

Big Data – Metody zpracování a analýzy velkých dat

Michal Karafiát

Bakalářská práce
2017



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal Karafiát**
Osobní číslo: **A12776**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Big Data - Metody zpracování a analýzy velkých dat**
Téma anglicky: **Big Data Processing and Analysis Methods**

Zásady pro vypracování:

1. Vypracujte obecnou literární rešerši na dané téma.
2. Provedte analýzu, popis a identifikaci oblastí, kde se velká data prakticky využívají.
3. Provedte analýzu a popis aktuálního stavu problematiky technologií zpracování a analýzy velkých dat, včetně souborových systémů inteligentních řešení.
4. Provedte analýzu použití dostupných technologií od Google, Apache, Microsoft, atd.
5. Provedte popis, včetně praktických příkladů technologie MapReduce pro práci s velkými daty.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Data science & big data analytics: discovering, analyzing, visualizing and presenting data.** Indianapolis: Wiley, 2015, xviii, 410 stran. ISBN 978-1-118-87613-8.
2. **HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK.** Big Data a NoSQL databáze. První vydání. Praha: Grada, 2015, 281 stran. ISBN 978-80-247-5466-6.
3. **MARZ, Nathan a James WARREN.** Big data: principles and best practices of scalable real-time data systems. Shelter Island: Manning, 2015, xx, 308 stran. ISBN 978-1-617290-34-3.
4. **LABERGE, Robert.** Datové sklady: agilní metody a business intelligence. 1. vyd. Brno: Computer Press, 2012, 350 s. ISBN 978-80-251-3729-1.
5. **MAYER-SCHÖNBERGER, Viktor a Kenneth CUKIER.** Big Data: revoluce, která změní způsob, jak žijeme, pracujeme a myslíme. 1. vyd. Brno: Computer Press, 2014, 256 s. ISBN 978-80-251-4119-9.
6. **EAGLE, Nathan a Kate GREENE.** Reality mining: using big data to engineer a better world. Cambridge, Massachusetts: The MIT Press, 2014, 1 online zdroj (vi, 199 pages). ISBN 9780262324564.

Vedoucí bakalářské práce:

doc. Ing. Roman Šenkeřík, Ph.D.

Ústav informatiky a umělé inteligence

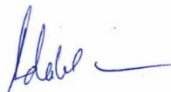
Datum zadání bakalářské práce:

24. února 2017

Termín odevzdání bakalářské práce:

24. května 2017

Ve Zlíně dne 24. února 2017



doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

.....
podpis diplomanta

ABSTRAKT

Bakalářská práce je rozdělená na teoretickou a praktickou část. V teoretické části se nachází vypracována obecná literární rešerše na téma Big Data, podávající základní přehled o tomto oboru, identifikaci oblastí, kde se tento přístup prakticky uplatňuje, včetně obecného popisu, jakým způsobem. Dále zahrnuje proces zpracovávání dat, včetně vyjmenování užívaných souborových systémů, databázových systémů, programovacích modelů a analytických metod. Praktická část podává porovnání současného stavu cloud computingových služeb od firmy Amazon, Google a Microsoft a v poslední části je představena současná verze technologie Apache Hadoop, jeho implementace a provedení benchmarkového testu TeraSort.

Klíčová slova: Big Data, Big Data Analytika, DFS, NoSQL, Programovací modely, Apache Hadoop, Cloud Computing, AWS, Google Cloud Platform, Microsoft Azure, MapReduce, Terasort

ABSTRACT

The bachelor thesis is divided into theoretical and practical part. The theoretical part consists of literature research on the topic of Big Data, providing general overview of the field and identifying areas this approach is practiced, including description of its practised ways. It further includes the process of data processing, as well as enumeration of file systems, database systems, programming models and analytical methods. The practical part include comparison of cloud computing services developer by Amazon, Google and Microsoft companies and in the last part is presented the current version of Apache Hadoop technology, its implementation and performance of the benchmark test TeraSort.

Keywords: Big Data, Big Data Analytics, DFS, NoSQL, Programming models, Apache Hadoop, Cloud Computing, AWS, Google Cloud Platform, Microsoft Azure, MapReduce, Terasort

Děkuji svému vedoucímu bakalářské práce, doc. Ing. Romanu Šenkeříkovi Ph.D., za poskytnuté cenné rady, připomínky a odborné vedení. Dále panu, Ing. Davidu Malaníkovi Ph.D., za poskytnutí přístupu na školní HW infrastrukturu. Také bych chtěl poděkovat své rodině, mým přátelům a přítelkyni za trpělivost a podporu.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 BIG DATA	11
1.1 DEFINICE.....	13
1.2 CHARAKTERISTIKA.....	14
1.2.1 Objem.....	14
1.2.2 Různorodost	15
1.2.3 Rychlost.....	15
1.2.4 Věrohodnost	16
1.2.1 Další uváděné vlastnosti.....	16
1.3 ROZDĚLENÍ ZDROJŮ DAT	16
1.4 OBLASTI VYUŽITÍ	17
1.4.1 Průmysl	17
1.4.2 Věda a výzkum.....	18
1.4.3 Zdravotní péče.....	18
1.4.4 Vývoj technických zařízení.....	18
1.4.5 Sociální sítě	19
1.4.6 Maloobchod.....	19
1.4.7 Finance a bankovníctví	19
2 ZPRACOVÁNÍ BIG DAT	20
2.1 SBĚR A NAHRÁVÁNÍ DAT	20
2.2 EXTRAKCE INFORMACÍ, ČIŠTĚNÍ, ANOTACE	21
2.3 DATOVÁ INTEGRACE, AGREGACE A REPREZENTACE	21
2.4 ZPRACOVÁVÁNÍ DOTAZŮ, VYTVÁŘENÍ MODELŮ A ANALÝZA.....	21
2.5 INTERPRETACE	22
3 MECHANISMY PRO SPRÁVU DAT	23
3.1 SOUBOROVÉ SYSTÉMY	23
3.1.1 GFS	24
3.1.2 HDFS.....	25
3.1.3 GlusterFS.....	27
3.2 DATABÁZOVÉ SYSTÉMY	28
3.2.1 CAP teorém.....	28
3.2.2 Rozdělení vlastností SQL a NoSQL databází v kontextu s CAP.....	29
3.2.1 NoSQL	30
3.3 PROGRAMOVACÍ PARADIGMATA.....	33
3.3.1 MapReduce	33
3.3.2 Funkcionální programování	34
3.3.3 Actor model.....	34
3.3.4 Statistické a analytické modely.....	34
3.3.5 Data-flow.....	35
3.3.6 Bulk Synchronous Parallel.....	35
4 BIG DATA ANALYTICKÉ METODY	36

4.1	DĚLENÍ PODLE HLOUBKY ANALÝZY	36
4.1.1	Deskriptivní analytika	36
4.1.2	Prediktivní analytika	37
4.1.3	Preskriptivní analytika	37
4.2	DĚLENÍ PODLE PŘEDMĚTU ANALÝZY	38
4.2.1	Analýza strukturovaných dat	38
4.2.2	Analýza webových dat	38
4.2.3	Analýza textu	39
4.2.4	Analýza multimédií	40
II	PRAKTICKÁ ČÁST	41
5	ÚVOD DO PRAKTICKÉ ČÁSTI.....	42
6	ANALÝZA CLOUDOVÝCH BIG DATA ŘEŠENÍ	43
6.1	CLOUD COMPUTING	43
6.2	AMAZON WEB SERVICES.....	44
6.3	GOOGLE CLOUD PLATFORM.....	45
6.4	MICROSOFT AZURE	45
6.5	POROVNÁNÍ ŘEŠENÍ OD FIRMY AMAZON, GOOGLE A MICROSOFT	46
6.5.1	Výpočetní služby.....	46
6.5.2	Uložiště	47
6.5.3	Databáze.....	47
6.5.4	Big Data Analytics	47
6.5.5	Shrnutí.....	48
6.5.1	Výběr vhodného řešení metodou TOPSIS	48
7	IMPLEMENTACE A TESTOVÁNÍ APACHE HADOOP.....	51
7.1	HADOOP MAPREDUCE	51
7.2	HADOOP YARN	52
7.3	IMPLEMENTACE.....	52
7.3.1	Základní požadavky	53
7.3.2	Módy nasazení	53
7.3.3	Příprava systému	53
7.3.4	Instalace.....	54
7.3.5	Modifikace konfiguračních souborů	55
7.3.6	Spouštění a základní informace	58
7.4	MĚŘENÍ A ANALÝZA PROCESŮ A VÝKONU	58
7.4.1	Konfigurace.....	58
7.4.2	TeraSort.....	58
	ZÁVĚR	62
	SEZNAM POUŽITÉ LITERATURY.....	63
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	66
	SEZNAM OBRÁZKŮ	67
	SEZNAM TABULEK.....	68
	SEZNAM PŘÍLOH.....	69

ÚVOD

„*Scientia potentia est*“, neboli česky „*vědění je moc*“, citát, jež je připisován siru Francisi Baconovi, poukazuje na jakýsi význam vědomostí pro člověka, a ať si pod mocí představíme cokoli, důležitost vědění je pro nás lidi nepopíratelná. Získání správné odpovědi v dnešní záplavě informací se ovšem stává čím dál komplikovanější. V důsledku tohoto informačního zahlcení se od druhé poloviny 20. století začal užívat termín „*informační exploze*“.

Z chronologického hlediska jsou nejčastěji uváděny 3 významné milníky. Prvním milníkem je období vzniku písma, které dnes datujeme cca do 6. tisíciletí př. n. l. Písmem lidstvo získalo schopnost informace vizuálně reprezentovat pomocí různých symbolů. Díky němu jsme schopni informace zaznamenávat, systematicky je zpracovávat a předávat dál. Za druhý významný bod se označuje masové rozšíření Gutenbergova knihtisku na přelomu let 1447 a 1448, díky kterému byly informace dostupnější, dokonce i nižším vrstvám společnosti. Poslední mezník je úzce spjatý s rozvojem informačních technologií.

Byla to právě potřeba zpracovávat velké objemy dat, která vynutila vznik prvních mechanických počítačů. Vynálezem a vývojem těchto strojů se postupně měnil i způsob záznamu informací a samotná podoba úložných médií. Hustota datového záznamu se několikanásobně zvýšila, ovšem současně s tím společnosti generují čím dál více dat. I když dnes tedy máme mnohonásobně výkonnější počítače než kdy dříve, pořád stojíme před otázkou, jak zpracovávat velké objemy dat a zejména co nejefektivnějším způsobem.

Kvůli tomuto datovému zahlcení postupně vznikají nové nástroje, nové programovací modely a nové metody, které mají pomoci k získání hodnotných poznatků. Rozdíl oproti předchozí době nyní tedy nespočívá hlavně ve sféře počítačů, ale zásadní změna spočívá hlavně v přístupu, jak s daty zacházíme a co z nich dokážeme získat. Termín „*Big data*“ zahrnuje všechny tyto aspekty a dá se tedy označit za jakousi koncepci, které se v poslední době dostává čím dál větší pozornosti.

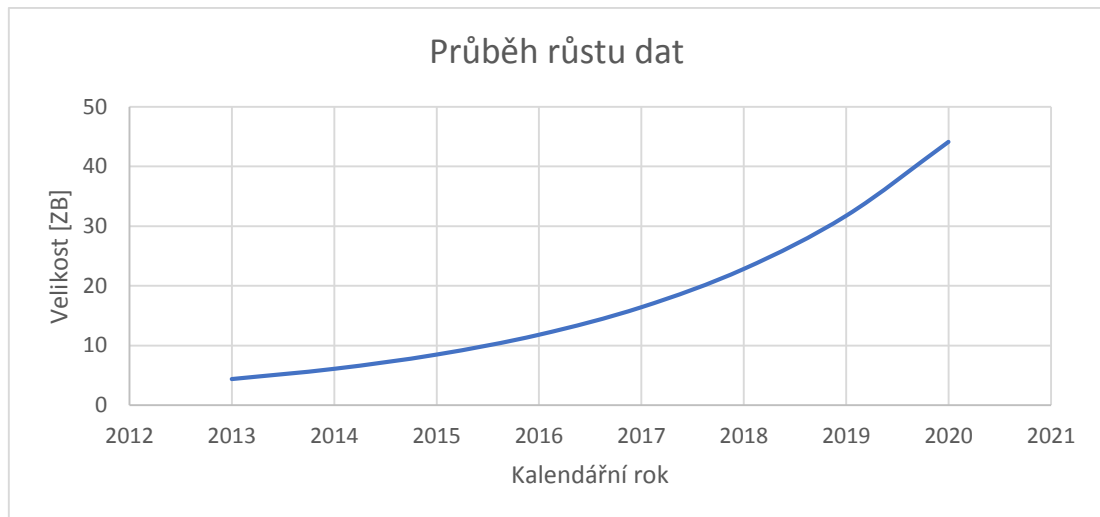
I. TEORETICKÁ ČÁST

1 BIG DATA

První, kdo se potýkaly s problémy spojené s datovou explozí, jsou vědní obory jako astronomie a genomika. [1] Za původcem užití termínu „*Big Data*“ se nejčastěji uvádí dva výzkumníci z NASA, Michael Cox a David Ellsworth, kteří v roce 1997 publikovali práci s názvem „*Application-Controlled Demand Paging for Out-of-Core Visualization*“. Během svého výzkumu byli natolik zahlceni daty, že tuto překážku označili jako problém Big Dat. Je to tedy nejspíše první užití tohoto termínu významově nejbližší, jak ho chápeme dnes. [2]

Organizace IDC sleduje velikost digitálního „vesmíru“ už od roku 2005, ve své v pořadí sedmé studii s názvem „*The Digital Universe of Opportunities*“ z roku 2014 plyne, že každým rokem roste počet dat o 40 % (viz. graf 1). To znamená, že každé dva roky se počet dat téměř zdvojnásobí. Jsou to data, která jednak lidé svou činností vědomě tvoří, nebo která jsou samočinně vytvářena lidskou interakcí, např. statistiky při procházení webových stránek apod. Na tomto růstu se již ale nepodílejí pouze lidé nebo společnosti, ale stále častěji jsou to i různá chytrá vestavěná zařízení se síťovým připojením a schopností sběru a výměny dat. Dále kvůli datům se vytváří data další, tzv. data o datech neboli metadata. Ta zastupují nejrychleji rostoucí množinu z celkového počtu dat. Roku 2020 má tento digitální vesmír celkově čítat 44 ZB (1 ZB = 10^{21} bajtů). Přičemž odhadem 90 % budou tvořit nestrukturovaná data, což jsou například e-maily, audio a video záznamy, obrázky, příspěvky na sociálních sítích apod. Dále spousta vytvořených dat se dá označit za tzv. šum, nebo též jako data neužitečná. V roce 2013 pouze 22 % dat z celkového množství se dala označit za užitečná a vhodná k analyzování, přičemž ve skutečnosti méně jak 5 % byla analyzována. Do roku 2020 se odhaduje, že by množství těchto užitečných dat mohlo vzrůst alespoň na 35 % a to především díky rozmachu vestavěných systémů a dále kvůli rozmachu IoT. [3]

Nárůst dat je jev, se kterým je lidstvo obeznámeno již delší dobu, nicméně dat již je tolik, že se kvantitativní problém změnil v problém kvalitativní, což v podstatě znamená, že je čím dál těžší nalézt správnou odpověď, navíc kvůli exponenciálně se zrychlujícímu počtu dat, data již nejsou tak statická, ale jsou neustále proudící, dynamická. [1] Dalším jevem, který je známý již delší dobu je nárůst výpočetního výkonu a kapacity úložných médií. Z tohoto pohledu nastolený trend jen umožňuje příchod éry Big Dat.



Graf 1 Předpovídané množství dat od roku 2013 do 2020 (zdroj: IDC)

Moorův zákon je empirické pravidlo, podle kterého se má každých 18 měsíců zdvojnásobit počet tranzistorů integrovaného obvodu. Nepřímo tím i udává, že výpočetní výkon se zdvojnásobí během 18 měsíců. Ačkoliv se tento vývoj v poslední době trochu zpomalil a výpočetní výkon se zdvojnásobí spíše každé 2 roky. Pořád je tohle pravidlo dosti přesné. Ovšem pomocí některých nových technologií bylo upozorováno, že najednou přišla možnost určité procesy provést 100krát až 1000krát rychleji. Dalo by se tedy říct, že taková změna se udála navzdory všem očekáváním. Jednou z hlavních příčin tohoto zrychlení byla změna výpočetního modelu, paralelizace úlohy s principem metody rozděl a panuj. Software svou pracovní zátěž rozšířil přes stovky až tisíce procesorů a díky tomu dosáhl tohoto efektu. [4]

Další významná změna se odehrála na poli statistiky. Ve statistice platí známé pravidlo „*korelace neimplikuje kauzalitu*“. V Big Data přístupu zásadní změna oproti klasickým metodám spočívá v tom, že společnost musí do jisté míry omezit svůj důraz na kauzalitu a spokojit se s jednoduchými korelacemi, kdy nevíme proč, ale pouze co. To představuje přelom oproti stovkám let zavedené praxe. [1] Existenci Big Dat zintenzivňuje právě hledání zajímavých korelací. Kauzalita vyžaduje modely a teorie, a dokonce i ty mají určité limity. Navíc pokud máme problém tak enormně komplexní, jakými se v obrovských datových množinách zabýváme, efektivita klasické kauzální statistiky je pochybná. Potvrdit kauzální vazbu v problému takového měřítka je příliš složité a je mnoho oblastí, v kterých jsou samotné korelace dostačující a neobyčejně užitečné k děláním zajímavých poznatků. [5]

Big Data jsou velmi mladým oborem, který v poslední době zažívá prudký nárůst. Pominou-li se výhody, které tato koncepce může pro společnost přinést. Za hlavní příčiny podněcující rozvoj této oblasti můžeme z výše uvedeného textu označit:

- Zlepšení výpočetních a statistických metod
- Množství stále rychleji přibývajících dat
- Pokrok v oblasti výpočetní techniky a paměťových médií

1.1 Definice

Z důvodu krátké existence tohoto oboru neexistuje nějaká přesná a všeobecně uznávaná definice. Většina definic popisuje celou problematiku pouze relativně nebo na ni nahlíží z různých úhlů pohledu. Big Data nejsou pouze o datech, ale jedná se o jakousi koncepci, která společně s daty zahrnuje více aspektů.

Společnosti publikují své vlastní definice, což může dávat poněkud matoucí vhled do celé problematiky. Nicméně napříč všemi těmito definicemi lze nalézt jakési společné rysy, kterými lze toto téma vymezit v obecné rovině.

Jedna z nejrozšířenějších definic pochází z roku 2012 od firmy Gartner, zabývající se výzkumem a poradenstvím v oblasti IS/ICT, která tvrdí, že:

„Big data jsou informačními aktivy o velkém objemu, velké proměnlivosti a/nebo velké různorodosti, která vyžadují nákladově efektivní inovativní formy zpracování k usnadňování rozhodovacího procesu, objevnému vhledu (do problematiky) a procesní optimalizaci.“ [6]

Tato definice poukazuje na všeobecně uznávané tři charakteristické vlastnosti Big Dat, a to na objem, rychlost a různorodost, většinou zkráceně označované podle počátečních písmen z anglického překladu těchto slov jako 3V (pozn.: angl. volume, velocity a variety). Tyto vlastnosti vytyčila v roce 2001 společnost META Group (pozn.: nyní Gartner).

V Oxfordském slovníku angličtiny jsou pak Big Data definována jako:

„Extrémně velké datové množiny, které můžou být výpočetně analyzovány k odhalení vzorů, trendů a asociací, zejména tykajících se lidského chování a interakcí.“ [7]

Ačkoliv Big Data analýzy se netýkají pouze lidského chování, spousta z nich se týká analýzou trendů ve společnosti a předmětem zájmu je právě hledání zajímavých vzorů, trendů a asociací.

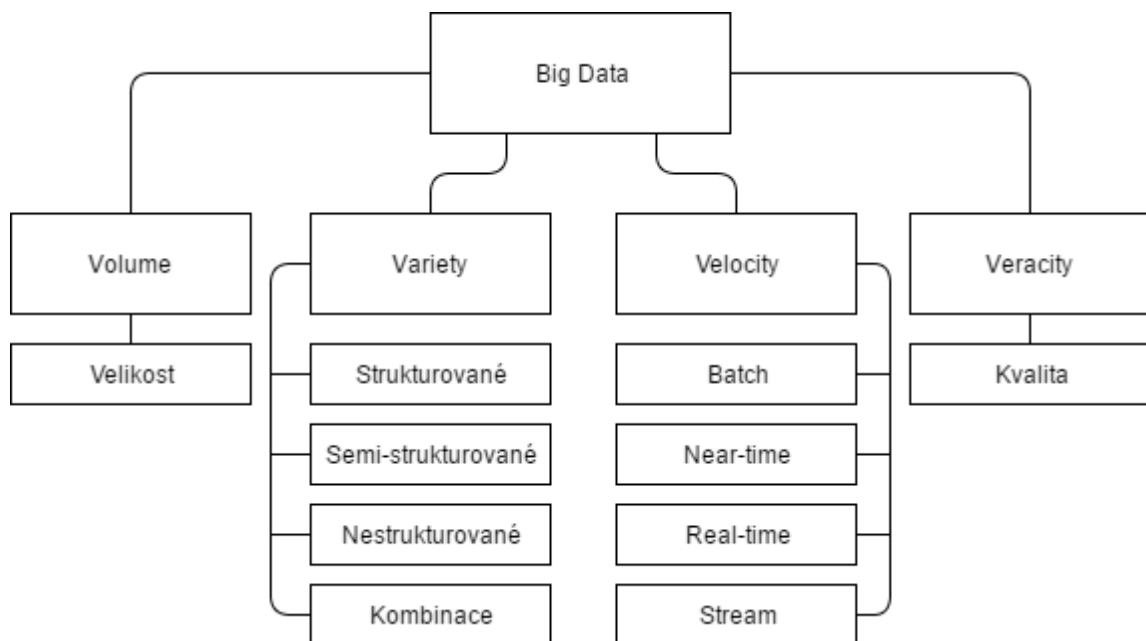
Další možnost, jak definovat Big Data přináší Viktor Mayer-Schönberger a Kenneth Cukier, autoři bestselleru Big data – Revoluce, která změní způsob, jak žijeme, pracujeme a myslíme s tvrzením, že Big Data jsou:

„Schopnost společnosti využívat informace novými způsoby k produkovaní užitečných poznatků nebo zboží a služeb významné hodnoty“ a dále: „Big data znamenají, že ve velkém měřítku lze provádět některé operace, které se v malém měřítku dělat nedají. Je například možné extrahovat nové poznatky nebo generovat novou hodnotu takovými způsoby, které mění trhy, organizace a vztahy mezi občany a úřady atd.“ [1]

Tato definice se do jisté míry odpošřuje od technických problémů, nicméně poukazuje na nové způsoby, díky kterým je možné získávat hodnotu tam, kde to dříve nebylo možné.

1.2 Charakteristika

Big Data se vyznačují jakýmsi charakteristickými vlastnostmi, které se označují 3V. Stejně tak jak tomu bylo u definice i zde neexistuje úplná jednotnost. Nicméně zde existuje pouze více modelů, které se snaží popsat Big Data ve více rozměrech. Model 3V je tedy všeobecně uznávaný a jen se přidávají další „V“. Takovým případem je označení 4V (viz. obrázek 1), které prosazuje společnost IBM, ta za čtvrté V označuje pravdomluvnost (angl. Veracity).



Obrázek 1 Charakteristika Big dat (pozn.: Pro zachování integrity jednotného názvosloví jsou uvedeny anglické názvy pro označení 4V)

1.2.1 Objem

Definovat přesnou hranici, kdy se z normálního objemu stává objem charakteristický pro Big Data je složité. Tato hranice je totiž pro každého subjektivní a postupem času se kvůli růstu kapacity paměti stále posouvá. Hovoříme-li o velikosti, můžeme mít na mysli

datové kolekce takových objemů, že je není možné uložit na jeden databázový server, ale potřebujeme jich několik desítek či stovek. [8]

1.2.2 Různorodost

Různorodostí dat je myšlena rozličnost v datové struktuře. Data podle struktury dělíme na [8]:

- **Strukturovaná** – Jednoduchá forma dat efektivně spravována relačními databázovými systémy, v kterých jsou snadně analyzována. Strukturovaná data obsahují písmena, čísla, a to ať už celá nebo i s plovoucí řadovou čárkou a data běžně používaná v systémech řízení vztahů se zákazníky.
- **Semi-strukturovaná** – Typ strukturovaných dat, který nemůže být zpracováván tradičními RDBMS nástroji. Jedná se např. o textové dokumenty, XML, JSON a soubory, které nejsou organizovány v tabulce podle relačního modelu.
- **Nestrukturovaná** – V současnosti se odhaduje, že je 90 % celkového počtu dat nestrukturovaných. [3] Tyto data jsou obtížně analyzovatelná a skladovatelná, nedají se zpracovávat tradičními relačními databázovými systémy, jedná se např. o obrázky, audio, video, emaily, senzorová data, data ze sociálních sítí, textové soubory apod.
- **Kombinovaná** – Výše uvedené typy je možné kombinovat, čímž se zvýší požadavky na celý systém, který musí být schopný s takovými typy dat pracovat.

1.2.3 Rychlost

Rychlostí je myšleno, jednak jakou rychlostí se data tvoří a prochází celým systémem, ale i jak rychle jsou analyzována.

Rychlost dělíme na [9]:

- **Batch** – Přijímání dat a zpracovávání je prováděno pouze v určitých intervalech.
- **Near-time** – Čas mezi tím, kdy data přijdou na vstup a jsou zpracována, je velmi malý, blízký real-time.
- **Real-time** – Data přicházejí a jsou zpracovávána nepřetržitě, což umožňuje analýzu v reálném čase.
- **Streaming** – Podobné real-time, data přichází a jsou zpracována ihned po příchodu.

1.2.4 Věrohodnost

Podle IBM každý třetí vedoucí manažer ne vždy důvěřuje informacím, z kterých obvykle vyvozuje rozhodnutí. Data ze zdrojů, jakými jsou např. sociální sítě, nám můžou poskytnout jen omezené množství odpovědí, navíc mnohdy pochybných, např. při sémantické analýze textu lze jen velmi těžko rozeznat ironii či sarkasmus. Dále může jít o systematický problém se senzory nebo chyby způsobující ztrátu dat apod. Věrohodnost tedy označuje datovou spolehlivost, přesnost, srozumitelnost a důvěryhodnost. [2]

1.2.1 Další uváděné vlastnosti

Jak bylo výše zmíněno, existuje více pohledů, které se snaží charakterizovat Big Data hlouběji. Nejčastěji se jedná např. o proměnlivost (angl. Variability), pojednávající o množství proměnných v datové sadě. A jako další vlastností se často označuje hodnota (angl. Value), popisující obchodní hodnotu. [2]

1.3 Rozdělení zdrojů dat

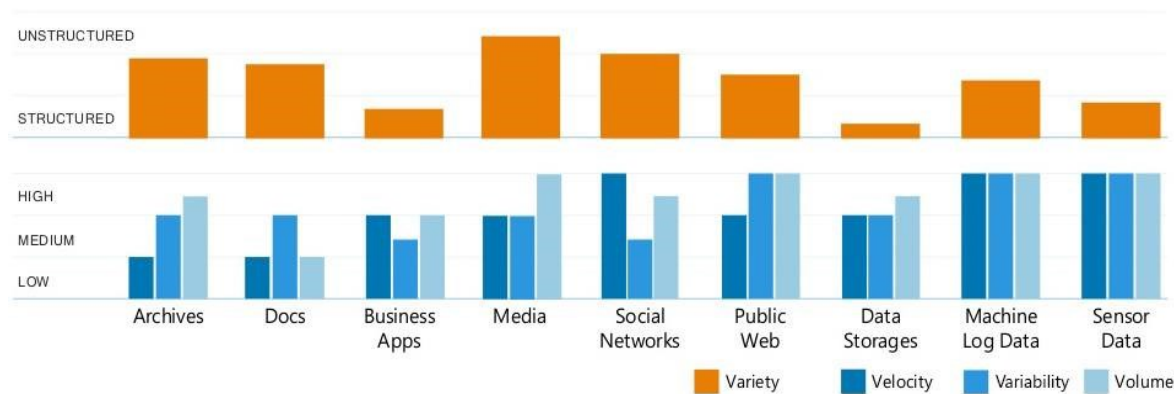
Každý zdroj dat je specifický svými vlastnostmi, tzn. každý představuje jiný objem, rychlost, různorodost nebo pravdomlupnost. To je potřeba zohledňovat při zahrnutí takového zdroje do datové kolekce. Kombinováním různých dat z různých zdrojů pro nás může být ovšem výhodné. Mnohdy totiž mluvit o potenciální hodnotě dat je dosti komplikované, protože určitá data se nám v jednom okamžiku nemusí zdát nijak přínosná, teprve až kombinací různých datových sad, třeba i z různých zdrojů nám může přinést hodnotu. [5]

Výčet zdrojů dat:

- **Archivy** – skenované dokumenty, prohlášení, pojištění, lékařské záznamy, zákaznická korespondence apod.
- **Dokumenty** – různé typy formátů jako XLS, PDF, CSV, dokumenty Microsoft Wordu, PPT, HTML, XML, JSON apod.
- **Podnikové aplikace** – Projektový management, marketingová automatizace, produktivita, CRM, ERP, HR procesy, intranet apod.
- **Média** – obrázky, videa, audia, flash, podcasty apod.
- **Sociální sítě** – Facebook, Twitter, LinkedIn, YouTube, Instagram apod.

- **Veřejný web** – vláda (úřady apod.), záznamy počasí, doprava, zdravotnické služby, veřejné finance, Světová banka apod.
- **Datové úložiště** – SQL, NoSQL, repositáře, souborové systémy apod.
- **Strojové záznamy** – logy událostí, aplikací, podnikových procesů, serverová data apod.
- **Senzorová data** – zdravotnická zařízení, chytré měřicí zařízení, automobilové senzory, satelity, kamery, zařízení zaznamenávající dopravu apod.

Výčet těchto zdrojů je s poměrným zastoupením charakteristických vlastností uveden v obrázku 2.



Obrázek 2 Poměrné rozložení charakteristických vlastností podle zdroje dat (zdroj: Soft-Serve Inc.)

1.4 Oblasti využití

Big Data přístup je v současnosti rozšířený především ve velkých společnostech a v oblastech, které se nějakým způsobem musí potýkat s obrovským množstvím dat. Nicméně tento koncept postupně nachází uplatnění ve všech oblastech lidské činnosti. Mnohdy se debaty vedou v duchu, že se lidstvo nachází na prahu jakési revoluce, kdy prostřednictvím tohoto přístupu dojde k mnoha změnám současného fungování světa. Popisem a identifikací oblastí, ve kterých se Big Data prakticky využívají, je ovšem možné poukázat na současný stav, v jakém tento koncept zaujímá místo pro naši společnost.

1.4.1 Průmysl

V této oblasti se v poslední době nejčastěji mluví o tzv. „Industry 4.0“ což je označení pro čtvrtou průmyslovou revoluci, která má spočívat hlavně v rozvoji tzv. „chytrých továren“.

Zde se má propojit v součinnost několik technologií a nedílnou součástí zde zaujímají právě poznatky z oblasti Big Dat. Jedná se zde především o jakýsi další stupeň automatizace, která prostupuje do současného trendu digitalizace, což znamená, že továrny budou generovat více dat, či je budou sbírat z okolí a na základě nich budou jednotlivá výrobní zařízení samočinně vyvozovat rozhodnutí. [10]

1.4.2 Věda a výzkum

Evropská organizace pro jaderný výzkum (CERN) je nejrozsáhlejším výzkumným centrem v oblasti částicové fyziky na světě a je dobrým příkladem praktického využití Big Data technologií ve vědě. Datový tok ze všech čtyř experimentů (Alice, Atlas, CMS, LHCb) pro tzv. „Run 2“, který je naplánován od roku 2015 do 2018, je odhadovaný na 25 GB/s, přičemž se ukládá pouze 0,01 %. V současné době na těchto experimentech pracuje kolem 8 000 analytiků. Všichni mohou vzdáleně přistupovat a analyzovat některé z údajů získané v téměř reálném čase. Pro analytické činnosti používají např. Oracle DB, Hadoop, R. [11]

1.4.3 Zdravotní péče

Ve zdravotnictví se předvídá velký nástup datových analýz. Zdravotní instituce totiž obsahují množství záznamů, které mezi sebou nejsou moc dobře nebo vůbec sdíleny. V těchto záznamech je možné nalézt spoustu skrytých vzorů. Zároveň nositelné chytré zařízení měřící základní životní funkce, mohou predikovat život ohrožující události jako zástavu srdce apod. Obecně vzato Big Data mohou zlepšit jak samotnou prevenci, tak včasnou diagnózu, léčení nemocí, ale např. i predikování výskytu různých epidemií, např. služba Google Flu Trends, a lepší krizová řešení v případě rozšíření epidemií, např. na základě měření pohybu obyvatelstva. [1]

1.4.4 Vývoj technických zařízení

Příkladem zde může být vývoj závodního auta formule F1. Tyto auta se skládají ze zhruba 25 000 součástí, 11 000 tvoří kostra auta, z 6 000 je tvořen motor a 8 500 dělá elektronika. Každá z těchto součástí představuje určité riziko poškození. Proto se auta různě testují a jsou vybaveny senzory, které měří všemožné aspekty jízdních vlastností a automobilu samotného. Není tedy divu, že během Velké ceny USA v roce 2014 závodní týmy nashromáždily 243 TB dat. Na těchto autech je zajímavé především to, že v době vývoje během jednoho roku se každé 2 týdny mění zhruba 5 000 komponent. Ty jsou na míru vyráběné na základě datových analýz. [12]

1.4.5 Sociální sítě

Díky obrovské popularitě jsou sociální sítě terčem mnoha analytických činností, z příspěvků na sociálních sítích organizace např. lépe vedou reklamy, cíleněji nabízí své produkty, lépe poskytují služby apod. Dále významným faktem je, že sociální sítě změnily běžně zavedený standard, mnoho informací, které dříve bylo možné získat jen prostřednictvím dotazníků nebo pozorování, je nyní možné masivně čerpat právě díky nim. Díky sociálním sítím se můžou provádět i např. různé psychometrické analýzy, lidé zde velmi často zveřejňují spoustu informací i osobního rázu.

1.4.6 Maloobchod

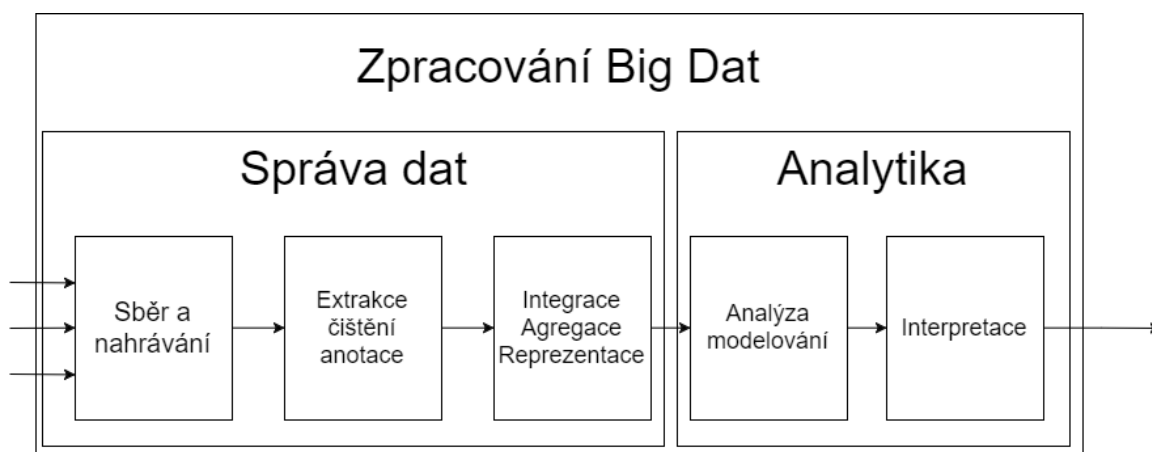
Společnosti v tomto sektoru pracují s různými druhy dat. Můžou to být údaje od zákazníků prostřednictvím věrnostních karet, ekonomické a demografické údaje, sociální sítě a web obecně. Díky těmto informacím pak mohou např. vytvářet obchodní strategie, předpovídat poptávku, optimalizovat ceny a identifikovat zákazníky. [1]

1.4.7 Finance a bankovníctví

Tento sektor je pod velkým dohledem regulačních orgánů, dále zákazníci vyžadují čím dál více individualizované služby, proto se zde nejčastěji používají na základě datových analýz techniky jako segmentace zákazníku, řízení rizik, monitoring obchodů, detekce proti podvodům apod. [13]

2 ZPRACOVÁNÍ BIG DAT

Konečná podoba zpracování dat probíhá několika fázemi. V praxi se skutečný model odvíjí od použitých technologií a od vlastností samotných dat, obecně ale tento proces můžeme rozdělit na podprocesy zahrnující správu dat a analytické podprocesy (viz. obrázek 3). Správa dat zahrnuje techniky a podporující technologie, k osvojení a uložení dat a jejich přípravě k analýze. Analytická část pak poukazuje na metody k získání poznatků z těchto dat.



Obrázek 3 Obecný průběh zpracování dat

2.1 Sběr a nahrávání dat

Sbíraná data mohou být z různých zdrojů. V této fázi se data nijak nezpracovávají, před samotným zpracováním je důležité určení, která data se budou sbírat a jak se budou nahrávat.

V této fázi se tedy řeší [15]:

- **Výběr relevantních dat** – Omezení výběru pouze na relevantní data se zredukuje celková velikost datové sady, což pozitivně přispívá rychlosti celého následujícího průběhu. Relevantní data mohou být vybírána dvěma způsoby, jedním z nich je vyhledávání informací. Někdy ovšem není předem jasné, které data jsou nejvhodnější, a proto se využívá různých filtrů, která se musí vhodně nastavit, aby nedocházelo ke ztrátě užitečných dat.
- **Automatické vytváření metadat** – Co nejpřesnější popsání, co je nahráno, jak je to nahráno, jak je to změřeno apod.

2.2 Extrakce informací, čištění, anotace

Nasbírané data většinou nejsou ve formátu vhodném pro analýzu. Aby bylo možné data co nejefektivněji analyzovat, je potřeba je zpracovat do vhodné podoby, tzn. odstranit šum a nesrovnalosti. Špatná datová kvalita, různá datová komplexnost způsobena nestrukturovanými daty a rozličnými formáty dělá tuto fázi časově nejnáročnější. Nicméně protože je zde často potřeba pracovat právě s takovými složitými daty, často se používá proces extrakce určité informace a transformace do strukturované podoby vhodné pro analýzu (technika zvaná ETL – „*Extract Transform Load*“). [15]

2.3 Datová integrace, agregace a reprezentace

Pro efektivní analýzu velkého měřítka dat. kompletně provedenou automatickým způsobem, musí být rozdílnosti v datové struktuře zaznamenány ve formě, která je počítačově srozumitelná a automaticky řešitelná. Dokonce i pro jednodušší datovou analýzu, která závisí pouze na jedné datové sadě, je důležitou otázkou vhodný návrh databáze. Obvykle bývá mnoho alternativ, jakým způsobem uložit stejná data. Jeden návrh přináší určité výhody nad ostatními pro nějaké účely, ale současně sebou nese i nějaké nedostatky pro jiné účely. [15]

2.4 Zpracovávání dotazů, vytváření modelů a analýza

Metody pro zpracovávání dotazů a data mining jsou fundamentálně odlišné od tradičních statistických analytických metod používaných na malém měřítku. Big Data jsou často plné tzv. šumu, dynamické, heterogenní, vnitřně provázané a nedůvěryhodné. Nicméně i tato chaotická Big Data mohou být hodnotnější než malé vzorky, protože i základní statistiky získané z frekvence vzorů a korelační analýzou, obvykle převažují nad individuálními výkyvy a často odhalí víc důvěryhodných skrytých vzorů a vědomostí. [15]

V této fázi se tedy řeší analytická úloha, která obvykle zahrnuje jednu nebo i více způsobů analytických metod. Tato fáze může být iterativní, obzvláště pokud naše analýza je výzkumného charakteru, v tomto případě analýza neustále probíhá, dokud např. nejsou odhaleny vhodné vzory nebo korelace. Dále může být poměrně jednoduchá anebo naopak velmi složitá v závislosti s daným účelem analýzy. Můžeme například kombinovat data mining s komplexními techniky statistické analýzy k odhalení vzorů a anomálií nebo k vytvoření statistického nebo matematického modelu k vykreslení vztahů mezi proměnnými. [15]

2.5 Interpretace

Schopnost analyzovat Big Data naráží na limity ve správném pochopení výsledků analýzy. Osoba dělající rozhodnutí na základě těchto výsledků, je musí správně interpretovat. To znamená, že musí být obeznámena s analyzovanými daty a se způsobem jakým byla získána a konec konců to obvykle zahrnuje i rozbor všech vytvořených předpokladů a rozbor všech kroků analýzy. Protože je zde mnoho možných zdrojů chyb, výpočetní systémy mohou obsahovat chyby, modely mají téměř vždy nějaké odhady a výsledky mohou být založeny na chybových datech. [15]

Počítačové systémy musí podávat výsledky v lidsky co nejsrozumitelnější podobě, což zejména kvůli komplexnosti Big Dat není vůbec jednoduché. Poskytnutí pouhých výsledků obvykle není dost. Je potřeba k těmto výsledkům podat informace o původu dat, vysvětlující, jak byl každý výsledek odvozen a na základě jakých vstupů. Vytvářením adekvátních metadata tedy je získána možnost nejen výsledky analýzy lépe interpretovat, ale i možnost opakování analýzy s jinými předpoklady, parametry nebo datovými množinami. [15]

Systémy s bohatou škálou vizualizačních možností jsou důležité v poskytování výsledků v co nejsrozumitelnější možné podobě. V této vizuální reprezentaci jsou informace předkládány abstraktně nějakou schématickou formou. Navíc zde bývá možnost snadného dohledání jednotlivých částí dat a obeznámení se tak s jejich původem. Vizualizace je v tomto směru velmi nápomocná, protože zde bývá existence možnosti si analýzu přehrát v krocích. [15]

3 MECHANISMY PRO SPRÁVU DAT

Tradiční řešení pro práci s daty v Big Data měřítku nejsou efektivní, v důsledku tohoto problému vznikají nové mechanismy, které poskytují rychlejší a celkově efektivnější práci.

Existující mechanismy lze klasifikovat do tří skupin [16]:

1. **Souborové systémy**
2. **Databázové systémy**
3. **Programovací paradigmaty**

3.1 Souborové systémy

V oblasti Big Data mluvíme především o distribuovaných souborových systémech (zkráceně „DFS“) patří mezi síťové souborové systémy, kde jsou data distribuovaná mezi více uzly, využívá se zde tedy model **M:N**. Což znamená, že do tohoto souborového systému může současně přistupovat více uživatelů, přičemž data jsou rozprostřena mezi několika uzly. Výhodou DFS je, že kapacita může být poměrně jednoduše škálovatelná přidáním uzlů.

Existující architektury DFS systémů lze popsat jako [17]:

- **Client-Server architektura** – Umožňuje sdílení souborů mezi několika stroji v síti, pomocí standardizovaného pohledu na lokální souborový systém. Typickým příkladem NFS, GlusterFS.
- **Cluster-Based architektura** – Skládá se z jednoho master serveru starající se o metadata a z několika tzv. „chunk“ serverů, napříč kterými se ukládají data. Typickým příkladem GFS, HDFS.
- **Symetrická architektura** – Správa metadat je plně decentralizovaná, klienti sami se podílejí na hostování správy metadat, což vede k rozpoznání celkové struktury systému. Tato architektura je základem peer-to-peer technologií.
- **Asymetrická architektura** – Existence jednoho nebo více dedikovaných správců metadat, kteří se starají o souborový systém a jeho přidružené diskové struktury. Typickým příkladem je LustreFS a tradiční NFS systémy.
- **Paralelní architektura** – Datové bloky jsou umístěny paralelně, napříč několika úložnými zařízeními na několika serverech. Typickým příkladem GFS, HDFS, GlusterFS, LustreFS apod.

3.1.1 GFS

Google File System (GFS) je škálovatelný distribuovaný souborový systém pro rozsáhlé distribuované aplikace náročné na přenos dat. Poskytuje odolnost vůči chybám, je určen na provoz pro levný komoditní hardware a poskytuje velký agregovaný výkon k velkému počtu klientů. Mnoho distribuovaných souborových systémů vychází právě z GFS. Byl vyvinutý firmou Google jako pro vlastní účely, např. jako pro svůj vyhledávač, který generuje obrovské množství dat.

Základní vlastnosti [18]:

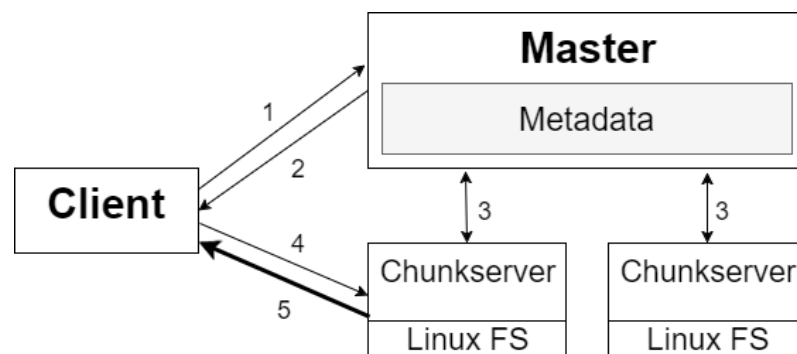
- **Odolnost proti chybám** – Systém je postavený z mnoha levných komoditních komponentů, které často selžou. Proto musí na rutinní bázi neustále monitorovat vlastní chod a detekovat, tolerovat a obnovovat se z těchto selhání. Proto je zde selhání komponent bráno spíše jako pravidlo než výjimka.
- **Optimalizován pro větší soubory** – Systémy Googlu pracují se soubory typicky velkými kolem 100 MB nebo i většími. Na jejich efektivní správu jsou jiné nároky než na menší soubory, ty jsou sice podporovány, ale systém pro ně není optimalizován, proto velký počet těchto malých souborů snižuje výkon celého systému.
- **Zápis/čtení** – Systém je uzpůsoben především pro čtecí operace. Jakmile je soubor jednou zapsán, je jen zřídkakdy přepisován. Mále zápisy v libovolných polohách jsou podporovány, ale nemusí být účinné. Změny souborů jsou tedy častěji prováděny přidáním nových souborů spíše než přepsáním existujících dat, proto přepisování souborů prakticky neexistuje.
- **Souběžný přístup** – Souběžný přístup několika klientů do jednoho souboru bez dodatečných synchronizací mezi nimi je dosažen díky atomocitě přístupových operací.
- **Důraz na propustnost na úkor odezvy** – Většina cílových aplikací Googlu klade důraz především na zpracování dat vysokou rychlostí, zatímco jen pár má přísnější požadavky na čas odpovědi pro jednotlivé operace čtení nebo zápisu.

Systémový klastr tvoří jeden master a několik tzv. „chunkserverů“ dle obrázku 4.

Chunkservery obsahují veškerá vlastní data. Soubory jsou rozděleny do tzv. „chunků“ pevné délky 64 MB. Každý chunk je identifikován nezaměnitelným a globálně unikátním 64bitovým číslem, které mu je přiřazeno masterem v okamžiku vytvoření. Chunkservery ukládají tyto chunky na lokální disky jako běžné linuxové soubory, čtou a zapisují data na základě požadavků specifikovaných číslem chunku a rozsahem, data v chunku nejsou přepisována,

ale připojována na konec. Pro spolehlivost je každý chunk replikován alespoň na třech chunkserverech. [18]

Master spravuje veškerá metadata (jmenný prostor, údaje o přístupových právech, mapování souborů na chunky a aktuální umístění chunků). Dále zajišťuje aktivity systému jako mazání osiřelých chunků, migrace chunků mezi chunkservery, pravidelně komunikuje s každým chunkserverem a dává mu instrukce a kontroluje jeho stav. Aby se tento master nestal úzkým místem v komunikaci, nejsou žádné operace s daty prováděny skrz něj, ale přímo mezi chunkserverem a klientem. Spolehlivost systému je posílena záložním master serverem. [18]



Obrázek 4 GFS Architektura (zdroj: <https://research.google.com>)

Při požadavku na čtení celý proces probíhá dle obrázku 4 [18]:

1. Nejprve klientská aplikace vznesne požadavek na čtení souboru. Požadavek na soubor obsahuje jméno souboru a jeho index a je klientem odeslán masteru.
2. Master nalezne v mapování souborů číslo chunku a všechny lokace jeho replik.
3. Tyto údaje odešle zpátky klientovi, který je uchovává v cache paměti.
4. Klient odešle jedné z replik (nejčastěji té nejbližší), požadavek na čtení dat, specifikováním čísla chunku a rozsahem. Následující požadavky na čtení ze stejného chunku již nebudou potřebovat klient-master interakci, dokud klient v cache paměti má potřebné informace.
5. Chunkserver nalezne na svých lokální discích požadovaná data a odešle je klientovi.

3.1.2 HDFS

Hadoop Distributed File System (HDFS) je distribuovaný souborový systém, inspirovaný právě GFS, proto je zde značná podobnost, nicméně GFS byl navržen hlavně pro vlastní potřebu společnosti Google, zatímco HDFS je open-source a může jej využívat kdokoliv. Je taktéž navržený především pro provoz na levném komoditním hardwaru a je vysoce odolný

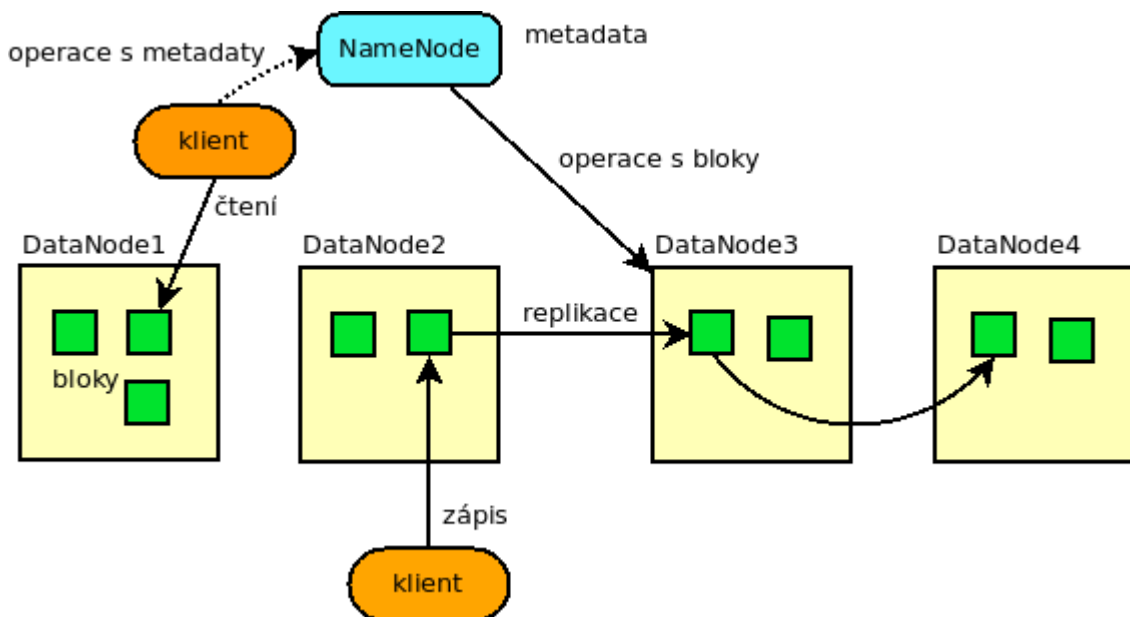
proti chybám, poskytuje vysokou propustnost v přístupu k aplikačním datům a je vhodný pro aplikace, které obsahují velké datové sady v měřítku od gigabytů po terabyty.

Základní vlastnosti: [19]

- **Odolnost proti chybám** – Chybovost hardwaru je stejně tak jako u GFS brána jako norma, ne jako výjimka. Instance HDFS může obsahovat stovky nebo tisíce serverů. Skutečnost, že existuje obrovské množství komponent s nezanedbatelnou pravděpodobností selhání, znamená, že některá část HDFS je stále nefunkční. Z toho důvodu je detekce poruch a rychlá automatická obnova architektonickým cílem HDFS.
- **Optimalizován pro větší soubory** – Typická velikost souboru, s kterými pracují aplikace běžící na HDFS je od gigabytů až po terabyty. Proto je HDFS optimalizován pro práci s velkými soubory a měl by poskytovat vysokou agregovanou propustnost dat a rozšiřitelnost na stovky uzlů v jediném klastru, což má podpořit práci s desítkami milionů souborů v jedné instanci.
- **Zápis/čtení** – HDFS aplikace vyžadují write-once/read-many model, tedy soubor, který je jednou vytvořen, vyplněn a uložen, už není dále modifikován. To velmi zjednodušuje problémy koherence dat a umožňuje zvýšit propustnost
- **Důraz na propustnost na úkor odezvy** – HDFS je navržen spíše pro dávkové zpracování, namísto interaktivního uživatelského přístupu. Standard POSIX vyžaduje náročné požadavky, které nejsou potřebné pro aplikace zaměřené na HDFS, proto byla POSIX sémantika v některých klíčových oblastech změněna, s cílem zvýšit propustnost v přístupu k datům. Aplikace jako MapReduce dokonale zapadají do takového modelu.
- **„Moving Computation is Cheaper than Moving Data“** – Základní předpoklad je, že je mnohem efektivnější provádět výpočty blíže místu, kde jsou data uložena, nežli je posílat do místa, kde je běžící aplikace a tím tak zbytečně zahlcovat celou síť.

HDFS je master/slave architektury (viz. obrázek 5). Stejně tak jako u GFS i zde v jednom klastru je jediný master server zvaný tentokrát „NameNode“, který spravuje veškerá metadata, tedy seznam souborů a adresářů, jejich mapování na bloky a umístění daných bloků. NameNode provádí souborové operace jmenného prostoru jako je otevírání, zavírání a přejmenovávání souborů a adresářů, jejich mapování na bloky a umístění daných bloků. DataNode je slave, který neví vůbec nic o souborech, slouží jen jako úložiště bloků, které jsou standardně o velikosti 64 MB, je tedy zodpovědný za poskytování čtecích a zapisovacích

požadavků. DataNode dále vytváří, maže nebo replikuje bloky na základě instrukcí od NameNodu. [19]



Obrázek 5 HDFS architektura (zdroj: <https://hadoop.apache.org>)

3.1.3 GlusterFS

GlusterFS je distribuovaný souborový systém škálovatelný do několika petabytů se schopností spravovat tisíce klientů. Oproti ostatním DFS má unikátní architekturu. Je typu klient-server, přičemž zde není žádný metadata server, proto je označován jako tzv. „FUSE“ (File System in User Space). Nepřítomností metadata serveru je dosaženo lepšího výkonu, lineární škálovatelnosti a spolehlivosti. [20]

Data a metadata se ukládají na několika zařízeních fyzicky připojených k různým serverům. Tyto servery mohou lokálně používat jakýkoliv souborový systém, který podporuje rozšířené atributy (např. ext4, XFS, apod.) Sada zařízení, na které se ukládají data a metadata se nazývá „volume“. Na každém tomto serveru běží tzv. „glusterfsd“ démon, který zpřístupňuje lokální souborový systém klientům jako oddíl. Klientský proces „glusterfs“ pak na straně klientů tyto zpřístupněné oddíly z více serverů skládá do virtuálního oddílu za užití tzv. překladačů („translatorů“). [21]

GlusterFS nepoužívá dedikovaný a centralizovaný server ke správě metadat, místo toho lokalizuje soubory algoritmicke pomocí tzv. „Elastic Hashing Algorithm“. EHA využívá hashovací funkce k převedení názvu cesty k souboru do pevné, jednotné a unikátní hodnoty,

díky čemuž se napříč celým systémem využívá jednotný globální jmenný prostor. Podle této přiřazené hodnoty souboru se určuje zařízení, na kterém je soubor uložen. [21]

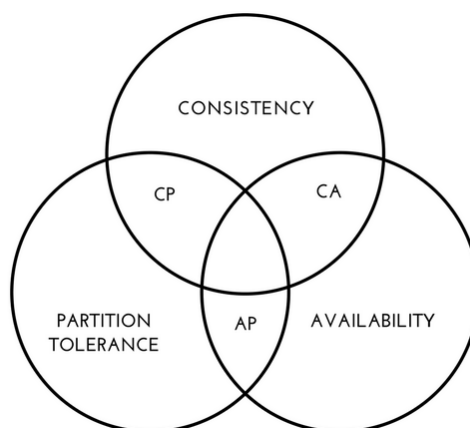
3.2 Databázové systémy

Po dlouhou dobu byla naprostá většina funkcionalit datové správy řešena výhradně prostřednictvím relačních databázových systémů. Nicméně v poslední dekádě vznikly nové aplikace a byly vzneseny nové požadavky, jako zejména na schopnost pracovat s nestrukturovanými daty, které relační databáze nejsou schopny splnit. Fundamentální vlastnosti relačních databází tedy byly změněny, čímž vznikla jakási odnož označovaná NoSQL. Z tohoto pohledu tedy máme relační SQL databáze a nerelační NoSQL.

3.2.1 CAP teorém

Začátkem 2. tisíciletí Eric Brewer formuloval tzv. Brewerův teorém nebo taky častěji nazýván jako CAP teorém. Tato formulace uvádí, že distribuovaný počítačový systém může současně poskytovat 2 ze 3 následujících záruk: [8]

- **Consistency (konzistence)** – V jeden okamžik jsou na všech uzlech dostupná stejná data.
- **Availability (dostupnost)** – Všechny požadavky na čtení a zápis dat jsou systémem vždy obslouženy.
- **Partition tolerance (odolnost vůči rozpadu sítě)** – Distribuovaný systém funguje, i když se rozpadne na několik izolovaných částí.



Obrázek 6 CAP teorém

Ideálem je mít systém s průnikem všech 3 vlastností (viz. obrázek 6), to však není možné. CA systémy bez odolnosti vůči rozpadu sítě běžně slouží jako tradiční relační databáze, běžící na jednom serveru. Tyto systémy ovšem nemůžou být rozšířené napříč více servery. U distribuovaných systémů je tedy P složka nutná a zbývající A složka nebo C složka je volitelná. [8]

CP systémy z důvodu vysokých požadavků na zajištění konzistence nemůžou poskytnout řádnou dostupnost, proto jsou vhodné do situací s přísným požadavkem na přesnost dat. Naopak AP systémy nemůžou poskytnout silnou konzistenci dat. Používají se spíše v situacích s častými žádostmi ale s menšími požadavky na přesnost. [16]

Nicméně CAP teorém nefunguje takhle doslovně, nemusí se volit striktně mezi dvěma složkami. Tento teorém nám spíše říká, že v distribuovaném prostředí musíme počítat s určitými omezeními v kontextu těchto tří podmínek. Například některé systémy nabízí odolnost vůči rozpadu sítě a vhodný kompromis mezi zbylými vlastnostmi. [2]

3.2.2 Rozdělení vlastností SQL a NoSQL databází v kontextu s CAP

Z pohledu CAP teorému většinu distribuovaných RDBMS označujeme primárně za CP (konzistentní s odolností vůči rozpadu sítě) s eventuální dostupností, zatímco NoSQL databáze označujeme za AP (dostupné s odolností vůči rozpadu sítě) s eventuální konzistencí. S tímto souvisí i provádění transakcí, čímž je myšleno provedení jedné nebo více CRUD (create, read, update, delete) operací. Provádění transakcí u relačních databází se označuje pod akronymem ACID, zatímco u NoSQL databází jako BASE. [16]

RDBMS transakce typu ACID:

- **Atomicity** – Databázová transakce je nedělitelná (atomární). Pokud transakce selže, než se dokončí operace, databáze se vrátí do původního stavu, před provedením transakce.
- **Consistency** – Při a po provedení transakce není porušeno žádné integritní omezení.
- **Isolation** – Souběžné transakce se vzájemně neovlivňují.
- **Durability** – Po uskutečnění každé transakce, změny jsou uloženy trvale a již nemohou být ztraceny.

NoSQL transakce typu BASE:

- **Basic Availability** – Systém garantuje odpověď na každý požadavek, nicméně není zaručena konzistence, tedy v podstatě funguje po celou dobu.
- **Soft state** – Systém se může kdykoliv měnit, není konzistentní po celou dobu.
- **Eventual consistency** – Časem ovšem vždy dosáhne určitého konzistentního stavu.

3.2.1 NoSQL

NoSQL databáze bývají často označovány jako „bezschématové“ ovšem, aby data byla pro stroj srozumitelná a řešitelná, je nutné nějaké schéma vytvářet. Toto schéma nemusí být nijak předem definováno a spíše než o schématu v klasickém pojetí, hovoříme o meta informacích k datům. Oproti relačním databázím, NoSQL databáze často přenášejí zodpovědnost za správu schématu na aplikační logiku. [8]

NoSQL rozdělujeme na základě modelu ukládání dat. V současnosti se nejčastěji používají čtyři takové koncepty: databáze typu klíč-hodnota, dokumentové databáze, sloupcové databáze, grafové databáze.

Databáze klíč-hodnota

Tento typ databáze má jednoduchý model založený na páru klíč-hodnota. Hodnota je zde přiřazena danému klíči, podle kterého je možné hodnotu vyhledat, není tedy možné vyhledávat podle hodnoty (viz. obrázek 7). Hodnota může být nějaký řetězec, ukazatel, nebo nějaká bajtová sekvence, jako třeba obrázek. [8]

Typický příklad: Dynamo, Redis, Voldemort.

Key_1	Value_1
Key_2	Value_2
Key_3	Value_1
Key_4	Value_3

Obrázek 7 Databáze klíč-hodnota (zdroj: *Big Data: Concepts, Methodologies, Tools, and Applications*)

Sloupcové databáze

Většina sloupcově orientovaných databází vychází z BigTable databáze od firmy Google. V tomto uspořádání se datová sada skládá z několika řádků, přičemž každý řádek je adresován primárním klíčem a je složen ze skupiny sloupců, důležitý je fakt, že různé řádky mohou mít různé skupiny sloupců (viz. obrázek 8). Tento druh databáze je vhodný pro data, která rychle proudí, jako např. webové logy, data přicházející současně z různých zařízení, senzory apod. [2]

Typický příklad: BigTable, Cassandra, HBase, HyperTable.

Row Key_1	Column-Family-1		Column-Family-2
	Column Name-1	Column Name-2	Column Name-2
	Column Value-1	Column Value-2	Column Value-2

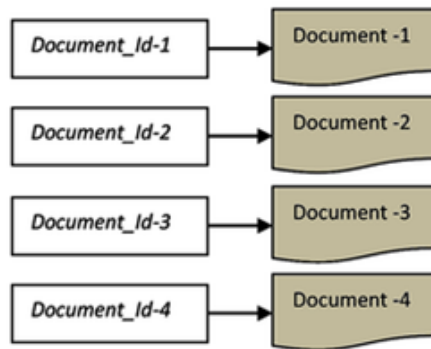
Row Key_2	Column-Family-1		
	Column Name-4	Column Name-5	Column Name-6
	Column Value-4	Column Value-5	Column Value-6

Obrázek 8 Sloupcové databáze (zdroj: *Big Data: Concepts, Methodologies, Tools, and Applications*)

Dokumentové databáze

Tento typ databází je jakýmsi derivátem od modelu klíč-hodnota. Klíč je zde použit k nalezení pozice dokumentu (viz. obrázek 9). Tento dokument je nějaká složená struktura většinou reprezentovaná v JSON formátu nebo nějakým podobným jako je BSON. Některé databáze tohoto typu nabízí možnost napsat si vlastní MapReduce aplikaci. [8]

Typický příklad: CouchDB, MongoDB.

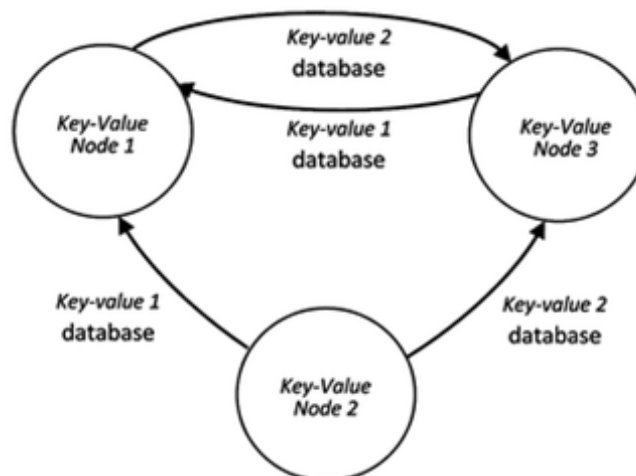


Obrázek 9 Dokumentové databáze (zdroj: *Big Data: Concepts, Methodologies, Tools, and Applications*)

Grafové databáze

Grafové databáze jsou speciálním typem NoSQL databází, graf je zde použitý jako datový model. Datová sada je reprezentována prostřednictvím vrcholů (nebo též uzlů) a vazeb (nebo též hran), které propojují tyto vrcholy (viz. obrázek 10). Tento typ databází vznikl spíše pro správu dat s komplexní strukturou než pro Big Data. Většina databází jako např. GraphDB a Neo4j běží na jediném serveru, nicméně jsou i databáze, jako např. Titan, který je optimalizován pro ukládání a vyhledávání grafů obsahujících stovky až miliardy vrcholů a vazeb distribuovaných napříč několika stroji. [8]

Typický příklad: GraphDB, Neo4j, Titan.



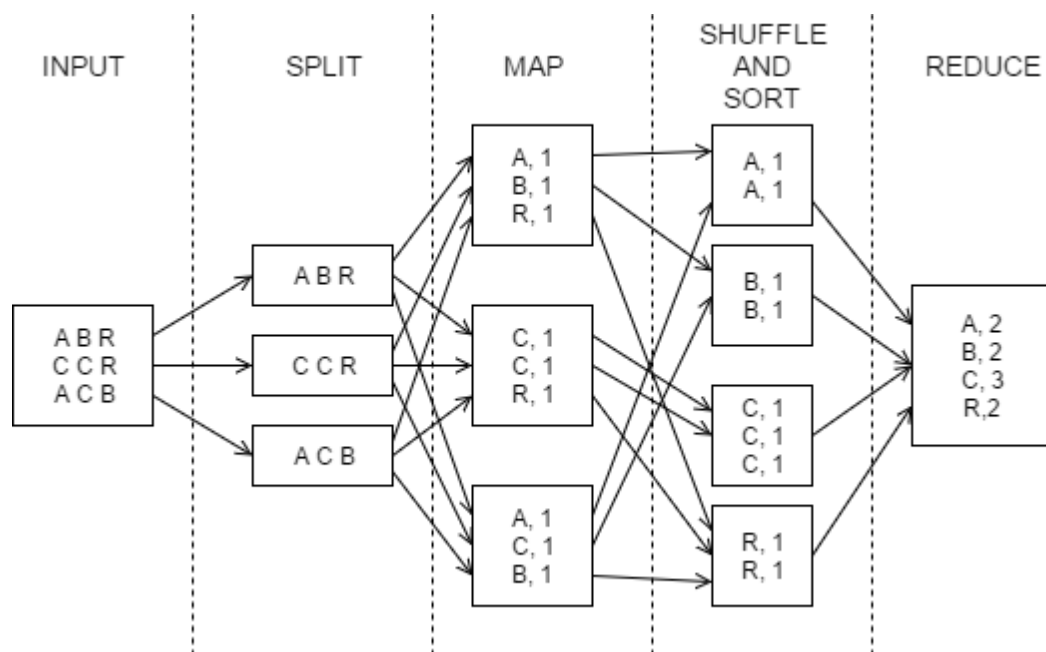
Obrázek 10 Grafové databáze (zdroj: *Big Data: Concepts, Methodologies, Tools, and Applications*)

3.3 Programovací paradigmatata

Programovací paradigmatata jsou fundamentálním prvkem pro vývojáře, udávající, jakým stylem psát programy a aplikace. V Big Data přístupu je záměr psát především datově poháněné paralelní programy, které mohou být řešeny ve velkém měřítku. Mnoho těchto modelů je určeno s různým zaměřením, proto přináší rozdílné výhody a nevýhody.

3.3.1 MapReduce

MapReduce je programovací model, který je navržen pro paralelní distribuované zpracování velkých datových sad. Principiálně vychází z běžně užívaných funkcí mapování a redukce v kombinaci s paralelním provedením metody zvané rozděl a panuj. Procesy založené na tomto modelu obecně prochází postupně 4 stavy (viz. obrázek 11): rozdělení vstupních dat, mapování, míchání/seřazení a redukce. [2]



Obrázek 11 MapReduce model

Z programátorské perspektivy je však důležitá pouze mapovací a redukční funkce, ty jsou totiž uživatelsky specifikovatelné. Běhové prostředí se stará o detaily rozdělení vstupních dat, seřazení, vypořádání se s chybovostí a řízení vnitřní komunikace mezi stroji.

Mapovací funkce zpracovává pár klíč/hodnota ke generování sady nezávislých dalších párů klíč/hodnota. Redukční funkce pak sloučí všechny hodnoty přiřazené ke stejnému klíči.

Hlavní výhodou tohoto modelu je schopnost „shared-nothing“, což znamená, že všechny mapovací funkce mohou být vykonávány nezávisle na sobě. Tato schopnost umožňuje paralelní běh jednoduchého programu napříč tisíci nebo dokonce až miliony nespolehlivých strojů s vyřešením úlohy ve velmi krátkém čase. [2]

Příklad: Hadoop MapReduce.

3.3.2 Funkcionální programování

Big Data zpracovávací systémy využívající funkcionálního programovacího modelu umožňují programátorům psát aplikace jednoduchou a deklarativní cestou. Programovací rozhraní jsou specifikována jako funkce a jsou aplikována na vstupní datové zdroje. Výpočet je pak prováděn jako vyhodnocování matematických funkcí. Funkcionální programování je deklarativní a zabraňuje sdílení měnitelných stavů. [22]

Příklady frameworků: Spark, Flink.

3.3.3 Actor model

Actor model je programovací model pro souběžný výpočet úlohy, kde „Actor“ je univerzální základní jednotkou pro tyto paralelní výpočty. Actor na základě obdrženého souboru zpráv reaguje specifickou procesní logikou. Na základě kontextu zprávy dělá rozhodnutí, vytváří více actorů a posílá zprávy ostatním actorům. Tento model je někdy též označován jako reaktivní programovací model, protože programátory vytvořená akční logika reaguje na události a kontextuální změny. Reakce actoru se dějí v jakémkoliv pořadí a akce ostatních actorů jsou také paralelní. [22]

Příklady frameworků: Akka, Storm, S4.

3.3.4 Statistické a analytické modely

V posledních letech vzrostla potřeba co nejjednodušším způsobem řešit statistické a analytické výpočty. Na základě toho vzniklo mnoho knihoven a frameworků poskytujících programovací modely na bázi lineární algebry, pracující s vektory, maticemi a tenzory jako datovými strukturami k vypořádání se s algebraicky definovanými matematickými problémy např. ve strojovém učení a data miningu. [22]

Příklady: R, Mahout.

3.3.5 Data-flow

Data-flow programovací paradigma modeluje programy jako orientované grafy dat s operacemi a závislostmi mezi vrcholy a vazby. Program je zde chápan jako řada propojení, mezi nimiž se pohybují data. Každý operátor a procesor má obvykle explicitně definované vstupy a výstupy a funkce jako černé skřínky. Operace běží v momentě, jakmile jsou validní všechna vstupní data. Každý uzel je zde chápan jako malá úloha, zatímco hrany reprezentují závislosti dat mezi různými úkoly (uzly). Tento programovací model zpracovává úlohy neoddělitelně paralelně, je dobře uplatnitelný ve velkých decentralizovaných systémech. [22]

Příklady: Oozie, Dryad

3.3.6 Bulk Synchronous Parallel

BSP je výpočetní a programovací model pro návrh paralelních algoritmů. Výpočet algoritmu pokračuje v sérii globálních tzv. super-kroků, které se skládají ze tří složek: [22]

- **Souběžný výpočet** – Každý procesor může provádět lokální výpočty. Každý procesor může využívat pouze hodnoty uložené v rychlé lokální paměti procesoru.
- **Komunikace** – Procesy si mezi sebou vyměňují data za účelem snížení nároků na datové úložiště.
- **Bariérová synchronizace** – V okamžiku kdy proces dosáhne určitého bodu (tzv. bariéry) čeká, dokud všechny ostatní procesy nedosáhnou stejné bariéry.

Příklady: Giraph, Pregel, Hama.

4 BIG DATA ANALYTICKÉ METODY

Analytika je jedna z oblastí, kde se vyskytuje spousta termínů, které zní podobně, jejich význam je však odlišný. Analýza je vědecká metoda, založená na dekompozici celku na elementární části. Cílem analýzy je identifikovat podstatné a nutné vlastnosti elementárních částí celku, poznat jejich podstatu a zákonitosti. Big Data Analytics je nový termín, který označuje analýzu nestrukturovaných velkých a rychlých datových množin, které nemůžou být analyzovány pomocí konvenčních nástrojů a metod. [27]

V Big Data analýzách se uplatňují hlavně tyto přístupy: [5]

- **N = VŠECHNO** – Nevybírání se nějakou zástupnou množinou, ale datová sada se analyzuje celá. Správné vybrání zástupné množiny je dosti komplikované, neboť podléhá z teorie pravděpodobnosti na zcela náhodný výběr zástupných členů. Analýzou celé datové sady se navíc nepřipravíme o různé detaily skryté uvnitř.
- **Integrace dat z různých zdrojů** – Integrovaná data z různých zdrojů v sobě ukrývají více proměnných než separované menší datové sady.
- **Důraz na korelace** – Hledat kauzální vazby v problémech tak komplexních jaké se řeší v obrovských datových sadách je příliš komplikované, proto je kladen důraz především na pouhé korelace, které se ukázaly být např. v systémech doporučování výjimečně účinné.

Z technického hlediska lze klasifikovat analytické metody dvěma způsoby, z pohledu hloubky analýzy a podle analyzovaného předmětu.

4.1 Dělení podle hloubky analýzy

Z pohledu hloubky analýzy je možné vytyčit tři základní podskupiny: Deskriptivní, prediktivní a preskriptivní.

4.1.1 Deskriptivní analytika

Deskriptivní analýza patří mezi nejjednodušší formy analytických metod, zahrnuje jednoduché statistické metody jako průměr, medián, modus, směrodatnou odchylku, rozptyl a měření frekvence určitých událostí. Deskriptivní analytika se především používá k vytváření reportů, dále k tzv. „*dashboardům*“, tedy k rychlým přehledům a k datové vizualizaci, což jsou techniky, které tvoří převážnou část činností v tradičních BI aplikacích. [23]

Deskriptivní analytika nám tedy zpětným pohledem odpovídá na otázku, co se stalo.

4.1.2 Prediktivní analytika

Prediktivní analytika se zabývá vytvářením předpovědí a vytváření statistických modelů k určení budoucích případných událostí. Principiálně je založena na statistických metodách a snaží se odhalit vzory a zachytit všechny spojitosti v datech. Je možné ji rozdělit do dvou základních celků, regresní analýza (např. multinomický logistický model) a strojové učení (např. neuronové sítě). K predikování budoucích hodnot je možné použít např. klouzavý průměr a pomocí identifikovaných historických vzorů jej následně extrapolovat do budoucnosti. [23]

Následující techniky jsou nejčastěji užívané při řešení prediktivních analytických úloh [24]:

- **Klasifikace** – Predikování kategorií položek třídy, nejčastěji s užitím techniky tzv. „Rozhodovacího stromu“
- **Clustering** – Objevování přirozených skupin nebo datových shluků.
- **Asociace** – Co dělají proměnné společně, např. analýza nákupního košíku (angl. „*market basket analysis*“).
- **Detekce divergence** – Odhalení změn nebo odchylky.
- **Aproximace a časové řady** – Předpovídání budoucího průběhu hodnoty.
- **Analýza závislostí** – Objevování vztahů a jejich vlastností.
- **Web mining** – Dolování relevantních informací ze strukturovaných, semi-strukturovaných a nestrukturovaných dat z webu.

4.1.3 Preskriptivní analytika

Preskriptivní analýza je pokročilá analytická koncepce závislá i na schopnostech právě deskriptivní a prediktivní analýzy. Preskriptivní úloha je optimalizačního charakteru. Zabývá se zkoumáním příčinných vazeb výsledků analýz s podnikovými optimalizačními procesy. Výsledky preskriptivní analýzy tedy ukazují důsledky užití jednotlivých řešení, vycházející z dané situace. Tyto výsledky se získávají pomocí pokročilých technik jako je grafová analýza, simulace, komplexní zpracování událostí, systémy doporučení, heuristiky a strojové učení.

Existuje velmi omezené množství dobrých příkladů preskriptivní analýzy běžně užívané v reálném světě. Podle výzkumu společnosti Gartner pouze 3 % organizací užívá tuto techniku. [25]

4.2 Dělení podle předmětu analýzy

Z pohledu podle předmětu analýzy, můžeme uvést 4 základní podskupiny: analýza strukturovaných dat, analýza textu, analýza multimédií a analýza webových dat.

4.2.1 Analýza strukturovaných dat

Díky dobře definované struktuře jsou tyto data efektivně analyzovatelná pomocí technologií, jakými jsou relační databázové řídicí systémy, datové sklady, OLAP a BPM. Datová analýza je pak nejčastěji založena na data miningu a statistické analýze. [14]

4.2.2 Analýza webových dat

Analýza webových dat (někdy též jako web mining) je spojena s několika dalšími oblastmi jako např. s databázemi, vyhledáváním informací, NLP a analýzou textu. Jedná se o aplikaci dolování dat k odhalení vědomostí a vzorů ze sítě WWW. Analýzu webových dat dělíme do 3 základních celků [16]:

- **Dolování obsahu (web content mining)** – Dolování obsahu je proces k odhalení užitečných vědomostí z webových stránek, které obecně zahrnují množství rozličných typů dat, jako text, obrázky, audio, video, kód, metadata a hypertextové odkazy. Dolování obsahu se nejčastěji provádí dvěma různými přístupy: přístupem vyhledávání informací a databázovým přístupem. Systémy na vyhledávání informací slouží k získávání informací nebo k jejich filtraci. Databázový přístup se pak zaměřuje spíše na simulaci a integraci webového obsahu, aby bylo možné provádět složitější dotazy než jenom vyhledávání založená na klíčových slovech.
- **Dolování struktury (web structure mining)** – Dolování struktury slouží k odhalení propojení webové sítě. Strukturou je zde myšlen schématický diagram propojení webové stránky napříč webovou sítí. K odhalení takového propojení je třeba zkoumat topologii hypertextových odkazů mezi dokumenty na webu. Výsledky těchto propojení pak odhalují důležitost, nebo určitou popularitu jednotlivých webových stránek a slouží k jejich klasifikaci. Jsou to systémy jako je např. PageRank od firmy Google nebo Clever od IBM.
- **Dolování použití (web usage mining)** – Dolování použití se zaměřuje na analyzování dat, které vznikají uživatelskými interakcemi s webovou stránkou. Jsou to tedy data jako serverové logy, prohlížečové záznamy historie prohlížení, uživatelské pro-

filí, záznamy o dotazování, pohyby myši apod. Cílem tedy je studovat chování uživatelů, což slouží k poskytování lepších služeb, zlepšení bezpečnosti, cíleným reklamám apod.

4.2.3 Analýza textu

Textová analýza (text mining) je technika, pomocí které se extrahují informace z textových dat. Analýza textu zahrnuje především statistickou analýzu, počítačovou lingvistiku a strojové učení. V této technice se uplatňují především tyto metody [14]:

- **Extrakce informací** – Systémy IE se skládají ze tří subčástí a to: rozpoznávání pojmenovaných entit, řešení koreferencí a detekci vztahů. Jedná se o typ vyhledávání informací, cílem je z nestrukturovaného textu extrahovat informace.
- **Sumarizace textu** – Existují dva způsoby automatického vytváření shrnutí textu a to: extraktivní a abstraktivní. Extraktivním způsobem, se pouze extrahují klíčové informace z textu, o důležitosti těchto informací rozhoduje např. pozice v textu, nebo frekvence výskytu. Není tedy vyžadováno porozumění daného textu. Oproti tomu abstraktivní způsob vyžaduje vyšší stupeň umělé inteligence. Vytváří se souhrnné parafrázování originálního textu. Využívá se technik zpracování přirozeného jazyka (NLP). Výsledek abstraktivní sumarizace je více souvislý, nicméně systémy extraktivní sumarizace jsou snadněji osvojitelné zvláště pro Big Data.
- **Analýza sentimentu** – Technika zahrnuje analyzování textů, jejichž obsahem jsou lidské názory na produkty, organizace, jednotlivce a události. Z těchto údajů poté nejvíce těží v oblasti marketingu, finančnictví, politiky a sociálních věd. Základní úloha této techniky je určení polaritý daného textu na třech úrovních, a to na úrovni dokumentu, na úrovni věty a na úrovni aspektu nebo funkce. Tyto úrovně jsou ohodnoceny buďto jako pozitivní, negativní nebo neutrální, nebo někdy též i větší škálou hodnotících metrik jako třeba pětihvězdičkovým systémem.
- **QA systémy** – Systémy odpovídající na otázky dělíme do tří skupin: systémy vyhledávání informací, odpovídání na bázi znalostí a hybridní. Jedná se např. o osobní asistenty jako iOS Siri od firmy Apple, která odpovídá na bázi znalostí. Tyto systémy jsou ovšem pořád ve fázi vývoje, nicméně hybridní superpočítačový QA systém Watson od IBM, dokázal porazit člověka v televizní soutěži Jeopardy (pozn.: česky známá jako Riskuj).

4.2.4 Analýza multimédií

Multimediální data (obrázky, audio, video apod.) jsou specifické především heterogenní strukturou, proto je velmi obtížné extrahovat z nich užitečné informace a porozumět sémantice. Analýza multimédií zahrnuje spoustu technik, z nichž nejběžnější jsou [14]:

- **Souhrny** – Automatická extrakce nejdůležitějšího vzorku shrnující daný objekt.
- **Anotace** – S pomocí sémantiky obsahu vytvoření popisu daného objektu.
- **Indexace a vyhledávání** – Zahrnuje metody popisování, ukládání a organizaci multimediálních informací pro rychlé vyhledání.
- **Doporučování** – Napříč uživatelským preferencím doporučení multimediálního obsahu.

II. PRAKTICKÁ ČÁST

5 ÚVOD DO PRAKTICKÉ ČÁSTI

Pouze velké IT společnosti jako je Facebook, Yahoo!, Twitter, Amazon aj. si můžou dovolit investovat velké množství svých prostředků na vývoj proprietárního nebo open-source softwaru k zvládnutí problémů Big Data, taková činnost totiž zahrnuje velké požadavky na IT infrastrukturu, na odborníky a další. Ovšem tyto analýzy mohou být velmi významné i pro menší nebo středně velké společnosti. V kapitole 6 se nachází analýza cloudových služeb z pohledu uživatele v kontextu s aplikací Big Data technologií. Zahrnuje nezbytné teoretické pozadí dané oblasti s cílem podat povědomí při výběru demonstrativně zvolených kritérií a jejich vah.

Mezi open-source projekty je jednoznačně Apache Hadoop vedoucí vysoce distribuovanou platformou. Proto větší část praktické části je zaměřena právě na tuto technologii. V kapitole 7 je ukázán způsob implementace a otestování Apache Hadoop, včetně jednoduchých případových studií na benchmarkové sadě.

6 ANALÝZA CLOUDOVÝCH BIG DATA ŘEŠENÍ

6.1 Cloud Computing

Cloud computing je nový způsob výpočetního modelu, uživatelé zde přes internet přistupují k výpočetnímu výkonu, datovému uložení, online aplikacím a síťové infrastruktuře jako ke sdílené službě.

Cloud computingové služby můžeme popsat třemi distribučními modely („*service models*“), čtyřmi nasazovacími modely („*Deployment models*“) a pěti charakteristikami. Zkráceně tedy 3S-4D-5C. [2]

Rozdělení podle distribučního modelu [2]:

- **Software-as-a-Service (SaaS)** – Software je zde pronajímán jako služba. Zákazník si tedy pronajímá přístup k aplikaci, ne aplikaci samotnou. Poskytovatel zajišťuje provoz a podporu služby. Příkladem SaaS jsou Gmail, Facebook, Twitter, Microsoft Office 365.
- **Platform-as-a-Service (PaaS)** – Prostředí zahrnující databázové a aplikační servery, vývojová IDE a API pro vytváření, testování a běh vlastních aplikací. Vývojáři se tak mohou soustředit pouze na vývoj aplikací, zatímco cloudový poskytovatelé jsou zodpovědní za správu a optimalizaci prostředí a vlastní infrastruktury. Příkladem PaaS je např. Google App Engine.
- **Infrastructure-as-a-Service (IaaS)** – Uživatelé si pronajímají infrastrukturu formou služby, nejčastěji je tato služba zajišťována s využitím virtualizace. Hlavní výhodou je, že o provoz infrastruktury se stará provozovatel. Typicky se jedná o virtuální stroje, uložení, load balancery, sítě, např. Amazon EC2, RackSpace Cloud.

Rozdělení podle modelu nasazení [2]:

- **Private** – Provozován výhradně pro jednu organizaci.
- **Public** – K dispozici libovolným organizacím nebo jednotlivcům.
- **Community** – Cloud sdílen skupinou organizací se společnými požadavky nebo zájmy.
- **Hybrid** – Kombinace dvou nebo více cloudů.

Rozdělení podle charakteristiky [26]:

- **Samospráva na vyžádání (On-demand self service)** – Uživatel může automaticky využívat službu podle potřeby na vyžádání, aniž by potřeboval interakci s poskytovatelem služby.
- **Široký přístup k síti (Broad network acces)** – Služba je dostupná prostřednictvím sítě a je možné k ní přistupovat pomocí standardních mechanismů podporujících rozličné klientské platformy, např. mobilní telefony, tablety, aj. pracovní stanice.
- **Fond prostředků (Resource pooling)** – Seskupování většího množství prostředků takovým způsobem, aby je bylo možno sdílet a distribuovat v požadované kvantitě.
- **Rychlá pružnost (Rapid elasticity)** – Schopnost rychle a snadno zajistit a uvolnit zdroje podle aktuální potřeby. Uživatel tak platí za výpočetní výkon pouze tolik, kolik potřebuje.
- **Měřitelná služba (Measured service)** – Umožňuje účtování, tj. sledování, o kvantitě a kvalitě využitých („spotřebovaných“) výpočetních prostředků a služeb. Kvantitou prostředků rozumíme mimo jiné výpočetní výkon serverů, přenosovou kapacitu sítě, kapacitu úložiště apod. Kvalita služeb zahrnuje mimo jiné různé stupně neboli úrovně výkonu, dostupnosti, ochrany, uchování dat, rychlosti vyhledávání nebo zabezpečení.

6.2 Amazon Web Services

Amazon nabízí výpočetní a úložné zdroje jejich vlastní IT infrastruktury vývojářům ve formě webových služeb. Nabízí velké množství cloudových služeb, které mohou být složeny uživateli k vytváření jejich vlastní SaaS aplikací nebo k integrování tradičního softwaru s cloudovými schopnostmi. AWS poskytuje SDK pro většinu hlavních programovacích jazyků a platform (např. Java, .NET, PHP, Android apod.), a tak lze snadno interagovat s jejich službami.

Výpočetní část zahrnuje Elastic Compute Cloud (EC2) pro vytváření a běh virtuálních serverů a Amazon Elastic MapReduce pro vytváření a provádění MapReduce aplikací. Úložné řešení je založeno na Amazon Simple Storage Service (S3), s velkým rozsahem úložných tříd navržených tak, aby se vyrovnaly s různými případy užití. Sada databázových systémů je také k dispozici. Relational Database Service (RDS) pro relační tabulky, DynamoDB pro nerelační tabulky, SimpleDB pro správu malých datových sad.

6.3 Google Cloud Platform

Cloud computingová služba, využívající stejnou infrastrukturu, kterou Google interně užívá pro své produkty, jako je jejich internetový vyhledávač nebo YouTube. Tato platforma poskytuje vývojářům nástroje pro vytváření velkého rozsahu aplikací od jednoduchých webových aplikací po komplexní programy.

Výpočetní prostředky poskytuje Compute Engine a App Engine. Compute Engine poskytuje IaaS, tedy virtuální počítače pro hostování pracovní zátěže. App Engine nabízí PaaS, umožňující přístup k SDK pro rychlý vývoj a nasazení produktů, které běží právě na App Engine, dále nabízí jednoduchou administraci bez nutnosti řešit hardwarové požadavky. Úložné řešení je poskytováno pomocí Cloud Storage, platformy, navržené pro ukládání velkého množství nestrukturovaných datových sad. Dále nabízí databázové úložné řešení Cloud Datastore pro NoSQL nerelační uložení, Cloud SQL pro MySQL relační uložení a Cloud BigTable databázi. K dispozici je také pestrá paleta dalších služeb a aplikací k rychlému umožnění široké škály funkcionalit.

6.4 Microsoft Azure

Microsoft Azure je prostředí poskytující velkou sadu služeb, které mohou být použity vývojáři k vytvoření různých cloudově orientovaných aplikací nebo ke zlepšení stávajících aplikací s cloudově založenými schopnostmi. Tato platforma poskytuje on-demand výpočetní a úložné zdroje využívající výpočetní výkon a úložný prostor datových center firmy Microsoft. Azure je navržen pro poskytování vysoce dostupných a dynamicky škálovatelných služeb, které slouží uživatelům s tzv. „*pay-per-use*“ účtovacím modelem. Poskytuje uložení pro velké datové sady a s možností provádění dávkových výpočtů nad nimi. Dále poskytuje standardní rozhraní, které umožňuje vývojářům snadno interagovat s jejich službami. Vývojáři používající IDE jako Microsoft Visual Studio nebo např. Eclipse, mohou navrhovat a publikovat Azure aplikace.

Microsoft Azure zahrnuje tři základní komponenty/služby:

- **Compute** – Výpočetní část zajišťující chod cloudových aplikací. Každá aplikace je rozdělena do tří rolí: Webová role pro webově orientované aplikace, Worker role pro dávkové aplikace, Virtual-machines role pro image virtuálních počítačů.
- **Storage** – Poskytuje škálovatelné uložení ke správě binárních a textových dat (tzv. „*Blobs*“), nerelační tabulky („*Tables*“), fronty pro asynchronní komunikaci mezi

komponenty („*Queues*“). A pro relační databáze Microsoft poskytuje jejich vlastní cloudové databázové služby zvané Azure SQL Database.

- **Fabric controller** – Zajišťuje rychlé a spolehlivé propojení s internetem a datacentry Microsoftu. Compute a Storage služby jsou postaveny právě na této komponentě.

6.5 Porovnání řešení od firmy Amazon, Google a Microsoft

Existuje celá řada dalších cloudových poskytovatelů např. IBM, Alibaba, Rackspace, CenturyLink apod. Nicméně za největší zástupce v tomto odvětví můžeme označit právě Amazon Web Services, Google Cloud Platform a Microsoft Azure, např. výzkumná společnost Gartner je označuje za přední poskytovatele IaaS. [28]

V tabulce 1 je možné vidět přehled klíčových užívaných technologií.

	Amazon	Google	Microsoft
Uložiště	S3	Google cloud storage	Azure storage
MapReduce	Elastic MapReduce	AppEngine	Hadoop on Azure
Big Data Analytics	Elastic MapReduce	BigQuery	HDInsight (Hadoop a Spark)
Streaming processing	Amazon Kinesis Streams	Cloud Dataflow	Azure Stream Analytics
Relační databáze	Amazon RDS (MySQL, Oracle, aj.)	Cloud SQL (MySQL, PostgreSQL)	Azure SQL Database
NoSQL databáze	DynamoDB	Cloud Datastore, Cloud Big Table	DocumentDB, Table storage
Strojové učení	Amazon Machine Learning	Cloud Machine Learning Services	Azure Machine Learning

Tabulka 1 Přehled užívaných technologií

6.5.1 Výpočetní služby

AWS bývá často považován za lídra na trhu v poskytování IaaS s jeho technologií EC2 (Elastic Compute Cloud). Nabízí volbu v rozsahu 40 virtuálních serverových instancí od levných zařízení až po špičkové high-end stroje nabízejících 128 virtuálních CPU s operační pamětí až neuvěřitelných 2 TB. Instance jsou možné pronajímat na vyžádání na hodinu. Je také možná podpora automatického škálování těchto instancí.

Google s Compute Enginem nabízí pestrou nabídku možností výběru virtuálního stroje. V nabídce jsou konfigurace, až po 64 virtuálních CPU s 416 GB operační pamětí. Podpora

operačních systémů Windows i Linux. Compute Engine nabízí k pronajímání na minutu a nabízí slevy pro prodloužené používání.

Azure prostřednictvím Azure Virtual Machine taktéž nabízí 40 různých virtuálních instancí s konfigurací až po 32 jader a s 440 GB operační paměti. Virtuální stroje je možné volit mezi platformou Windows nebo různými Linuxovými distribucemi. Podpora automatického škálování je taktéž zahrnuta. Pronajímání je možné taktéž na každou hodinu.

Ačkoliv Amazon nabízí nejlepší high-end řešení, to se pochopitelně výrazně promítne v ceně. Nabízené služby ovšem nejsou vždy stejné ve vztahu k regionu, AWS působí v 16 regionech, Azure ve 34 regionech a Google Cloud Platform v 7 regionech po celém světě. Co se tedy týče nabídky AWS má pestrou nabídku s nejvýkonnějšími konfiguracemi, ale zároveň nejdražší. Azure nabízí podobné množství výběru také za poměrně vysokou cenu. Google nabízí o něco menší výběr ovšem je cenově nejdostupnější.

6.5.2 Uložiště

Všichni poskytovatelé nabízí dobré objektové uložení s dobrým přístupem k datům. Amazon S3 díky nejdelší době na trhu však nabízí rozsáhlou dokumentaci, velkou komunitní podporu a je díky tomu i nejlépe odzkoušený a otestovaný a zároveň Amazon tedy i díky delší době působení na tomto trhu poskytuje lepší služby v oblasti úložného prostoru, ale opět je nejdražší.

6.5.3 Databáze

Všichni tři poskytovatelé nabízí relační databáze, NoSQL databáze a datové sklady petabytového měřítka. Nicméně volba databázového řešení není jednoduchá, záleží totiž na individuálních požadavcích. Nutno ovšem dodat, že Google Cloud Platform trochu zaostává v nabídce možností koncovým uživatelům.

6.5.4 Big Data Analytics

Všechny tři cloudové služby poskytují základní pilíře pro analýzy Big Dat, a to pro zpracování dat, streamované analýzy, strojové učení a vizualizaci. AWS nabízí široké množství služeb, nejslabším článkem je však poměrně menší zastoupení v před-trénovaných modelů strojového učení. Azure je na tom rozhodně lépe, nicméně v tomto poli Google nabízí nejvíce pokročilé nástroje pro analýzy Big Dat a strojové učení.

6.5.5 Shrnutí

Obecně vzato žádný z těchto poskytovatelů jednoznačně nepřevažuje nad druhým, protože každý má svou individuální silnou stránku, která spočívá v jiné oblasti. Amazon na trhu s cloudovými službami působí nejdéle, a to od roku 2006 a dá se tedy říci, že má v tomto ohledu náskok, Microsoft oficiálně vydal svou platformu Azure v roce 2010 a Google Cloud Platform oficiálně funguje od roku 2011.

Nejsilnější stránkou AWS je jejich nejširší nabídka služeb a nejlepší nabídka IaaS. Microsoft na druhou stranu má taktéž dobrou nabídku IaaS, nicméně hlavní síla spočívá v poskytovacích platformách, poskytuje spoustu nástrojů pro podniky a v této sféře má poměrně silné zastoupení, navíc se stává stále více otevřenější open-sourcovým distribucím. Síla Google Cloud Platform však spočívá především v analytických nástrojích a ve strojovém učení.

6.5.1 Výběr vhodného řešení metodou TOPSIS

TOPSIS („*Technique for Order Preference by Similarity to Ideal Solution*“) analýza pomáhá při výběru nejlepšího řešení při vícekritériálním rozhodování. Hlavní kritéria a jejich ohodnocení bude zavedeno na základě informací uvedených z kapitoly 6.5, další kritérium, které musí být zahrnuto je cena, která z pohledu uživatele, může hrát klíčovou roli. Pro následnou aplikaci této metody jsou ohodnocena jednotlivá kritéria (viz. tabulka 2), přičemž vyšší hodnota je lepší než nižší, jedná se tedy o maximalizační typ úlohy. K těmto kritériím je přidána váha, udávající důležitost jednotlivého kritéria.

Kritérium	Váha	AWS	GCP	Microsoft Azure
Výpočetní služby	0,2	9	6	7
Uložiště	0,15	6	4	5
Databáze	0,15	7	5	7
BDA	0,2	6	8	6
Cena	0,3	4	9	6

Tabulka 2 Kritériální matice

Vytvoření normalizované matice $R = (r_{ij})$ dle vztahu 1:

$$r_{ij} = \frac{y_{ij}}{\sqrt{\sum_{i=1}^m y_{ij}^2}} \quad (1)$$

Kde $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$; y_{ij} jsou kritéria.

Na základě váhového rozdělení kritérii se dále vytvoří vážená normalizovaná matice (tabulka 3), kde $Z = (z_{ij})$ dle vztahu 2:

$$z_{ij} = v_j \cdot r_{ij} \quad (2)$$

Kde v_j je váha j -tého kritéria

Kritérium	AWS	GCP	Microsoft Azure
Výpočetní služby	0,14	0,09	0,11
Uložiště	0,10	0,07	0,09
Databáze	0,09	0,07	0,09
BDA	0,10	0,14	0,10
Cena	0,10	0,23	0,16

Tabulka 3 Vážená normalizovaná matice

Vytvoření ideální varianty dle vztahu 3 kde $H = (h_1, h_2, \dots, h_n)$ a bazální varianty dle vztahu 4 kde $D = (d_1, d_2, \dots, d_n)$:

$$h_j = \max z_{ij} \quad (3)$$

$$d_j = \min z_{ij} \quad (4)$$

Vypočtení vzdálenosti jednotlivých variant od ideální varianty IV dle vztahu 5 a vzdálenosti od bazální varianty BV dle vztahu 6, kde

$$IV = \sqrt{\sum_{j=1}^n (z_{ij} - h_j)^2} \quad (5)$$

$$BV = \sqrt{\sum_{j=1}^n (z_{ij} - d_j)^2} \quad (6)$$

Tabulka 4 ukazuje výsledek výpočtu vzdálenosti od ideální varianty dle vztahu 5.

Kritérium	AWS	GCP	Microsoft Azure
Výpočetní služby	0,0000	0,0022	0,0010
Uložiště	0,0000	0,0010	0,0002
Databáze	0,0000	0,0005	0,0000
BDA	0,0014	0,0000	0,0014
Cena	0,0159	0,0000	0,0055
IV	0,13	0,06	0,09

Tabulka 4 Vzdálenost od ideální varianty (Pozn.: červená značí místo výskytu maxima)

Tabulka 5 ukazuje výsledek výpočtu vzdálenosti od bazální varianty dle vztahu 6.

Kritérium	AWS	GCP	Microsoft Azure
Výpočetní služby	0,0025	0,0000	0,0003
Uložiště	0,0011	0,0000	0,0002
Databáze	0,0006	0,0000	0,0006
BDA	0,0000	0,0014	0,0000
Cena	0,0000	0,0180	0,0031
BV	0,06	0,14	0,07

Tabulka 5 Vzdálenost od bazální varianty (Pozn.: modrá značí místo výskytu minima)

Relativní ukazatel UV je dán vztahem 7, konečný výsledek pak udává tabulka 6.

$$UV = \frac{BV}{(IV + BV)} \quad (7)$$

	AWS	GCP	Microsoft Azure
IH	0,13	0,06	0,09
BH	0,06	0,14	0,07
UV	0,32	0,70	0,44

Tabulka 6 Výsledek porovnání

Výsledek analýzy

Metodou TOPSIS v této modelové situaci, kde cena hrála poměrně významnou roli, bylo zjištěno, že nejlepší poskytovatel cloudových služeb v kontextu s možnostmi poskytování Big Data analýz je Google. Google Cloud Platform sice nenabízí tak výkonnou infrastrukturu jako jeho konkurenti, nicméně poskytuje pokročilé analytické nástroje a v poměru právě k ceně dopadl nejlépe. Na druhé pozici se umístil produkt firmy Microsoft a jako třetí dopadl Amazon.

7 IMPLEMENTACE A TESTOVÁNÍ APACHE HADOOP

Hadoop je jeden z předních open-source projektů Apache Software Foundation, je to framework pro spolehlivé, škálovatelné, distribuované výpočty. Tento framework umožňuje zpracovávat velké množství dat napříč klastry počítačů za použití jednoduchých programovacích modelů. Je navržen na možnost škálování z jednotlivých serverů až na tisíce strojů, přičemž každý nabízí lokální výpočty a úložný prostor. Aplikační vrstva se stará o detekci a zvládání chyb, díky čemuž je možné navrhovat vysoce dostupné služby.

Tento framework zahrnuje 4 základní moduly:

- **Hadoop Common** – Sada knihoven a nástrojů pro ostatní moduly Hadoopu.
- **HDFS** – Distribuovaný souborový systém poskytující rychlý přístup k aplikačním datům.
- **Hadoop YARN** – Framework plánující rozdělení zátěže v celém klastru.
- **Hadoop MapReduce** – Implementace programovacího modelu MapReduce.

Celý framework je napsán v programovacím jazyce Java, s nějakým nativním kódem v C a s různými utilitami příkazového řádku napsanými jako shell skripty. Nicméně ačkoliv tedy MapReduce je napsán v Javě, díky „*Hadoop Streaming*“ může být použit jakýkoliv jiný programovací jazyk k implementaci mapovacích a redukčních úloh.

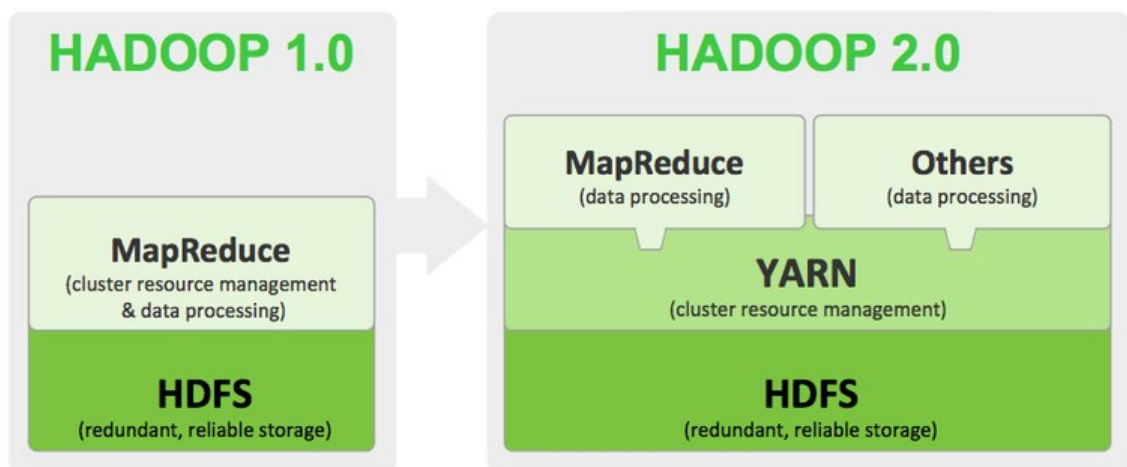
S Apache Hadoop frameworkem je možné využívat celou řadu souvisejících Big Data projektů od Apache Foundation jako např.: Cassandra, HBase, Hive, Mahout, Pig, Spark, Storm, ZooKeeper. Velké společnosti jako např. Amazon, Apple, Facebook, Google, HP, IBM, Microsoft užívají Hadoop pro své potřeby. S různým stupněm úprav pak existuje řada Hadoop distribucí jako např. Hortonworks, Cloudera, MapR.

7.1 Hadoop MapReduce

Hadoop MapReduce je framework na zpracování velkého množství dat, uložených v HDFS. Obecný popis je popsán v kapitole 3.3.1. Ve verzi Hadoop 1 se MapReduce procedura skládala z JobTracker procesu, který běžel na master uzlu a kde se staral se o řízení a chod pracovní úlohy a TaskTrackers procesů, které běžely na každém slave uzlu a staraly se o koordinování a řešení úloh v rámci jednotlivého uzlu. V Hadoop v2 došlo s příchodem technologie YARN ke změně, zodpovědnost za koordinování práce převzal YARN od JobTrackeru.

7.2 Hadoop YARN

YARN (Yet Another Resource Manager) je jedním z hlavních vylepšení verze 2. Jedná se o systém řízení zdrojů, který se stará o efektivní sdílení výpočetního výkonu napříč klastrem a efektivním využíváním dat uložených v HDFS. Je ústřední součástí Hadoop systému, proto je někdy označován jako OS Hadoopu. Poskytuje společnou platformu pro mnoho různých typů distribuovaných aplikací. Tato skutečnost rozsáhle zvyšuje celkovou schopnost celého systému, umožňuje totiž využití více programovacích modelů. V předešlé verzi byl na zpracování dat nativně podporován pouze MapReduce framework, který principiálně pracuje na bázi dávkového zpracování, což se ukázalo být velmi účinné k analyzování velkého množství dat, nicméně nebyla podpora ostatních způsobů zpracování, jako např. real-time, iterativního a grafového zpracování. YARN, jak je zobrazeno na obrázku 12, tedy umožňuje široké fungování distribuovaných aplikací, které poskytují takové funkce vedle sebe, sdílením jediného klastru a souborového systému HDFS, označení „Others“ v obrázku 12. udává např. frameworky jako: Dryad, Giraph, Hoya, Tez, Spark, Storm aj.



Obrázek 12 YARN (zdroj: hortonworks.com)

7.3 Implementace

Hadoop je volně ke stažení na stránce: <http://hadoop.apache.org/>

Je k dostání ve více verzích, které se mohou lišit jak ve funkčnosti, tak v postupu implementace, následný postup je uveden především pro aktuální verzi 2.8.0 dostupnou od 22. března 2017.

7.3.1 Základní požadavky

Elementární vlastností je funkčnost na komoditním hardwaru, jako vývojová a produkční platforma jsou podporovány GNU/Linux distribuce, Windows je sice taktéž podporován, nicméně není tolik zastoupen v použití, a tedy existuje jen malé množství dokumentace pro chod na tomto operačním systému. Pro běh je vyžadován od verze Apache Hadoop 2.7. Java Development Kit verze 7 a vyšší. Dále je vyžadován Secure Shell (SSH), aby docházelo ke komunikaci se vzdálenými démony Hadoopu.

7.3.2 Módy nasazení

Hadoop je možné nasadit ve třech různých módech:

- **Lokální mód** – Nedistribovaný mód, jednoduchý na nasazení, vhodný zejména pro lokální testování a debugování aplikací.
- **Pseudo-distribovaný mód** – Hadoop běží na jediném stroji, každý Hadoop démon ovšem běží jako separovaný proces, čímž je emulován distribuovaný mód.
- **Distribovaný mód** – Klasický distribuovaný chod, podporující rozproštění strojů od několika až po tisíce uzlů.

7.3.3 Příprava systému

Nejprve je tedy nutné mít nainstalovanou Javu verze 7 a vyšší. Dále je vhodné na každé stanici vytvořit dedikovaného uživatele, přiřadit ho do skupiny a tomuto uživateli přidělit oprávnění superuživatele editací „*/etc/sudoers*“. Editace se provede přidáním řádku: „*<user> ALL=(ALL) ALL*“, pod řádek s textem: „*##Allow member of group sudo to execute any command anywhere in the format*“. Po provedení těchto úkonů je nutný restart a další postup se provádí již pod vytvořeným novým uživatelem.

Pro komunikaci Hadoop démonů je nutné, aby byl sdílený veřejný klíč, vygenerovaný prostřednictvím SSH, ten je po vygenerování uložen v adresáři „*~/.ssh/id_rsa.pub*“ a je tedy nutné jej rozkopírovat všem stanicím, včetně té lokální, do adresáře „*~/.ssh/authorized_keys*“. Po tomto procesu je nutné přidat čtecí a zapisovací oprávnění k této složce („*authorized_keys*“) pro vytvořeného uživatele a spustit SSH, případně jej restartovat. Na konec je vhodné zkontrolovat konektivitu SSH. Na každou stanici, včetně té lokální, by mělo být možné se připojit bez zadávání hesla.

7.3.4 Instalace

Instalace spočívá ve stažení balíčku ze stránek výrobce a po stažení jeho rozbalení např. do adresáře „*/usr/local/*“. Po rozbalení je nutné změnit vlastníka a přidělit oprávnění k dané složce, dále je vhodné vytvořit link k tomuto adresáři, který se bude jmenovat např. „*hadoop*“.

Příklad (při použití následujících příkladů je nutno brát v potaz aktuální verzi Hadoopu, aktuální verzi Javy a také lokálního umístění Javy):

```
sudo mv ~/Desktop/hadoop-2.8.0.tar.gz /usr/local/  
cd /usr/local  
sudo tar -xvf hadoop-2.8.0.tar.gz  
sudo rm hadoop-2.8.0.tar.gz  
sudo ln -s hadoop-2.8.0 hadoop  
sudo chown -R hduser:hadoop hadoop-2.8.0  
sudo chmod 777 hadoop-2.8.0
```

Po těchto krocích je nutné editovat soubor „*hadoop-env.sh*“ definováním lokální proměnné „*JAVA_HOME*“ a zakázáním použití IPv6.

Příklad editace „*hadoop-env.sh*“:

```
sudo vi /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

Přidáním těchto řádků na konec souboru:

```
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true  
export JAVA_HOME=/usr/local/java/jdk1.8.0_131
```

Dále přidání některých souvisejících proměnných prostředí Hadoop příkazem „*vi ~/.bashrc*“ (editace opět na konci souboru):

```
export HADOOP_HOME=/usr/local/hadoop  
export HADOOP_PREFIX=/usr/local/hadoop  
export HADOOP_MAPRED_HOME=${HADOOP_HOME}  
export HADOOP_COMMON_HOME=${HADOOP_HOME}  
export HADOOP_HDFS_HOME=${HADOOP_HOME}  
export HADOOP_YARN_HOME=${HADOOP_HOME}  
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop  
export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native  
export HADOOP_OPTS="-Djava.library.path=${HADOOP_PREFIX}/lib"  
export JAVA_HOME=/usr/local/java/jdk1.8.0_131  
export  
PATH=$PATH:$HADOOP_HOME/bin:$PATH:$JAVA_HOME/bin:$HADOOP_HOME/sbin
```

Pro zaznamenání změn je nutné provést příkaz „*source ~/.bashrc*“.

7.3.5 Modifikace konfiguračních souborů

Hadoop je ve výchozím nastavení nakonfigurován pro lokální mód, následující ukázky jsou tedy pro pseudo-distribuívaný a distribuovaný mód. Konfigurace Hadoop systému se provádí prostřednictvím souborů, které jsou v xml formátu. Lokace jednotlivých souborů je individuální dle verze, nicméně od verze 2.6 jsou hlavní konfigurační soubory jednotně uloženy v adresáři: „*hadoop/etc/hadoop/*“. Všechny uvedené nastavení se vkládají mezi tagy `<configuration> ... </configuration>`. V pseudo-distribuívaném módu se adresa hostující stanice nechává jako „*localhost*“, v distribuovaném módu se ve všech případech udává buďto IP adresa nebo jméno domény stanice, která bude plnit úlohu mastera, proto je vhodné v případě distribuovaného módu editovat DNS záznamy „*vi /etc/hosts*“, přidáním názvů a adres pro master a všechny slave uzly.

core-site.xml – Konfigurace vlastností, které Hadoop používá při spouštění.

Pseudo-distribuívaný mód:

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:9000</value>
</property>
```

Dále vytvoření náležitých adresářů a udělení patřičných oprávnění:

```
sudo mkdir -p /app/hadoop/tmp
sudo chown hduser:hadoop -R /app/hadoop/tmp
sudo chmod 750 /app/hadoop/tmp
```

Distribuívaný mód (nutné změnit „*localhost*“ na adresu master instance):

```
<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:9000</value>
</property>
```

hdfs-site.xml – Defínování počtu replikací, adresu namenodu a adresu datanodu, velikosti bloku souborového systému apod.

Při pseudo-distribuívaném módu zůstává editace celá, při distribuovaném je pak blíže specifikované v tagu „*<description>*“, který element platí pro master uzel a který pro slave uzly.

```
<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>MASTER + SLAVES</description>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
  <description>POUZE MASTER</description>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
  <description>POUZE SLAVES</description>
</property>
```

Vytvoření náležitých adresářů a udělení patřičných oprávnění.

Pseudo-distribuovaný mód:

```
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
```

Distribuovaný mód pouze pro master uzel:

```
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
sudo chmod 777 /usr/local/hadoop_tmp/hdfs/namenode
```

Distribuovaný mód pouze pro slave uzly:

```
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
sudo chmod 777 /usr/local/hadoop_tmp/hdfs/datanode
```

yarn-site.xml – Konfigurace YARNU, jako např. adresa a port běhu Resource Manageru.

Pseudo-distribuovaný mód:

```
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

Distribuovaný mód (nutné změnit „localhost“ na adresu master instance):

```
<property>
  <name>yarn.nodemanager.aux-services</name>
```



```

    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>localhost:8025</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>localhost:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>localhost:8050</value>
  </property>

```

mapred-site.xml – Definování, který MapReduce framework bude použit a adresa a port běhu Job History serveru (Job History server není nutné uvádět, ale poskytuje podrobnější přehled o proběhlých procesech a pro běh některých platforem jako např. pro Pig je nezbytný).

Pro pseudo-distribuovaný mód zůstává „localhost“, pro distribuovaný mód nutné změnit adresu master instance:

```

  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>localhost:10020</value>
    <description></description>
  </property>

```

Pro distribuovaný mód je nyní nutné dát Hadoop systému informace o adresách master a slave instancí. To se provede v souborech na adrese „*hadoop/etc/hadoop/slaves*“ připsáním adres nebo názvů domén slave uzlů a na adrese „*hadoop/etc/hadoop/masters*“, připsáním adresy nebo názvem domény master instance.

Před spuštěním Hadoop démonů je nutné nejprve provést formát HDFS. Ten se provede příkazem: „*hdfs namenode -format*“ (při distribuovaném módu provádí pouze master instance.)

7.3.6 Spouštění a základní informace

Spouštěcí příkazy zadává pouze master instance. Příkazem „*start-dfs.sh*“ se spouští HDFS procesy. Na master instanci tedy spouští služby NameNode a SecondaryNameNode a na slave instancích DataNode. Příkazem „*start-yarn.sh*“ se pak spouští YARN procesy, na master uzlu to je ResourceManager a na slave uzlech NodeManager. Spuštěné služby je možné zkontrolovat příkazem „*jps*“. Dále je možné spustit příkazem „*mr-jobhistory-daemon.sh start historyserver*“ Job History Server. Příkazem „*hdfs dfsadmin -report*“ je pak možné získat informace o klastru a tedy o funkčnosti všech uzlů.

K dispozici je webové rozhraní, udávající přehled o systému:

http://localhost:50070	webové UI služby NameNode
http://localhost:8088	webové UI služby ResourceManageru
http://localhost:19888	webové UI služby JobHistory Serveru

7.4 Měření a analýza procesů a výkonu

Hadoop sám nabízí několik předpřipravených nástrojů k provedení benchmarkových testů. Mezi jeden z nejznámějších a nejuniverzálnějších patří TeraSort.

7.4.1 Konfigurace

Pro měřicí účely byly vytvořeny 4 virtuální stroje na školním serveru. Každý stroj měl tyto parametry:

- OS: Debian 8.8
- CPU: Xeon E3 dual core
- RAM: 8 GB
- HDD: 750 GB

7.4.2 TeraSort

TeraSort poskytuje možnost ověřit současně výkonnostní schopnosti MapReduce frameworku a HDFS Hadoop klastru. Skládá se ze tří navzájem se doplňujících částí:

- **TeraGen** – Generuje libovolné množství dat. Testování především zapisovacích operací.
- **TeraSort** – Třídění dat. Testování rychlosti jak čtecích, tak zapisovacích operací.

- **TeraValidate** – Ověření výsledku třídění. Testování rychlosti především čtecích operací.

TeraSort samotný se spouští příkazem:

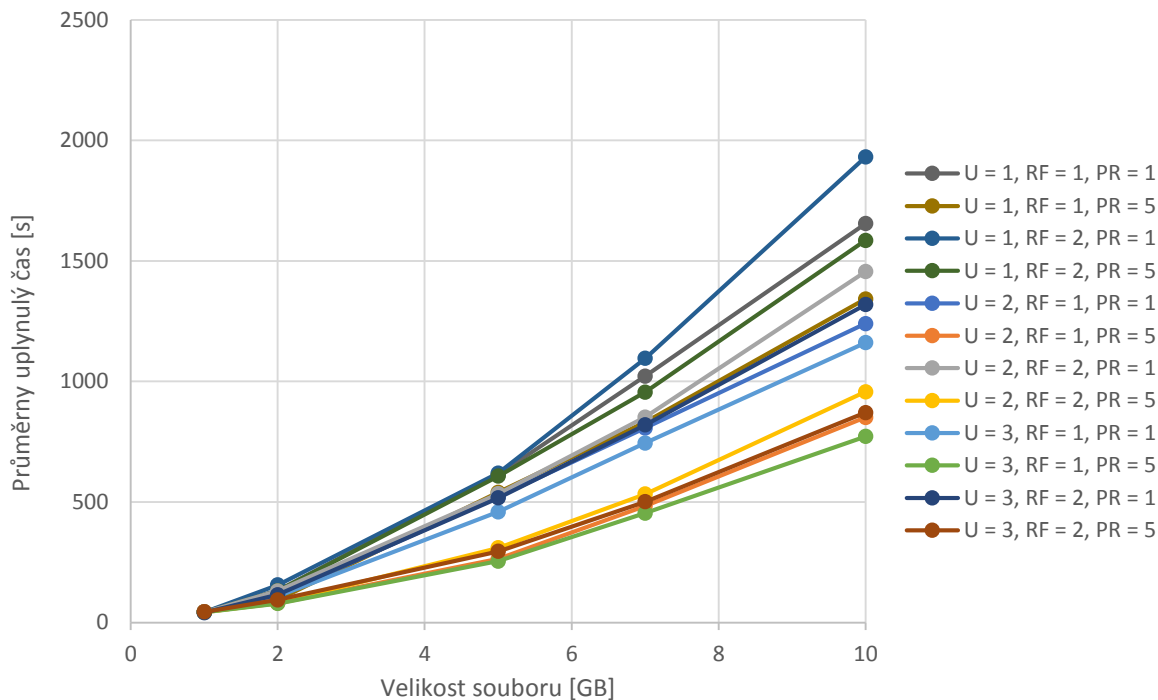
- `hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar terasort <input dir> <output dir>`

Výstupem je setříděný obsah vstupní složky, defaultně tento proces probíhá tak, že Hadoop sám si rozhodne, kolik pro tento program vynaloží mapovacích úloh, počet reduktorů je nastaven na 1. Počet uzlů by měl navyšovat výkon systému, standardně se tato procedura provádí s replikačním faktorem nastaveným na 1, pro tento test totiž není potřebné mít nastavený replikační faktor, protože ten snižuje výkon. V tabulce 7 jsou vidět výsledky měření provedené v zapojení distribuovaného módu s maximálně třemi slave uzly a vliv nastavení replikačního faktoru a počtu redukčních úloh. V rámci statistické validnosti bylo každé měření třicetkrát opakováno.

Počet uzlů	1				2				3			
Replikační faktor	1		2		1		2		1		2	
Počet reduktorů	1	5	1	5	1	5	1	5	1	5	1	5
Velikost [GB]	Průměrný uplynulý čas [s]											
1	44	45	43	43	43	44	42	43	43	42	41	44
2	133	94	156	132	111	80	131	83	109	79	115	94
5	613	540	620	607	522	264	534	310	459	255	517	295
7	1022	832	1096	956	807	484	852	533	745	454	820	501
10	1655	1342	1931	1585	1240	851	1456	957	1161	772	1319	871

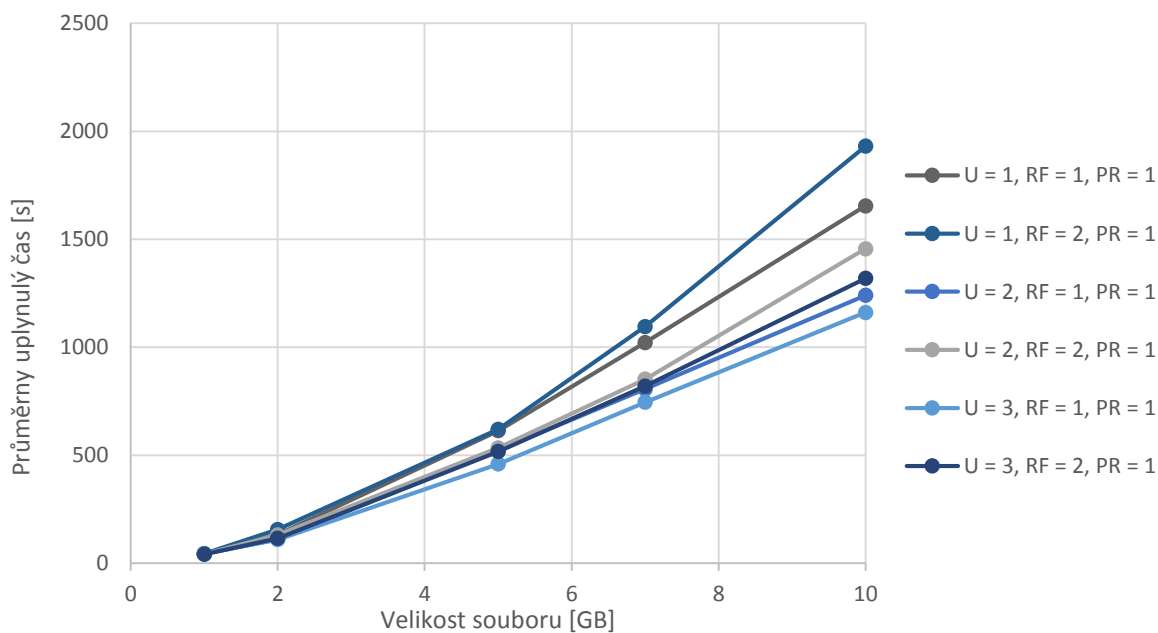
Tabulka 7 Naměřené hodnoty s různým stupněm konfigurace

Graf 2 ukazuje průměrné výsledky jednotlivých měření a vliv nastavených parametrů.



Graf 2 Vliv jednotlivých parametrů na dobu provedení programu (pozn.: U = počet uzlů, RF = replikační faktor, PR = počet reduktorů)

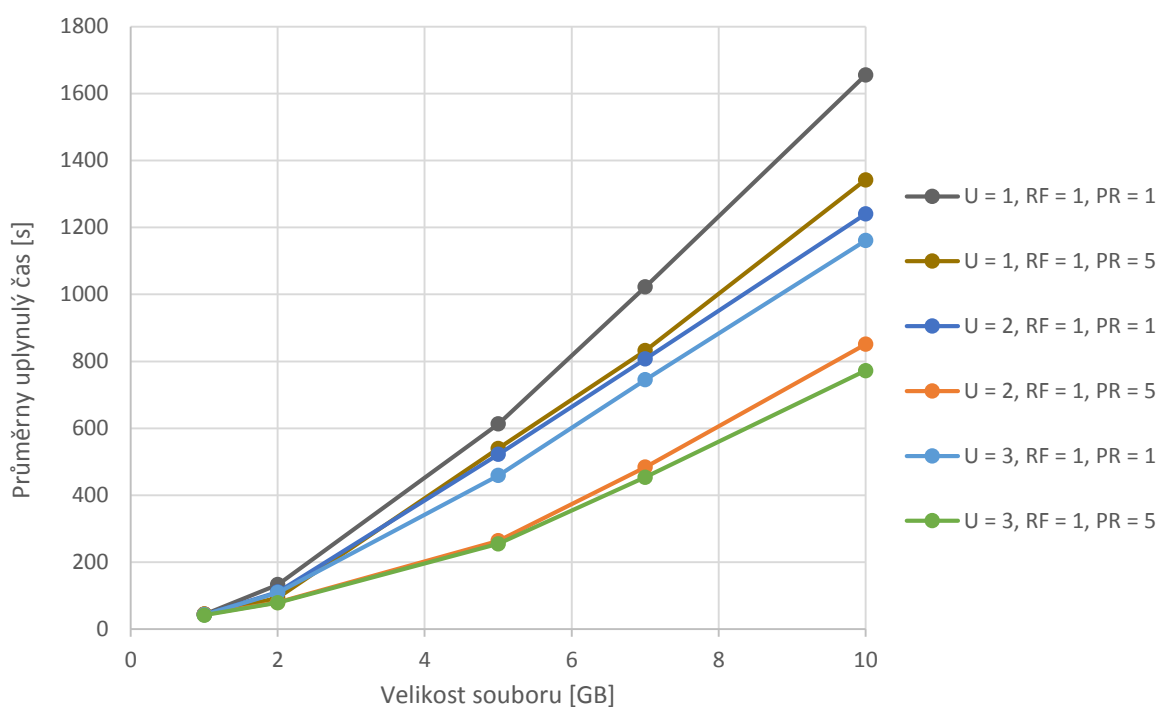
Replikační faktor se používá k zajištění bezpečnosti a dostupnosti dat, ve větších systémech standardně bývá nastaven na úroveň tři, nicméně logicky snižuje celkovou rychlost provedení programu. Během měření byl určen vliv replikačního faktoru na rychlost, z výsledků bylo zjištěno, že v každém případě došlo ke snížení rychlosti o 15 %.



Graf 3 Vliv replikačního faktoru na rychlost provedení (pozn.: U = počet uzlů, RF = replikační faktor, PR = počet reduktorů)

Z grafu 3 je dále možné vidět vliv rychlosti provedení jednotlivých měření při použití různého počtu uzlů. Při třídění datové sady o velikosti 1 GB se rychlost pohybovala na přibližně stejné hodnotě 43 s, se zvyšujícím se počtem dat se však rozdíl zvyšoval. S vyšším počtem uzlů třídění proběhlo rychleji.

Graf 4 ukazuje vliv změny počtu reduktorů. Díky zvýšení jejich počtu se snížila průměrná doba trvání redukce, což měla za následek zrychlení celkové doby trvání. Se zvyšujícím se počtem uzlů se také zvýšila účinnost, zatímco u v zapojení s jedním slave uzlem došlo ke zrychlení jen o 22 % u zapojení se třemi uzly to činilo 50 %.



Graf 4 Vliv změny počtu reduktorů s různým počtem uzlů (pozn.: U = počet uzlů, RF = replikační faktor, PR = počet reduktorů)

ZÁVĚR

Práce v teoretické části je rešeršního charakteru, poskytuje základní a klíčové informace o daném tématu. První kapitola se zabývá jak samotným vznikem koncepce Big Data, tak základním popisem, tedy definicemi, charakteristickými vlastnostmi a kde se Big Data prakticky využívají. Druhá kapitola se zabývá popisem zpracování velkých dat. Třetí kapitola obsahuje souborové systémy, databázové systémy a programovací modely, které se využívají při práci s Big Data. Poslední kapitola teoretické části se zabývá rozбором analytických technik z pohledu hloubky analýzy a z pohledu předmětu, který je analyzován.

K vypracování této části byly použity vědecké články, odborná literatura a záznamy z provedených výzkumů a přednášek, většina použitých informačních zdrojů nejsou staršího roku vydání než 2015. Z důvodu obrovského rozsahu oblastí, které toto samotné téma zahrnuje, jsou tyto kapitoly vypracovány v co nejobecnější rovině.

Praktická část zobrazuje cloud computingové služby jako jeden z prostředků řešení Big Data analýz. Rozбором nabídky služeb od předních společností v tomto sektoru jako Amazonu, Googlu a Microsoftu a následnou TOPSIS analýzou se došlo k závěru, že Google poskytuje nejvýhodnější řešení, nicméně je nutno říci, že každý produkt má svou individuální sílu v jiné oblasti. Závěrem byla představena současná verze Apache Hadoop, včetně její implementace a provedení základního benchmarkového testu TeraSort.

Jedním z přínosů této práce je poukázat na současný stav tohoto velmi rychle se rozvíjejícího oboru. Dalším přínosem je rozbor používaných technologií a nabízených služeb největších cloudových poskytovatelů v kontextu s Big Data a v poslední řadě popis a postup implementace aktuální verze Apache Hadoop technologie, včetně jednoduchých případových studií na benchmarkové sadě.

SEZNAM POUŽITÉ LITERATURY

- [1] MAYER-SCHÖNBERGER, Viktor a Kenneth CUKIER. *Big Data: Revoluce, která změní způsob, jak žijeme, pracujeme a myslíme*. Brno: Computer Press, 2014. ISBN 978-80-251-4119-9.
- [2] BUYYA, Rajkumar, Rodrigo N. CALHEIROS a Amir Vahid DASTJERDI, ed. *Big data: principles and paradigms*. Cambridge: Morgan Kaufmann, 2016. ISBN 978-0-12-805394-2.
- [3] *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things* [online]. IDC, 2014 [cit. 2017-05-18]. Dostupné z: <https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm>
- [4] BLOOR, Robin a Rebecca JOZWIAK. *THE BIG DATA INFORMATION ARCHITECTURE: An Analysis of the Consequences of the Big Data Trend*. The Bloor Group, 2014.
- [5] BOLLIER, David. *The promise and peril of Big Data*. Washington, DC: Aspen Institute, Communications and Society Program, 2010. ISBN 08-984-3516-1.
- [6] Gartner. *Big Data* [online]. [cit. 2017-05-18]. Dostupné z: <http://www.gartner.com/it-glossary/big-data>
- [7] *Big data - definition of big data in English | Oxford Dictionaries* [online]. [cit. 2017-05-18]. Dostupné z: https://en.oxforddictionaries.com/definition/big_data
- [8] HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK. *Big Data a NoSQL databáze*. Praha: Grada, 2015. Profesionál. ISBN 978-80-247-5466-6.
- [9] *Big-Data Processing Techniques and Their Challenges in Transport Domain* [online]. ResearchGate, 2016, [cit. 2017-05-18]. Dostupné z: https://www.researchgate.net/publication/275518375_Big-Data_Processing_Techniques_and_Their_Challenges_in_Transport_Domain
- [10] Industry 4.0. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-18]. Dostupné z: https://en.wikipedia.org/wiki/Industry_4.0
- [11] BRAEGER, Matthias a Manish DEVGAN. *Unlocking Big Data at CERN* [online]. Terracotta, 2014 [cit. 2017-05-18]. Dostupné z: <http://blog.terracotta.org/wp-content/uploads/2014/10/Unlocking-Big-Data-at-CERN.pdf>
- [12] Better baby care -- thanks to Formula 1. In: *TED: Ideas worth spreading* [online]. TED, 2013 [cit. 2017-05-18]. Dostupné z: https://www.ted.com/talks/peter_van_manen_how_can_formula_1_racing_help_babies
- [13] 5 Big Data Use Cases in Banking and Financial Services. *Ingram Micro Advisor* [online]. Ingram Micro Advisor, 2017 [cit. 2017-05-18]. Dostupné z: <http://www.ingrammicroadvisor.com/data-center/5-big-data-use-cases-in-banking-and-financial-services>
- [14] GANDOMI, Amir a Murtaza HAIDER. Beyond the hype: Big data concepts, methods, and analytics. In: *International Journal of Information Management* [on-

- line]. 2015, **35**(2), s. 137-144 [cit. 2017-05-18]. DOI: 10.1016/j.ijin-fomgt.2014.10.007. ISSN 02684012. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0268401214001066>
- [15] *Challenges and Opportunities with Big Data* [online]. USA: Computing Research Association, 2012 [cit. 2017-05-18]. Dostupné z: <http://cra.org/ccc/wp-content/uploads/sites/2/>
- [16] CHEN, Min, Shiwen MAO a Yunhao LIU. Big Data: A Survey. In: *Mobile Networks and Applications* [online]. Springer, 2014, **19**(2), s. 171-209 [cit. 2017-05-18]. DOI: 10.1007/s11036-013-0489-0. ISSN 1383-469x. Dostupné z: <http://link.springer.com/10.1007/s11036-013-0489-0>
- [17] RANI, I. Sudha, K. SUDHAKAR a S. Vinay KUMAR. *Distributed File Systems: A Survey* [online]. International Journal of Computer Science & Information Technologies, 2014 [cit. 2017-05-18]. ISSN 0975-9646. Dostupné z: <http://ijcsit.com/docs/Volume%205/vol5issue03/ijcsit20140503234.pdf>
- [18] GHEMAWAT, Sanjay, Howard GOBIOFF a Shun-Tak LEUNG. *The Google File System* [online]. In: . NY, USA: Google, 2003 [cit. 2017-05-18]. Dostupné z: <https://research.google.com/archive/gfs.html>
- [19] HDFS Architecture Guide. *Apache Hadoop* [online]. Apache, 2013 [cit. 2017-05-19]. Dostupné z: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
- [20] *Introducing Gluster File System* [online]. Red Hat [cit. 2017-05-19]. Dostupné z: <https://gluster.readthedocs.io/en/latest/Administrator%20Guide/GlusterFS%20Introduction/>
- [21] GlusterFS. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-19]. Dostupné z: <https://en.wikipedia.org/wiki/GlusterFS>
- [22] ZOMAYA, Albert Y. a Sherif SAKR, ed. *Handbook of big data technologies*. Switzerland: Springer, 2017. ISBN 978-3-319-49339-8.
- [23] SIVARAJAH, Uthayasankar, Muhammad Mustafa KAMAL, Zahir IRANI a Vishanth WEERAKKODY. Critical analysis of Big Data challenges and analytical methods. In: *Journal of Business Research* [online]. 2017, **70**, s. 263-286 [cit. 2017-05-19]. DOI: 10.1016/j.jbusres.2016.08.001. ISSN 01482963. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S014829631630488X>
- [24] Data science & big data analytics discovering, analyzing, visualizing and presenting data. Indianapolis: Wiley, 2015, xviii, 410 stran. ISBN 978-1-118-87613-8.
- [25] Prescriptive analytics takes analytics maturity model to a new level. *TechTarget* [online]. TechTarget, 2013 [cit. 2017-05-19]. Dostupné z: <http://searchbusinessanalytics.techtarget.com/feature/Prescriptive-analytics-takes-analytics-maturity-model-to-a-new-level>
- [26] Cloud computing. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-19]. Dostupné z: https://en.wikipedia.org/wiki/Cloud_computing
- [27] PRASAD, Y. Lakshmi Prasad. *Big data analytics: Made Easy*. Notion Press, 2016. Wiley and SAS business series. ISBN 978-1-946390-72-1.

- [28] Magic Quadrant for Cloud Infrastructure as a Service: Worldwide. *Gartner* [online]. Gartner, 2016 [cit. 2017-05-20]. Dostupné z: <https://www.gartner.com/doc/reprints?id=1-2G2O5FC&ct=150519>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

IoT	Internet věcí
XML	Rozšířený značkovací jazyk
PDF	Přenosový formát dokumentů
CSV	Čárkou oddělované hodnoty
PPT	Prezentace PowerPoint
HTML	Hypertextový značkovací jazyk
IS/ICT	Informační služby/Informační a komunikační technologie
UI	Uživatelské rozhraní
CRM	Řízení vztahů se zákazníky
ERP	Plánování podnikových zdrojů
HR	Lidské zdroje
SQL	Strukturovaný dotazovací jazyk
NoSQL	Ne jenom SQL
RDBMS	Relační databázový systém
JSON	JavaScript Object Notation
BSON	Binary JSON
IE	Vyhledávání informací
QA	Odpovídání otázek
OLAP	Online Analytical Processing
BPM	Business Process Management
SDK	Software Development Kit
GNU	GNU není UNIX

SEZNAM OBRÁZKŮ

Obrázek 1 Charakteristika Big dat (pozn.: Pro zachování integrity jednotného názvosloví jsou uvedeny anglické názvy pro označení 4V).....	14
Obrázek 2 Poměrné rozložení charakteristických vlastností podle zdroje dat (zdroj: SoftServe Inc.).....	17
Obrázek 3 Obecný průběh zpracování dat.....	20
Obrázek 4 GFS Architektura (zdroj: https://research.google.com)	25
Obrázek 5 HDFS architektura (zdroj: https://hadoop.apache.org)	27
Obrázek 6 CAP teorém	28
Obrázek 7 Databáze klíč-hodnota (zdroj: Big Data: Concepts, Methodologies, Tools, and Applications)	30
Obrázek 8 Sloupcové databáze (zdroj: Big Data: Concepts, Methodologies, Tools, and Applications)	31
Obrázek 9 Dokumentové databáze (zdroj: Big Data: Concepts, Methodologies, Tools, and Applications)	32
Obrázek 10 Grafové databáze (zdroj: Big Data: Concepts, Methodologies, Tools, and Applications)	32
Obrázek 11 MapReduce model.....	33
Obrázek 12 YARN (zdroj: hortonworks.com)	52

SEZNAM TABULEK

Tabulka 1 Přehled užívaných technologií.....	46
Tabulka 2 Kriteriační matice	48
Tabulka 3 Vážená normalizovaná matice	49
Tabulka 4 Vzdálenost od ideální varianty (Pozn.: červená značí místo výskytu maxima).....	49
Tabulka 5 Vzdálenost od bazální varianty (Pozn.: modrá značí místo výskytu minima)	50
Tabulka 6 Výsledek porovnání	50
Tabulka 7 Naměřené hodnoty s různým stupněm konfigurace	59

SEZNAM PŘÍLOH

PI CD

PŘÍLOHA P I: CD

Obsahuje bakalářskou práci s názvem fulltext.pdf