

Univerzita Pardubice

Fakulta ekonomicko-správní

Ústav systémového inženýrství a informatiky

Zpracování velkých objemů prostorových dat

Jakub Špidlen

Diplomová práce

2013

Univerzita Pardubice
Fakulta ekonomicko-správní
Akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jakub Špidlen**
Osobní číslo: **E10247**
Studijní program: **N6209 Systémové inženýrství a informatika**
Studijní obor: **Informatika ve veřejné správě**
Název tématu: **Zpracování velkých objemů prostorových dat**
Zadávací katedra: **Ústav systémového inženýrství a informatiky**

Zásady pro vypracování:

Cílem práce je navrhnout vhodný způsob využití projektu Apache Hadoop pro zpracování velkých objemů prostorových dat a navržené řešení prakticky ověřit.

Zásady:

- Prostorová data a jejich ukládání.
- Současné přístupy ke zpracování velkoobjemových prostorových dat.
- Návrh vhodného postupu využití Apache Hadoop a jeho realizace.

Rozsah grafických prací:

Rozsah pracovní zprávy: cca 55 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

GHADIRIAN, Payam. GIS-based Augmented Reality. Saarbrücken: Lambert Academic Publishing, 2009. 143 s. ISBN 978-3-8383-2227-8.


LONGLEY, Paul A., et al. Geographic information systems and science. Chichester : John Wiley & Sons, 2001. 454 s. ISBN 0-471-89275-0.

TUČEK, Ján. Geografické informační systémy : Principy a praxe. Praha: Computer Press, 1998. 424 s. ISBN 80-7226-091-X.

WADE, Tasha; SOMMER, Shelly. A to Z GIS : an illustrated dictionary of geographic information systems. Redlands: ESRI Press, 2006. 268 s. ISBN 1-58948-140-2.

WALFORD, Nigel. Geographical data: Characteristic and Sources. Chichester: John Wiley & Sons, 2002. 274 s. ISBN 0-471-97085-9.


Vedoucí diplomové práce:


doc. Ing. Jitka Komárková, Ph.D.


Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce: 1. října 2012

Termín odevzdání diplomové práce: 30. dubna 2013


doc. Ing. Renáta Myšková, Ph.D.
děkanka

L.S.


prof. Ing. Jan Čapek, CSc.
vedoucí ústavu

V Pardubicích dne 3. října 2012

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 30. 4. 2013

Jakub Špidlen

PODĚKOVÁNÍ:

Rád bych poděkoval vedoucí mé diplomové práce doc. Ing. Jitce Komárkové, Ph.D. za její odborné rady, názory a především za projevenou trpělivost. Dále bych chtěl poděkovat Ing. Oldřichovi Horákovi a Devanjanu Bhattacharya, Ph.D. za poskytnutou technickou i morální podporu při sestavování více-počítačového clusteru a při následných experimentech.

ANOTACE

Diplomová práce upozorňuje na značný nárůst dat ve světě informatiky a s tím spojené problémy při jejich zpracování. Některé metody pro zpracování velkého objemu dat představuje po teoretické stránce tato práce, z nichž největší pozornost je věnována zřejmě nejmladšímu z trendů, Cloud computingu. Pro samotné zpracování velkého objemu prostorových dat je využit systém Apache Hadoop, naimplementovaný v rámci více-počítačového clusteru, uzpůsobeného jak pro ukázkou zpracování, tak pro zátěžové testování jeho výkonnosti.

KLÍČOVÁ SLOVA

Big data, Cloud computing, Hadoop, MapReduce, velkoobjemová data

TITLE

Processing of large volume spatial data

ANNOTATION

The master's thesis highlights the exponential growth of data with occurring problems during their processing. There are introduced some of the large volume data processing methods, while the most important present phenomena is called Cloud computing. Thesis describes the utilization of multi-node cluster of Apache Hadoop system, where the large volume of spatial data is processed. Final results offer the illustration of Hadoop implementation and some performance testing presented by charts as well.

KEYWORDS

Big data, Cloud computing, Hadoop, MapReduce, large volume of data

OBSAH

Úvod	11
1 Objemná data a geografické informační systémy	12
2 Ukládání prostorových dat	15
2.1 Shapefile	15
2.2 Geodatabáze	16
3 Objemná prostorová data	18
3.1 Trendy pro zpracování velkých objemů prostorových dat	18
3.2 Cloud computing	20
3.2.1 Content delivery network	20
3.2.2 Přínosy cloud computingu	22
3.2.3 Využití cloud computingu	23
3.2.4 Cloud computing při práci s prostorovými daty	24
4 Modely služeb informačních technologií v cloudu	27
5 Cloudová řešení v praxi	29
5.1 Administrativa a finance v cloudu	29
5.2 Cloudová řešení společností Google a Microsoft	30
5.3 ArcGIS v cloudu	31
6 Hype křivka	34
7 Způsoby zpracování velkého objemu dat	36
8 Apache Hadoop systém	38
8.1 Funkce systému Hadoop	39
8.1.1 Distribuce dat	39
8.1.2 Model pro zpracování dat	40
8.2 Hadoop Distributed File System	41
8.2.1 Popis architektury	41
8.2.2 Replikace dat	43
8.3 Hadoop model MapReduce	44
8.3.1 Spojení metod Map a Reduce	46
8.3.2 Datový tok modelu MapReduce	47
8.4 Hadoop více-počítačový cluster	50
8.4.1 Architektura a potřebný software	50
8.4.2 Instalace vývojové sady Java	53

8.4.3	Vytvoření clusteru	53
8.4.4	Instalace Hadoop	54
8.4.5	Konfigurace clusteru.....	55
8.4.6	Start HDFS	58
8.4.7	Interakce s HDFS	59
8.5	Hadoop streaming	60
8.5.1	Zdrojový kód mapper.py	61
8.5.2	Zdrojový kód reducer. py	61
9	Zpracování velkého objemu prostorových dat	63
9.1	Četnost nadmořské výšky	63
9.1.1	Příprava dat.....	63
9.1.2	Zpracování dat	64
9.2	Zátěžové testy	66
9.2.1	Testování ve výchozím nastavení.....	67
9.2.2	Testování v optimálním nastavení.....	68
	Závěr.....	70
	Použitá literatura.....	71

SEZNAM TABULEK

Tabulka 1: Přehled použitých adresářů a souborů.....	52
Tabulka 2: Interakce klienta s HDFS	59
Tabulka 3: Hadoop streaming	60
Tabulka 4: Vstupní soubor 71.txt.....	63
Tabulka 5: Vstupní soubor 71_vse.txt.....	67

SEZNAM ILUSTRACÍ

Obrázek 1: Typy geodatabází v ArcGIS	16
Obrázek 2: Flow mapa - Migrace mezi Kalifornií a ostatními státy USA	18
Obrázek 3: CDN – mirror úložiště	20
Obrázek 4: Distribuční body CDN sítě CloudFront.....	21
Obrázek 5: Rámec prostorového cloud computingu	25
Obrázek 6: Schéma koncepce SaaS	28
Obrázek 7: ArcGIS Online.....	32
Obrázek 8: Hype křivka technologie cloud computing.....	35
Obrázek 9: Funkce a role Hadoop systému.....	39
Obrázek 10: Rozdělené datové vstupy na jednotlivé uzly.....	40
Obrázek 11: Schéma programovacího modelu MapReduce	40
Obrázek 12: HDFS architektura.....	41
Obrázek 13: Replikace dat na uzlech DataNodes	43
Obrázek 14: Metoda Map.....	45
Obrázek 15: Metoda Reduce	45
Obrázek 16: Nezávislá agregace s různými klíči	46
Obrázek 17: Prvky modelu MapReduce	47
Obrázek 18: Architektura vybraného řešení clusteru	51
Obrázek 19: Výpis adresáře HDFS se vstupním souborem	64
Obrázek 20: Streaming úloha.....	65
Obrázek 21: Výpis adresáře se vstupními i výstupními daty	66
Obrázek 22: Ukázka finálního výstupu.....	66
Obrázek 23: Průměrná doba zpracování vzhledem k počtu zapojených počítačů	67
Obrázek 24: Průměrná doba zpracování vzhledem k parametru mapred.reduce.tasks	68
Obrázek 25: Průměrná doba zpracování vzhledem k optimu.....	69

SEZNAM ZKRATEK A ZNAČEK

API	aplication programming interface
ArcSDE	Arc spatial database engine
BAO	business analyst online
BI	business intelligence
BISE	business and information system engineering
CDN	content delivery network
DaaS	data as a service
DNS	domain name system
FS	file system
GIS	geographical information system
HDFS	Hadoop distributed file system
IaaS	infrastructure as a service
IP	internet protocol
IT	information technology
JDK	java development kid
MS	Microsoft
PaaS	platform as a service
RDBS	relational database system
SaaS	software as a service
SDM	spatial data management
SQL	structured query language
XaaS	x as a service

ÚVOD

Pojmy jako big data, velký objem dat, rychlost jejich vzniku nebo doba zpracování, jsou v dnešní moderní době využívání elektronických dat, takřka ve všech odvětvích poskytujících produkty a služby, považovány za kritické a prioritní k zamyšlení, protože žádný malý, či středně velký podnik nebo i velká korporátní společnost by neměly exponenciální nárůst dat přehlížet, ba naopak ho akceptovat a efektivně využít ve svůj prospěch. Data, informace i znalosti jsou nedílnou součástí každého informačního systému. Aktuálním požadavkem na informační systém je centralizace dat, která snižuje nároky na jejich zabezpečení a zálohování. Zatímco se data hromadí, zálohují a centralizují, vyvstává nespočet otázek nad jejich efektivním uložením a zpracováním. Moderní přístupy vyzdvihují cloud computing jako účinnou podporu komplexního informačního systému. Lídři v oblasti informačních technologií paralelně vyvíjejí hardware i software pro nalezení vysoce výkonných řešení, mezi které patří např. NoSQL databáze nebo distribuované souborové systémy. Jedním z nejdiskutovanějších a nejdynamičtěji se rozvíjejících systémů pro zpracování velkého objemu dat je projekt Apache Hadoop.

Před zpracováním velkých objemů prostorových dat se zdá jako priorita nejprve rozeznat, za jakým účelem budou geografická data zpracována a jaké konkrétní způsoby lze pro jejich ukládání využít. Důležitým nástrojem pro koncové uživatele geografických informačních systémů je program ArcGIS, který dokáže data načíst, modifikovat a vytvářet z nich podrobné analýzy. Cloud computing se jeví jako moderní informační trend, do kterého se vyplatí investovat čas, peníze i výpočetní prostředky. Mnoha systémům, včetně ArcGISu, tak přibývá nová funkcionalita v podobě lepší synchronizace a on-line využití, které s sebou cloud computing přináší.

Projekt Apache Hadoop vyvíjí open-source software pro spolehlivé, škálovatelné a distribuované výpočty. Po zhlédnutí úvodních multimediálních prezentací byl Hadoop vybrán jako stěžejní software pro experimentální zpracování výpočetních úloh nad velkým objemem dat a jeho použití čistě vyplývá z rozšiřujícího zadání diplomové práce.

Cílem diplomové práce je implementace Hadoop více-počítačového clusteru, na kterém lze prezentovat zpracování velkoobjemových prostorových dat, v rámci dostupného vybavení. Mezi dílčí cíle patří tvorba praktického návodu pro začínající uživatele Hadoop systému, včetně teoretického popisu jeho hlavních i elementárních komponent.

1 OBJEMNÁ DATA A GEOGRAFICKÉ INFORMAČNÍ SYSTÉMY

Pro označení velkého objemu dat se stal velmi populární pojem big data. Rozumí se jím velké množství údajů, které mají být transportovány z datového skladu do smysluplné analýzy. Informační manažeři se často soustředí na problém velkého objemu dat, zatímco ztrácejí kontrolu nad přístupem a kvalifikačními aspekty. Ovšem nechat tuto otázku bez povšimnutí by v současném exponenciálně rostoucím trendu znamenalo v blízkém horizontu hrozbu finančně náročných reinvestic. Termín big data je definován pomocí tří „V“ [19]:

Volume (objem) – zvýšení datového objemu v podnikových systémech je způsobeno jednak množstvím informačních transakcí a jednak tradičními a nově vznikajícími datovými typy. Otázkou je pak vhodné řešení při ukládání dat, ale i při tvorbě analýz.

Variety (druh) – problém zpracování velkého objemu dat řeší analytici již delší dobu, další výzvy přicházejí se stále nově se objevujícími druhy dat, které zahrnují tabulkové záznamy z databází, hierarchické údaje, dokumenty, e-maily, naměřené údaje, videa, obrázky, zvukové záznamy, burzovní data, finanční transakce a další.

Velocity (rychlost) – zaprvé myšlená jak rychle data vznikají a zadruhé jak rychle by měla být zpracována pro uspokojení poptávky.

Podle [12] lze uvést čtvrté „V“ – **Value (hodnota)**, které znamená smysluplnost a užitečnost obsahu informace. Alokovat výpočetní výkon v čase a mít k dispozici metody pro okamžité analýzy a najít tak ve velkém množství dat skutečně hodnotnou informaci znamená další výzvu a rozšíření pohledu na big data o čtvrtou dimenzi.

Mezi vůdčí vědní obory zabývající se návrhem efektivních a výkonných aplikací pro zpracování velkého objemu dat patří business and information systems engineering (BISE) úzce spolupracující s oborem operations research. Síla BISE spočívá ve více kolaborujících vědních disciplínách, které tak mohou přispět k odhalení jak teoretického, tak praktického potenciálu big dat. [7]

Nové příležitosti výzkumu (teoretický potenciál) – vzrůstající nároky na datový objem, druh i rychlost, stejně jako na jejich kvalitu, kladou různé otázky pro výzkum ohledně ukládání, analyzování a zpracování dat. Kromě technologických otázek (optimalizace databáze, sémantická analýza) je nutné zaměřit se na management, optimalizaci podnikových procesů a zjišťování ekonomické hodnoty podnikových modelů operujících s big daty. V tomto směru lze BISE považovat za lídra mezi vědními obory, protože je schopen sestavit

různorodou skupinu výzkumných metod a nabízet tak odůvodněná teoretická řešení použitelná v praxi. [7]

Nové školící příležitosti (praktický potenciál) – ve srovnání s minulou dobou, kdy se podniky spoléhaly především na interní informaci, současná a budoucí podniková rozhodnutí zahrnují velký objem informací externích. To vede podniky k povolávání drahých datových expertů, anebo k zaučování mladých talentovaných studentů, kteří by mohli v budoucnu naplno využít svůj potenciál v oblasti podnikání, informačních systémů nebo počítačových technologií, obzvláště když poptávka po datových specialistech stále poroste. [7]

Big data odpovídají enormnímu objemu, druhu a rychlosti dat, přičemž efektivní přístup se může změnit v konkurenční výhodu podniku. Výzva při zpracování big dat spočívá v destilování člověku užitečné informace z velkého množství dat. Mnoho podniků akumuluje astronomický rozsah dat, který již dávno překročil alarmující meze. Podle [12] vzrostl objem veškerých digitálních dat v roce 2012 na 2,7 miliard terabytů, meziročně nárůst o 48% a podle prognóz dosáhne hranice 8 miliard terabytů v roce 2015. V současné době lze IT svět rozdělit podle přístupu společností k datům na podniky, které data ukládají a skladují, jiné data dolují pro jejich pochopení a další data agregují pro snazší interpretaci. Shromažďovat, standardizovat nebo očišťovat data od nesrovnalostí v takto velkém rozsahu předkládá podnikům další překážky, jejichž odstranění se stává prioritou. [12], [59]

Významnou roli v mnoha oblastech moderního světa hrají geografické informační systémy (GIS), kde by ztráta kontroly nad daty omezila nebo zpomalila jejich vývoj, implementaci i využití, a to by s sebou např. v případě krizového řízení neslo fatální důsledky. Zhruba 80% všech informací obsahuje prostorovou komponentu. Spousta důležitých funkcí je tak závislých na prostorových datech. Proto vznikají nové integrační služby nebo služby spojené se sdílením prostorových informací. Existuje mnoho faktorů, jako drahé náklady na software, nákladná údržba, malý hardwarový výkon nebo nízká systémová bezpečnost, které brzdí vývoj služeb pracujících s prostorovou informací. Příkladem využití GIS při práci s prostorovými daty mohou být funkce spojené s environmentálním plánováním, funkce krizového managementu nebo funkce spojené s demografií. Důležité uplatnění nachází GIS také v oblasti pojišťovnictví, na jejíž funkce se zaměřuje následující kapitola. [62]

Geografické informační systémy v ČSOB Pojišťovně

Ve světě i v České republice se člověk setkává stále častěji s přírodními katastrofami různých rozměrů, které způsobují značné škody na majetku i ztráty na životech. Vzhledem ke stoupající četnosti povodní, vichřic, zemětřesení a dalších přírodních úkazů roste zájem o pojištění proti těmto přírodním nebezpečím. Jedno z využití GIS v pojišťovnictví spočívá při analýze možného rizika v oblasti risk managementu. Slouží jako důležitý přehledný nástroj v rozhodovacím procesu o pojištění a přijetí možného rizika. [49]

Díky implementovanému systému ArcGIS lze prostřednictvím přehledových map a **prostorových analýz** získat ucelenou geografickou představu o rozložení portfolia občanského pojištění nebo podnikatelských rizik, či v případě pojištění průmyslových subjektů se vyhodnocují oblasti s největší pravděpodobnou maximální škodou. **Přehledové mapy** oblastí zasažených přírodní katastrofou využívají informaci o historii škod v místě pojištění. Uplatnění nacházejí především při upisování pojistného nebezpečí katastrofy v pojištění obcí a měst. Výsledky analýz, které zjišťují a zobrazují geografické souvislosti mezi zkoumanými prvky, se vizualizují na statických nebo dynamických mapách. [49], [61]

Jednu z nejdůležitějších rolí hraje GIS v ČSOB Pojišťovně při **vyhodnocování rizika povodně** a tím i při sjednávání smlouvy o pojištění. Riziko povodně je klíčovým kritériem pro určení sazby pojistného. Pro minimalizování škod se připravují tzv. povodňové plány, které zahrnují jak mapové podklady, tak i textové části. V dnešní době převládá digitální podoba povodňového plánu, která je připravována za pomoci nástrojů GIS pro modelování povodňových situací. Elektronické mapy připravené v GIS mají tu výhodu, že lze jejich obsah pružně upravovat dle nových skutečností. Závěrečná **analýza škod** tvoří pojistné události objektivními, předchází sporům, vysvětluje případná nedorozumění a dokonce může odhalit pojistný podvod. [14], [49], [56]

Shrnutí

Efektivní zpracování velkého množství prostorových dat klade velké nároky jak na hardware, tak na software. Existuje velké množství překážek pro rychlou manipulaci s daty v nejrůznějších praktických oborech. Zejména v pojišťovnictví je využití prostorových informací velmi názorné, protože GIS zde skýtají řadu možností a uplatnění. Mezi nejčastěji využívané funkce patří prostorové analýzy, přehledové mapy, vyhodnocování rizika a analýza škod. Předním nástrojem pro komplexní zpracování prostorových dat je systém ArcGIS. Pro práci v systému ArcGIS je nezpochybnitelnou výhodou znalost způsobů ukládání dat.

2 UKLÁDÁNÍ PROSTOROVÝCH DAT

System ArcGIS podporuje dva způsoby uložení dat. Někdy záleží na charakteru řešené úlohy, jindy na preferencích uživatele, jestli si vybere mobilnější způsob uložení ve formátu shapefile nebo naopak bude preferovat centrální úložiště v podobě geodatabáze. Oba způsoby nabízejí konkrétní výhody, mají ovšem i své nedostatky.

2.1 Shapefile

Shapefile je datový formát pro ukládání prostorových dat ve struktuře několika souborů nebo adresářů popisujících jednu tematickou vrstvu. Jedna taková vrstva podle [17], [31] sestává z několika souborů:

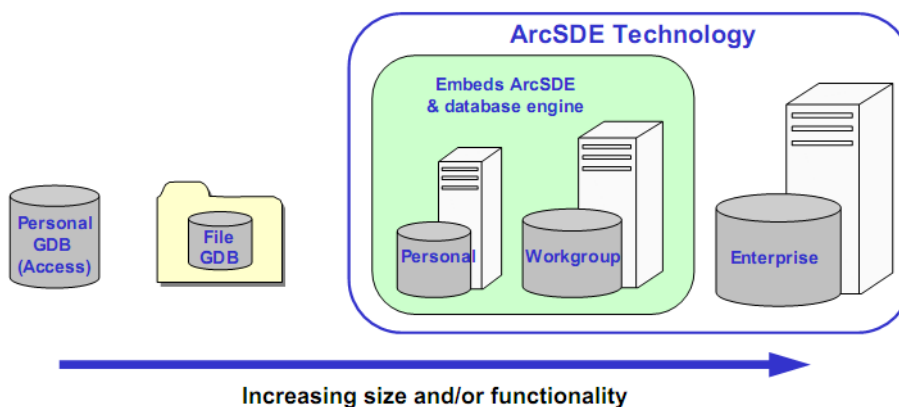
- *.shp – soubor obsahující geometrické popisy prostorových prvků obsažených ve vrstvě,
- *.dbf – soubor obsahující atributová data vztahující se k prostorovým prvkům obsažených ve vrstvě,
- *.shx – soubor obsahující uložení prostorového indexu nad geometrickými popisy prvků obsažených ve vrstvě.

Minimálně tyto tři soubory se objeví v každém správně fungujícím shapefile.

Velkou **výhodou** shapefileu oproti souborové geodatabázi je jeho snadná přenositelnost mezi software, dále jeho jednoduchost nebo přístupnost. Shapefile totiž nabízí možnost přístupu k atributům dat a jejich úpravy bez potřeby „GIS klienta“. Prostřednictvím GIS klienta uživatel pracuje s daty na svém aplikačním prostředí, ale změna dat se projevuje na pracovním počítači serveru. V podstatě stačí otevřít soubor v některém databázovém editoru Microsoft (MS) Excel nebo MS Access, kde lze zpřístupnit, číst nebo aktualizovat atributy podle libosti a změny uložit. Pro uložení většího počtu dat **je** však shapefile **nevhodný**. Stejně tak i při ukládání více datových druhů. Neumožňuje totiž práci s topologií, doménami nebo prostorově závislými atributy. Shapefile totiž obsahuje data vždy jen k jednomu geometrickému typu (bod, linie nebo polygon).[53]

2.2 Geodatabáze

Jak ukazuje Obrázek 1, v ArcGIS 10 se rozlišují tři hlavní typy geodatabází: Personal geodatabase, File geodatabase a Arc spatial database engine (SDE) geodatabase.



Obrázek 1: Typy geodatabází v ArcGIS

Zdroj: [47]

Personal geodatabase ukládá data do jediného mdb souboru (MS Access) až do velikosti 2 GB, ale v MS Windows pracuje efektivně jen při 500 MB. [47], [55]

File geodatabáze je tvořena adresářem název.gdb, který obsahuje soubory s daty. Struktura adresáře není podstatná. Výhodou je navýšení velikosti až na 1 TB a rychlosti oproti osobní geodatabázi. Obě geodatabáze oproti shapefilu umožňují práci s topologií, doménami, či prostorově závislými atributy. Za nevýhodu lze považovat horší přenositelnost dat mezi software. [47], [55]

Třetím typem geodatabáze v ArcGIS jsou **ArcSDE geodatabase**. Podle [47] se rozlišují další tři typy ArcSDE geodatabází:

1. ArcSDE Personal geodatabase,
2. ArcSDE Workgroup geodatabase,
3. ArcSDE Enterprise geodatabase.

Jejich společným rysem je využití relačních databázových systémů (RDBS) pro uložení geodat. Hlavní důvody využití RDBS pro ukládání geodat podle [47] stručně shrnují následující body:

- data (atributová i prostorová) jsou integrována v jednom datovém úložišti,
- větší možnosti při ochraně, sdílení a publikování dat,

- umožněna současná práce (editace/prohlížení) více uživatelů najednou,
- snadná tvorba záloh a replikací,
- centralizovaný přístup k datům,
- relativně snazší údržba software i hardware.

Formát dat v geodatabázi

V prostředí ArcGIS se v geodatabázích nejčastěji podle [55] objevují následující prvky:

- tabulky popisující atributy geoprvků,
- třídy prvků (Feature class),
- soubory rastrů a rastrové catalogy.

V **tabulce** se geoprvky vyskytují v jednotlivých řádcích a jejich atributy ve sloupcích. Každý sloupec je definován konkrétním datovým typem, např. číselný datový typ integer nebo znakový typ character. Na tabulky a jejich záznamy lze použít řadu vztahových funkcí a operátorů. [55]

Geoprvky ve **třídách** spadají do společné geografické oblasti. V ArcGIS jde o kolekci geoprvků se stejným typem tvarů (bod, linie, polygony), atributů a souřadnicovým systémem. [60]

Raster datová sada je vytvořena z jednoho či více samostatných rastrů. V případě využití násobného rastru, jsou tato data sloučena do jedné bezešvé datové sady. Vstupní rastry v takovém případě musí mít shodný souřadnicový systém, jednotnou velikost buňky a datový formát. Pro každou datovou sadu je vytvořen soubor s příponou img. [30]

Raster catalog obsahuje kolekci rastrů, které na sebe nemusí navazovat, mohou být uloženy v odlišných formátech a mít jiné vzájemné difference. Raster catalog je definován v geodatabázi jako tabulka. [30]

Shrnutí

Základ pro efektivní zpracování dat tvoří bezpochyby smysluplný způsob jejich uložení. Systém ArcGIS nabízí mnoho variant, jednoduchým ukládáním geoprvků zvláště, až po tvoření tříd geoprvků se společnými vlastnostmi. Dobrá orientace v jednotlivých formátech obou způsobů uložení dat předurčuje kvalitnější výstup při jejich zpracování, zejména pak velkého objemu dat.

3 OBJEMNÁ PROSTOROVÁ DATA

Na objemnost prostorových dat lze nahlížet ze dvou úhlů pohledu. Jednak je způsobena množstvím údajů v atributových tabulkách, kde se ke každému bodovému záznamu objevují, kromě jednoznačného identifikátoru, jeho souřadnice, k liniovým prvkům lze přidat nebo vypočítat délku a k polygonům rozlohu. Každý geoprvek může obsahovat nepřehledné množství atributů pro různé účely zpracování a v reálném případě několika set tisíc záznamů tvořících mapový list se velikost prostorových dat pohybuje ve stovkách MB až GB. Druhý úhel pohledu tvoří kvalita a rozměr pořizovaných snímků zemského povrchu. Zatímco velikost snímku do rodinného alba se pohybuje okolo 5–10 MB, originální velikost leteckého snímku ve formátu tiff může být až desetinásobná. Při zpracování družicových dat může činit velikost přijímaných dat až 540 MB na hodinu, denní objem všech původních dat přijatých z družice tak představuje objem kolem 14 GB. [38], [41]

3.1 Trendy pro zpracování velkých objemů prostorových dat

Jedním ze zástupců velkého objemu prostorových dat je shromažďování záznamů o pohybu lidí. Např. středně velká datová sada obsahující stovky uzlů v případě migrace obyvatel v České republice, USA nebo jiných zemích zahrnuje tisíce cest a směrů pohybu, jež chtějí být zkoumány. Vznikají ovšem daleko větší soubory v podobě přehledů a simulačních modelů nad epidemií zasaženými oblastmi. Již k tak velkému objemu dat je třeba přičíst i množství nejrůznějších proměnných v podobě různých skupin věku obyvatelstva, pohlaví, atd. Pro vizualizaci prostorových interakcí se využívá **flow mapping**. Flow mapy pracují nad datovými sadami a využívají se k zobrazení migračních pohybů nebo k zaznamenávání letecké dopravy v časových intervalech. K vizualizaci se využívají přímky nebo uměle zahnuté křivky, jak zobrazuje Obrázek 2. [15]



Obrázek 2: Flow mapa - Migrace mezi Kalifornií a ostatními státy USA

Zdroj: [15]

S popularitou GIS v mnoha organizacích roste i množství prostorových dat ukládaných v databázích nebo jiných informačních úložištích. V důsledku toho se objevují nové výzvy pro vytváření inovativních metod pro extrahování informace z velkého objemu dat. Data mining nabízí několik metod pro dolování znalostí z rozsáhlých databází nebo datových skladů. Pro rozsáhlé analýzy dat se využívají datové kostky, ve kterých jsou data uspořádána podle důležitosti jejich atributů pro konkrétní analýzu. Kostka je tvořena naměřenými hodnotami, které jsou kategorizovány do dimenzí. Dimenze jsou uspořádány do hierarchií s agregovanými daty podle zvolené rozlišovací úrovně. Ačkoliv míry ani dimenze nejsou prostorovými údaji, lze podle [32] zavést nový datový model **spatial data cube** (datová kostka pro práci s prostorovými daty), ve kterém míry i dimenze podporují jak neprostorová, tak i prostorová data. Dotaz na spatial data cube by mohl vypadat např. takto: „Ukaž evropské státy s podobnou teplotou v jarních měsících.“ Státy představují prostorové míry, protože podle jejich prostorového umístění bude zjištěna teplota. Pro sestavení datové kostky pro práci s prostorovými daty se využívají speciální techniky a algoritmy, kterými se zabývají pokročilejší data miningové metody. [6], [32], [68]

V GIS se prostorová data vytváří, aktualizují a spravují prostřednictvím spatial data management (SDM). Ten ovšem v důsledku neustálého růstu objemu prostorových dat potřebuje stále více a více místa v datových úložištích. Technologie **grid computing** nabízí další způsob, jak zvládnout současnou poptávku po úložných a výpočetních kapacitách. Grid je druh infrastruktury charakterizované sdílením zdrojů. Grid computing umožňuje sdílet kapacity a řešit problémy spoluprací v rámci distribuovaného multiorganizačního prostředí. Při práci s prostorovými daty lze grid computing uplatnit jak na víceúrovňová data v rastrové reprezentaci, tak na geometricky přesnější vektory. Všechna data jsou tradičně ukládána v geodatabázích nebo souborech formátu shapefile, vždy podle současných požadavků. Grid computing při práci s prostorovými daty nabízí využití široké škály služeb, prohlížením a dotazováním nad ekonomickými statistickými daty počínaje, a placenými službami jako upravování dat využitím superpočítače v rámci jednoho či více uzlů v grid architektuře konče. [39], [54]

Z grid computingu postupně začala vznikat nová myšlenka a posléze i nový technologický trend **cloud computing**, který nabízí organizacím masivní zvýšení kapacit datových úložišť bez velkých investic do nové architektury. Cloud computing lze označit za fenomén současných informačních technologií (IT), protože představuje základní změny ve způsobu vnímání IT, resp. ve způsobu vývoje, rozmístění a placení za IT služby. [40], [43]

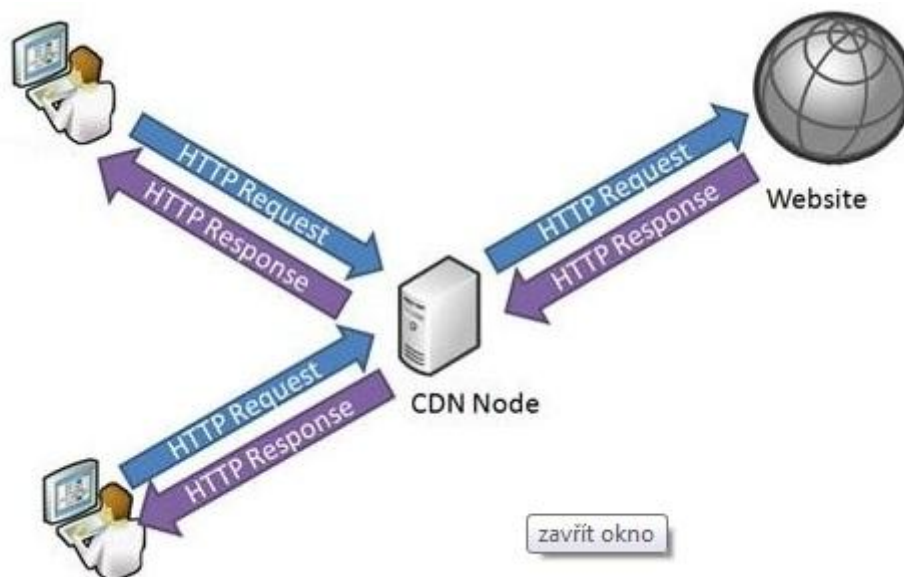
3.2 Cloud computing

Dnešní svět počítačů přináší paradox: Na jedné straně nutí konkurence jednotlivé výrobce výpočetní techniky tlačít ceny dolů, na druhé straně stále více pronikající počítačové technologie do společností zvyšující nároky na komplexní ovládání firemní infrastruktury svoji hodnotu zdražují. V tomto směru cloud computing slibuje zajištění veškeré funkcionality existujících IT a současně snížení počátečních nákladů společnosti. [40]

Předpokladem úspěšného akceptování nového IT směru je dostupnost neméně moderního vybavení, které zahrnuje superpočítače, rychlý internet, nové generace megadatových center, či vhodně zvolenou architekturu sítě. Nejhojněji využívaným typem sítě je content delivery network (CDN). [13]

3.2.1 Content delivery network

Jednoduchá CDN síť pracuje jako běžné mirror úložiště. Návštěvníci nestahují náročný obsah z původního serveru ale z geograficky nejbližšího bodu, na kterém je uložena kopie originálu. Zjednodušený model komunikace koncového uživatele a webové stránky přes CDN Node (uzel) ukazuje Obrázek 3. [1]



Obrázek 3: CDN – mirror úložiště

Zdroj: [1]

Obsahově lze podle [46] síť CDN rozdělit do tří nezávislých částí - content delivery, content routing a performance measurement.

- content delivery – zajišťuje dodávku požadovaného obsahu a jeho bezpečnost,
- content routing – sestává z technologií, které "nasměrují" uživatelův požadavek na správné místo,
- performance measurement – zajišťuje požadovaný výkon celé CDN a její měření.

Akamai je jednou z mnoha CDN sítí, které se starají o distribuci prakticky libovolného internetového materiálu. Síť Akamai, kterou pro své služby využívá např. společnost Microsoft, se stará o distribuci několika procent celkové každodenní výměny dat na globálním internetu. Slouží ji k tomu datová centra rozestá po celém světě. [13]

O distribuci výměny dat se starají desítky dalších menších sítí a to především Amazon, který vedle prodeje knížek a dalšího zboží prodává také cloud computingový superpočítačový výkon, databázový prostor a CDN síť pod názvem **CloudFront**. [13]



Obrázek 4: Distribuční body CDN sítě CloudFront

Zdroj: [13]

Obrázek 4 nastiňuje rozložení čtrnácti distribučních bodů ve městech po celém světě. Nejvíce, tedy osm, bodů se nachází v Severní Americe, čtyři body jsou v Evropě a dva v Asii.

3.2.2 Přínosy cloud computingu

Zavádění cloud computingu znamená změnu v základech obchodní a informační infrastruktury, protože končí nákupy služeb a instalování programů. Moderní doba přináší outsourcing datových úložišť, IT služeb i výpočetního výkonu. V případě využití outsourcového programu bude na příslušném počítači nainstalován pouze spouštěč, či zavaděč konkrétní aplikace pronajímáný na určitou dobu. [13], [66]

Jednou z výhod cloudu je tak snadný a ucelený přístup uživatelů ke službám, které ke své práci potřebují. Uživatelé, tedy např. zaměstnanci, zaměstnavatelé, podnikatelé a další, získají vše, co firma potřebuje - hardware, software, čas IT odborníků, údržbu, podporu, aktualizace, a to ve formě služby s pravidelným měsíčním poplatkem. Dochází tak ke snížení nákladů společností, protože hradí pouze to, co skutečně využívají – počet přesně stanovených licencí v závislosti na počtu zaměstnanců, kteří se službami pracují. [57]

Další výhodou cloud computingového řešení je revoluční změna přístupu k datům. Především společnosti, které využívají z velké části data z externích zdrojů, ocení hosting řešení, díky němuž data nemusí vlastnit, ale pouze pronajímat. Šetří tak čas i peníze. Navíc cloud computing neutralizuje problém, kdy data dorazí do společnosti v různých formátech, na různých médiích, či v různých časových intervalech. Datovému správci se tak o mnoho usnadní práce, protože se může připojit k hostované databázi kdykoli je potřeba, pracovat s pronajatými daty a jakmile práci skončí, jednoduše se odpojí. [18]

Výše pospané výhody cloudu přinášejí užitek hlavně zákazníkům. Avšak i na druhém konci trhu, tedy na straně prodejce, představuje cloud computingový model data as a service (DaaS) výhodu. Ta spočívá v centralizaci operačního prostředí a datový dodavatelé se tak mohou soustředit na rozvíjení nových distribučních kanálů, zjednodušovat cenové modely, sledovat trendy, předvídat poptávku a uspokojit tak potřeby svých klientů. [18]

Cloud computing se jeví jako důležitý moderní IT model pro přístup a distribuci výpočetních zdrojů a jeho přínos sumarizují následující body [37]:

- Dovoluje uživateli rozložit svoje výpočetní prostředky (počítače, síť, datová úložiště atd.) bez zbytečného zdržení a komplikací typicky spojenými se získáváním zdrojů.
- Identifikuje různorodé potřeby jednotlivých uživatelů ve společnosti a snáze uspokojuje poptávku po systémových zdrojích.
- Umožňuje společnosti rychle reagovat na měnící se podmínky podnikání – zaměstnanec dané společnosti nepotřebuje pokročilé znalosti v oblasti IT.

- Vyrovnává poměr mezi výdaji na zdroje a jejich aktuálním využitím – organizace platí za zdroje tehdy, když jsou opravdu potřeba.

Výše v textu byl představen cloud computing jako nový technologický směr, který se do podvědomí mnoha společností teprve zapracovává, ale současně neméně firem jeho výhod již trvale využívá. Služby cloudu lze využívat různými způsoby pro různé účely. Jak těchto služeb využít co nejefektivněji, popisuje následující kapitola.

3.2.3 Využití cloud computingu

Jedním ze způsobu využití je významná open source platforma, jejímž prostřednictvím lze vybudovat vlastní privátní cloud. Na druhé straně stojí veřejné cloudy poskytující aplikace nebo infrastrukturu formou služby. [44]

Privátní cloud poskytuje uživatelům okamžitý přístup k výpočetním zdrojům hostovaných ve firemní architektuře. Uživatelé si sami vybírají data získaná z privátního cloudu, nejčastěji skrze rozhraní webových služeb. Tímto se privátní a veřejné cloudy příliš neliší. Na druhou stranu, protože je privátní cloud umístěn v již existujícím firemním datovém centru a za firemním firewallem, jsou jeho devízou fyzická i elektronická bezpečnostní opatření, která zajišťují ochranu i pro citlivá data společnosti. [37], [51]

Pro vytvoření privátního cloudu je potřeba nejprve vyhledat vhodnou platformu. Za nejvýznamnější je podle [9] označován open source cloudový projekt OpenStack, za kterým zpočátku stály společnosti RackSpace a NASA, ale postupně se v něm začalo angažovat mnoho dalších firem, mimo jiné Citrix, Dell, AMD, Intel nebo Cisco. Jako další přední open source cloudovou alternativu, která je o několik měsíců starší než OpenStack, lze uvést projekt Eucalyptus společnosti Eucalyptus Systems.

Veřejné cloudy se uplatňují v situacích, kdy zákazník upřednostňuje výpočetní výkon před hotovými aplikacemi. Poskytují totiž pro širokou veřejnost přístup k výpočetním zdrojům skrze internet. Poskytovatel veřejného cloudu umožňuje zákazníkům přístup k datům pomocí rozhraní webových služeb, jak bylo popsáno výše u privátního cloudu. Zde ale zákazník zaplatí za pronájem přístupu k datovým zdrojům. Veřejné cloudy nabízí přístup ke skladům s různorodými datovými zdroji bez potřeby vysoké investice do infrastruktury datových center. [37], [51]

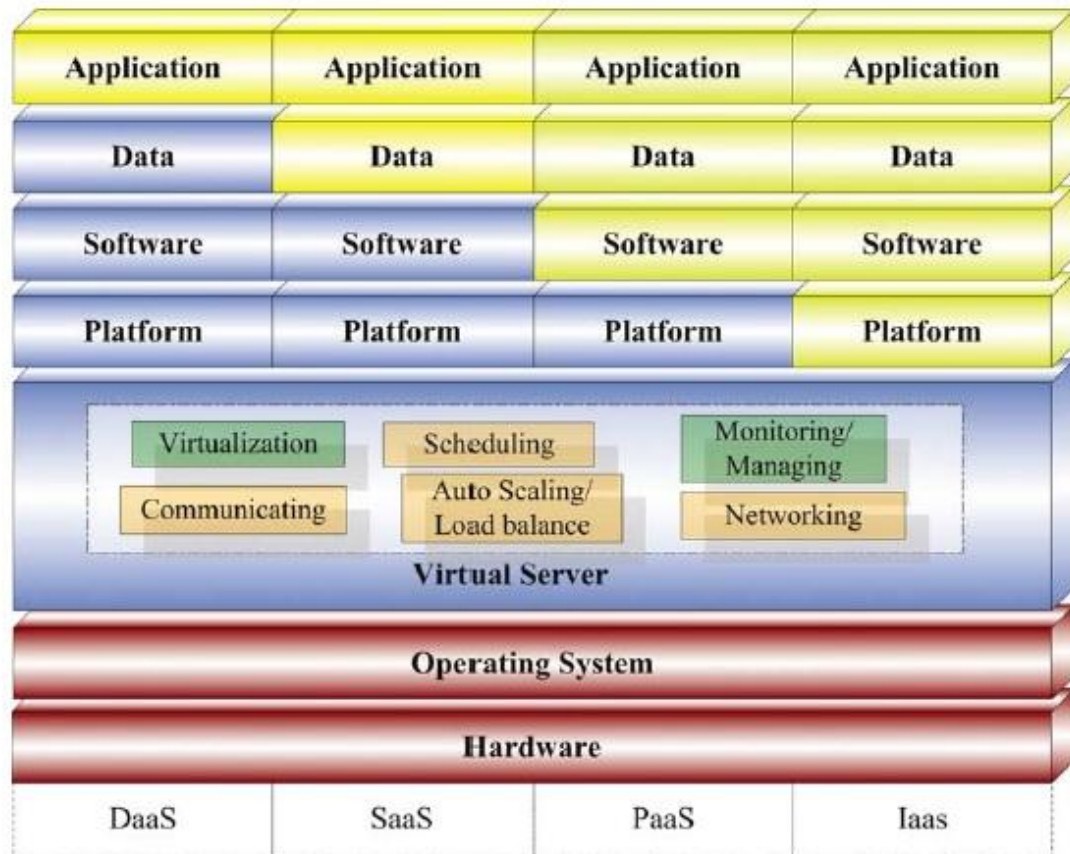
Výše popsané základní principy a znaky cloud computingu lze rovněž aplikovat na práci s prostorovými daty.

3.2.4 Cloud computing při práci s prostorovými daty

Spatial cloud computing (Cloud computing při práci s prostorovými daty) se stává další generací počítačových platforem a jeho přijetím lze dosáhnout snížení nákladů na zahájení projektů a jejich údržbu. Komerční entity jako Microsoft, Amazon a ESRI zkoumají, jak pracovat s prostorovými aplikacemi v prostředí cloud computingu a zjišťují, jak se co nejlépe adaptovat na tento nový fenomén v oblasti IT. Dřívější výzkumy objevily, že cloud computing může podpořit nejen prostorové technologie, ale může být maximálně využit s časoprostorovými principy, aby bylo co nejlépe naloženo s použitelnými výpočetními prostředky. [67]

V odkazu na výše popsané výhody, dosáhne cloud computing svého maximálního potenciálu právě při práci s prostorovými daty. Umožňuje totiž mimo jiné interakci uživatelů, resp. výměnu jejich velkoobjemových dat, aniž by musela být věnována nadměrná pozornost jejich bezpečnosti. Otázku bezpečnosti řeší samotný cloud, který je navrhnout v rámci několika spolehlivých uživatelů. [62]

Kromě zjednodušení architektury aplikací by měla mít cloud computingová platforma, podporující prostorové technologie, vliv na optimalizaci časoprostorových principů, a zapracovat prostorový cloud computing jako moderní časoprostorový trend. Lze si jej představit jako rámec, který zahrnuje fyzickou počítačovou infrastrukturu – hardware a operační systém, dále systémové vizualizační, komunikační nebo síťové zdroje rozdělené na několika místech a konečně virtuální server, který systémové zdroje ovládá na podporu cloudových služeb pro koncového uživatele. [67]



Obrázek 5: Rámec prostorového cloud computingu

Zdroj: [67]

Červená barva v Obrázku 5 vyznačuje elementární komponenty počítačového systému. Virtuální server virtualizuje základní výpočetní komponenty a poskytuje podporu pro platformu, software, data a aplikace. Mezi jeho rozšířené funkce patří [67]:

- Poskytovat fungující vizualizaci a upřednostňovat virtuální stroje nad fyzickými počítači tak, aby co nejlépe podporoval technologie cloud computingu.
- Optimalizovat síťové funkce, aby co nejnáze poskytoval a automatizoval dostupnost veřejných a privátních IP (internet protocol) adres a domén výpočetních zdrojů.
- Určit, který fyzický počítač použít, když je požadována cloudová služba.
- Automatizovat škálovatelnost a nastavit rovnováhu výpočetních instancí založených na zvolených uživatelských kritériích.
- Připojit se k veřejnému cloudu (např. Amazon EC2) a vytvořit tak hybridní cloud computing k obslužení více cloudových potřeb.

Shrnutí

Přínos cloud computingu dnešnímu světu strojů a počítačů je nezpochybnitelný. Nejen že dokáže zmírnit kapacitní nároky na celosvětová datová úložiště, ale zajišťuje šetření finančních prostředků při budování složitých počítačových infrastruktur, při vyvíjení nových aplikací a především při jejich zavádění a využívání. Cloudová řešení, která lze rozdělit podle primární myšlenky využití na privátní, veřejné nebo jejich kombinací na hybridní, stojí na architektuře CDN sítí. V mnohém se neliší od serverových řešení, vycházejí ze stejného principu poskytnutí dat mnoha klientům, ale moderní přístup k externím zdrojům činí cloud computing přinejmenším více perspektivním, především pak při práci s velkoobjemovými prostorovými daty. Cloud computing zkvalitňuje IT služby, především v oblasti modelů software as a service (SaaS) a DaaS. Ovšem bylo by chybou nezmínit další dva důležité modely, ve kterých mají technologie cloudu také svá opodstatnění.

4 MODELÝ SLUŽEB INFORMAČNÍCH TECHNOLOGIÍ V CLOUDU

Podle [11], [37], [67] lze definovat čtyři koncepce poskytovaných IT služeb, jež lze rozlišit na základě priorit uživatele cloudové služby:

IaaS (Infrastructure as a Service) – je populární službou, která prostřednictvím počítačové sítě dodává výpočetní architekturu, včetně fyzicky uloženého hardware, sítě, úložného a systémového software, jako virtuální výpočetní prostředek. IaaS umožňuje uživatelům spravovat, rozvíjet a spouštět operační systém a jeho aplikace, aniž by věděli, kde se pronajímáný hardware fyzicky nachází. Uživatelé služeb IaaS by měli mít znalosti systémového správce nutné pro plnou kontrolu nad virtuálním zdrojem. Projekt, který na této bázi běží je např. Elastic Compute Cloud od Amazonu.

PaaS (Platform as a Service) – nabízí službu, díky které mohou vývojáři využít základ aplikace k vývoji vlastního programu. Uživatelé mohou vyvíjet nebo spouštět aplikace pouze na poskytnuté platformě a nemusí brát v úvahu vlastnosti operačního systému, serverového hardware, ani výpočetní kapacity. PaaS podporuje prostřednictvím internetu kompletní životní cyklus webové aplikace od jejího vzniku až po její distribuci. Jako příklad lze uvést projekty Microsoft Azure nebo Google App Engine.

SaaS (Software as a Service) – je typ služby, která přibližuje koncovému uživateli různorodé funkce sofistikovaných aplikací prostřednictvím webového prohlížeče. Podrobněji bude tato, v cloud computingu nejpoužívanější, koncepce popsána níže.

DaaS (Data as a Service) – je typ služby, jejíž podstata je založena na poskytování dat koncovému uživateli bez ohledu na zeměpisný nebo organizační rozdíl mezi poskytovatelem a zákazníkem.

Typy služeb IaaS, PaaS, SaaS, DaaS jsou na Obrázek 5 zbarveny v závislosti na míře začlenění koncového uživatele v komponentách. Systémový správce nebo síťový architekt využívající model IaaS má kontrolu nad platformou operačního systému, software, daty i aplikacemi, jak ilustruje žlutě zbarvený pravý sloupeček, a zabývá se infrastrukturou úložných systémů, serverů nebo bezpečnostních nástrojů. Model SaaS využívají koncoví uživatelé, kteří neřeší otázku infrastruktury ani vývojové platformy (modrá), ale pronajímají od prodejců aplikace a data, se kterými dále pracují (žlutá). [67]

Software jako služba

Software jako služba (software as a service) představuje s určitou mírou nadsázky klasickou formu cloud computingu. V koncepci SaaS poskytovatel služeb hostuje aplikaci, ke které klient přistupuje na webu. Schematický princip koncepce SaaS vystihuje Obrázek 6, kde je aplikace hostovaná na vzdáleném serveru dostupná klientovi prostřednictvím internetu. Koncepci SaaS lze uplatnit na mnoha webových službách nebo aplikacích. Nejběžnějším využitím se zdají být elektronické poštovní služby, kdy poskytovatel (Microsoft – Hotmail, Google – Gmail) hostuje všechny programy a data v centrálním umístění a nabízí koncovým uživatelům přístup k datům a software přes web. Jiným příkladem využití modelu je elektronické bankovníctví. Velkou výhodou koncepce SaaS je rychlost nasazování aplikací v organizaci a tím jejich dřívější uvedení na trh, přičemž podniky nemusí programy instalovat ani spravovat hardware a to vede k výrazným finančním úsporám. Ke komplikaci může dojít tehdy, když aplikace stojí na složitých datových schématech, služba se tak rychle prodraží vzhledem k nákladům na šířku pásma a výkon sítě nutný k přenosu dat. Jedná se např. o aplikace typu business intelligence (BI), u kterých mohou být u každého zákazníka jiné požadavky na konfiguraci. [58]



Obrázek 6: Schéma koncepce SaaS

Zdroj: [58]

Shrnutí

Základní myšlenka cloud computingu tkví ve snazším a levnějším poskytování výpočetních zdrojů. Podle druhu zdroje lze vyčlenit několik modelů poskytujících zdroje jako služby – x as a service (XaaS). Z mnoha nejrůznějších modelů se cloudově využívají zejména IaaS, PaaS, DaaS a především SaaS, který díky velkému množství koncových uživatelů dosahuje největšího využití v praxi.

5 CLOUDOVÁ ŘEŠENÍ V PRAXI

Technologii cloud computingu lze uplatnit v mnoha oborech, ve kterých se orientují malé i velké společnosti. Všechny firmy si na trhu vzájemně konkurují a předhánějí se v lepších nabídkách pro své zákazníky. Cloudová řešení nabízí nejednu konkurenční výhodu. Následující kapitola popisuje několik praktických aplikací situovaných v cloudu zpracovaných zejména koncepcí SaaS.

5.1 Administrativa a finance v cloudu

Aplikace **QuickBooks** od společnosti Intuit slouží pro vedení podnikového účetnictví malých firem. Existuje několik verzí rozdělených podle nabízených funkcí, mezi které patří přístup k finančním datům odkudkoli 24 hodin denně a 7 dní v týdnu, automatizované online bankovníctví, či automatické zálohování dat na podnikových serverech. Není potřeba kupovat, vlastnit, ani instalovat žádný software, ten je hostován online a uživatelé se nemusí starat o žádnou údržbu nebo aktualizace. [58], [64]

IBM uvádí na trh řadu nabídek cloud computingu pod názvem **Blue Cloud**. Tato služba umožňuje propojení podnikových datových center s internetem. Umožní tak lepší přístup a snazší zpracování prostředků, které se právě nenacházejí v lokálních počítačích. Služba Blue Cloud je založena na otevřených standardech a softwaru open-source, který doplňují programy, systémové technologie a služby IBM. Společnost IBM vyvinula službu Blue Cloud, aby mohli klienti využívat cloud computing, včetně možnosti integrovat aplikace cloudu do stávající infrastruktury prostřednictvím webových služeb založených na architektuře SaaS. Blue Cloud se zaměřuje zejména na nová vylepšení správy IT, která poskytují zejména bezpečnost, soukromí a spolehlivost stejně jako vysoce efektivní úroveň využití. Cloud computing je určen pro existující pracovní úkoly a také pro budoucí masivně škálovatelné úkoly s intenzivním zpracováním dat. [58]

Jiný produkt od IBM, **Entry Cloud**, je dodávka cloudových řešení typu IaaS. Jedná se o dodávku cloudových hardware, software a služeb, kde cloudové prostředí je spravováno nástrojem Service Delivery Manager, jež datovému centru umožňuje urychlit vytváření platform pro široké spektrum typů pracovních zátěží s vysokým stupněm integrace a flexibility. Také automatizuje implementaci služeb IT a poskytuje monitorování prostředků a správu nákladů v prostředí cloudu. Výhodou řešení Entry Cloud je možnost využití stávající infrastruktury, čímž dojde k optimalizaci prostředků pro správu služeb. [27]

Vývoj nových technologických služeb nutí finanční instituce zajistit otevřený a pružný přístup k integraci různého elektronického finančního obsahu a služeb. Strategie **OpenChannel** od společnosti CheckFree kombinuje moderní funkce online bankovníctví a elektronických plateb s mimořádně pružným přístupem k integraci, implementaci a správě financí, které zahrnuje různá zařízení, kanály a platformy pro vylepšení uživatelského komfortu. Toto řešení může být jako nabídka SaaS plně hostováno u společnosti CheckFree nebo může být součástí hybridního modelu integrovaného do stávajícího prostředí. OpenChannel umožňuje finančním institucím využití datových skladů společnosti CheckFree i externích datových zdrojů, a zaměřit se na analýzu chování zákazníků ve specifických segmentech poptávky. [25], [58]

5.2 Cloudová řešení společností Google a Microsoft

Google Apps jsou jednoduché a intuitivní aplikace, které umožňují využívání technologických novinek, přičemž jejich cena je mnohonásobně nižší než v případě tradičního nainstalovaného řešení. Mezi služby Google Apps od společnosti Google patří velmi populární, zdarma nabízená sada aplikací jako např. webový e-mail Gmail, či sdílený správce dokumentů Google Disk. [20], [58]

Google pro vyšší úroveň zabezpečení a snadnější správu navrhnul vlastní počítače. „**Chromebooky**“ jsou vyvíjeny a optimalizovány speciálně pro Google Apps. Jejich velká výhoda spočívá ve snadné administraci, protože data aplikací jsou umístěna v cloudu, a nikoli v samotném zařízení. Informace tak zůstávají v bezpečí a jsou dostupné i v případě, že dojde ke ztrátě nebo odcizení zařízení. [26]

Google prostřednictvím služby **Cloud Connect for Microsoft Office** nabízí vylepšení populární sady Microsoft Office o možnost společných úprav více uživateli současně. Spolupracovníci spolu sdílí, zálohují a upravují dokumenty aplikací MS Word, PowerPoint, či Excel. [20]

Od společnosti Microsoft byla atraktivní, na cloudové koncepci SaaS postavená, služba **Microsoft Office Live Small Business**, která podporovala malé podniky zejména v elektronickém obchodování a distribuci reklamních nabídek, či e-mailových zpravodajů. Koncem roku 2011 byla na základě zpětné vazby od zákazníků zjištěna potřeba rozšíření a reorganizace služeb, proto Microsoft vyvinul vysoce zabezpečený, flexibilní balíček **Office 365**, který nabízí kombinaci služeb cloudové produktivity. [58]

Balíček služeb Microsoft Office 365 zajišťuje [42]:

- tradiční služby jako e-mail nebo sdílený kalendář (Exchange Online),
- synchronizaci týmů díky centrálnímu úložišti sdílených dokumentů (SharePoint Online),
- cloudovou komunikační službu na propojení lidí prostřednictvím zpráv, či audiovizuálních hovorů (Lync Online),
- webové aplikace na prohlížení a upravování dokumentů sady MS Office přímo ve webovém prohlížeči (Office Web Apps).

5.3 ArcGIS v cloudu

Cloud computing je stále velmi mladý IT směr, avšak doba počítačů čítá už desítky let. Tato nová technologie má velmi blízko k serverovým řešením, někdy jich využívá, jinde spolu obě technologie spolupracují, a naopak místy cloud computing servery zcela nahrazuje. [57]

Jako jedno z mnoha zaběhnutých serverových řešení lze uvést produkt **ArcGIS Server** od Esri. ArcGIS Server poskytuje nástroje pro správu geografických dat založených na geodatabázovém modelu ArcGIS. Poskytuje mapové služby, které slouží k vizualizaci geografických dat, dále pak vektorové či rastrové funkce nebo síťové analýzy, jež mohou být využívány v různých klientských aplikacích (desktopových, mobilních i webových). Prostřednictvím těchto aplikací lze také provádět sběr dat a jejich aktualizace. Všechny tyto funkce a služby nejsou omezeny na specializovaný software a lze je provádět prakticky odkudkoli, což je podstatou serverového řešení. Nicméně filosofie cloud computingu se příliš neliší a ArcGIS server lze provozovat i v cloudovém prostředí. [5]

Funkce, nástroje, datové vrstvy, databáze, tedy vše, co GIS analytici, ale i non-GIS uživatelé (např. business analytici), potřebují ke své práci, lze zpřístupnit kdykoli a odkudkoli. Namísto spuštění GIS na vlastní výpočetní infrastrukturu, v cloudu je požadovaný software udržován mimo, běžně u pronajímatele, který ho zpřístupňuje prostřednictvím webových služeb. Zůstávají tak zachovány základní výhody cloudu v podobě stále aktuálního software, pohodlného celodenního přístupu ke službám a snazšímu sestavení rozpočtu za poskytnuté služby, protože po rozmyšlení projektu platí uživatel pouze za to, co skutečně využívá. Společnost Esri za tímto účelem vyvinula produkt **ArcGIS Online**. [10]

Obrázek 7 zachycuje základní myšlenku přístupu libovolného oprávněného uživatele ke GIS webovým službám prostřednictvím webového prohlížeče.



Obrázek 7: ArcGIS Online

Zdroj: [10]

Produkt ArcGIS Online je software připravený k použití, aniž by byla provedena jakákoli instalace nebo detailní nastavení. Splňuje charakteristiky koncepce SaaS a poskytuje nástroje pro vytváření a publikování map nebo aplikací. Využitím centrálního úložiště a přístupu k datům v intuitivním pracovním prostředí cloudu mohou členové týmu z různých kanceláří, měst i států po definování práv a rolí snáze kooperovat a zefektivnit tak své výsledky. Díky online přístupnosti mapových služeb, které nabízí podkladové, demografické, referenční a specializované mapy, mohou uživatelé i vývojáři získat snadný přístup ke geometrické i tématické složce popisu geoprůvku, ve kterých jsou popsány jeho tvar a atributy, včetně prostorových vztahů s okolními objekty – tzv. topologie. Při integraci v MS Office má koncový uživatel možnost prezentovat různé výstupy s prostorovými daty pomocí přehledných tabulek a grafů. [10], [36]

Významným a současně nejdynamičtěji se rozvíjejícím oddělením jsou aplikace určené pro mobilní zařízení, protože svojí jednoduchostí a snadným integrováním jsou jedním z nejvhodnějších segmentů pro rozšíření GIS mezi neodbornou veřejnost. Za zmínku stojí minimálně ArcPad, ArcGIS Mobile a ArcGIS for Smartphones. ArcPad a ArcGIS Mobile jsou samostatné aplikace pro mobilní přístroje a přenosné počítače, které jsou určeny

především pro sběr a aktualizaci dat v terénu. „Chytré telefony“, fungující na bázi sofistikovaných operačních systémů, dokáží dokonce využívat mapové služby, či editovat data v terénu. [50]

Model SaaS zajišťuje prostřednictvím konkrétních aplikací a jejich nástrojů snadné komplexní řešení mnoha problémů GIS spojených s podnikáním. Jako příklad takových služeb, které Esri nabízí, lze uvést ArcLogistic nebo Business analyst online (BAO). ArcLogistic pomáhá navrhnout optimální trasu na základě vybraných kritérií. BAO nabízí webové řešení, které urychluje a hlavně usnadňuje vyhodnocení údajů o zákaznících a trhu jako takovém. BAO poskytuje detailní informace o životních stylech zákazníků, jejich zvycích a chování při nakupování. [50]

Shrnutí

Popis vybraných produktů předních IT společností se věnoval zejména koncepci SaaS, která je v oblasti zpracování velkého objemu data a zvláště v cloud computingu nejrozšířenější. Společnost, specializující se na nabízení služeb, vytváří aplikace, jejichž prostřednictvím své služby za poplatek poskytuje svým odběratelům. Někteří zákazníci ale nemusí být s nabídkou služeb docela spokojeni, proto si díky koncepcím IaaS a PaaS mohou vyvinout vlastní počítačovou infrastrukturu, vývojové prostředí nebo samotnou aplikaci. Jedním z nejkompexnějších a nejhojněji využívaných systémů pro práci s prostorovými daty je bezpochyby systém ArcGIS od společnosti Esri, který svými aplikacemi a nástroji umožňuje analýzu, editaci a správu prostorových dat. Ve spojení s technologií cloud computingu ještě obohacuje svůj přínos.

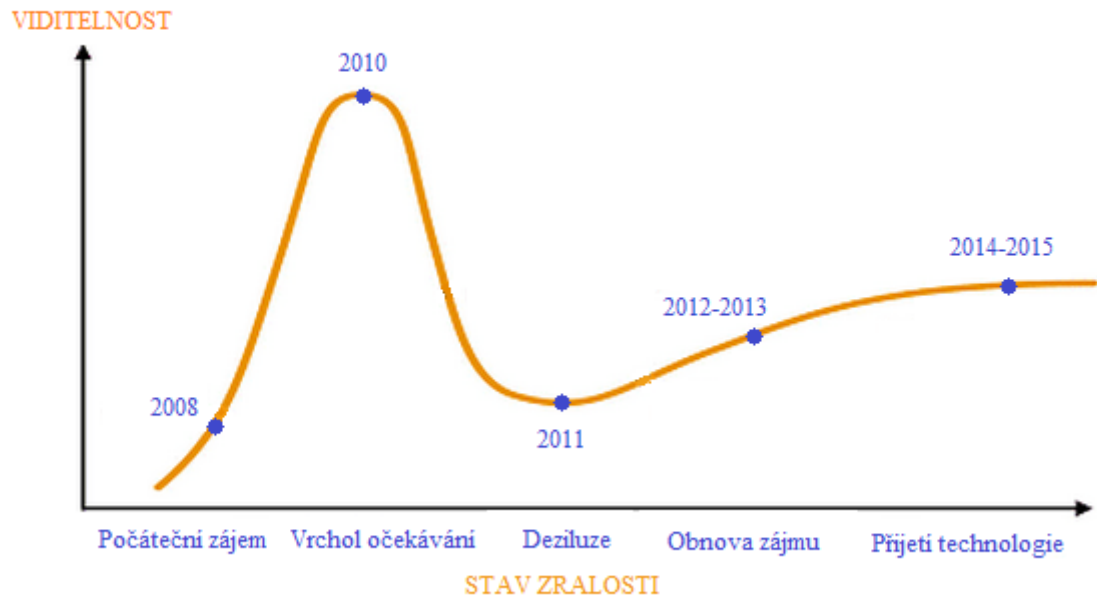
6 HYPE KŘIVKA

Analytická společnost Gartner zavedla do informační teorie graf, jež hodnotí vyspělost konkrétních IT uvedených na informačním trhu, popř. jejich připravenost na implementaci. Jednotlivé body zralosti spojuje hype křivka („křivka humbuku“). Hype křivka tak ukazuje grafickou reprezentaci stavu zralosti, přijetí technologie a její potenciál řešit konkrétní úlohy. Grafické vyjádření hype křivky rozděluje technologii do pěti fází vývoje [24]:

1. **Počáteční zájem** – média nadšeně popularizují novou technologii a vkládají do ní občas až přehnaně velký potenciál, přitom technologie zatím většinou nemá žádné praktické využití.
2. **Vrchol očekávání** – časná publicita technologie a opěvování očekávaných pozitivních výsledků je doprovázena kritickými ohlasy; naděje a očekávání začínají klesat.
3. **Deziluze** – zájmu ubývá, mnoho experimentů a pokusů o implementaci nové technologie se na první pokus nezdaří.
4. **Pozitivní sklon osvětlení** – pochopení technologie více do hloubky vede k nalezení jejích výhod. Objevují se další generace produktů od poskytovatelů technologických služeb.
5. **Plošina produktivity** – technologie je plně přijata a využívána.

Sledované technologie v různých fázích svého vývoje vykazují určitou míru „viditelnosti“, a to nejen ve smyslu viditelnosti v médiích, ale především ve smyslu vkládaných očekávání a nadějí. Tato závislost „viditelnosti“ na „stavu zralosti“ je prakticky nezávislá na tom, o jakou konkrétní technologii jde. [48]

Obrázek 8 zaznamenává aplikaci hype křivky na technologii cloud computing.



Obrázek 8: Hype křivka technologie cloud computing

Zdroj: upraveno podle[24]

Shrnutí

Na konferenci Business & Information Fórum 2011, věnované informačním a komunikačním technologiím jako nástroji pro úspěšné podnikání, bylo hned několika řečníky zmíněno, že po prvotním zájmu se nyní cloud computing nachází na hype křivce v nejnižším bodu, totiž v údolí Deziluze, a nyní ho čeká pozvolný vzestup než se technologie cloud computingu stane užitečnou a běžně užívanou součástí společností využívajících IT služby. [34]

7 ZPŮSOBY ZPRACOVÁNÍ VELKÉHO OBJEMU DAT

Pojem big data existuje již přes deset let. Jeho první definování se vztahuje k roku 2001, ale dnes se pod tímto termínem neskrývá jen velké množství dat, ale i jejich různorodost, rychlost, s jakou jsou generována, a v neposlední řadě také hodnotnost. Zpracovat a analyzovat data, ze kterých lze lépe využít informace pro byznys představuje pro podniky velikou konkurenční výhodu. Od roku 2009 se big data skloňují stále častěji a vyvíjejí se nové metody, databáze a technologie pro zpracování velkého objemu dat. [8]

Technologie **high-performance analytic appliance** (HANA) od společnosti SAP je platforma rozložitelná v cloudu, která je vhodná pro tvorbu real-time analýz a vývoj real-time aplikací nad velkým objemem operačních a transakčních dat. Replikační a synchronizační server zajišťuje paralelní in-memory zpracování, kdy data jsou zavedena přímo v operační paměti místo na pevném disku a přístup k nim je mnohonásobně rychlejší než při standardních přístupech. HANA podporuje různé technologické standardy, mezi které patří i structured query language (SQL). [52], [65]

Apache Hadoop, open-source softwarový rámec odvozený z Google MapReduce a Google File System, byl vyvinut za účelem efektivního zpracování velkého objemu dat. Hlavní podstata spočívá v rozdělení zadaných úloh na drobnější úkoly, které se v několika počítačovém clusteru zpracovávají paralelně a na závěr vybranou funkcí agregují požadovaný výstup. Jedná se o vylepšení již zaběhnuté technologie grid computingu s novým kabátem pod názvem cloud computing. Jako příklad poslouží společnost s třiceti miliony zákazníků, která doluje údaje z datových skladů, reportů a záznamů call center pro vylepšení svých služeb. Operace, která trvala týden, zabere použitím nové techniky jen pár hodin. V Hadoop rámci jsou zabudované další softwarové komponenty pro práci s velkým objemem dat. [59]

Apache Cassandra je masivně škálovatelná NoSQL databáze navržená tak, aby zvládla práci s velkým objemem real-time dat na několika datových centrech s žádným kritickým bodem pro selhání celého systému a poskytovala podnikům vysoký databázový výkon a neustálou dostupnost. Hlavním komerčním distributorem software, technické podpory a školení je společnost DataStax. Cassandra na rozdíl od Hadoop nepracuje jako key – value (klíč – hodnota) úložiště, nýbrž jako column – family (sloupec – rodina) úložiště, kde se ukládaný objekt skládá z trojice: název, klíč, časový otisk. V praxi to znamená upřednostnění využití projektu Cassandra na úkor Hadoop při zpracování dat z naměřených časově podložených údajů nebo bezpečnostních logů, kde čas hraje významnou roli. Není vyloučena

spolupráce obou projektů v rámci jednoho clusteru, přičemž časově založené (real-time) aplikace spouští Cassandra a dávkově založené analýzy a dotazy zpracuje Hadoop. [23], [63]

Mezi účinné způsoby zpracování velkých objemů dat lze zařadit i dokumentově orientovanou NoSQL databáze **MongoDB**, v několika aspektech koncepčně blízkou relačním databázím, poskytující vysoký výkon, dostupnost a snadnou škálovatelnost. Celé rozestavení hostuje mnoho databází, databáze drží řadu kolekcí, kolekce pak několik dokumentů a dokumenty jsou množiny dvojic klíč – hodnota. MongoDB používá vlastní dotazovací jazyk podobný SQL. Pro hromadné operace nad daty má k dispozici implementaci návrhového vzoru MapReduce. MongoDB pro provozování velkých nebo zatížených aplikací využívá vlastnosti sharding a replication. Sharding distribuuje jeden logický databázový systém přes více strojů v clusteru. Dokumenty jsou rozdělovány podle specifického „shard“ klíče. Replikace databáze zajišťuje redundanci, zálohování a automatické překonávání chyb. Při úvahách nad architekturou nové aplikace by tak NoSQL princip neměl být vynechán, zvláště při ukládání velkého objemu nestrukturovaných dat. [29], [33]

Shrnutí

Mnoho metod pro zpracování velkého objemu dat se vymýšlí a realizuje, ale jen pár z nich se po implementaci v praxi skutečně osvědčí. Jestli zvolit NoSQL databázi nebo distribuovaný souborový systém, využít technologie cloud computingu či nikoliv, závisí na prioritách samotných společností. Nejen IT společnosti, ale i firmy podnikající v telekomunikacích, dopravě či strojírenství, se exponenciálnímu nárůstu dat nevyhnou, a které ho nepřijmou a nezačnou ho v blízké době efektivně řešit, mohou se dostat do velkých nesnází.

8 APACHE HADOOP SYSTÉM

Hadoop je rozsáhlá, distribuovaná a dávkově zpracovávající infrastruktura. I když může být používán na jednom počítači, jeho pravá síla spočívá ve schopnosti škálovat data na stovky či tisíce počítačů, každý s několika procesorovými jádry. První plně funkční verze se na trhu objevily na konci roku 2007. Od samého počátku vývoje Hadoop systému až do přítomnosti bylo vydáno téměř 50 releasů, které postupně obohacují a doplňují jeho funkcionalitu. [21]

Hadoop je primárně navržen pro linuxové distribuce tak, aby efektivně krátkodobě zpracoval údaje o řádu stovek gigabytů, terabytů až petabytů a distribuoval velké množství práce v celé řadě strojů. Hadoop proto disponuje distribuovaným souborovým systémem, který rozdělí objemná vstupní data a odesílá podstatně menší části (bloky) originálních dat do několika počítačů v clusteru (skupině počítačů), kde pak dochází k paralelnímu zpracování přijatých dat a tím i k maximálně efektivnímu výpočtu zadané úlohy. [21]

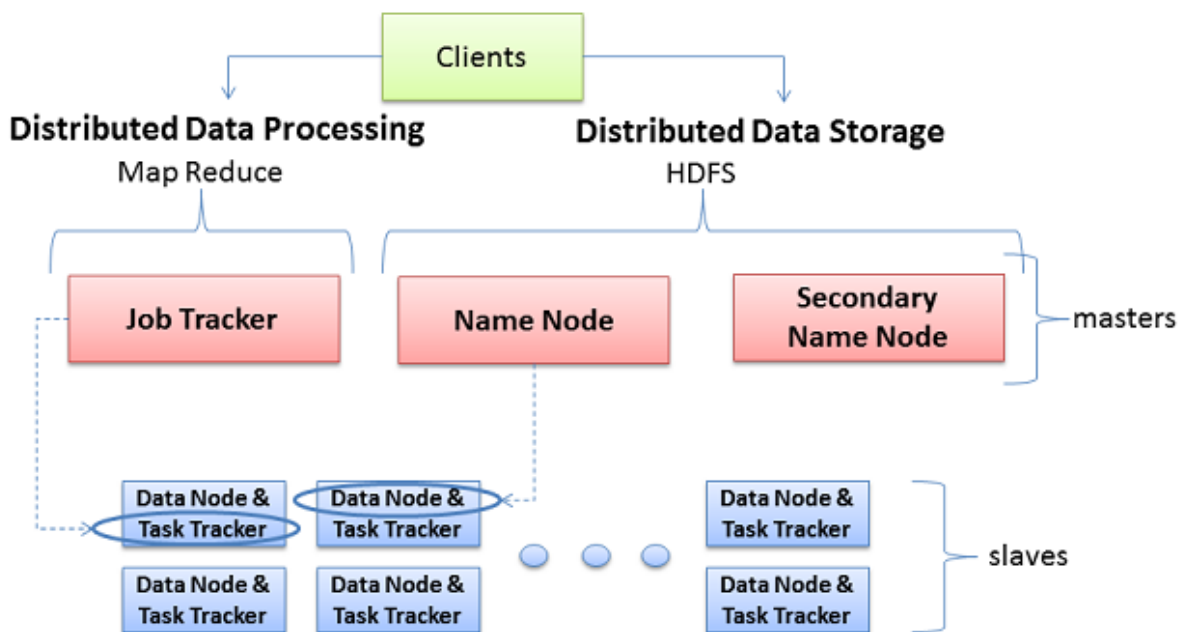
Se zvyšujícím se počtem počítačů ovšem roste pravděpodobnost výskytu chyb. Příkladem překážek, které se mohou v distribuovaném prostředí objevit, mohou být různé verze klientského softwaru nebo dočasná ztráta připojení k síti několika transakcí. Kromě obav z možných chyb a tedy akceptování několika druhů problémů je potřeba zajistit dostatečné výpočetní zdroje. Především se jedná o procesor, paměť, pevný disk a šířku pásma sítě. I přesto, že distribuovaný souborový systém s sebou nese množství rizik a výzev ke zdokonalení efektivního využití, lze jeho přínos demonstrovat na jednoduchém příkladu. Při současných výpočetních a technologických kapacitách čte běžný pevný disk data rychlostí 100 MB/s. Procesor se čtyřmi jádry poskytuje teoreticky rychlost čtení 400 MB/s. Data o kapacitě 4 TB by tak mohla být načtena za 10 000 s, tedy přibližně za 3 hodiny. 100 samostatných strojů s polovičním výkonem by stejné množství dat svedlo načíst přibližně za 3 minuty. Hadoop je tedy navržen tak, aby efektivně zpracoval velké množství informací vytvořením paralelně pracující komunity počítačů, i vzhledem k tomu, že hypotetický počítač s tisícem jader by byl mnohonásobně dražší než tisíc počítačů s jedním jádrem. [3]

Výhody Hadoop oproti ostatním metodám [3]:

- zjednodušený programovací model – umožní rychle psát a testovat distribuované systémy,
- automatická distribuce dat – efektivně využívá základních podobností procesorových jader.

8.1 Funkce systému Hadoop

System Hadoop se primárně skládá ze dvou základních celků. Pro ukládání velkého objemu dat používá **distribuovanou souborovou databázi** a k paralelnímu spouštění výpočetních úloh nad uloženými daty slouží **programový rámec MapReduce**. Na správu a ukládání dat do databáze dohlíží master uzel NameNode, zatímco daemon JobTracker má na starosti zpracování dat. Jejich příkazy vykonávají podřízené slave uzly DataNodes pro NameNode a podřízené slave daemons TaskTrackers pro JobTracker, jak ilustruje Obrázek 9. Hadoop preferuje rychlost zpracování, ale také velký stupeň odolnosti proti chybám. „Úlohové“ uzly Tasktrackers jsou v neustálé komunikaci s hlavním uzlem JobTracker, který má přehled o rozdělení úkolů a také pravomoci k případnému restartu uzlu nebo k vyřazení, v případě jeho selhání. [22]



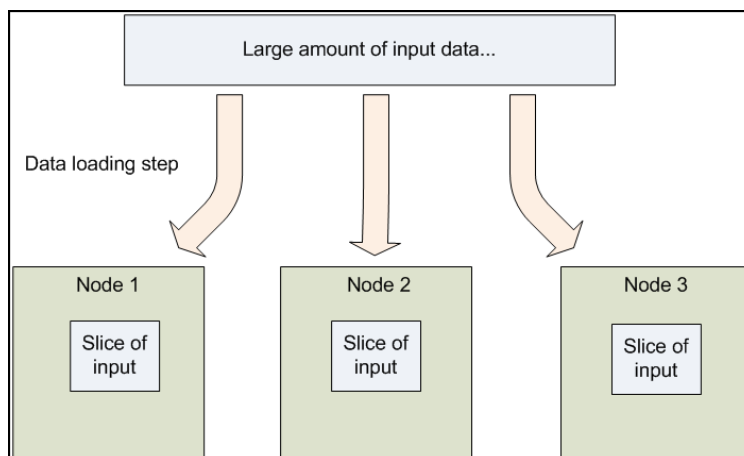
Obrázek 9: Funkce a role Hadoop systému

Zdroj: [22]

8.1.1 Distribuce dat

Distribuovaná souborová databáze – Hadoop distributed file system (HDFS) rozděluje velký objem dat do několika bloků, které jsou řízeny různými uzly v clusteru. Přitom každý blok je replikován na více strojích, takže selhání jednoho z nich nezpůsobí ztrátu přístupu k datům. Soubory ze jmenného prostoru jsou rozděleny do specifických formátů pro vybranou aplikační logiku a každý proces běžící na uzlu v clusteru pak zpracovává podмноžinu těchto záznamů, a to vzhledem ke své lokalitě k uzlu. Vhodná datová lokalita zmírňuje napětí

na šířku pásma, zabraňuje zbytečným síťovým přenosům a zvyšuje tak samotný výkon při zpracování velkobjemových dat. Schéma rozdělení datového souboru na menší části vystihuje Obrázek 10. [3]

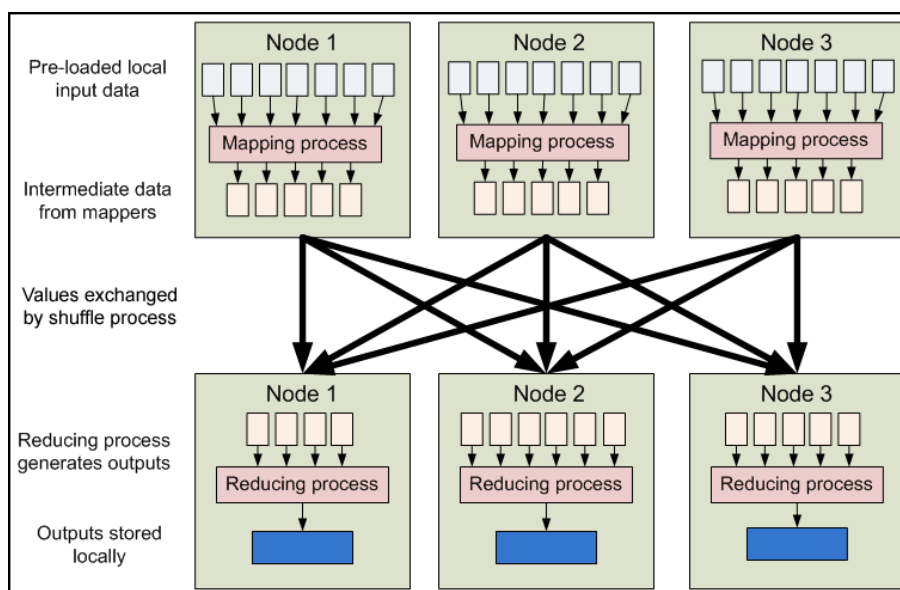


Obrázek 10: Rozdělené datové vstupy na jednotlivé uzly

Zdroj: [3]

8.1.2 Model pro zpracování dat

Aby výpočet mohl být spuštěn a distribuován přes cluster, musí vyhovovat konkrétnímu programovacímu modelu s názvem MapReduce. Záznamy jsou pro větší spolehlivost zpracovávány samostatně v izolaci v úlohách, tzv. Mappers. Jednotlivé výstupy z Mappers slouží dále jako vstupy do druhé sady úloh s názvem Reducers, které výsledky agregují. Celý proces názorně nastiňuje Obrázek 11, avšak podrobněji budou obě funkce popsány níže. [3]



Obrázek 11: Schéma programovacího modelu MapReduce

Zdroj: [3]

8.2 Hadoop Distributed File System

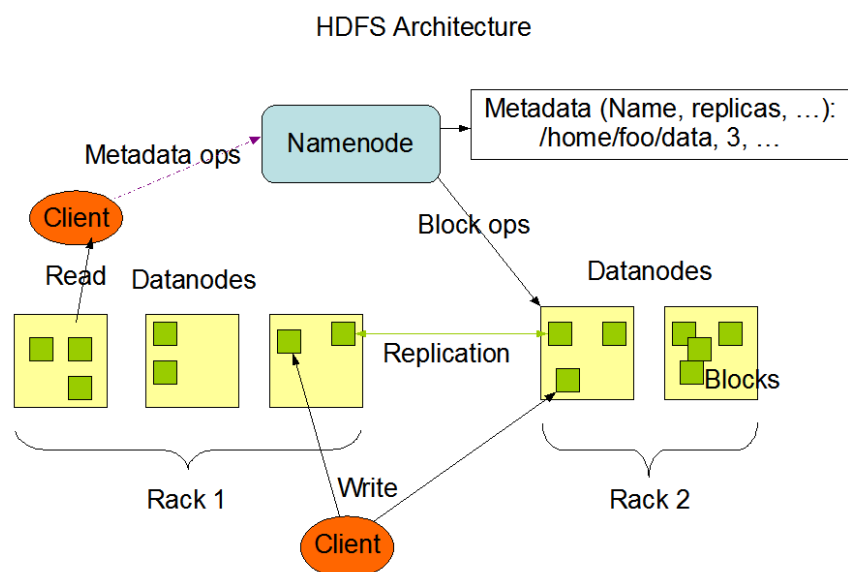
HDFS je distribuovaný souborový systém navržený k držení velmi velkého objemu dat, který svojí vysokou propustností poskytuje přístup klientovi ke všem uloženým informacím. Soubory jsou uloženy redundantním způsobem, aby byla zvýšena odolnost proti chybám a umožněna práce paralelně spuštěným aplikacím. [21]

HDFS je založen na konstrukci Google file system (GFS) a je navržen tak, aby byl odolný k co nejvíce problémům. Vyznačuje se zejména těmito vlastnostmi [3]:

- poskytnutí rychlého, stálého a škálovatelného přístupu k velkému množství informací,
- šíření dat v rámci velkého počtu strojů (uzlů) začleněných do jednotlivých clusterů,
- cluster odolný kompletnímu selhání několika strojů, aniž by ztratil informaci nebo výrazně zpomalil výkon.

8.2.1 Popis architektury

HDFS funguje na bázi master/slave architektury. Cluster HDFS se skládá z jednoho master uzlu NameNode, který spravuje systém souborů a reguluje přístup klienta k souborům. Kromě toho se v clusteru nachází množství slave uzlů DataNodes, které jsou zodpovědné za vyřizování žádostí klientů. Uzel NameNode provádí operace se soubory, jako otevření, zavření, a přejmenování souborů a adresářů, a v neposlední řadě také určuje mapování bloků na uzlech DataNodes, na kterých lze provádět blokovou tvorbu souboru, mazání dat a replikace dat. Nejdůležitější členy architektury a jejich vazby znázorňuje Obrázek 12. [21]



Obrázek 12: HDFS architektura

Zdroj: [21]

HDFS je blokově strukturovaný souborový systém. Jednotlivé soubory jsou rozděleny do bloků pevné velikosti a jsou uloženy na jednotlivých počítačích v clusterech. Tyto počítače jsou označovány jako DataNodes. Bloky tvořící jeden soubor by se neměly nacházet na jednom stroji, proto opětovný přístup k datům může vyžadovat spolupráci několika strojů. [21]

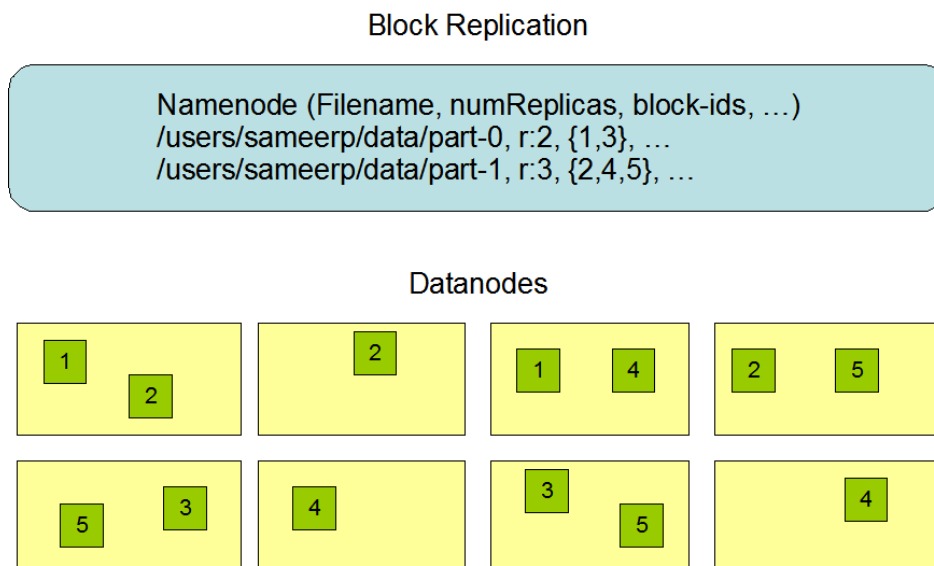
Příkaz `ls` v příkazovém řádku slouží k vypsání obsahu adresáře. Psaní `ls` na počítači, který je zároveň jeden z uzlů DataNodes, zobrazí obsah běžného souboru v linuxovém systému používaného jako hostitele pro Hadoop služby. Výpis ale nebude obsahovat žádný ze souborů uložených uvnitř HDFS, protože HDFS běží v samostatném jmenném prostoru, izolovaném od lokálních souborů. Bloky, na které jsou vstupní soubory rozděleny, jsou uvnitř HDFS uloženy v konkrétním adresáři, který je spravován DataNode službami pomocí blokových ID. Prostřednictvím běžných linuxových příkazů (`ls` – zobraz obsah, `cp` - kopíruj, `mv` – vyjmi/přejmenuj, atd.) tedy nelze komunikovat se soubory uloženými v HDFS, ta nabízí vlastní nástroje pro správu souborů, které se ovšem chovají velmi podobně. [3]

Pro rozsáhlý souborový systém je velmi důležité spolehlivé ukládání metadat. Veškerou synchronizaci informací, které mohou být využívány současně několika klienty, obstarává jen jeden stroj, master uzel NameNode. NameNode ukládá všechna metadata pro systém souborů. Mezi uchovávaná „data o datech“ patří názvy souborů, oprávnění a umístění každého souboru rozděleného na bloky. Všechny tyto informace jsou uloženy v hlavní paměti přístroje NameNode, umožňující rychlý přístup k metadatům. [22]

Pro samotné otevření souboru uloženého v HDFS musí klient nejprve kontaktovat uzel NameNode pro načtení seznamu lokalit jednotlivých uzlů DataNodes, které drží všechny bloky tvořící požadovaný soubor. Čtení dat tedy probíhá přímo z uzlů DataNodes, režijní uzel NameNode je do hromadného přenosu dat zapojen minimálně. Z uvedeného vyplývá, že selhání uzlu NameNode je pro databázi daleko závažnější než selhání některého z uzlů DataNodes. Proto se v Hadoop architektuře vyskytuje další role zvaná Secondary NameNode. Ta zajišťuje vysokou měrou podporu zálohování pro master uzel NameNode. Secondary NameNode se při výchozím nastavení jednou za hodinu připojí k NameNode uzlu a vytvoří aktuální kopie dat a metadat, která NameNode právě využívá. V případě selhání master uzlu NameNode, využije Hadoop cluster zálohovaná data na Secondary NameNode pro jeho znovuoobnovení. [3], [22]

8.2.2 Replikace dat

Aby Hadoop předešel problémům chybného načtení souboru vzniklým selháním strojů s několika bloky, předem replikuje každý blok v celé řadě strojů. Obrázek 13 přibližuje situaci, jak DataNodes drží bloky více souborů s replikačním faktorem 2 a 3. [21]



Obrázek 13: Replikace dat na uzlech DataNodes

Zdroj: [21]

Bloky s ID 1 a 3 jsou replikovány dvakrát, bloky s ID 2, 4, 5 mají každý tři repliky.

Jeden blok v HDFS dosahuje ve výchozím nastavení maximálně velikosti 64 MB. Všechny bloky v souboru kromě posledního bloku jsou stejné velikosti a je-li to možné, každý blok je umístěn na jiném z uzlů DataNodes. Očekává se tedy ukládání a čtení malého počtu velkých souborů. Při zahájení komunikace nejprve klient ukládá data do dočasného lokálního souboru. Jakmile jeho velikost dosáhne 64MB, dojde ke kontaktování uzlu NameNode, který vytvoří soubor v hierarchické struktuře s novým názvem na jednom z uzlů DataNodes a přiřadí mu data z dočasného souboru. Za předpokladu, že HDFS má nastaven replikační faktor tři, klient načte seznam uzlů DataNodes prostřednictvím master uzlu NameNode a tento seznam obsahuje uzly DataNodes, které budou hostit repliku tohoto bloku. První DataNode začne přijímat data po malých dávkách (4 kB), zapíše každou část na své lokální úložiště a současně ji přepošle na druhý DataNode v pořadí. Druhý a třetí DataNodes se chovají stejně s tím rozdílem, že třetí jen zapíše data po malých částech na své lokální úložiště. [3]

Optimalizací umístění replik lze dosáhnout zvýšení spolehlivosti dat, zlepšení jejich dostupnosti a zdokonalit využití šířky pásma sítě. Při použití funkce na umístění replik

v HDFS je potřeba mnoho testování a neméně zkušeností. Pro minimalizování globální spotřeby šířky pásma sítě se HDFS snaží uspokojit požadavek na čtení z repliky, která je nejbližší ke čtečce. [22]

Součástí architektury jsou nejrůznější protokoly a reporty ohlašující stav nebo nastalou změnu v HDFS. Mezi nejdůležitější patří [21]:

- Protokol EditLog – NameNode trvale zaznamená každou změnu souborových metadat (např. vytvoření nového souboru, změna faktoru replikace, atd.).
- Soubor FsImage – Obsahuje celý namespace, včetně mapování bloků na soubory a vlastnosti souborového systému.

Když uzel NameNode nastartuje, přečte soubor FsImage a protokol EditLog z disku. Použije všechny transakce z protokolu EditLog na „in-memory“ reprezentaci souboru FsImage místo toho, aby využíval paměť disku. Aktualizovanou verzi souboru FsImage pak odešle na disk. Tento proces se nazývá kontrolní bod (Checkpoint). Několik reportů doplňuje kontrolní funkci HDFS [21]:

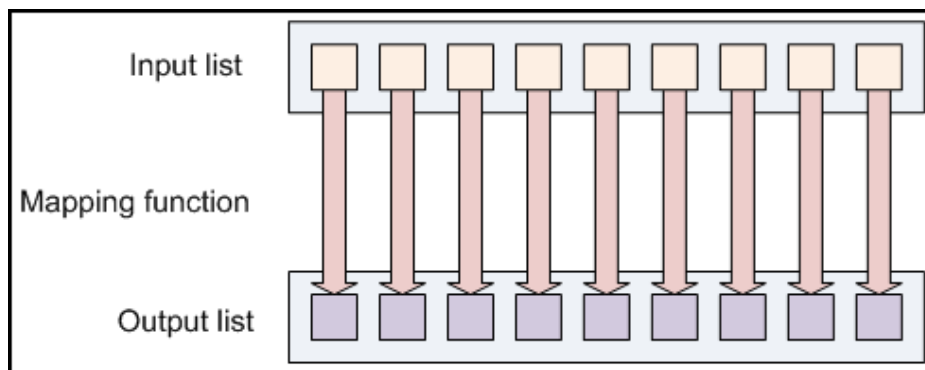
- Report Heartbeat – Každé tři vteřiny odešle DataNode report na NameNode se zprávou, že funguje správně.
- Report BlockReport – Každý desátý Heartbeat, který obsahuje sdělení od DataNode pro NameNode, obsahuje informace o všech blocích, které vlastní.

Všechny HDFS komunikační protokoly jsou vrstveny na protokolech TCP/IP. [21]

8.3 Hadoop model MapReduce

MapReduce je programovací model navržený pro zpracování velkého objemu dat paralelním způsobem. Podle přesně definovaných stylů spouští aplikace, které běží v Hadoop prostředí. Vstupní data jsou pomocí MapReduce funkcí transformovány na výstupní datové prvky. Při zpracování dat programovacím modelem MapReduce se tak bude dít hned dvakrát, protože jsou použity dvě na sobě závislé metody Map a Reduce. [21]

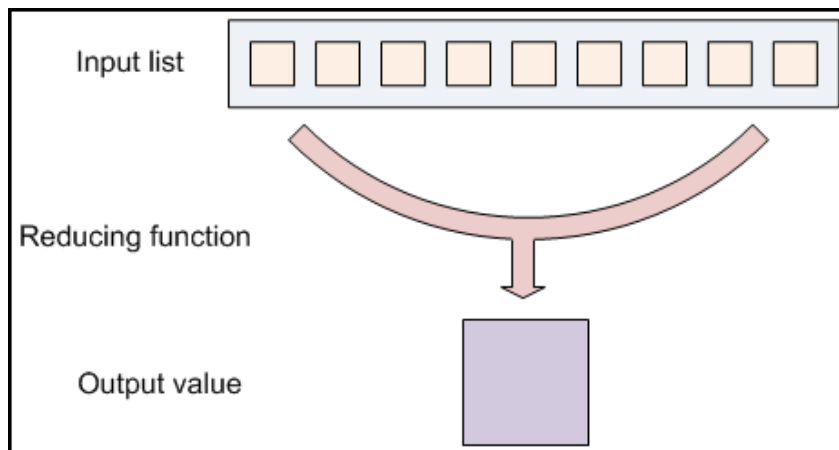
První ze dvou fází programu MapReduce se nazývá **Map**. Její funkce spočívá v postupném transformování každého ze vstupních datových prvků na výstupní datové elementy, jak ilustruje Obrázek 14. Jako příklad užitečného využití metody Map lze uvést funkci `toUpper(str)`, která vytvoří (ne modifikuje) ze vstupního textového řetězce nový text s výhradně velkými písmeny. [3]



Obrázek 14: Metoda Map

Zdroj: [3]

Druhou fází programu MapReduce je metoda **Reduce**. Ta spojuje vstupní datové prvky v jednu výstupní hodnotu podle zadaného parametru, jak nastiňuje Obrázek 15. Běžným využitím metody je sumarizace vstupních dat, kdy funkce Reduce vrátí celkový součet zadaných hodnot v podobě jednoho čísla. [3]

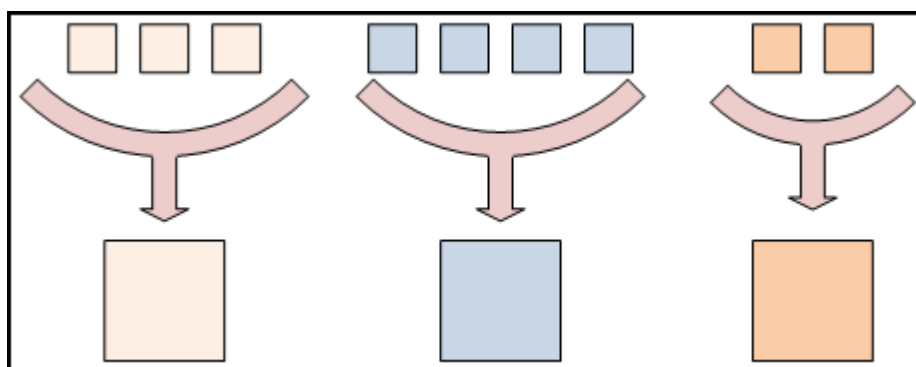


Obrázek 15: Metoda Reduce

Zdroj: [3]

8.3.1 Spojení metod Map a Reduce

Rámec MapReduce využívá obě metody a rozšiřuje jejich možnosti pro zpracování velkého objemu dat. Funkce Map a Reduce nepracují pouze s hodnotami, ale vždy s dvojicí hodnota – klíč. Každá z obou funkcí přijímá jako vstup dvojici hodnota – klíč a rovněž ji poskytuje jako výstup. Funkce Reduce sjednocuje vstupní zadané prvky do jedné výstupní hodnoty. Na výstupu z modelu MapReduce se obvykle nachází více výstupních prvků podle definovaných klíčů. Všechny hodnoty se stejným klíčem jsou agregovány nezávisle na ostatních, jak zobrazuje Obrázek 16, a jsou součástí celkového výstupu modelu MapReduce. [3]



Obrázek 16: Nezávislá agregace s různými klíči

Zdroj: [3]

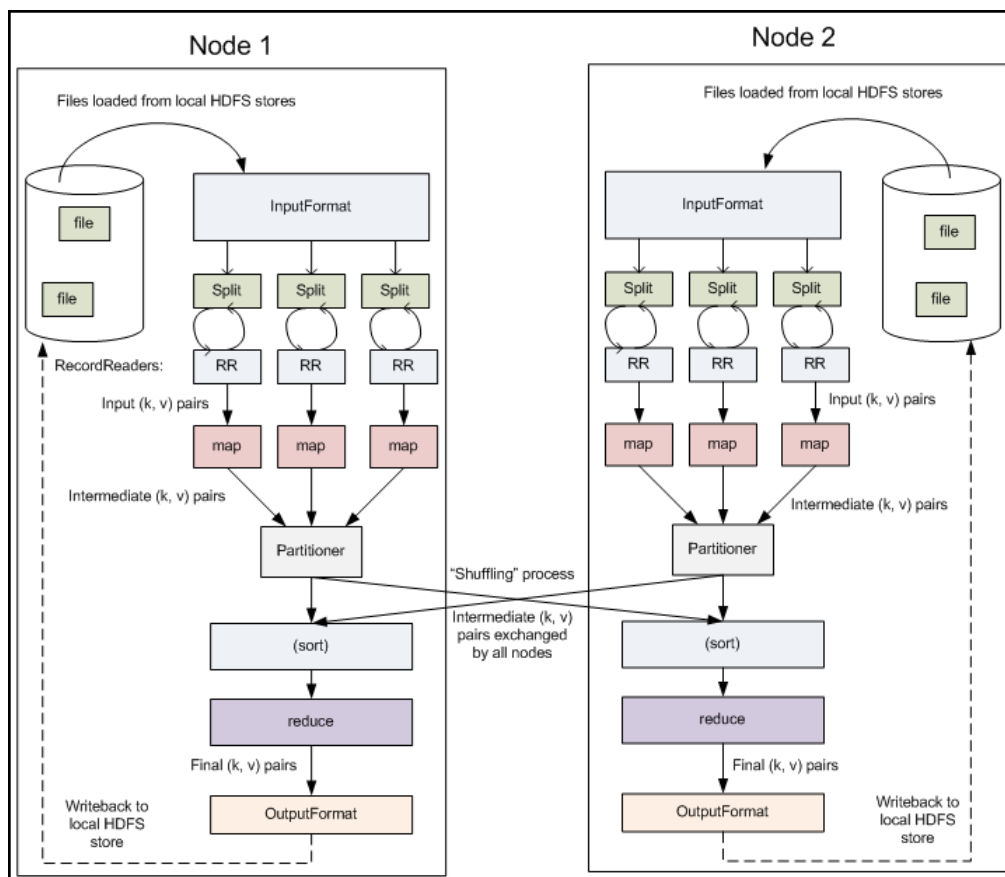
V programu Hadoop MapReduce existuje důležitá komponenta **Driver**, která vyvolává spouštění úloh a provádí kontrolu umístění vstupních i výstupních souborů. Metoda Driver pro svoji práci podle využívá detailní nastavení několika argumentů [3]:

- `inputPath` – argument definuje, nad jakými daty bude úloha spuštěna,
- `outputPath` – výstup z modelu bude uložen do souboru v adresáři,
- `JobConf` – objekt obsahuje informace o konfiguraci pro spuštěnou úlohu,
- `setMapperClass()`, `setReducerClass()` – prostřednictvím metod jsou označeny použité konkrétní funkce Map a Reduce,
- `setOutputKeyClass()`, `setOutputValueClass()` – obsahují emitované datové typy funkcí Reduce, ve výchozím nastavení odpovídají výstupům funkce Map,
- `InputFormat` – volí vstupní datové typy do funkce Map.

8.3.2 Datový tok modelu MapReduce

Souhrnný pohled na práci a návaznost obou metod Map a Reduce ukazuje Obrázek 11. Vstupní vrstva do modelu MapReduce obvykle pochází přímo ze souborů načtených do HDFS a je rovnoměrně rozdělena na všechny uzly v clusteru. Spuštění MapReduce programu proto vyvolá zahájení map úloh na všech uzlech. V clusteru existuje naprostá ekvivalence mezi uzly a instancemi funkce Map, každý vstupní soubor je tedy zpracováván libovolným uzlem. Po dokončení fáze Map dochází k výměně vzniklých dvojic klíč – hodnota tak, aby každá instance funkce Reduce obdržela všechny hodnoty se stejným klíčem. Tento proces lze označit jako jediný komunikační krok v modelu MapReduce, protože navzájem nedochází k žádné výměně informace mezi map úlohami, stejně tak mezi reduce úlohami. Pokud selže některý z uzlů v clusteru, zpracovávaná úloha se po restartu spustí znovu. [21]

Podrobnější popis modelu MapReduce a jeho rozdělení na dílčí prvky nabízí Obrázek 17, kde jsou pro názornost zobrazeny jen dva uzly, ale podobné schéma lze replikovat do velkého počtu uzlů. Následující odstavce obsahují podle [3], [21] zevrubný popis fází modelu MapReduce.



Obrázek 17: Prvky modelu MapReduce

Zdroj: [3]

Data, která se nachází v HDFS a jsou k dispozici pro MapReduce program, se označují jako **Vstupní soubory**. Pro velké vstupní soubory (desítky gigabytů a více) je typická různorodost vstupních formátů, které od sebe rozlišuje a poté načítá třída *InputFormat*, která podle Obrázek 17 vybere z HDFS soubory pro vstup, definuje jednotky *InputSplits* rozkládající soubor do úkolů a poskytuje je třídě *RecordReader* ke čtení.

Výchozí **InputFormat** je textový *TextInputFormat*, který je užitečný zejména při načítání neformátovaných dat nebo line-based záznamů (logy), protože považuje každý řádek vstupního souboru za samostatný záznam. Jiný vstupní formát *KeyValueInputFormat* také zachází s každým řádkem jako se samostatným záznamem, ale zatímco *TextInputFormat* považuje celý řádek za hodnotu, *KeyValueInputFormat* řádek rozdělí na dvojici klíč – hodnota. Tento formát je prospěšný hlavně pro čtení výstupu z jedné MapReduce úlohy a její použití jako vstupu do jiné. Speciální rychle načítané binární soubory, pro Hadoop specifické, čte třetí důležitý vstupní formát *SequenceFileInputFormat*. Sekvenční soubory tvoří blokově komprimovaná data libovolných datových typů.

Každá jednotka práce **InputSplit** tvoří jeden *map* úkol v MapReduce programu. Řešení příkladu modelem MapReduce obvykle zahrnuje spuštění několika úkolů. *Map* úkoly mohou zahrnovat čtení celého souboru, avšak mnohem častěji jsou soubory čteny po částech o velikosti 64 MB (jako bloky HDFS). Zpracování souboru v blocích umožňuje paralelní postup a v případě datově objemného souboru zvyšuje výkon při řešení úlohy.

Prvek *InputFormat* definuje podle zadání úlohy seznam úkolů a tím spouští fázi **Map**. Každý úkol odpovídá jedné jednotce práce *InputSplit*. Úkoly jsou poté systematicky přiřazeny k uzlům v clusteru podle fyzického úložiště bloků vstupního souboru. O načtení a převedení úkolů do dvojice klíč – hodnota se stará třída *RecordReader*. Výchozí *InputFormat*, *TextInputFormat*, podporuje třídu *LineRecordReader*, která zachází s každým řádkem jako s novou hodnotou. Klíč každého nového řádku tvoří jeho byte offset ve vstupním souboru, kde byte offset je číslo indikující vzdálenost počátečního písmene slova od začátku řádku. Třída *RecordReader* je na vstupu opakovaně vyvolávána, dokud nejsou spotřebovány všechny jednotky *InputSplits* a tedy vyzvány k činnosti všechny instance metody *Map*. Každý úkol *InputSplit* tvoří jen část celkového řešení úlohy, protože úkoly jsou odděleně zpracovávány v samostatných Java procesech. Jednotlivé *map* instance však nejsou vybaveny schopností komunikovat, aby se veškerá odpovědnost za chyby odrážela pouze od spolehlivosti lokálního počítače (uzlu). Kromě parametrů klíč a hodnota pracuje metoda *Map* ještě s objekty *OutputCollector* a *Reporter*. Metoda *Map* provede uživatelem zadané práce a prostřednictvím

metod objektu *OutputCollector* předá průběžné dvojice klíč – hodnota dále do metody *Reduce*. Objekt *Reporter* shromažďuje a poskytuje informace o konkrétních úlohách.

I přes paralelní zpracování neprobíhají všechny úkoly ve fázi *Map* současně. Aniž by byla kompletně dokončena úvodní část *MapReduce* programu, mohou být procesem **Shuffling** některé průběžné dvojice klíč – hodnota rozmisťovány do fáze *Reduce* podle svého klíče bez ohledu na to, ze které *map* instance pocházejí, zatímco jiné úlohy jsou v *Map* fázi teprve zpracovávány.

Pro každý *reduce* uzel je přiřazena jiná podmnožina klíče označovaná jako **Partition**. Ve výchozím nastavení je pro každou průběžnou dvojici klíč – hodnota vypočítán otisk funkce *hash*, podle kterého je dvojice přiřazena příslušnému *reduce* oddílu.

Každá *reduce* úloha je odpovědná za snížení hodnoty vzhledem k průběžným klíčům. Ještě před zahájením *reduce* úloh, jsou ale průběžné klíče (na jednotlivých uzlech) v rámci prvku **Sort** automaticky seřazeny.

Pro každou *reduce* úlohu je vytvořena jedna *reduce* instance s přiřazeným oddílem jednoho klíče. Metoda **Reduce** takto ke všem hodnotám obdrží průběžný klíč včetně iterátoru, který vrací hodnoty spojené s klíčem v nedefinovaném pořadí. Rovněž metoda *Reduce* pracuje s objekty *OutputCollector* a *Reporter*, které plní stejné funkce jako v metodě *Map*. Metody objektu *OutputCollector* poskytují výstupní dvojice klíč – hodnota třídě *OutputFormat*, která stanovuje způsob zápisu do výstupních souborů.

Třída **OutputFormat** funguje analogicky jako třída *InputFormat*, tedy řídí zápis dat zpět do *HDFS*. Každá *reduce* instance zapíše část výstupního souboru do společného adresáře. Výchozí *OutputFormat* je opět textový *TextOutputFormat*, který zapisuje výstupní dvojice klíč – hodnota na jednotlivé řádky textového souboru, což je jednak dobře čitelné pro uživatele a jednak vhodné pro použití jako dalšího vstupu do jiného *MapReduce* programu. Pokročilejším výstupním formátem je *SequenceFileOutputFormat*, který rychle třídí libovolné datové typy a zapisuje je v binární podobě do souboru. Odpovídající *SequenceFile-InputFormat* soubor rozdělí zpět na původní datové typy a poskytne je novému *MapReduce* programu.

Prvek **RecordWriter** je nástroj, který slouží pro vlastní zápis jednotlivých záznamů do souborů podle toho, jak je definována třída *OutputFormat*. Výstupní soubory uložené do *HDFS* jsou poté připraveny k zobrazení, případně k dalšímu načtení.

Podle potřeby a druhu úlohy lze využít volitelnou komponentu **Combiner**, která je umístěna mezi fázemi Map a Reduce. Proces *Combiner* pracuje pouze s daty získanými od *map* instance na daném stroji a předzpracovává je pro *reduce* oddíly.

8.4 Hadoop více-počítačový cluster

Vlastní řešení konstrukce více-počítačového clusteru v univerzitní laboratoři a jeho popis se opírá o [2], [3], [21], [45].

8.4.1 Architektura a potřebný software

Systém Apache Hadoop klade velmi malé nároky na hardware i software a lze jej tedy nakonfigurovat téměř na každý moderní počítač. Systém Hadoop byl z experimentálních důvodů úspěšně nakonfigurován a spuštěn dokonce i na netbookové jednotce s operační pamětí 2 GB RAM a procesorem Intel Celeron 743 (1.3 GHz).

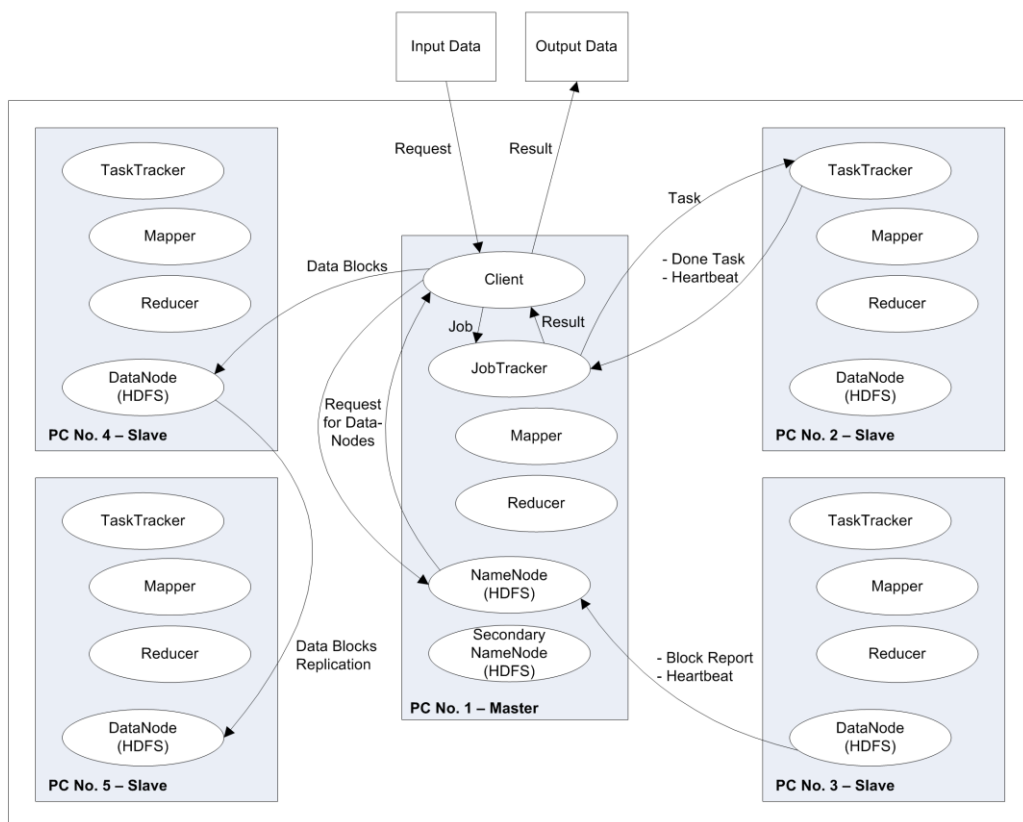
Hardware

Cluster pro vlastní zpracování v univerzitní laboratoři se skládá z pěti identických počítačů s podstatně vyššími výkonnostními předpoklady. Konkrétní konfigurace je tvořena operační pamětí 4 GB RAM a procesory Intel Dual Core i3-3220 (3.30 GHz) doplněné technologií hyper-threading, která umožní operačnímu systému pracovat s dvěma logickými procesory navíc.

Software

Důležitou komponentou pro instalaci Hadoop je software Java verze 1.6 nebo vyšší, bez které by distribuovaná databáze, ani MapReduce programy nemohly být realizovány. Proto je vyžadována její instalace na všech Ubuntu 12.10 operačních systémech, na kterých se spustí uzly DataNodes a NameNode a také daemons TaskTrackers a JobTracker. Všechny počítače jsou připojeny k 100 Mbps Ethernet síti, která je začleněna do lokální univerzitní sítě.

Model architektury vybraného řešení clusteru ukazuje Obrázek 18.



Obrázek 18: Architektura vybraného řešení clusteru

Zdroj: [35]

Tabulka 1 vypisuje pro snazší orientaci s root cestami přehled adresářů a souborů použitých při Hadoop a Java instalaci, konfiguraci i interakci.

Tabulka 1: Přehled použitých adresářů a souborů

/usr/local/hadoop	adresář Hadoop instalace
- /bin	adresář spouštěcích skriptů
start-all.sh	skript spustí HDFS i rámec MapReduce
stop-all.sh	skript ukončí HDFS i rámec MapReduce
start-dfs.sh	skript spustí HDFS
stop-dfs.sh	skript ukončí HDFS
start-mapred.sh	skript spustí rámec MapReduce
stop-mapred.sh	skript ukončí rámec MapReduce
+ /c++	
- /conf	adresář Hadoop konfigurace
core-site.xml	definice vlastností
hadoop-env.sh	konfigurace Java úložiště
hdfs-site.xml	definice vlastností
mapred-site.xml	definice vlastností
masters	seznam master uzlů
slaves	seznam slave uzlů
- /contrib	adresář adaptačních vrstev
- /streaming	adresář Hadoop streaming
hadoop-streaming-1.0.4.jar	balíček Hadoop streaming
+ /libexec	
+ /logs	
+ /src	
/usr/lib/jvm/	
+ /jdk1.7.0	adresář Java instalace
/app/hadoop/tmp	adresář dočasných souborů
+ /dfs	HDFS
+ /mapred	MapReduce
/etc	
hostname	soubor pro pojmenování uzlu
hosts	soubor pro překlad jmen na IP
.bashrc	systemový konfigurační soubor

Zdroj: vlastní zpracování

8.4.2 Instalace vývojové sady Java

Při instalaci vývojové sady Java 1.7 Development Kit (JDK) lze postupovat podle [28].

Přesným dodržáním postupu na všech uzlech se vývojová sada nainstaluje a zároveň se stane kompatibilní s ostatními konfiguračními soubory.

1. Stáhni archiv instalačního balíčku `jdk-7u9-linux-x64.tar.gz`.
2. Proveď dekomprimační příkaz `sudo tar -xvf ~/Stažené/jdk-7u9-linux-x64.tar.gz`.
3. Vytvoř adresář příkazem `sudo mkdir -p /usr/lib/jvm/jdk1.7.0`.
4. Přesuň všechny instalované soubory příkazem `sudo mv jdk1.7.0_09 /usr/lib/jvm/jdk1.7.0`.
5. Proveď postupně tři příkazy pro zpřístupnění Java vývojové sady.

```
sudo update-alternatives --install "/usr/bin/java" "java"  
"/usr/lib/jvm/jdk1.7.0/bin/java" 1
```

```
sudo update-alternatives --install "/usr/bin/javac" "javac"  
"/usr/lib/jvm/jdk1.7.0/bin/javac" 1
```

```
sudo update-alternatives --install "/usr/bin/javaws" "javaws"  
"/usr/lib/jvm/jdk1.7.0/bin/javaws" 1
```

6. Vytvoř adresář pro Mozilla pluginy příkazem `sudo mkdir ~/.mozilla/lugins/`.
7. Vytvoř symbolický link do Mozilla pluginů příkazem `ln -s /usr/lib/jvm/jdk1.7.0/jre/lib/amd64/libnpjp2.so ~/.mozilla/plugins/`.

8.4.3 Vytvoření clusteru

Více-počítačový cluster tvoří pět počítačů. Jeden z nich, označený `master1`, koordinuje HDFS a řídí výpočty, další čtyři, označené `slave2`, `slave3`, `slave4` a `slave5`, jsou mu podřízeny. Pojmenování uzlů lze provést v souboru `/etc/hostname` každého z nich. Počítače jsou jako síťové uzly identifikovány přidělenými IP adresami. Protože počítače použité pro účely experimentu nejsou identifikovány pomocí domain name system (DNS) a místo IP adres jsou dále používána jejich jména, je třeba do souboru `/etc/hosts` na všech používaných uzlech zanechat údaje o překladu jmen na IP adresy.

```
10.84.230.153 master1  
10.84.230.152 slave2  
10.84.230.151 slave3  
10.84.230.156 slave4  
10.84.230.154 slave5
```

Z důvodu přehlednosti a prevence nežádoucích vlivů Hadoop na administrátorský účet je pro každý počítač vytvořen pracovní uživatelský účet `hduserx`, kde `x` je pořadové číslo počítače. Spojením názvu účtu a jména uzlu v podobě `hduser1@master1`, `hduser2@slave2`, `hduser3@slave3`, `hduser4@slave4` a `hduser5@slave5` vzniknou jednoznačné identifikátory uživatelských účtů na uzlech v Hadoop clusteru.

K zajištění zabezpečené komunikace mezi uzly slouží komunikační protokol Secure Shell (SSH). Prvním krokem je provedení příkazů pro vygenerování RSA klíče na master uzlu a jeho přidání mezi autorizované klíče na všech slave uzlech tak, aby master uzel mohl otevírat SSH spojení ke všem slave uzlům s ověřením pomocí klíče a nevyžadoval ověření pomocí hesla.

```
ssh-keygen -t rsa --P
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Vygenerovaný veřejný RSA klíč uživatele uzlu `master1` je zapotřebí nakopírovat do adresáře autorizovaných klíčů na uzlech `slave2`, `slave3`, `slave4` a `slave5` provedením příkazů na uzlu `master1`.

```
ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser2@slave2
ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser3@slave3
ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser4@slave4
ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser5@slave5
```

Po úspěšném provedení příkazů nebude vyžadováno heslo pro připojení na slave uzly a uživatel `hduser1` master uzlu získá přístup k terminálu ostatních uživatelů na slave uzlech. Správně nastavené spojení lze otestovat následujícím příkazem.

```
ssh hduserx@slavex
```

8.4.4 Instalace Hadoop

Přesným dodržením postupu na všech uzlech se systém Hadoop verze 1.0.4 nainstaluje a zároveň se stane kompatibilní s ostatními konfiguračními soubory.

1. Přesuň se do adresáře příkazem `cd /usr/local`.
2. Proveď dekomprimační příkaz `sudo tar xzf hadoop-1.0.4.tar.gz`.
3. Přejmenuj adresář příkazem `sudo mv hadoop-1.0.4 hadoop`.
4. Změň vlastníka adresáře příkazem `sudo chown -R hduserx hadoop`.
5. Vytvoř adresář příkazem `sudo mkdir -p /app/hadoop/tmp`.
6. Změň vlastníka adresáře příkazem `sudo chown hduserx /app/hadoop/tmp`.

HDFS organizuje uložená data strukturovaně ve formě souborů a adresářů. Interakce uživatele s HDFS probíhá skrz vybranou klientskou aplikaci nebo službu. V případě dobré orientace v linuxových příkazech dostatečně poslouží terminálové rozhraní file system (FS) Shell. Pro synchronizaci příkazů Shellu a Hadoopu je o následující řádky doplněn, v domovském adresáři uživatele nacházející se, systémový konfigurační soubor .bashrc.

```
# Nastavení proměnné Hadoop = adresář, kde je Hadoop nainstalován
export HADOOP_HOME=/usr/local/hadoop

# Nastavení proměnné Java = adresář, kde je Java nainstalována
export JAVA_HOME=/usr/lib/jvm/jdk1.7.0

# Přidání několika praktických aliasů a funkcí pro spuštění Hadoop
#příkazů
unalias fs &> /dev/null
alias fs="hadoop fs"
unalias hls &> /dev/null
alias hls="fs -ls"

# Přidání Hadoop bin/ adresáře do proměnné PATH
export PATH=$PATH:$HADOOP_HOME/bin
```

8.4.5 Konfigurace clusteru

Pro zajištění Java podpory je nutné správně nadefinovat cestu nově nainstalovaného JDK aktualizací proměnné v souboru ./conf/hadoop-env.sh, který se nachází v konfiguračním adresáři domovského adresáře Hadoop.

```
export JAVA_HOME=/usr/lib/jvm/jdk1.7.0
```

Celá konfigurace HDFS se nachází ve skupině souborů xml v adresáři /usr/local/hadoop/conf. Soubor ./conf/capacity-scheduler.xml obsahuje výchozí hodnoty pro každý parametr v Hadoop databázi. Tento soubor je jen pro čtení a je netřeba ho měnit. Pro aktualizaci konfigurace a nastavení nové hodnoty lze využít další xml soubory v adresáři ./conf.

Pro základní konfiguraci HDFS je nutné v konkrétním formátu [název] – [hodnota] nastavit několik vlastností v souborech core-site.xml, hdfs-site.xml a mapred-site.xml.

Stěžejní vlastnosti adresa master uzlu a cesta pro ukládání datových bloků jsou definovány v souboru **core-site.xml**.

- `fs.default.name` – plní funkci uniform resource identifier (URI), tedy textový řetězec s definovanou strukturou, který slouží k přesné specifikaci master uzlu NameNode. Každý uzel DataNode je srozuměn s touto adresou a poskytne pro ni svoje data. Jednotlivé klientské programy se připojí na tuto adresu pro načtení umístění skutečných souborových bloků.

```
<property>
  <name>fs.default.name</name>
  <value>hdfs://master1:54310</value>
</property>
```

- `hadoop.tmp.dir` – cesta na místním souborovém systému, ve kterém by měly uzly NameNode a DataNodes ukládat svá data. Vlastnost lze rozdělit podle potřeby na specifické cesty pro NameNode i DataNode pomocí `dfs.name.dir`, resp. `dfs.data.dir`.

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
</property>
```

Parametry zahrnující změny v rodině HDFS vlastností jsou nastaveny v souboru **hdfs-site.xml**.

- `dfs.replication` – konfigurační parametr, který nastavuje replikační faktor pro každý blok dat v HDFS. Zadaná hodnota parametru by neměla přesáhnout počet uzlů.

```
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>
```

- `dfs.permissions` – volitelný parametr s hodnotou `false` zbavuje HDFS povinnosti ptát se na zadaná práva. V případě malého clusteru, který je spravován jedním administrátorem, však není kontrola práv prioritou.

```
<property>
  <name>dfs.permissions</name>
  <value>>false</value>
</property>
```


- `dfs.http.address` – po nastavení parametru začne webový prohlížeč naslouchat zadané adrese a zpřístupní tak přehlednou tabulku webového rozhraní se základními informacemi o spuštěné HDFS.

```
<property>
  <name>dfs.http.address</name>
  <value>master1:40012</value>
</property>
```

Všechny aktualizace MapReduce vlastností jsou prováděny v souboru **mapred-site.xml**, kde lze v případě většího clusteru specifikovat desítky dalších možností. Pro potřeby malého clusteru se jeví jako nejdůležitější pouze dvě.

- `mapred.job.tracker` – parametr, který definuje hostitele a port pro spuštění MapReduce aplikací.

```
<property>
  <name>mapred.job.tracker</name>
  <value>master1:54311</value>
</property>
```

- `mapred.reduce.tasks` – Nastavení parametru na master uzlu se odvíjí podle aktuální konfigurace clusteru, tedy podle počtu uzlů a počtu jejich jader, resp. vláken. Přesně lze podle [2] hodnotu vypočítat rovnicí:

$$\text{mapred.reduce.tasks} = \text{počet uzlů} * (\text{počet jader} - 1) * \text{číslo z intervalu } \langle 0.95; 1.75 \rangle$$

- 1 jádro je rezervováno pro daemons, které na uzlu běží.

Z výše uvedeného intervalu bylo vybráno číslo 0.95, protože všechny počítače jsou výkonem na stejné úrovni, a proto v konfiguraci figuruje hodnota 15.

```
<property>
  <name>mapred.reduce.tasks</name>
  <value>15</value>
</property>
```

- `mapred.map.tasks` – počet map úloh se ve výchozím nastavení odvíjí od počtu bloků v HDFS vzniklých ze vstupních dat, jejich počet je roven počtu jednotek `InputSplits` a není nutné tuto hodnotu detailněji nastavovat.

Všechny provedené změny v konfiguračních souborech musí být zaznamenány na master uzlu. Aby slave uzly `DataNodes` znaly hostitelskou adresu master uzlu `NameNode` i úložiště blokových souborů `/app/hadoop/tmp`, je nutné konfigurační soubor `core-site.xml` nakopírovat

na všechny uzly v clusteru. Stejně se chovají slave daemons TaskTrackers vůči primárnímu daemonu JobTracker, proto musí být soubor `mapred-site.xml` se stěžejní vlastností `mapred.job.tracker` nakopírován na ostatní počítače. Soubory `.bashrc` a `hadoop-env.sh` ukrývající aliasy shell příkazů a domovských adresářů Hadoop a Javy také musí být součástí konfigurace každého počítače.

Před spuštěním Hadoop daemons zbývá ještě nadefinovat seznamy masters a slaves v adresáři `./conf` na master uzlu, které obsahují jednoznačné identifikátory uživatelských účtů na uzlech v Hadoop clusteru.

Seznam **masters** obsahuje pouze identifikátor master uzlu.

```
hduser1@master1
```

Seznam **slaves** čítá pět položek. Výpis nabízí čtyři slave uzly a jeden master uzel, který může zároveň plnit roli slave uzlu. V případě malého clusteru tak lze dosáhnout většího výkonu při provádění výpočtů.

```
hduser1@master1
hduser2@slave2
hduser3@slave3
hduser4@slave4
hduser5@slave5
```

8.4.6 Start HDFS

Po dokončení instalace systému Hadoop, včetně přípravy vývojového prostředí a konfigurace xml souborů, přichází na řadu před prvním spuštěním formát master uzlu NameNode, při kterém jsou do konfigurace zaznamenány nové ID a název jmenného prostoru. Všechny příkazy nad HDFS (start i interakce) jsou vykonávány v adresáři uživatele `hduser1` na `master1` uzlu (`hduser1@master1`), kde je Hadoop nainstalován, tedy v `/usr/local/hadoop`.

```
./bin/hadoop namenode -format
```

Příkaz provede formát uzlu NameNode, ovšem při opakovaném spuštění příkazu je nutné ověřit zformátování i všech uzlů DataNodes, případně jejich dočasné úložiště `/app/hadoop/tmp/data` manuálně rekurzivně smazat, aby i jim byly při každém prvním startu HDFS přiděleny nová ID a jmenný prostor.

Spuštění clusteru lze pro přehlednost rozdělit do dvou kroků:

1. Start HDFS daemons – NameNode, Secondary NameNode a DataNodes
2. Start MapReduce daemons – JobTracker a TaskTrackers

Pro start HDFS daemons se používá spouštěcí skript `./bin/start-dfs.sh` a pro start MapReduce daemons skript `./bin/start-mapred.sh`. Oba skripty využívají nastavenou konfiguraci v xml souborech ve složce `./conf`. Po zadání obou spouštěcích skriptů nainstalují master daemons NameNode, Secondary NameNode a JobTracker, včetně slave daemons DataNode a TaskTracker, protože master uzel je současně využíván i jako slave uzel (seznam `./conf/slaves`). S nepatrnou prodlevou proběhne start daemons DataNode a TaskTracker na ostatních čtyřech slave uzlech.

Po ukončení práce v HDFS lze všechny daemons ukončit analogickými skripty `./bin/stop-dfs.sh` a `./bin/stop-mapred.sh`. Pro zrychlení spuštění, resp. ukončení všech daemons najednou slouží spouštěcí skript `./bin/start-all.sh`, resp. `./bin/stop-all.sh`.

8.4.7 Interakce s HDFS

Interakce uživatele s HDFS probíhá skrz klientskou terminálovou aplikaci FS Shell. Standardní podobu příkazů pro komunikaci s HDFS lze nastínit v následujícím modelu.

```
./bin/hadoop NázevModulu -Příkaz Argumenty
```

Název modulu řekne programu Hadoop, kterou množinu funkcí má použít. *Příkaz* je název konkrétního příkazu, který má program v rámci tohoto modulu se zadanými *Argumenty* provést. Přehled nejčastěji používaných modulů, konkrétních příkazů se specifickými argumenty přibližuje Tabulka 2.

Tabulka 2: Interakce klienta s HDFS

Název modulu	-Příkaz	Argumenty	Poznámka
dfs	-copyFromLocal	Z: /cesta Do:/cesta	Zkopíruje soubor nebo adresář z lokálního disku do adresáře HDFS
	-copyToLocal	Z: /cesta Do:/cesta	Zkopíruje soubor nebo adresář z adresáře HDFS na lokální disk
	-ls	Co: /cesta	Vypíše soubory a adresáře
	-cat	Co: /cesta	Vypíše obsah souboru na obrazovku
	-getmerge	Kde: /cesta Kam:/cesta	Spojí obsah všech souborů v adresáři a výstup uloží na lokální disk

Zdroj: upraveno podle [45]

8.5 Hadoop streaming

System Hadoop je vytvořen programovacím jazykem Java, proto primárně spouští programy Mapper a Reducer s kódem napsaným v Javě. Hadoop ovšem přichází s adaptačními vrstvami pipes a **streaming**, které umožní rámci MapReduce spuštění kódů napsaných i jiným programovacím jazykem. Pipes lze označit za knihovnu, která umožňuje použití zdrojového kódu C++. Streaming zahrnuje application programming interface (API) několika dalších programovacích jazyků, mezi které patří např. Bash, Perl nebo Python. Pro implementaci programů Mapper a Reducer a zpracování velkého objemu prostorových dat je využít právě jazyk Python. [3], [21]

Spuštění streaming úlohy provede následující příkaz.

```
./bin/hadoop jar ./contrib/streaming/hadoop-*streaming*.jar -argumenty
```

Symbol hvězdičky nahrazuje libovolný počet znaků a v příkazu tak odstraňuje problém s výběrem správné verze streaming balíčku. Popis relevantních argumentů, které budou použity dále v textu ke spuštění streaming úlohy, zřehledňuje Tabulka 3.

Tabulka 3: Hadoop streaming

Argumenty	Poznámka
-file	cesta k souboru s kódem programu
-mapper	název a umístění souboru s kódem mapper.py
-reducer	název a umístění souboru s kódem reducer.py
-input	cesta ke vstupním datům v HDFS
-output	vytvoření adresáře v HDFS pro výstup

Zdroj: upraveno podle [45]

Oba soubory ukrývající zdrojový kód programu MapReduce se nacházejí na lokálním úložišti master uzlu, odkud jsou klientem neomezeně spouštěny v rámci Hadoop streaming API.

8.5.1 Zdrojový kód mapper.py

Inspirace pro zdrojový kód programovacího jazyku Python byla čerpána z [3], [16], [45].

```
import sys

# vstup je načten ze STDIN (standard input)
for line in sys.stdin:
    # odstraň úvodní a závěrečné mezery
    line = line.strip()
    # rozděl řádek na slova - použito u více souřadnic
    words = line.split()
    # spust' cyklus
    for word in words:
        print '%s\t%s' % (word, 1)
        # vytiskni / vypiš výstup v zadaném formátu string-tabulátor-
        #string a zapiš výsledek do STDOUT (standard output);
```

Výstup z fáze Map slouží zároveň jako vstup do fáze Reduce.

8.5.2 Zdrojový kód reducer.py

Inspirace pro zdrojový kód programovacího jazyku Python byla převzata z [3], [16], [45].

```
from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

# vstup je načten ze STDIN
for line in sys.stdin:
    # odstraň úvodní a závěrečné mezery
    line = line.strip()
    # rozděl tabulátorem vstup obdrženy od mapper.py
    word, count = line.split('\t', 1)
    # konvertuj proměnnou count z datového typu string (řetězec) na int
    # (číslo) v rámci výjimky
    try:
        count = int(count)
```

```

except ValueError:
    # když count není číslo, položku ignoruj a pokračuj
        continue

if current_word == word:
    #ANO: inkrementuj proměnnou current_count o count, tedy 1
    current_count += count

else:
    #JINAK: vytiskni/zapiš výsledek do STDOUT, kde proměnná
    #current_word je naplněna aktuální hodnotou proměnné word a
    #proměnná current_count je naplněna aktuální hodnotou proměnné
    #count

    if current_word:
        # zapiš výsledek do STDOUT
        print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

#Poslední záznam se sice inkrementuje, protože podmínka bude splněna, ale
#již se nevytiskne v rámci cyklu. Poslední záznam je tedy nutné
#vytisknout/vypsat samostatně mimo hlavní cyklus.

if current_word == word:
    print '%s\t%s' % (current_word, current_count)

```

Oba programy obdrží vstup jako `stdin` (standard input) a emitují výstupní pár klíč – hodnota jako `stdout` (standard output). Výstup z fáze Reduce se již zapíše do nově připraveného adresáře v HDFS.

Shrnutí

Základem správného sestavení, spuštění i využívání každého rozsáhlého stroje či aplikace je manuál. Stejně tak pro Hadoop systém bylo vytvořeno několik dokumentací, konfigurací i tutoriálů, které mají za úkol ho zpřístupnit začínajícím uživatelům. Dostatečné porozumění stěžejním oblastem systému Hadoop, tedy HDFS a MapReduce modelu, ale i jejich elementárním prvkům, znamená dobrý předpoklad pro jejich efektivní užívání. Pro sestavení malého Hadoop clusteru podle strukturovaného popisu je očekávána základní orientace v počítačových sítích a linuxových operačních systémech. S využitím knihoven Hadoop streaming lze zkompileovat zdrojový kód takřka kteréhokoli programovacího jazyku nebo jeho hybridní modifikace.

9 ZPRACOVÁNÍ VELKÉHO OBJEMU PROSTOROVÝCH DAT

Vstupy i výstupy do MapReduce streaming rámce jsou vždy reprezentovány textovým formátem. Pro editaci, manipulaci, vizualizaci či provádění statistických výpočtů nad prostorovými daty se na „akademické půdě“ Univerzity Pardubice používá systém ArcGIS. Podle způsobu ukládání prostorových dat v ArcGISu lze data pro provedení výpočtu získat ve formátech shapefile nebo geodatabáze.

9.1 Četnost nadmořské výšky

Prezentace zpracování velkého objemu dat systémem Hadoop vychází z distribuovaného výpočtu četnosti nadmořské výšky bodů vybrané oblasti. Data pro rutinní činnost v programu ArcGIS nedosahují závratné velikosti a nelze je označit za velkoobjemová. Aplikovat na taková data distribuovaný výpočet by postrádalo smysl. Pravá síla HDFS a rámce MapReduce spočívá ve zpracování několika set megabytových nebo dokonce gigabytových souborů. Pro dosažení potřebné velikosti vstupního souboru bylo spojeno více mapových listů s pravidelně rozmístěnými body, jejichž vzdálenost ve skutečnosti odpovídá deseti metrům. Každý bod je reprezentován x, y a z-souřadnicí, přičemž pro účely zpracování byly souřadnice x a y vyřazeny. Celkem bylo sjednoceno 71 mapových listů, tedy území o rozloze přibližně 1,3 tisíc km², které odpovídá přibližně 1/3 pardubického kraje.

9.1.1 Příprava dat

Formáty shapefile nebo geodatabáze si lze jen těžko představit jako vstupy do MapReduce programu. Pro konverzi souboru z databázových formátů gdb, dbf a mdb byl využit program MS Access, ze kterého byla data vyexportována v požadovaném txt formátu. Přehled parametrů vstupního souboru 71.txt nastiňuje Tabulka 4.

Tabulka 4: Vstupní soubor 71.txt

Parametry vstupního souboru 71.txt	
počet záznamů v jednom mapovém listu	181 922
spojeno mapových listů	71
celkový počet záznamů	12 921 454
počet sloupců	1
zaokrouhlení	0,01
velikost (Byte)	103 371 632
rozloha (tisíc km ²)	1,3

Zdroj: vlastní zpracování

9.1.2 Zpracování dat

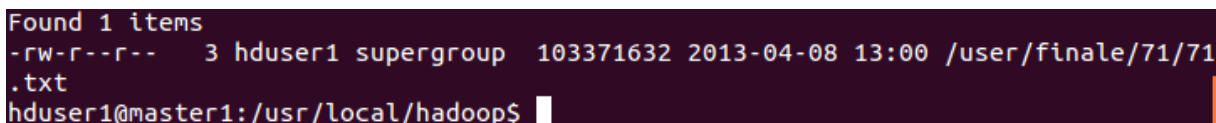
Jakmile je vstupní soubor 71.txt připraven na lokálním disku pro zpracování MapReduce programem v Hadoop systému, příkazem copyFromLocal se nakopíruje do HDFS.

```
./bin/hadoop dfs -copyFromLocal /home/hduser1/71.txt \  
/user/finale/71/71.txt
```

Příkaz copyFromLocal automaticky v distribuovaném souborovém systému u uživatele finale vytvoří adresář 71 a do něho umístí kopii originálního souboru 71.txt z lokálního disku. Pro ověření úspěšné kompilace lze příkazem ls zobrazit obsah vytvořeného adresáře.

```
./bin/hadoop dfs -ls /user/finale/71
```

Obsah adresáře s výpisem čtecích, zapisovacích a spouštěcích práv k souboru, včetně vlastníka souboru, velikosti v Bytech a časového údaje vložení, ukazuje Obrázek 19.



```
Found 1 items  
-rw-r--r--  3 hduser1 supergroup  103371632 2013-04-08 13:00 /user/finale/71/71  
.txt  
hduser1@master1:/usr/local/hadoop$
```

Obrázek 19: Výpis adresáře HDFS se vstupním souborem

Zdroj: vlastní zpracování

Po ověření přítomnosti vstupního souboru následuje spuštění Hadoop streaming úlohy s konkrétními parametry. Použité parametry vysvětluje Tabulka 3.

```
./bin/hadoop jar ./contrib/streaming/hadoop-*streaming*.jar \  
-file /home/hduser1/mapper.py -mapper /home/hduser1/mapper.py \  
-file /home/hduser1/reducer.py -reducer /home/hduser1/reducer.py \  
-input user/finale/71/* -output user/finale/71_vystup
```


Celý průběh streaming úlohy s výpisem procentuálních kroků fází Map a Reduce v klientském terminálu zobrazuje Obrázek 20.

```
packageJobJar: [/home/hduser1/mapper.py, /home/hduser1/reducer.py, /app/hadoop/t
mp/hadoop-unjar5340853661321987965/] [] /tmp/streamjob2903794502325237372.jar tm
pDir=null
13/04/08 13:01:38 INFO util.NativeCodeLoader: Loaded the native-hadoop library
13/04/08 13:01:38 WARN snappy.LoadSnappy: Snappy native library not loaded
13/04/08 13:01:38 INFO mapred.FileInputFormat: Total input paths to process : 1
13/04/08 13:01:38 INFO streaming.StreamJob: getLocalDirs(): [/app/hadoop/tmp/map
red/local]
13/04/08 13:01:38 INFO streaming.StreamJob: Running job: job_201304081131_0021
13/04/08 13:01:38 INFO streaming.StreamJob: To kill this job, run:
13/04/08 13:01:38 INFO streaming.StreamJob: /usr/local/hadoop/libexec/./bin/had
oop job -Dmapred.job.tracker=master1:54311 -kill job_201304081131_0021
13/04/08 13:01:38 INFO streaming.StreamJob: Tracking URL: http://master1:50030/j
obdetails.jsp?jobid=job_201304081131_0021
13/04/08 13:01:39 INFO streaming.StreamJob: map 0% reduce 0%
13/04/08 13:01:54 INFO streaming.StreamJob: map 21% reduce 0%
13/04/08 13:01:56 INFO streaming.StreamJob: map 40% reduce 0%
13/04/08 13:01:57 INFO streaming.StreamJob: map 49% reduce 0%
13/04/08 13:01:59 INFO streaming.StreamJob: map 61% reduce 0%
13/04/08 13:02:00 INFO streaming.StreamJob: map 72% reduce 0%
13/04/08 13:02:02 INFO streaming.StreamJob: map 82% reduce 0%
13/04/08 13:02:03 INFO streaming.StreamJob: map 90% reduce 0%
13/04/08 13:02:04 INFO streaming.StreamJob: map 99% reduce 0%
13/04/08 13:02:07 INFO streaming.StreamJob: map 100% reduce 0%
13/04/08 13:02:18 INFO streaming.StreamJob: map 100% reduce 1%
13/04/08 13:02:20 INFO streaming.StreamJob: map 100% reduce 4%
13/04/08 13:02:21 INFO streaming.StreamJob: map 100% reduce 7%
13/04/08 13:02:23 INFO streaming.StreamJob: map 100% reduce 11%
13/04/08 13:02:24 INFO streaming.StreamJob: map 100% reduce 12%
13/04/08 13:02:29 INFO streaming.StreamJob: map 100% reduce 39%
13/04/08 13:02:30 INFO streaming.StreamJob: map 100% reduce 56%
13/04/08 13:02:32 INFO streaming.StreamJob: map 100% reduce 61%
13/04/08 13:02:33 INFO streaming.StreamJob: map 100% reduce 67%
13/04/08 13:02:35 INFO streaming.StreamJob: map 100% reduce 70%
13/04/08 13:02:36 INFO streaming.StreamJob: map 100% reduce 73%
13/04/08 13:02:41 INFO streaming.StreamJob: map 100% reduce 90%
13/04/08 13:02:42 INFO streaming.StreamJob: map 100% reduce 100%
13/04/08 13:02:48 INFO streaming.StreamJob: Job complete: job_201304081131_0021
13/04/08 13:02:48 INFO streaming.StreamJob: Output: /user/finale/71_vystup
hduser1@master1:/usr/local/hadoop$
```

Obrázek 20: Streaming úloha

Zdroj: vlastní zpracování

Program Mapreduce vytvoří v adresáři 71_vystup uživatelského účtu finale za 1 min a 10 s celkem 17 souborů. Na základě vypočítaného optimálního počtu mapred.reduce.tasks = 15 bylo vytvořeno odpovídajících 15 souborů označených part-00000 až part-00014, nabízejících hledané řešení, tedy četnost výskytu jednotlivých nadmořských výšek vstupních bodů. Zbýlé dva soubory ve výstupním adresáři informují o úspěšném provedení úlohy a jejím systémovém nastavení.

Vznik výstupního adresáře lze potvrdit příkazem `ls`, který opět vypíše obsah, nyní již bohatšího, adresáře uživatele finale.

```
./bin/hadoop dfs -ls /user/finale
```

Obrázek 21 ilustruje výpis adresáře se vstupními i výstupními daty.

```
Found 2 items
drwxr-xr-x - hduser1 supergroup 0 2013-04-08 13:00 /user/finale/71
drwxr-xr-x - hduser1 supergroup 0 2013-04-08 13:02 /user/finale/71_vy
stup
```

Obrázek 21: Výpis adresáře se vstupními i výstupními daty

Zdroj: vlastní zpracování

Pro umístění získaných výstupů na lokální disk a následnou snazší manipulaci lze použít příkazy `copyToLocal` nebo `getmerge`. Nejprve je ale nutné adresář pro umístění výstupu na lokálním úložišti vytvořit příkazem.

```
mkdir /home/hduser1/71_vystup_merge
```

Teprve potom se použije příkaz pro kopírování výstupu. Zatímco příkaz `copyToLocal` pouze nakopíruje všechny soubory `part-xxxxx` z HDFS do lokálního úložiště, příkaz `getmerge` všechny soubory sjednotí a poté jeden výstupní soubor nakopíruje na disk.

```
./bin/hadoop dfs -getmerge /user/finale/71_vystup \
/home/hduser1/71_vystup_merge
```

Finální výstup čte, příp. edituje libovolný textový editor. Ukázkou finálního výstupu, který v konečném výsledku obsahuje výčet **30 544** z-souřadnic a k nim přiřazenou četnost jejího výskytu, zachycuje Obrázek 22.

```
203,99 1872
204,02 1035
204,11 517
204,20 413
204,89 658
204,98 2147
205,01 2932
```

Obrázek 22: Ukázka finálního výstupu

Zdroj: vlastní zpracování

9.2 Zátěžové testy

Pro testování výkonnosti sestaveného clusteru byla vybrána velmi podobná data, jako v případě měření četnosti nadmořské výšky. Pro zvýšení objemu dat však nebyly ze vstupního souboru odstraněny `x` ani `y`-souřadnice a počet se tak ztrojnásobil na téměř 39 milionů

úplných záznamů obsahujících x, y i z-souřadnici. Velikost vstupního souboru se přirozeně také zvýšila.

Přehled parametrů vstupního souboru 71_vse.txt vypisuje Tabulka 5.

Tabulka 5: Vstupní soubor 71_vse.txt

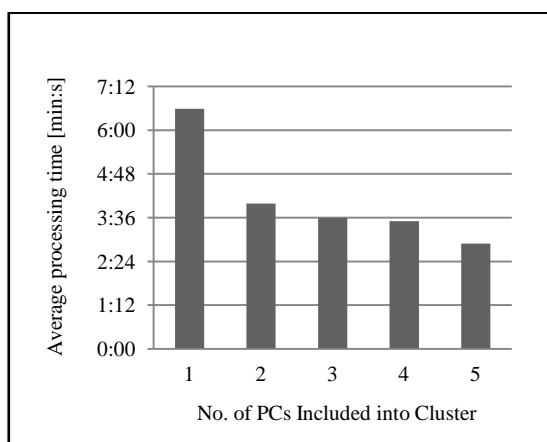
Parametry vstupního souboru 71_vse.txt	
počet záznamů v jednom mapovém listu	181 922
spojeno mapových listů	71
celkový počet záznamů	38 749 386
počet sloupců	3
zaokrouhlení	0,01
velikost (Byte)	607 313 920
rozloha (tisíc km ²)	1,3

Zdroj: vlastní zpracování

Samotné testování je rozděleno na dvě části. První se zaměřuje na testování sestaveného clusteru při zpracování velkého objemu dat ve výchozím nastavení, druhá využívá vypočítaného optimálního nastavení parametru `mapred.reduce.tasks`.

9.2.1 Testování ve výchozím nastavení

Ve výchozím nastavení parametru `mapred.reduce.tasks = 1` nedochází během fáze Reduce k žádné distribuci výpočtu. Rozdělení výpočtu mezi fyzické a logické procesory využívá pouze fáze Map a jen ta tak snižuje dobu trvání celého procesu. Z Obrázku 23 je patrné, že nejmarkantnější rozdíl nastal mezi jedním a dvěma v clusteru začleněnými počítači. Z toho vyplývá, že i sebemenší možnost distribuce prospívá ke zrychlení výpočtu.



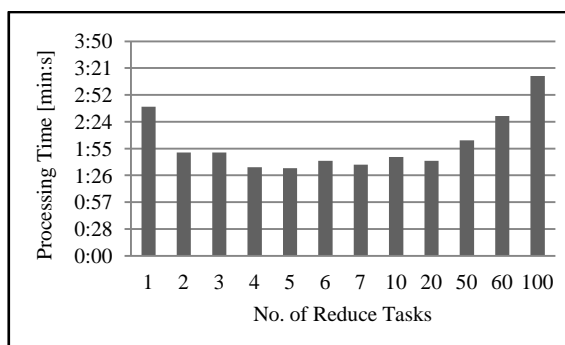
Obrázek 23: Průměrná doba zpracování vzhledem k počtu zapojených počítačů

Zdroj: [35]

9.2.2 Testování v optimálním nastavení

Pro optimalizaci výkonu clusteru lze uvažovat hned několik konfiguračních parametrů, mezi které patří minimální a maximální počet map či reduce tasks, chybová tolerance nebo bloková replikace dat. Detailní nastavení optima u všech stěžejních parametrů stojí za úvahu v případě obrovských objemů dat a velmi rozsáhlých clusterů čítající desítky počítačů.

V případě malého clusteru se jako stěžejní jeví konfigurace optimální hodnoty parametru `mapred.reduce.tasks`. Obrázek 24 poukazuje na časové změny doby zpracování v odlišném nastavení tohoto parametru. Protože se nejedná o pro Hadoop výrazně velký soubor ke zpracování, drží se průměrné doby výpočtu s nastavením parametru na hodnotách 2 až 20 na velmi podobné časové úrovni. Při zadání nedistribuční hodnoty 1, nebo naopak při zadání podstatně větší hodnoty převyšující počet jader clusteru, dochází k nárůstu doby zpracování.

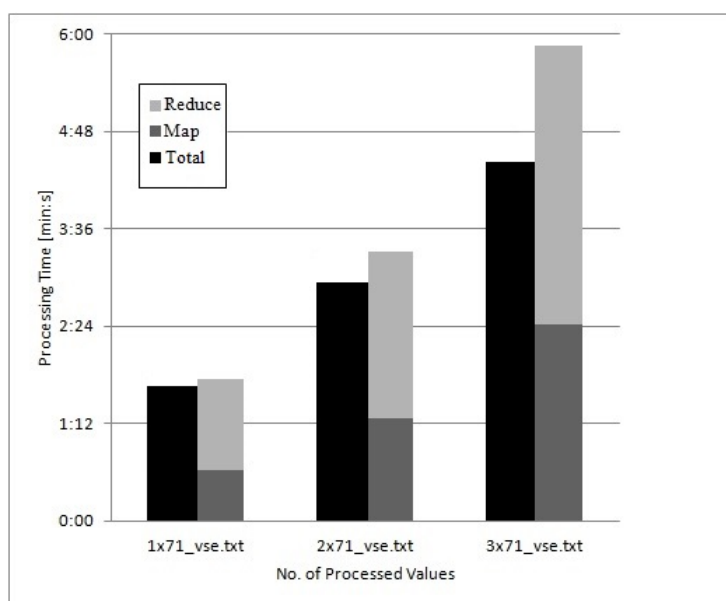


Obrázek 24: Průměrná doba zpracování vzhledem k parametru `mapred.reduce.tasks`

Zdroj: [35]

Další test rychlosti zpracování, nyní s pevným nastavením parametru `mapred.reduce.tasks` na optimální hodnotu 15, sleduje výkon clusteru při lineárním zvyšování objemu vstupních dat. Do HDFS adresáře se vstupními daty byly pro snazší kontrolu a přesnou interpretaci výsledků postupně přidávány další kopie vstupního souboru `71_vse.txt`.

Dosažené výsledky graficky reprezentuje Obrázek 25.



Obrázek 25: Průměrná doba zpracování vzhledem k optimu

Zdroj: upraveno podle [35]

Součet doby zpracování obou fází Map a Reduce radikalizuje exponenciální nárůst oproti celkové době zpracování (Total) a zároveň poukazuje na úsporu času při zpracování dat distribuovaným výpočtem, protože součet časů fází Map a Reduce vždy převyšuje celkovou dobu zpracování. Tak jest, protože fáze Reduce nemusí vždy začít až po kompletním dokončení fáze Map. Jakmile fáze Map uvolní dostatečný počet výstupních dvojic klíč – hodnota, fáze Reduce s nimi okamžitě začne pracovat jako se vstupy a šetří tak celkový čas zpracování.

Shrnutí

V rámci prezentace zpracování velkého objemu prostorových dat byl podrobně popsán postup při provádění výpočtu MapReduce programem v systému Hadoop. Pro ilustraci výpočetního rozdílu byla stejná úloha zadána statistické analýze programu ArcGIS for Desktop 10, kde sice hardware konfigurace dual-core AMD Opteron 8220 CPU, 48 GB RAM výkonnostně převyšuje počítače použité v clusteru, výpočet četnosti z-souřadnice však i tak trval přes 36 min [35]. Do zátěžového testu v obou případech vstoupil soubor 71.txt, jehož zpracování při distribuovaném výpočtu zabralo 1 min a 10 s, jak lze odečíst z Obrázku 20. Zátěžové testy přinesly hodnotné, leč nepřekvapující výsledky. Systém Hadoop rozhodně dokáže zvýšit výpočetní kapacitu po softwarové stránce při využití relativně levného hardware.

ZÁVĚR

Problematika velkých objemů a exponenciálního růstu dat i možnosti pro jejich zpracování byly v rámci diplomové práce představeny a na základě mnoha studií nemůže být pochyb o důležitosti jejího řešení.

Cílem diplomové práce bylo naimplementovat Hadoop více-počítačový cluster, na kterém lze prezentovat zpracování velkoobjemových prostorových dat v podmínkách malé laboratoře. Dosažené výsledky vlastního řešení jednak při výpočtu četnosti z-souřadnice, jednak při zátěžových testech výkonnosti clusteru, potvrzují potenciál systému Hadoop při zpracování velkého objemu prostorových dat a z toho lze usuzovat, že cíl práce byl splněn.

Přínos práce ovšem netkví pouze v pozitivních výsledcích zpracování dat v podmínkách malé laboratoře, nýbrž i v realizaci podrobného unikátního česky psaného návodu na sestavení Hadoop clusteru, jež by měl při dodržení jednotlivých kroků naimplementovat i začátečník v oblasti distribuovaných systémů.

Během zkoumání zadaného tématu vyvstalo mnoho souvisejících otázek a práce si bezpochyby zaslouží další rozšíření v oblasti ukládání dat do HDFS nebo v možnosti zapojení příbuzných projektů (Ambari, Pig a dalších) paralelně vyvíjených týmem The Apache Software Foundation. Jejich efektivní využití by mohlo značně usnadnit práci ve složitějších programech na rozsáhlejších clusterech zabývajících se obrovským objemem dat a výpočetně ještě náročnějším zpracováním. Jinou neméně zajímavou možností, jak navázat na bádání, navrhuji začlenit cluster do privátního cloudu a spojit tak oba IT trendy. V případě úspěšné implementace a naplnění předpokladů obou technologií, by mohla být data velkých objemů zpracována ještě efektivněji.

POUŽITÁ LITERATURA

- [1] AL ZABIR, Omar. Make your website faster using Content Delivery Network. *Omaralzabir.com* [online]. 2007 [cit. 2012-03-04]. Dostupné z: http://omaralzabir.com/make_your_website_faster_using_content_delivery_network/
- [2] Apache Hadoop. *Hadoop Wiki* [online]. 2012 [cit. 2013-04-14]. Dostupné z: <http://wiki.apache.org/hadoop/>
- [3] Apache Hadoop: Tutorial. *Yahoo!* [online]. 2009 [cit. 2013-02-25]. Dostupné z: <http://developer.yahoo.com/hadoop/tutorial/>
- [4] ArcGIS online. *Esri* [online]. 2011 [cit. 2013-02-19]. Dostupné z: <http://www.esri.com/software/arcgis/arcgisonline/features>
- [5] ArcGIS Server. *Arcdata Praha* [online]. 2004 [cit. 2012-03-04]. Dostupné z: <http://www.arcdata.cz/produkty-a-sluzby/software/esri/serverova-reseni-esri/arcgis-server/>
- [6] BOGORNY, Vania, SHEKHAR, Shashi. Spatial and Spatio-temporal Data Mining. *Data Mining (ICDM), 2010 IEEE 10th International Conference* [online]. 2010, [cit. 2013-02-22]. DOI: 10.1109/ICDM.2010.166. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5694111&isnumber=5693950>
- [7] BUHL, Hans, et al. Big Data. *Business & Information Systems Engineering* [online]. 2013, [cit. 2013-02-22]. ISSN 1867-0202. DOI: 10.1007/s12599-013-0249-5. Dostupné z: <http://link.springer.com/10.1007/s12599-013-0249-5>
- [8] CASONATO, Regina. Big Data optikou analytika Gartner. *Inside* [online]. 2012 [cit. 2013-02-25]. Dostupné z: <http://inside.cz/2012/06/casonato-regina-big-data-optikou-analytika-gartner/>
- [9] Cloud computing pro divoké lovce, manažery a IT specialisty. *BusinessIT* [online]. 2011 [cit. 2012-03-04]. Dostupné z: <http://www.businessit.cz/cz/cloud-computing-pro-manazery-a-it-specialisty-tema.php>
- [10] Cloud GIS. *Esri* [online]. 2011 [cit. 2013-02-19]. Dostupné z: <http://www.esri.com/technology-topics/cloud-gis>

- [11] Cloud service models. *Esri* [online]. 2008 [cit. 2012-03-04]. Dostupné z: <http://www.esri.com/technology-topics/cloud-gis/service-models.html>
- [12] *Computerworld: Ucelený informační zdroj pro IT profesionály*. Praha: IDG Czech, a.s., 2013, XXIV, č. 3. ISSN 1210-9924.
- [13] ČÍŽEK, Jakub. Cloud computing a zrod megadatových center. *Živě.cz* [online]. 2009 [cit. 2012-03-04]. Dostupné z: <http://www.zive.cz/Clanky/Cloud-computing-a-zrod-megadatovych-center/sc-3-a-145934/default.aspx>
- [14] ČSOB Pojišťovna: pojistné události ze záplav lze hlásit i o víkendu. *Mesec.cz: Tiskové zprávy* [online]. 2006 [cit. 2012-03-04]. Dostupné z: <http://www.mesec.cz/tiskove-zpravy/csob-pojistovna-pojistne-udalosti-ze-zaplav-lze-hlasit-i-o-vikendu/>
- [15] DIANSHENG Guo. Flow Mapping and Multivariate Visualization of Large Spatial Interaction Data. *IEEE Transactions on Visualization and Computer Graphics* [online]. 2009, [cit. 2013-02-22]. ISSN 1077-2626. DOI: 10.1109/TVCG.2009.143. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5290710>
- [16] Download ALGLIB. *ALGLIB: numerical analysis library* [online]. 2012 [cit. 2013-04-14]. Dostupné z: <http://www.alglib.net/>
- [17] File extensions: The Source for File Extensions Information. *File-Extensions*. 2011 [cit. 2011-12-23]. Dostupné z: <http://www.file-extensions.org>.
- [18] FRALEY, Keith. Is the cloud right for your organisation?. *ArcRevue*. 2011, č. 3.
- [19] Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data. *Gartner* [online]. 2011 [cit. 2013-02-23]. Dostupné z: <http://www.gartner.com/newsroom/id/1731916>
- [20] Google Apps pro firmu. *Google* [online]. 2013 [cit. 2013-02-18]. Dostupné z: <http://www.google.com/apps/intl/cs/business/>
- [21] Hadoop Documentation. *Hadoop* [online]. 2008 [cit. 2013-02-25]. Dostupné z: <http://hadoop.apache.org/docs/r0.18.3/index.html>
- [22] HEDLUND, Brad. Understanding Hadoop Clusters and the Network. *Brad Hedlund* [online]. 2011 [cit. 2013-02-25]. Dostupné z: <http://bradhedlund.com/2011/09/10/understanding-hadoop-clusters-and-the-network/>

- [23] HOWARD, Philip. Cassandra and Hadoop. *Bloor Research* [online]. 2012 [cit. 2013-02-24]. Dostupné z: <http://www.bloorresearch.com/blog/IM-Blog/2012/1/cassandra-hadoop.html>
- [24] Hype cycles. *Gartner* [online]. 2007 [cit. 2012-03-04]. Dostupné z: <http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp>
- [25] CheckFree's OpenChannel. *CheckFree Corporation* [online]. 2007 [cit. 2012-03-04]. Dostupné z: http://ir.checkfreecorp.com/phoenix.zhtml?c=94799&p=irol-newsArticle_print&ID=1077958&highlight=
- [26] Chrome devices. *Google* [online]. 2013 [cit. 2013-02-18]. Dostupné z: <http://www.google.com/intl/en/chrome/devices/>
- [27] IBM Cloud řešení. *IBM* [online]. 2012 [cit. 2012-03-04]. Dostupné z: <http://www-304.ibm.com/businesscenter/smb/cz/cs/cloud>
- [28] Install Oracle Java JDK 7 in Ubuntu 12.04. *Liberian Geek* [online]. 2012 [cit. 2013-02-27]. Dostupné z: <http://www.liberiangeek.net/2012/04/install-oracle-java-jdk-7-in-ubuntu-12-04-precise-pangolin/>
- [29] Introduction to MongoDB. *MongoDB* [online]. 2013 [cit. 2013-02-24]. Dostupné z: <http://www.mongodb.org/about/introduction/>
- [30] JANEČKA, Karel, PACINA, Jan. Výukové materiály k předmětu KMA/UGI: Další prvky personal geodatabase. *Geomatika*. 2009 [cit. 2011-12-23]. Dostupné z: <http://gis.zcu.cz/studium/ugi/cviceni/ch05s04.html>.
- [31] JANEČKA, Karel, SOUČEK, Petr. Převod prostorových dat katastru nemovitostí do formátu shapefile. *GIS Ostrava*. 2009 [cit. 2011-11-07]. Dostupné z: http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2009/sbornik/Lists/Papers/011.pdf.
- [32] KAY, Russel. QuickStudy: Data Cubes; *Computerworld* [online]. 2004 [cit. 2012-02-29]. Dostupné z: http://www.computerworld.com/s/article/91640/Data_Cubes
- [33] KLOBÁSA, Pavel. Letmý technologický pohled na MongoDB. *MoroSystems* [online]. 2012 [cit. 2013-02-24]. Dostupné z: <http://vsadnajavu.cz/2012-01/databaze/letmy-technologicky-pohled-na-mongodb/>
- [34] KOČÍ, Petr. Cloud v českém podniku? Už je to tady. *Lupa.cz* [online]. 2011 [cit. 2012-03-04]. Dostupné z: <http://www.lupa.cz/clanky/cloud-v-ceskem-podniku-uz-je-to-tady/>

- [35] KOMÁRKOVÁ, Jitka, et al. *Distributed processing of elevation data by means of Apache Hadoop in a small cluster*. Univerzita Pardubice. 2013, 5 s.
- [36] KOMÁRKOVÁ, Jitka, KOPÁČKOVÁ, Hana. *Geografické informační systémy*. Pardubice: Univerzita Pardubice, 2008. 55 s. ISBN 978-80-7395-120-7.
- [37] Learn about cloud computing. *Eucalyptus* [online]. 2009 [cit. 2012-03-04]. Dostupné z: <http://open.eucalyptus.com/learn>
- [38] Letecké snímkování. *Panoramas.cz* [online]. 2006 [cit. 2012-03-04]. Dostupné z: <http://www.panoramas.cz/letecke-snimkovani.html>
- [39] LIU, Hua, LI, De-ren, ZHU, Xin-yan. Large volume spatial data management based on grid computing. *Geoscience and Remote Sensing Symposium, 2005. IGARSS'05. IEEE International* [online]. 2005, [cit. 2013-02-22]. DOI: 10.1109/IGARSS.2005.1525208. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1525208&isnumber=32596>
- [40] MARSTON, Sean, et al. Cloud computing : The business perspective. *System Sciences* [online]. 2011, [cit. 2011-11-24]. DOI: 10.1109/HICSS.2011.102. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5718593&isnumber=5718420>
- [41] Meteosat druhé generace. *Český hydrometeorologický ústav* [online]. 2005 [cit. 2012-03-04]. Dostupné z: <http://old.chmi.cz/meteo/sat/msg/msg05a.html>
- [42] Microsoft Office 365. *Microsoft* [online]. 2011 [cit. 2012-03-04]. Dostupné z: <http://www.microsoft.com/cs-cz/office365/online-services.aspx>
- [43] MYERSON, Judith. Cloud computing versus grid computing. *IBM: DeveloperWorks* [online]. 2009 [cit. 2012-03-04]. Dostupné z: <http://www.ibm.com/developerworks/web/library/wa-cloudgrid/>
- [44] Nasazení cloudu v praxi. *BusinessIT* [online]. 2011 [cit. 2012-03-04]. Dostupné z: <http://www.businessit.cz/cz/nasazeni-instalace-cloud-openstack-amazon-google-apps.php>
- [45] NOLL, Michael. Tutorials. *Michael-noll* [online]. 2010 [cit. 2013-04-14]. Dostupné z: <http://www.michael-noll.com/tutorials/>

- [46] ODVÁRKA, Petr. Architektura CDN sítí. *Svět sítí*[online]. 2001 [cit. 2012-03-04]. Dostupné z: <http://www.svetsiti.cz/clanek.asp?cid=Architektura-CDN-siti-3082001>
- [47] PEJŠA, Jiří. (SDE) Geodatabáze: Seminární práce z předmětu Aplikace GIS. *Geomatika*. 2008 [cit. 2011-12-23]. Dostupné z: http://gis.zcu.cz/studium/agi/referaty/2008/Pejsa_Geodatabases/html/#id430672.
- [48] PETERKA, Jiří. Znáte křivku humbuku?. *DSL.cz* [online]. 2007 [cit. 2012-03-04]. Dostupné z: <http://www.dsl.cz/clanek/833-znate-krivku-humbuku>
- [49] POCHOBRADSKÁ, Jana. GIS v ČSOB Pojišťovně. *ArcRevue: informace pro uživatele software ESRI a Leica Geosystems*. Praha: Arcdata Praha, 2011, roč. 20, č. 2. ISSN 1211-2135.
- [50] Products. *Esri* [online]. 2008 [cit. 2012-03-04]. Dostupné z: <http://www.esri.com/products/>
- [51] Public versus Private Cloud. *Esri* [online]. 2009 [cit. 2012-03-04]. Dostupné z: <http://www.esri.com/technology-topics/cloud-gis/public-vs-private.html>
- [52] REZEK, Jan. Big data: na velké objemy dat musíme jinak. *Lupa.cz* [online]. 2012 [cit. 2013-02-25]. Dostupné z: <http://www.lupa.cz/pr-clanky/big-data-na-velke-objemy-dat-musime-jinak/>
- [53] SEEMANN, Pavel, JANATA, Tomáš. Datový model. *Kartografie: e-learningový portál o tvorbě map*. 2011 [cit. 2011-12-23]. Dostupné z: <http://kartografie.fsv.cvut.cz/2-1-0-datovy-model.php>.
- [54] Spatial Data Management Overview. *U. S. General Services Administration* [online]. 2011 [cit. 2012-03-04]. Dostupné z: <http://www.gsa.gov/portal/content/104479>
- [55] ŠMÍDA, Jiří. Návrh geodatabáze. *Slideshare*. 2009 [cit. 2011-12-14]. Dostupné z: <http://www.slideshare.net/jirms/nvrh-geodatabze>.
- [56] ŠMÍDA, Jiří. Využití GIS v dnešním světě. *GIS do škol* [online]. 2007 [cit. 2012-03-04]. Dostupné z: <http://gisdoskol.fp.tul.cz/index.php/vyuzitigisvdenesnimsvete>
- [57] Váša, Petr. Cloud computing v praxi. *ITBiz.cz* [online]. 2011 [cit. 2012-03-04]. Dostupné z: <http://www.itbiz.cz/zpravicky/server-nebo-cloud-to-je-to-oc-tu-bezi#>
- [58] VELTE, Anthony. *Cloud computing: praktický průvodce*. Vyd. 1. Brno: Computer Press, 2011, 344 s. ISBN 978-80-251-3333-0.

- [59] VOGT, Jim. The Future of Big Data. *Cloud computing journal* [online]. 2012 [cit. 2013-02-23]. Dostupné z: <http://cloudcomputing.sys-con.com/node/2273725>
- [60] VOJTEK, David. Úvod do GIT a Základy geoinformatiky: Studijní opory pro cvičení oboru Geoinformatika. *Institut geoinformatiky*. 2009 [cit. 2011-12-23]. Dostupné z: http://gis.vsb.cz/vojtek/index.php?page=git_c/cviceni07.
- [61] Využití GIS. *GIScom.cz: Geoinformační řešení* [online]. 2010 [cit. 2012-03-04]. Dostupné z: <http://www.giscom.cz/priklady-vyuziti-gis/>
- [62] WANG, Yonggang, WANG, Sheng, ZHOU, Daliang. Retrieving and Indexing Spatial Data in the Cloud Computing Environment. *Cloud computing* [online]. 2009, [cit. 2011-11-26]. ISSN 0302-9743. DOI: 10.1007/978-3-642-10665-1_29. Dostupné z: http://link.springer.com/chapter/10.1007/978-3-642-10665-1_29#
- [63] What is Apache Cassandra?. *DataStax* [online]. 2011 [cit. 2013-02-24]. Dostupné z: <http://www.datastax.com/technologies/cassandra>
- [64] What is QuickBooks. *Intuit* [online]. 2011 [cit. 2012-03-04]. Dostupné z: <http://quickbooks.intuit.com/what-is-quickbooks/?navClick=true>
- [65] What is SAP HANA?. *SAP HANA* [online]. 2012 [cit. 2013-02-25]. Dostupné z: <http://www.saphana.com/docs/DOC-2272>
- [66] XU, Xun. From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing* [online]. 2012, [cit. 2013-02-22]. ISSN 0736-5845. DOI: 1016/j.rcim.2011.07.002. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0736584511000949>
- [67] YANG, Chaowei, et al. Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing?. *International journal of digital earth* [online]. 2011, [cit. 2011-11-17]. ISSN: 1753-8947. DOI: 10.1080/17538947.2011.587547. Dostupné z: <http://www.tandfonline.com/doi/abs/10.1080/17538947.2011.587547>
- [68] ZHANG, Yuanzhi, et al. Spatial data cube: provides better support for spatial data mining. *Geoscience and Remote Sensing Symposium* [online]. 2005, [cit. 2013-02-22]. DOI: 10.1109/IGARSS.2005.1525227. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1525227&isnumber=32596>