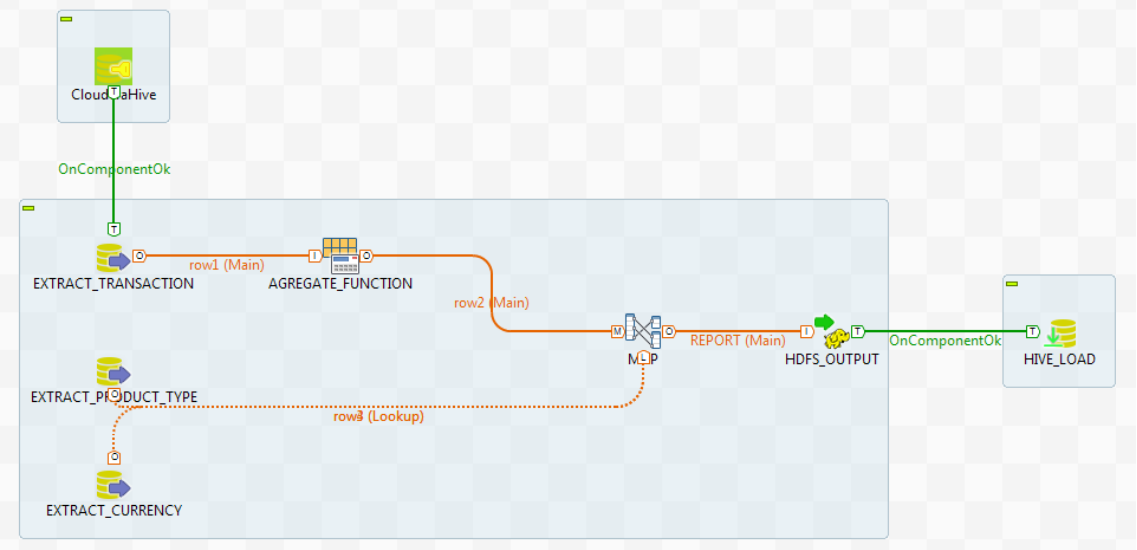
- **tHiveConnection**
- **tHiveInput**
- **tAggregateRow**
- **tMap**
- **tHDFSOutput**
- **tHiveLoad**

Komponenta tHiveConnection slouží k vytvoření připojení do Hive databáze. Připojení bylo vytvořeno na základě hosta a portu. Při úspěšném připojení byly načteny data ze zdrojových tabulek pomocí komponenty tHiveInput. V komponentě tHiveInput byl nadefinován select pro čtení z tabulek TRANSACTION, PRODUCT_TYPE, CURRENCY.

Po načtení dat byla data transformována do výsledného reportu pomocí agregační komponenty tAggregateRow. Tato komponenta data sjednotila data na základě klíče ACCOUNT_ID, PRODTP_UID, CURR_UID, TRN_DATE a vytvořila nové sloupce AMOUNT a TRN_COUNT. Komponenta pracuje na podobném principu jako GROUP BY v SQL databázích. Tímto způsobem zpracovaná data byla zaslána do komponenty tMap, která slouží k připojení číselníků.

Pokud data úspěšně prošla transformací, byly zapsány do HDFS pomocí komponenty tHDFSOutput. Následně došlo k nahrání dat do databáze Hive, tabulky REPORT, komponentou tHiveLoad.

Připojení do Hive a HDFS je zajištěno vestavěným JDBC ovladačem.



Obrázek 7.3: Talend ETL zpracování.

7.5.3 Pentaho Data Integration

Zavedený komplexní nástroj pro zpracování dat formou ETL/ELT, který nabízí možnost vývoje zpracování dat pomocí spustitelných SQL scriptů, bez nutnosti programování nebo kombinací obojího. Implementaci předcházela instalace nástroje (viz 7.3.4) a nahrání zdrojových dat do databáze Hive, stejným způsobem jako v případě TOS.

Pro implementaci byly využity oba režimy PDI. Samotné načtení a zpracování dat je součástí transformace. Spuštění ETL zpracování, transformace a následné nahrání do Hive obsluhuje modul pro řízení úloh (Job).

Implementovaný Job osahuje tyto komponenty:

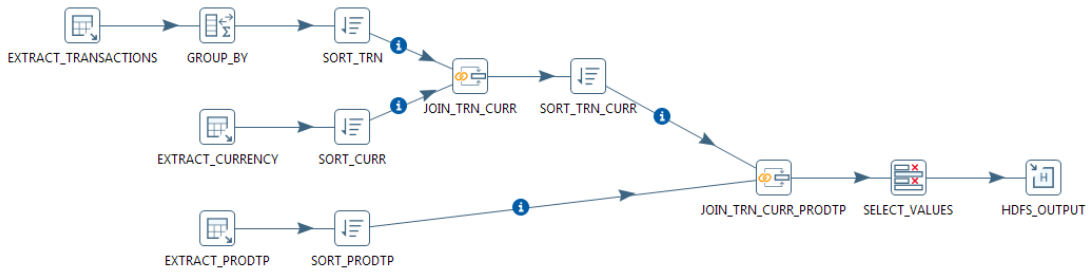
- **Start/Job Scheduling**
- **Transformation**
- **Execute SQL Script**

Obrázek 7.4: Pentaho ETL job.

Komponenta Start slouží ke spuštění Jobu a umožňuje periodické plánování. Poté byla spuštěna samotná transformace. Pokud byla úspěšně dokončena, byl spuštěn SQL script pro nahrání dat do Hive databáze.

Transformace obsahuje tyto komponenty:

- **Table input**
- **Memory Group By**
- **Sort rows**
- **Select/Rename values**
- **Hadoop File Output**



Obrázek 7.5: Pentaho ETL transformace.

Po spuštění transformace jsou data načteny z Hive pomocí komponenty Table input, kde je definovaný SQL SELECT pro získání dat. Načtené transakce jsou následně agregovány podle klíče ACCOUNT_ID, PRODTP_UID, CURR_UID, TRN_DATE a jsou vytvořeny nové sloupce AMOUNT a TRN_COUNT.

Při spojení tabulek je zapotřebí, aby byla data seřazena dle klíče, podle kterého se spojují. Nejprve jsou tedy seřazeny a spojeny transakce a měny, podle CURR_UID. Následně jsou transakce s měnami spojeny s typem produktu, podle PRODTP_UID. PDI obsahuje i komponentu, kde nemusí být spojované záznamy seřazeny podle klíče. Je však možnost chyby. Tato chyba se projevila i u mé implementace, proto byly tabulky vždy seřazeny dle klíče.

Po spojení tabulek jsou pomocí komponenty Select values sloupce seřazeny do požadovaného výstupního formátu a odstraněny nadbytečné sloupce. Následně jsou data zapsána do HDFS.

Připojení do HDFS je zprostředkováno vestavěným pluginem pro připojení do CDH. Do Hive se PDI připojuje pomocí vestavěného JDBC ovladače.

7.5.4 Možné rozšíření implementace

Pro porovnání Big Data ETL nástrojů byla zvolena velice jednoduchá transformace. V praktickém využité jsou transformace mnohem komplexnější. Proto bych jako první možnost rozšíření implementace zvolil transformaci nad komplexním datovým modelem.

V mé testovací implementaci byly samotné zpracování(Joby) spouštěny ručně. Pro reálné využití je potřeba mít Joby zaplánované, aby byly spuštěny automaticky a zpracovávaly data ve vhodnou dobu. Možným rozšířením je tedy plánování spouštění Jobů. MapReduce Joby lze řídit pomocí Cloudera HUE, který obsahuje plánovač, nebo pomocí cronu [62]. Joby v TOS lze vyexportovat jako samostatné aplikace, které lze poté zaplánovat pomocí cronu [31]. PDI obsahuje plánovač spouštění Jobů [29].

7.5.5 Problémy při implementaci

Během implementace se nevyskytly žádné zásadní problémy s funkcionalitou samotných nástrojů. Nejzásadnějším problém nastal při snaze vytvořit připojení do CDH na virtuálním stroji. Hlavním problémem se při vytvoření připojení stala starší verze VMware Workstation Playeru 12, která špatně propojovala síť s CDH. Dalším problémem byly uzavřené porty. Po instalaci CDH na novou verzi VMware Workstation Player 14 byl tento problém vyřešen.

Kapitola 8

Měření vlastností Big Data ETL nástrojů

Navržená metodika pro porovnání Big Data ETL nástrojů byla zacílena na získání dostatečných informací k porovnání výkonu jednotlivých nástrojů a náročnosti implementace. Metodika obsahuje tyto měření a vyhodnocení:

- **Zjištění škálovatelnosti na základě zvyšujícího se objemu dat**
- **Zjištění pracnosti daných implementací**

8.1 Metodika měření

Pro měření škálovatelnosti bylo vygenerováno 8 testovacích sad dat o 5, 10, 20, 40, 60, 80, 100 a 120 milionech záznamech o velikosti od 256 MB do 6,2 GB. Každý nástroj byl otestován na jednotlivých sadách desetkrát, poté byla spočítána průměrná hodnota. Naměřená data byla následně vynesena do grafu. Během měření byly ukončeny veškeré nepotřebné procesy, a to jak na lokálním, tak virtualizovaném systému, aby nedocházelo ke spouštění nechtěných procesů, které by mohly ovlivnit výkon.

Činnost	**Pracnost [MD]**
Instalace a nastavení Hadoop Cluster	-
Zprovoznění vývojového prostředí	-
Seznámení s technologií/nástrojem	-
PoC	-
Připojení do Hadoop Clusteru	-
Implementace	-
Testování	-

Tabulka 8.1: Tabulka pro zápis pracnosti.

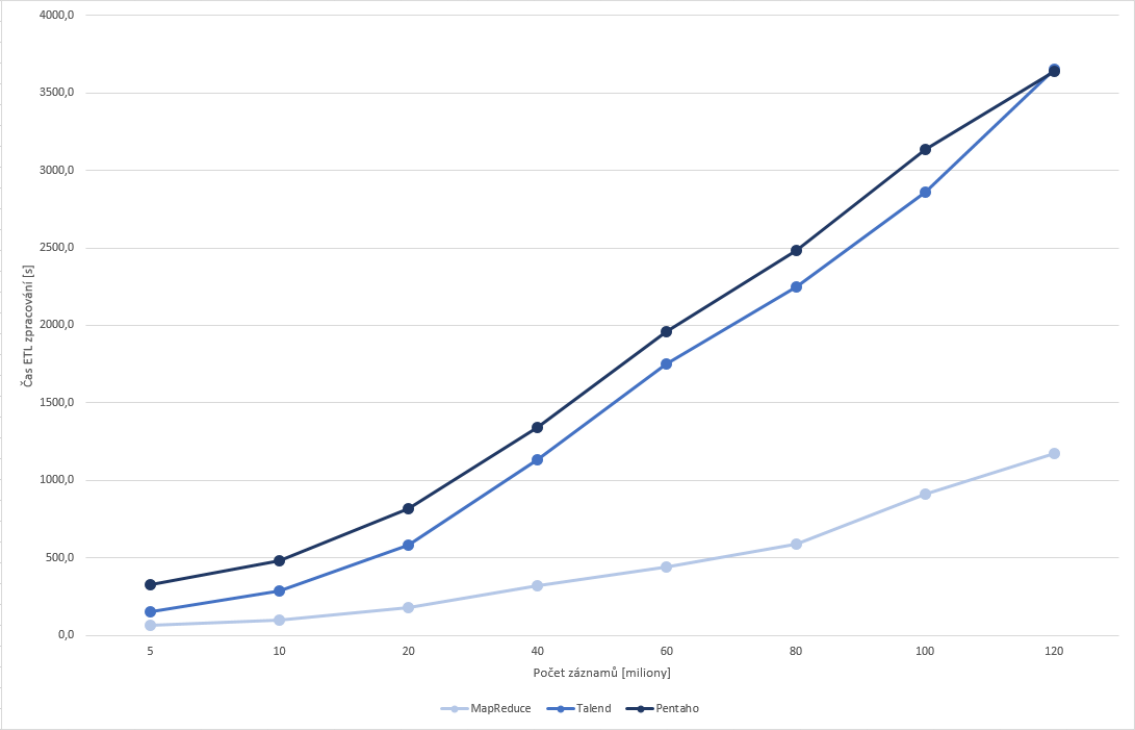
Pracnost byla zaznamenávána průběžně během implementace do navržené tabulky (viz 8.1). Některé položky jsou pro všechny nástroje shodné, jako například instalace Hadoop Clusteru. Čas spotřebovaný pro měření škálovatelnosti není v těchto hodnotách obsažen.

8.2 Měření škálovatelnosti na základě zvětšujícího se objemu dat

Myšlenka zpracování Big Data je zaměřena na zpracování dat velkého objemu za vysoké rychlosti, tak aby se data nestala nevalidními z důvodu jejich stáří. Cílem tohoto měření bylo porovnat rychlost zpracování jednotlivých nástrojů pro zpracování Big Data formou ETL.

Rozhodujícím činitelem pro uváděné měření je čas, za který jsou data zpracovány.

Následující graf zobrazuje časy ETL zpracování v závislosti na testovací datové instanci pro použité ETL nástroje. Hodnoty vynesené do grafu jsou hodnoty průměrné, které byly spočítány jako aritmetické průměry všech měřeních na dané instanci. K těmto měřením byla spočítána směrodatná odchylka. Konkrétní grafy pro daný nástroj s vynesenou směrodatnou odchylkou a zdrojovými daty lze najít v příloze (viz D).



Obrázek 8.1: Průměrné hodnoty ETL zpracování v závisloti na datové sadě pro ETL nástroje.

Nástroj/Dataset	5M	10M	20M	40M	60M	80M	100M	120M
MapReduce	65,3	98,7	177,7	320,1	441,5	592,8	910,2	1174,5
TOS	150,8	290,1	580,1	1132,6	1751,2	2246,1	2860,2	3648,9
PDI	329,5	484,1	821,1	1344,0	1962,8	2482,3	3131,9	3637,0

Tabulka 8.2: Zdrojová data pro graf závislosti času ETL zpracování na datové sadě pro ETL nástroje.

Dle výsledného grafu je zřejmé, že nativní zpracování pomocí MapReduce je mnohem rychlejší než ostatní nástroje. To je způsobeno tím, že nativní ETL zpracování BD odpovídá standardnímu ELT zpracování, jelikož transformace probíhá přímo na Hadoop clusteru a ne mimo. Díky této vlastnosti má velikou výhodu proti ostatním řešením – nedochází k přenosu dat mimo systém.

TOS a PDI má nevýhodu v přenosu dat mimo Hadoop cluster. K tomu dochází pomocí JDBC ovladače a přenosu dat po síti. Využití JDBC ovladače přináší nevýhodu v omezené propustnosti ovladače, záleží však na dané implementaci ovladače. Další zpoždění může nastat při přetížené síti nebo její malé propustnosti. Dle naměřených dat, čtení pomocí JDBC ovladače nástrojům zabralo cca 93-96 % času pro TOS, respektive 43-74 % času pro PDI, v závislosti na datové instanci. Samotné zpracování dat a následný zápis výsledku probíhá v řádu několika desítek až stovek sekund.

Nástroj/Dataset	**5M**	**10M**	**20M**	**40M**	**60M**	**80M**	**100M**	**120M**
TOS	9,4	14,3	28,6	46,4	70,4	89,5	115,2	139,4
PDI	186,9	198,1	201,2	208,9	210,1	212,4	213,2	216,9

Tabulka 8.3: Čas zpracování a následného zápisu dat v závislosti na datové sadě pro TOS a PDI.

Výsledný graf rovněž ukazuje, že se zvětšujícím se objemem dat se časy zpracování pomocí TOS a PDI více přibližují. Dle naměřených hodnot (viz 8.2), TOS obecně zpracovávalo data rychleji než PDI. To může být z určité části způsobeno tím, že PDI pro zpracování využívá více komponent než TOS. Na základě výsledků lze usuzovat, že PDI využívá JDBC ovladač s větší propustností dat a čtení ze systému je rychlejší, proto dokáže vyrovnat výkonovou ztrátu při zpracování. Dalším faktem je, že PDI má menší časové rozdíly ve zpracování nejmenší a největší datové sady cca 30 s, naproti tomu u TOS je rozdíl 130 s. Skutečnost lze vysvětlit využíváním přidělené paměti. TOS vždy využívalo 2-2,5 GB ze 4 GB přidělených, naproti tomu PDI vždy atakovalo horní hranici přidělených 4 GB. Lze tedy usuzovat, že PDI je sice pro menší datové sady pomalejší, ale celkově na dobu zpracování stabilnější než TOS.

Naměřená data prokazatelně ukazují, že nativní řešení dosahuje největší rychlosti zpracování na všech testovacích instancích. Rychlost zpracování by šla ještě více navýšit, a to rozšířením clusteru o více prvků a následnou distribucí dat. TOS a PDI oproti nativnímu zpracování zaostává hlavně z důvodu využití JDBC ovladače.

8.3 Porovnání pracnosti navržených řešení vůči dosaženému výkonu

Hlavní myšlenkou tohoto porovnání je srovnat, zda se vyplatí investovat čas do implementace, která dosahuje nejvyšších výkonových hodnot nebo zvolit implementaci méně časově náročnou na úkor nižšího výkonu.

Aby bylo porovnání co nejvíce přesné, všechna práce na vývoji řešení byla zaznamenávána do přehledné tabulky (viz 8.1). Zároveň jsem se snažil eliminovat znalost MapReduce programovacího modelu, která mi v této implementaci poskytovala velikou výhodu.

K vývoji jsem ve všech případech přistupoval stejným způsobem. Nejprve bylo zapotřebí nainstalovat Hadoop Cluster, poté nainstalovat nástroj a seznámit se s ním. Následně jsem na základě návodů dostupných na stránkách výrobců implementoval PoC, abych vyzkoušel, jak daný nástroj pracuje. Pak jsem řešil připojení do CDH [1], samotnou implementaci a otestoval správnou funkčnost.

Činnost	MapReduce	TOS	PDI
Instalace a nastavení Hadoop Cluster	1	1	1
Zprovoznění vývojového prostředí	-	0,5	0,5
Seznámení s technologií/nástrojem	1,5	0,5	0,5
PoC	0,5	0,5	0,5
Připojení do Hadoop Clusteru	-	1	1
Implementace	2	0,2	0,5
Testování	0,5	0,5	0,5
Celkem	**5,5**	**4,2**	**4,5**

Tabulka 8.4: Zaznamenaná pracnost implementace pro ETL nástroje.

Z porovnání zaznamenané pracnosti vychází nejlépe nástroj od Talendu. Přibližně stejnou dobu trval vývoj pomocí Pentaho. MapReduce je na implementaci náročnější než využité nástroje. Je nutné si ale uvědomit, že se jedná o malý testovací projekt a rozdíly v délce vývoje nejsou tak markantní.

Při úvaze většího projektu by byl vývoj pomocí MapReduce mnohem náročnější, jelikož se zde složitě programuje propojení více vstupních souborů. MapReduce už ze své podstaty není zcela vhodný pro malé projekty nebo projekty krátkodobého trvání, jelikož vývoj MapReduce programu je mnohem nákladnější než využití některých jiných nástrojů. Dále je zapotřebí také zvážit to, že je nutné mít v týmu jednoho nebo několik programátorů, kteří ovládají programovací jazyk, v němž lze MapReduce implementovat. Na základě těchto specifik lze MapReduce doporučit na projekty nebo aplikace dlouhodobého trvání, kde se vyplatí investovat čas do implementace, poněvadž se se z výkonu bude těžit dlouhou dobu.

Velice blízce výkonově i pracností vývoje se k sobě blíží nástroje od Talendu a Pentaho. Díky dosaženým výsledům lze tyto nástroje doporučit na projekty krátkodobé i středně dlouhé nebo dlouhodobé, pokud bude dostatečná rychlost zpracování dat. Z pohledu implementace řešení jsou tyto nástroje nezanedbatelnou mírou výhodnější než implementace MapReduce. Nástroj od Talendu poskytuje o něco lepší výkon a implementace je méně náročnější. Na druhou stranu je zapotřebí vzít v úvahu, že se vývojáři budou muset učit s novým nástrojem. PDI je oproti TOS velice rozšířený. Z tohoto pohledu je PDI i přes své výkonnostní rezervy dle mého názoru, lepší volbou.

Dle potřebné pracnosti pro vývoj a na základě dosaženého výkonu lze za nejvhodnější řešení pro testovací projekt považovat nástroj od Talendu.

[1] Do tohoto času nebyly započítány problémy s připojením způsobené starou verzí VMware Worksation Playeru

Kapitola 9

Závěr

V diplomové práci jsem popsal myšlenku, specifika a vlastnosti Big Data. Shrnutí vlastností této oblasti není zcela jednoduché, protože data jsou stále ve vývoji a jejich hlavní vlastnosti se mění. V práci byly rovněž uvedeny některé příklady využití a potenciálního prospěchu využití Big Data. S ohledem na zaměření diplomové práce jsem specifikoval ETL zpracování dat a popsal nástroje pro zpracování Big Data formou ETL.

Hlavním cílem práce bylo porovnat aktuálně dostupné nástroje pro zpracování Big Data formou ETL na základě jejich škálovatelnosti s ohledem na zvětšující se objem dat a porovnat pracnost vývoje řešení vůči dosaženému výkonu. V rámci porovnání bylo porovnáno nativní řešení ETL pro BD pomocí programovacího modelu MapReduce, nástroj Talend Open Studio for Big Data a Pentaho Data Integration. Pro porovnání byly připraveny testovací sady generovaných bankovních dat o objemu od 5 do 120 milionu záznamů (viz 8.1).

Během měření škálovatelnosti bylo jednoznačně prokázáno, že nativní řešení je mnohem rychlejší než ostatní, a to zejména z důvodu využití JDBC ovladače pro čtení dat z databáze. Řešení od Talendu a Pentaho s rostoucím množstvím dat dosahují podobných výkonnostních výsledků (viz 8.2).

Při porovnání pracnosti vyšla implementace MapReduce programu nejméně vhodná oproti ostatním řešením. Z tohoto důvodu byl MapReduce doporučen pro velké komplexní projekty, kde se vyplatí investovat čas pro daný výkon. Nejvýhodnějším řešením na základě dosaženého výkonu a nejméně potřebného času pro implementaci je řešení od Talendu. Tento nástroj lze doporučit na projekty krátkodobého trvání i projekty dlouhodobého trvání, pokud není nutný vysoký výkon. Na druhou stranu výkonový rozdíl TOS a PDI není tak markantní a s přihlédnutím na skutečnost, že PDI je velice známé a využívané, lze PDI doporučit pro stejné účely jako TOS.

Zdrojové kódy a implementovaná řešení pomocí nástrojů jsou přiloženy na CD, viz A.

Literatura

[1] ACCENTURE. *Většina manažerů považuje big data za velmi důležitá* [online]. Marketingovenoviny.cz, 2015. [cit. 8. 4. 2018]. Dostupné z: <http://www.marketingovenoviny.cz/vetsina-manazeru-povazuje-big-data-za-velmi-dulezita/>.

[2] AKHGAR, B. et al. *Application of Big Data for National Security.* Butterworth-Heinemann, 2015.

[3] AUGUSTÍN, J. *BIG DATA A MOŽNOSTI JEJICH VYUŽITÍ* [online]. Adastra, s.r.o., 2014. [cit. 26. 3. 2018]. Dostupné z: <http://www.adastra.cz/clanky/big-data-a-moznosti-jejich-vyuziti>.

[4] BEYER, M. – LANEY, D. *The Importance of 'Big Data': A Definition* [online]. Gartner, Inc., 2012. [cit. 6. 2. 2018]. Dostupné z: <https://www.gartner.com/doc/2057415/importance-big-data-definition>.

[5] BOLLIER, D. *The Promise and Peril of Big Data* [online]. THE ASPEN INSTITUTE, 2010. [cit. 6. 1. 2018]. Dostupné z: <https://www.emc.com/collateral/analyst-reports/10334-ar-promise-peril-of-big-data.pdf>.

[6] BRAEGER, M. – DEVGAN, M. *Unlocking Big Data at CERN* [online]. Terracotta, 2014. [cit. 19. 12. 2017]. Dostupné z: <http://blog.terra-cotta.org/wp-content/uploads/2014/10/Unlocking-Big-Data-at-CERN.pdf>.

[7] BREWER, E. *Towards robust distributed systems* [online]. PODC, 2000. [cit. 9. 3. 2018]. Dostupné z: <https://www.researchgate.net/publication/221343719_Towards_robust_distributed_systems>.

[8] BUYYA, R. – CALHEIROS, R. – DASTJERDI, A. V. *Big Data Principles and Paradigms.* Morgan Kaufmann, 2016.

[9] CARTER, P. *Big Data Analytics: Future Architectures, Skills and Roadmaps for the CIO* [online]. IDC, 2011. [cit. 20. 1. 2018]. Dostupné z: <https://triangleinformationmanagement.com/wp-content/uploads/2013/12/bigdata-idc-wp.pdf>.

[10] CUKIER, K. – MAYER-SCHöNBERGER, V. *Big Data.* Brno : Computer Press s.r.o, 2014.

[11] Cyclone Interactive. *The DIGITAL UNIVERSE of OPPORTUNITIES* [online]. IDC, 2014. [cit. 9. 1. 2018]. Dostupné z: <https://www.emc.com/collateral/analyst-reports/idc-digital-universe-2014.pdf>.

[12] D., A. et al. *Challenges and Opportunities with Big Data: A white paper prepared for the Computing Community Consortium committee of the Computing Research Association* [online]. 2012. [cit. 19. 2. 2018]. Dostupné z: <https://cra.org/ccc/wp-content/uploads/sites/2/2015/05/bigdatawhitepaper.pdf>.

[13] DEMCHENKO, Y. *Addressing Big Data Issues in the Scientific Data Infrastructure* [online]. SNE Group, University of Amsterdam, 2013. [cit. 6. 2. 2018]. Dostupné z: <https://tnc2013.terena.org/includes/tnc2013/documents/bigdata-nren.pdf>.

[14] DOLáK, O. *Big data Nové způsoby zpracování a analýzy velkých objemů dat* [online]. SystemOnLine.cz, 2011. [cit. 7. 4. 2018]. Dostupné z: <https://www.systemonline.cz/clanky/big-data.htm>.

[15] ELLIOTT, T. *More Big Data Vs Value And Veracity* [online]. Digitalist Magazine, 2014. [cit. 19. 12. 2017]. Dostupné z: <http://www.digitalistmag.com/technologies/big-data/2014/01/23/2-more-big-data-vs-value-and-veracity-01242817>.

[16] EVANS, B. *The Top 5 Cloud-Computing Vendors* [online]. Forbes, 2017. [cit. 24. 4. 2018]. Dostupné z: <https://tinyurl.com/yd8vpl3u>.

[17] GEWIRTZ, D. *Volume, velocity, and variety: Understanding the three V's of big data* [online]. ZDNet, 2018. [cit. 17. 1. 2018]. Dostupné z: <https://www.zdnet.com/article/volume-velocity-and-variety-understanding-the-three-vs-of-big-data/>.

[18] GHEMAWAT, S. – GOBIOFF, H. – LEUNG, S.-T. *The Google File System* [online]. 2003. [cit. 14. 3. 2018]. Dostupné z: <https://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf>.

[19] GILL, N. S. *Data Ingestion and Processing of Data For Big Data and IoT Solutions* [online]. XenonStack, 2017. [cit. 21. 1. 2018]. Dostupné z: <https://www.xenonstack.com/blog/data-engineering/ingestion-processing-data-for-big-data-iot-solutions>.

[20] GROSSMAN, L. *How Computers Know What We Want — Before We Do* [online]. Time, Inc., 2010. [cit. 21. 3. 2018]. Dostupné z: <https://tinyurl.com/ycg65l8r>.

[21] HASSANIEN, A. et al. *Big Data in Complex Systems*. Springer International Publishing, 2015.

[22] HOLUBOVá, I. et al. *Big Data a NoSQL databáze*. Praha : Grada Publishing, a.s., 2015.

[23] HUGG, J. *Fast data: The next step after big data* [online]. IDG Communications, Inc., 2014. [cit. 12. 5. 2018]. Dostupné z: <https://www.infoworld.com/article/2608040/big-data/fast-data--the-next-step-after-big-data.html>.

[24] HURWITZ, J. et al. *Big Data For Dummies* [online]. John Wiley & Sons, Inc., 2013. [cit. 21. 12. 2017]. Dostupné z: <https://tinyurl.com/yalunmru>.

[25] KIMBALL, R. – CASERTA, J. *The Data Warehouse ETL Toolkit* [online]. Wiley Publishing, Inc., 2004. [cit. 17. 1. 2018]. Dostupné z: <https://tinyurl.com/y8zlyqm3>.

[26] Komunita autorů Apache Hive wiki. *Apache Hive* [online]. Apache Software Foundation. [cit. 22. 1. 2018]. Dostupné z: <https://cwiki.apache.org/confluence/display/Hive/Home>.

[27] Komunita autorů Pentaho Community Wiki. *Pentaho Data Integration (Kettle) Tutorial* [online]. Pentaho Corporation. [cit. 28. 4. 2018]. Dostupné z: <https://tinyurl.com/y8uyyjog>.

[28] Komunita autorů Pentaho Documentation. *Working with Big Data and Hadoop in PDI* [online]. Pentaho Corporation, 2017. [cit. 19. 5. 2018]. Dostupné z: <https://help.pentaho.com/Documentation/7.1/0L0/040/020>.

[29] Komunita autorů Pentaho Documentation. *Schedule Jobs* [online]. Pentaho Corporation, 2016. [cit. 11. 5. 2018]. Dostupné z: <https://help.pentaho.com/Documentation/5.4/0J0/0C0/040>.

[30] Komunita autorů Talend Community. *Allocating more memory to Talend Studio* [online]. Talend, 2017. [cit. 11. 4. 2018]. Dostupné z: <https://community.talend.com/t5/Migration-Configuration-and/Allocating-more-memory-to-Talend-Studio/ta-p/21642>.

[31] Komunita autorů Talend Community. *Talend job scheduling* [online]. Talend, 2013. [cit. 4. 5. 2018]. Dostupné z: <https://community.talend.com/t5/Deployment/resolved-talend-job-scheduling/td-p/9787>.

[32] LANEY, D. *3D Data Management Controlling Data Volume Velocity and Variety* [online]. META Group, 2001. [cit. 20. 1. 2018]. Dostupné z: <https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>.

[33] MANYIKA, J. et al. *Big data: The next frontier for innovation, competition, and productivity* [online]. McKinsey Global Institute, 2011. [cit. 7. 1. 2018]. Dostupné z: <https://tinyurl.com/ycfpelwx>.

[34] MARR, B. *Big Data: The 5 Vs Everyone Must Know* [online]. Linked in, 2014. [cit. 9. 2. 2018]. Dostupné z: <https://www.linkedin.com/pulse/20140306073407-64875646-big-data-the-5-vs-everyone-must-know/>.

[35] MARR, B. *Big Data: 20 Free Big Data Sources Everyone Should Know* [online]. SmartData Collective, 2014. [cit. 9. 1. 2018]. Dostupné z: <https://www.smartdatacollective.com/big-data-20-free-big-data-sources-everyone-should-know/>.

[36] MOHANTY, S. – JAGADEESH, M. – SRIVATSA, H. *Big Data Imperatives* [online]. Apress, 2013. [cit. 12. 1. 2018]. Dostupné z: <https://tinyurl.com/ydaxdv6q>.

[37] NEWMAN, D. *Big Data: Why Facebook Knows Us Better Than Our Therapist* [online]. Forbes, 2015. [cit. 9. 3. 2018]. Dostupné z: <https://tinyurl.com/y9n8ego5>.

[38] NORMANDEAU, K. *Beyond Volume, Variety and Velocity is the Issue of Big Data Veracity* [online]. insideBIGDATA, 2013. [cit. 3. 1. 2018]. Dostupné z: <https://insidebigdata.com/2013/09/12/beyond-volume-variety-velocity-issue-big-data-veracity/>.

[39] O'BRIEN, J. *Big Data Is Changing the Game for Recruiters* [online]. Mashable, Inc., 2014. [cit. 20. 3. 2018]. Dostupné z: <https://tinyurl.com/y8qozbcw>.

[40] O'Reilly Radar Team. *Planning for Big Data* [online]. O'Reilly Media, Inc., 2012. [cit. 27. 1. 2018]. Dostupné z: <https://tinyurl.com/y937crgp>.

[41] PARKER, D. S. et al. *Detection of Mutual Inconsistency in Distributed Systems* [online]. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 1983. [cit. 13. 3. 2018]. Dostupné z: <http://zoo.cs.yale.edu/classes/cs422/2013/bib/parker83detection.pdf>.

[42] Prezentace: Talend Big Data Sandbox. Big Data Insights Cookbook. <https://info.talend.com/rs/talend/images/CB_EN_BD_BigData_Insights.pdf>, stav z 23. 4. 2018.

[43] ROGERS, S. *Big Data is Scaling BI and Analytics* [online]. Pearson Education, Inc, 2011. [cit. 28. 3. 2018]. Dostupné z: <https://www.information-management.com/news/big-data-is-scaling-bi-and-analytics>.

[44] ROUSE, M. *Extract, transform, load (ETL)* [online]. SearchData Management, 2005. [cit. 16. 1. 2018]. Dostupné z: <https://searchdatamanagement.techtarget.com/definition/extract-transform-load>.

[45] SADALAGE, P. J. – FOWLER, M. *NoSQL Distilled* [online]. Pearson Education, Inc, 2013. [cit. 8. 3. 2018]. Dostupné z: <http://bigdata-ir.com/wp-content/uploads/2017/04/NoSQL-Distilled.pdf>.

[46] SHRIVASTAVA, R. *Big Data : Parallelism and Hadoop:Basics* [online]. Codemphasis.wordpress.com, 2012. [cit. 8. 5. 2018]. Dostupné z: <https://codemphasis.wordpress.com/2012/08/13/big-data-parallelism-and-hadoopbasics/>.

[47] SILVA, L. M. M. *ETL in the Big Data Era* [online]. Instituto Superior Tcnico. [cit. 13. 1. 2018]. Dostupné z: <https://fenix.tecnico.ulisboa.pt/downloadFile/1689244997255767/Resumo.pdf>.

[48] SKLENáK, V. *Data, informace, znalosti a Internet* [online]. C. H. Beck, 2001. [cit. 24. 1. 2018]. Dostupné z: <https://tinyurl.com/yau9fw7s>.

[49] STAMFORD, C. *Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data* [online]. Gartner, Inc., 2011. [cit. 2. 2. 2018]. Dostupné z: <https://www.gartner.com/newsroom/id/1731916>.

[50] STEWARD, D. *Big Content: The Unstructured Side of Big Data* [online]. Gartner, Inc., 2013. [cit. 12. 2. 2018]. Dostupné z: <https://blogs.gartner.com/darin-stewart/2013/05/01/big-content-the-unstructured-side-of-big-data/>.

[51] SUTHAHARAN, S. *Big Data Classification: Problems and Challenges in Network Intrusion Prediction with Machine Learning* [online]. The University of North Carolina at Greensboro (UNCG), 2014. [cit. 29. 1. 2018]. Dostupné z: <https://libres.uncg.edu/ir/uncg/f/S_Suthaharan_Big_2014.pdf>.

[52] TECHROBA. *10 Open Source ETL Tools* [online]. Data Science Central, 2015. [cit. 9. 5. 2018]. Dostupné z: <https://www.datasciencecentral.com/profiles/blogs/10-open-source-etl-tools>.

[53] MANEN, P. *Better Baby Care - thanks Formula 1* [online]. TED, 2013. [cit. 21. 3. 2018]. Dostupné z: <https://www.ted.com/talks/peter_van_manen_how_can_formula_1_racing_help_babies>.

[54] RIJMENAM, M. *Why The 3V's Are Not Sufficient To Describe Big Data* [online]. Datafloq - Driving Innovation, 2013. [cit. 8. 2. 2018]. Dostupné z: <https://datafloq.com/read/3vs-sufficient-describe-big-data/166>.

[55] VAUGHN, C. *Multichannel vs Omnichannel Marketing* [online]. GRANIFY, INC., 2017. [cit. 13. 5. 2018]. Dostupné z: <https://www.granify.com/blog/multichannel-vs-omnichannel-marketing>.

[56] VERRILLI, M. *From Lambda to Kappa: A Guide on Real-time Big Data Architectures* [online]. Talend, 2017. [cit. 27. 4. 2018]. Dostupné z: <https://www.talend.com/blog/2017/08/28/lambda-kappa-real-time-big-data-architectures/>.

[57] WEB: Adastra.cz. ETL/ELT NÁSTROJE – SRDCE VAŠICH DATABÁZÍ. <http://www.adastra.cz/technologie/etl-elt>, stav z 12. 2. 2018.

[58] WEB: Cloudera.com. QuickStarts for CDH 5.12. <https://www.cloudera.com/downloads/quickstart_vms/5-12.html>, stav z 15. 4. 2018.

[59] WEB: Community.hitachivantara.com. Pentaho Community Edition 8.0. <https://community.hitachivantara.com/docs/DOC-1009931-downloads>, stav z 15. 4. 2018.

[60] WEB: Cs.calvin.edu. MapReduce Exercise: Hands-On Lab. <https://cs.calvin.edu/courses/cs/374/exercises/12/lab/>, stav z 7. 5. 2018.

[61] WEB: Datawarehouse4u.info. ETL process. <http://datawarehouse4u.info/ETL-process.html>, stav z 2. 1. 2018.

[62] WEB: Gethue.com. HUE.
<http://gethue.com/>, stav z 2. 5. 2018.

[63] WEB: Hadoop.apache.org. Apache Hadoop, HDFS Architecture, HDFS Architecture Guide, MapReduce Tutorial.
<http://hadoop.apache.org/>, stav z 7. 5. 2018.

[64] WEB: Impala.apache.org. Impala.
<https://impala.apache.org/>, stav z 1. 5. 2018.

[65] WEB: Internetlivestats.com. Internet Live Stats.
<http://www.internetlivestats.com/>, stav z 6. 2. 2018.

[66] WEB: Kudu.apache.org. Apache Kudu.
<https://kudu.apache.org/>, stav z 2. 5. 2018.

[67] WEB: Managementmania.com. Profilování (Profiling).
<https://managementmania.com/cs/profilovani-profiling>, stav z 5. 2. 2018.

[68] WEB: Oracle.com. Oracle Big Data.
<https://www.oracle.com/cz/big-data/index.html//>, stav z 27. 12. 2017.

[69] WEB: Sas.com. What Is ETL?
<https://tinyurl.com/yaxd8u7v>, stav z 12. 1. 2018.

[70] WEB: Talend.com. Talend Open Studio, Open Source Integration Software, Big Data Integration Products, ETL with Hadoop, Open Studio for Data Integration, Open Studio for Big Data.
<https://www.talend.com>, stav z 2. 5. 2018.

[71] WEB: Voltdb.com. Big Data.
<https://www.voltdb.com/why-voltdb/big-data/>, stav z 12. 5. 2018.

[72] WHITE, T. *Hadoop The Definitive Guide* [online]. O'Reilly Media, Inc., 2015. [cit. 16. 1. 2018]. Dostupné z: <http://javaarm.com/file/apache/Hadoop/books/Hadoop-The.Definitive.Guide_4.edition_a_Tom.White_April-2015.pdf>.

[73] WILSON, G. *Eight Fallacies of Distributed Computing - Tech Talk* [online]. Fog Creek Software, Inc., 2015. [cit. 16. 3. 2018]. Dostupné z: <https://blog.fogcreek.com/eight-fallacies-of-distributed-computing-tech-talk/>.

[74] YADAV, V. *Processing Big Data with Azure HDInsight: Building Real-World Big Data Systems on Azure HDInsight Using the Hadoop Ecosystem* [online]. Apress, 2017. [cit. 25. 4. 2018]. Dostupné z: <https://tinyurl.com/yalzba9k>.

[75] ZASLAVSKY, A. – PERERA, C. – GEORGAKOPOULOS, D. *Sensing as a Service and Big Data* [online]. The Australian National University. [cit. 5. 5. 2018]. Dostupné z: <https://arxiv.org/ftp/arxiv/papers/1301/1301.0159.pdf>.

[76] ZIKOPOULOS, P. et al. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data* [online]. The McGraw-Hill Companies, 2011. [cit. 22. 1. 2018]. Dostupné z: <https://www.immagic.com/eLibrary/ARCHIVES/EBOOKS/I111025E.pdf>.

Příloha A

Obsah přiloženého CD

Součástí diplomové práce je přiložené CD, které obsahuje samotnou diplomovou práci v elektronické podobě, zdrojové kódy Java aplikací, projekty nástrojů Pentaho a Talend, grafy s naměřenými daty a spustitelné bash scripty.

V kořenovém adresáři se nachází soubor DP_Slavicek_2018.pdf, který obsahuje diplomovou práci. Dále jsou zde adresáře Pentaho_ETL, Talend_ETL, MapReduce_ETL, Data_Generator, Data, Cloudera_Hadoop, Data_Hadoop, DP_tex.

Adresář DP_tex obsahuje diplomovou práci vypracovanou v LaTexu. Adresář Data obsahuje naměřené hodnoty, směrodatné odchylky a výsledné grafy, porovnávající ETL implementace.

Adresář Pentaho_ETL obsahuje soubory ETL jobu a transformace spustitelné v PDI. Adresář Talend_ETL obsahuje projekt BIG_DATA_ETL, který lze importovat do TOS a následně spustit. MapReduce_ETL obsahuje Java kód implementace programovacího modelu MapReduce a jar spustitelný na Hadoopu. Adresář Data_Generator obsahuje Java kód generátoru dat a pom.xml soubor pro Maven. Adresář Data_Hadoop obsahuje txt soubory – číselníky PRODUCT_TYPE, CURRENCY a ukázku vygenerovaných dat. V posledním adresáři Cloudera_Hadoop jsou uloženy spustitelné Bash scripty pro MapReduce, nahrání dat do HDFS a load dat do Hive.

Příloha B

Seznam použitých zkratek

- JVM – Java Virtual Machine
- PDI – Pentaho Data Integration
- TOS – Talend Open Studio
- IoT – Internet of Things
- BD – Big Data
- CDH – Cloudera Distribution Including Apache Hadoop
- MD – Man Day
- PoC – Proof of concept
- ETL – Extrakce, transformace, nahrání
- ELT – Extrakce, nahrání, transformace
- ID – Identifikátor
- JDBC – Java Database Connectivity

Příloha C

Generátor dat

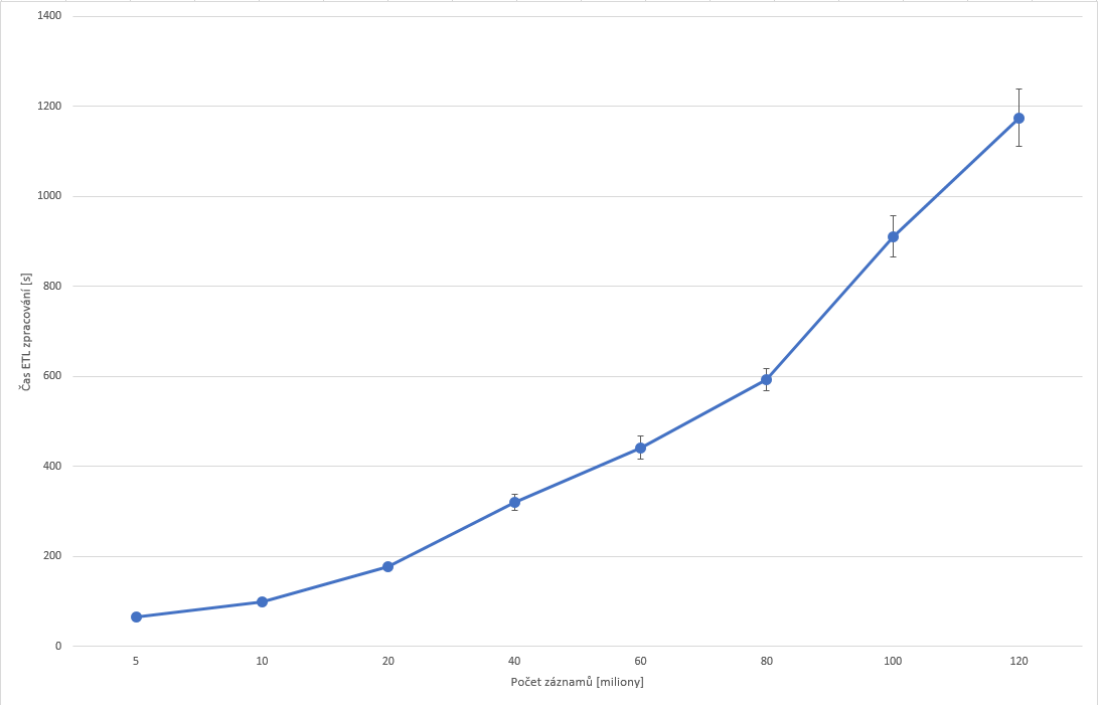
```
import org.fluttercode.datafactory.impl.DataFactory;

import java.io.FileWriter;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

/*
* @author Ondrej Slavicek
* Test data generator - Transactions
* */
public class Main {

   static int SEQUENCE_START = 0;
   static int DATA_COUNT = 1000;

   public static void main(String[] args) {

      DataFactory df = new DataFactory();

      Date minDate = df.getDate(2018, 4, 3);
      Date maxDate = df.getDate(2018,4,5);

      FileWriter fileWriter = null;

      try {

      fileWriter = new FileWriter("datasetTest.txt");

      } catch (IOException e) {
         System.out.println("Error while creating data file.");
         e.printStackTrace();
      }

      for (int i = SEQUENCE_START; i < (SEQUENCE_START + DATA_COUNT); i++) {
```

```
String[] productTypeUIDs = {"BU-FO","BU-FOP","BU-PO","PK-ST","PK-EX", "SP",
    "AH", "STU", "HU", "SU"};
String[] currencyUIDs = {"CZK", "EUR", "USD", "GBP"};

// Next int sequence
String transactionID = String.valueOf(i);

// Generate account ID
int accountID = df.getNumberBetween(100, 700100);

// Generate product type by divisor factor
String productTypeUID = productTypeUIDs[9];
if (accountID % 23 == 0){
   productTypeUID = productTypeUIDs[8];
} else if (accountID % 19 == 0) {
   productTypeUID = productTypeUIDs[7];
} else if (accountID % 17 == 0) {
   productTypeUID = productTypeUIDs[6];
} else if (accountID % 13 == 0) {
   productTypeUID = productTypeUIDs[5];
} else if (accountID % 11 == 0) {
   productTypeUID = productTypeUIDs[4];
} else if (accountID % 7 == 0) {
   productTypeUID = productTypeUIDs[3];
} else if (accountID % 5 == 0) {
   productTypeUID = productTypeUIDs[2];
} else if (accountID % 3 == 0) {
   productTypeUID = productTypeUIDs[1];
} else if (accountID % 2 == 0) {
   productTypeUID = productTypeUIDs[0];
}

// Generate currency by divisor factor
String currencyUID = currencyUIDs[0];
if (accountID % 37 == 0) {
   currencyUID = currencyUIDs[3];
} else if (accountID % 31 == 0) {
   currencyUID = currencyUIDs[2];
} else if (accountID % 29 == 0) {
   currencyUID = currencyUIDs[1];
}

// Generate transaction amount
String transactionAmount = String.valueOf(df.getNumberBetween(-10000,
    100000));

// Generate transaction date
Date transactionDate = df.getDateBetween(minDate, maxDate);
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

// Generate transaction note
String transactionNote = df.getRandomWord(4, 100);
```

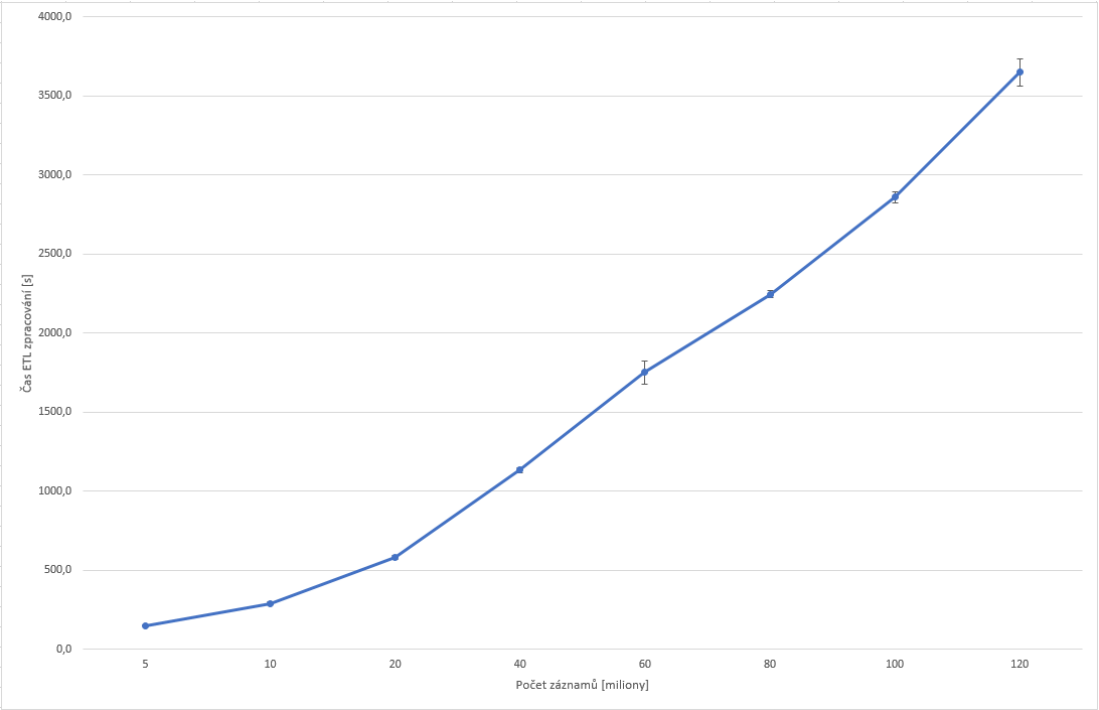
```
      String generatedTransaction = transactionID + "\t" +
          String.valueOf(accountID) + "\t" + productTypeUID + "\t" +
          transactionAmount
      + "\t" + currencyUID + "\t" + dateFormat.format(transactionDate) + "
          00:00:00" + "\t" + transactionNote + "\n";

      try {
         fileWriter.append(generatedTransaction);
      } catch (IOException e) {
         System.out.println("Error while writting to data file.");
         e.printStackTrace();
      }

      }

      try {
         fileWriter.flush();
         fileWriter.close();
      } catch (IOException e) {
         System.out.println("Error while flushing/closing fileWriter.");
         e.printStackTrace();
      }

   }

}
```

Příloha D

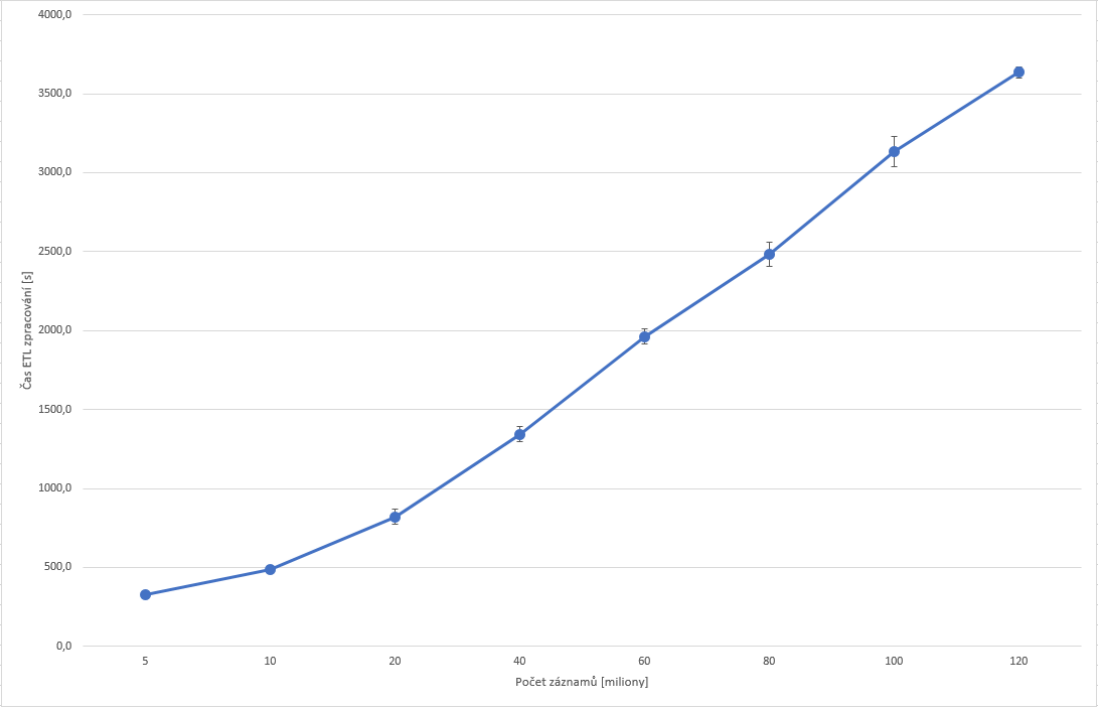
Vizualizace naměřených hodnot včetně směrodatných odchylek



Obrázek D.1: Průměrné hodnoty ETL zpracování v závisloti na datové sadě pro MapReduce.



Obrázek D.2: Průměrné hodnoty ETL zpracování v závisloti na datové sadě pro Talend.



Obrázek D.3: Průměrné hodnoty ETL zpracování v závisloti na datové sadě pro Pentaho.