VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

2017 **Jan Dvořáček**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Jan Dvořáček**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Tieto Czech s.r.o.
2. Struktura závěrečné zprávy:
   a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
   b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
   c) Zvolený postup řešení zadaných úkolů.
   d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
   e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
   f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Zdeněk Sawa, Ph.D.**

Konzultant bakalářské práce: Ing. Radek Šlachta

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 26.4.2017

Jan Dvořák

Podpis studenta

Poděkování

Rád bych poděkoval vedoucímu práce doc. Ing. Zdeňku Sawovi, Ph.D. za konzultaci a pomoc při vypracovávání této práce. Dále bych chtěl poděkovat firmě Tieto Czech s.r.o za to, že jsem zde mohl absolvovat svou praxi. A také bych chtěl poděkovat svému managerovi Ing. Josefu Mulkovi a tutorovi Ing. Radku Šlachtovi.

Abstrakt

V této bakalářské práci popisuji absolvování studentské stáže ve společnosti Tieto Czech s.r.o. Konkrétně zde popisuji oddělení Business Intelligence, technologie a nástroje, se kterými jsem se měl možnost setkat a které se v tomto oddělení používají.

V této práci také popisuji úkoly, které mi byly svěřeny, jejich výsledek a postupy, které jsem zvolil a které vedly k jejich řešení.

Klíčová slova

Business Intelligence, ETL, praxe, big data, IOT, Talend

Abstract

In this bachelor thesis, I am describing an Intership I have attended in Tieto Czech s.r.o company. Specifically I will disribe the business intelligence departement of the Tieto company, technologies and tools which I worked with and which are mostly used in this department.

In this bachelor thesis, I am describing the tasks I was working on during the internship and the methods I was using.

Key words

Business Intelligence, ETL, practice, big data, IOT, Talend

Seznam použitých zkratek

HDFS	Hadoop File System
T-SQL	Transact - SQL
IP	Internet Protocol
RDBMS	Relational Database Management System
SQL	Structured Query Language
HTTP	Hypertext Transfer Protocol

Seznam ilustrací a seznam tabulek

Obsah

Úvod

Jako formu vypracování bakalářské práce jsem si vybral provedení odborné praxe ve firmě Tieto Czech, kde jsem působil jako specialista Business Intelligence. Tuto možnost jsem si vybral, protože mě oblast Business Intelligence velmi zajímá, a protože vím, že získané znalosti a zkušenosti, které jsem získal během působení v této firmě pro mě do budoucna budou velikým přínosem.

Hlavní výhodou zvolení bakalářské praxe oproti klasické formě zpracování bakalářské práce jsem viděl v tom, že praxe mi umožní získat více v dnešní době tak ceněných praktických znalostí a zkušeností, které v budoucnu budu moct určitě uplatnit na trhu práce.

V první části této práce budu popisovat firmu, ve které jsem praxi vykonával, dále budu popisovat oddělení Business Intelligence, ve kterém jsem působil a úkoly, které mi byly zadány. V druhé částí této práce uvedu způsoby a metody, jakými jsem tyto úkoly řešil, software, který jsem při řešení úkolů použil, a nakonec uvedu znalosti, které mi byly při mé práci užitečné a naopak znalosti, které mi při vykonávání úkolů scházely.

1. O společnosti

V této kapitole uvádím informace o firmě Tieto, ve které jsem praxi vykonával a dále zde popisuji oddělení Business Intelligence, ve kterém jsem působil.

1.1. O firmě Tieto

Společnost Tieto je největším dodavatel IT služeb pro soukromý a veřejný sektor ve Skandinávii. Svými službami pokrývá mnoho segmentům, od průmyslového až po segment finanční či vládní. Jedná se o finskou firmu, založenou koncem 60. let se sídlem v Helsinkách s ročními tržbami 1,5 miliard eur, zaměstnává 13 000 expertů a působí ve více než 20-ti zemích světa. Tieto Czech je jednou z poboček společnosti Tieto, která zaměstnává přibližně 2000 expertů v oblasti IT a je třetí největší pobočkou této firmy, když první dvě místa zaujímá Finsko a Švédsko. S těmito parametry se může pyšnit jako jeden z největších zaměstnavatelů v oblasti IT služeb a největším v Moravskoslezském kraji.[1]

1.2. Působení na trhu

Společnost Tieto působí v oblasti informačních komunikací, vývoje a konzultací dostupných řešení. Tieto svým zákazníkům poskytuje vývoj softwaru podle jejich požadavků a následně stálou podporu tohoto softwaru. Dále poskytuje návrh řešení a konzultaci možných řešení, které by byly pro zákazníka nejvhodnější. Mezi největší zákazníky Tieta patří skandinávské firmy působící v oblasti zdravotnictví, dřevozpracujícího průmyslu, města a vlády.

Oddělení, na kterém jsem po dobu průběhu praxe působil, sídlí v Tieto Towers v centru Ostravy a nabízí návrh a realizaci Business Intelligence softwaru, jehož prostřednictvím se zákazníci dokáží lépe vyznat ve svých aktuálních business informacích. To umožňuje zákazníkům rychlé a kvalitní rozhodování o firemních věcech. Mezi konkrétní služby, které Business Intelligence v Tietu nabízí, například patří vytváření a budovaní datových skladů, integrace dat, reporting, vytváření sestav a jejich následná analýza.

2. Pracovní zařazení a zadané úkoly

V této kapitole budu popisovat pracovní zařazení, jednotlivé úkoly, které mi byly zadány a jejich časovou náročnost.

2.1 Pracovní zařazení

O firmě Tieto Czech s.r.o jsem se doslechl už dříve, jelikož mnoho mých přátel u této firmy absolvovalo stáž a také se jedná o jednoho z největších zaměstnavatelů v oblasti IT, díky čemuž je firma velice dobře známá. O volné pozici specialisty Business Intelligence jsem se dozvěděl z oficiálních webových stránek společnosti Tieto, na kterých jsou vypsány aktuálně otevřené pozice, a také ze seznamu pracovních pozic, ve kterém jsou nabízeny pozice pro absolvování bakalářské praxe. Po odeslání životopisu a motivačního dopisu jsem byl pozván na pohovor s personalistkou a svým budoucím manažerem Josefem Mulkou. Na pohovoru jsem musel prokázat své znalosti angličtiny, databází, jazyku T-SQL a také byla prověřena má schopnost analyticky myslet.

Po přijetí jsem se připojil k týmu Josefa Mulky, který sídlí v Tieto Towers v centru Ostravy. Po celou dobu působení v Tietu, jsem pracoval jako specialista BI(Business Intelligence). Náplní mé práce byla integrace a transformace senzorových dat a dat z WebLogů a clickstreamů, vše pomocí platformy pro datovou integraci, která se nazývá Talend.

2.2 Zadané úkoly

2.2.1 Zpracování a integrace dat získaných ze senzorů

Náplní tohoto úkolu bylo zpracovat a upravit data generována senzory, a ty následně nahrát do systému HDFS. Úkol byl rozdělen do čtyř částí, a to vytvoření schématu pro odchycení dat, vytvoření schématu pro integraci a transformaci dat a následně testování schématu a závěrečná prezentace mnou vytvořených schémat. Celkové trvání úkolu bylo osm dní.

2.2.2 Zpracování a integrace dat získaných ze senzorů s využitím MySQL

V tomto případě, bylo mým úkolem zpracovat údaje generována senzorem a ty následně nahrát do systému HDFS a databáze MySQL. Tento úkol byl také rozdělen na čtyři části, které jsou stejné jako u předchozího úkolu. Celkové trvání úkolu bylo deset dní.

2.2.3 Zpracování dat získaných pomocí clickstreamu

Mým úkolem v tomto případě bylo zpracování dat, která byla získána pomocí clickstreamu z webové stránky, na které jsou nabízeny určité produkty. Hlavní části tohoto úkolu bylo zpracovat tato data a následně z nich vypočítat zájem uživatelů o určitou kategorii produktu. Úkol byl rozdělen do dvou částí, a to vytvoření schématu pro integraci a dále prezentování mnou vytvořeného schématu. Celkové trvání úkolů bylo dvanáct dní.

2.2.4 Vytvoření schémat pro zpracování WebLogu použitím Apache Pig a metody MapReduce

V tomto úkolu jsem zpracovával data z Apache WebLogu pomocí dvou rozdílných metod a tyto metody nakonec porovnával. Úkol byl rozdělen do tří částí, a to vytvoření schématu pomocí metody MapReduce dále vytvoření schématu pomocí Pig komponentů a prezentace dosažených výsledků. Celkové trvání úkolu bylo 10 dní.

2.3 Časová náročnost

Tabulka 1: Časová náročnost

Název úlohy	Čas potřebný k vykonání vyjádřený ve dnech
Zpracování a integrace dat získaných ze senzorů	8
Zpracování a integrace dat získaných ze senzorů s využitím MySQL	12
Zpracování dat získaných pomocí clickstreamu	12
Zpracování WebLogu pomocí Apache Pig a Map Reduce	10

3. Používaný Software a použité technologie

V této kapitole popisuji software a technologie, se kterými jsem se za dobu výkonu praxe setkal a které jsem používal při řešení zadaných úloh.

3.1.1 Talend

Talend je software vytvořený společností Talend Corporation, který je určený pro datovou integraci. Typicky je používán pro zpracování big data, datové migrace, ETL (Extract Transform and Load) procesy, synchronizaci a replikace databází. Talend pracuje na principu generování kódu v jazyce Java, kdy grafické rozhraní tohoto softwaru nabízí uživateli více než 900 komponent, které se týkají běžné agregace a transformace dat, agregace dat použitím metody MapReduce a metody Pig, načítání dat z různých typů databází a datových uložišť a ukládání dat na datová uložiště. Princip práce se softwarem Talend spočívá tedy v tom, že uživatel volí vhodné komponenty, které můžeme vidět v pravé části obrázku č.1 a tyto komponenty mezi sebou propojuje. Pomocí zvolení a propojení vhodných komponent se uživatel snaží upravit vstupní data do požadovaného formátu a takto upravená data nahrát na určené datové uložiště. Jak již jsem zmínil, Talend pracuje na principu generování kódu v jazyce Java a to tak, že při zvolení určité komponenty v softwaru Talend se pro tuto komponentu automaticky vygeneruje metoda v jazyce Java, která je volána při spuštění dané komponenty. Tuto metodu může uživatel libovolně upravovat a následně použít jak v rámci Talendu tak jako samostatné skripty.

Obrázek 1: Grafické prostředí Talendu

3.1.2 Hadoop file distribuiton system

HDFS (Hadoop file distribution systém) je distribuovaný, škálovatelný a přenosný souborový systém napsaný v jazyce Java. Cluster HDFS obvykle obsahuje jeden hlavní server namenode a cluster podřízených serverů datanodes. Tyto servery mezi sebou komunikují klasicky pomocí socketů a protokolu TCP/IP. Princip použití je takový, že HDFS ukládá velká data (data v rozsahu gigabytů až terabytů) na více strojů zároveň. Tím, že data jsou rozložená na více strojích, umožňují paralelní a rychlejší zpracování[6]. Velkou výhodou HDFS oproti ostatním file systémům je fakt, že HDFS je velice odolný proti chybám a je určený k nasazení na low-cost hardwaru. Většinou HDFS běží na

linuxových systémech a dá se k jeho nastavení přistupovat jak přes webové rozhraní, tak přes příkazový řádek.

3.1.3 MySQL

MySQL je multiplatformní databáze, se kterou se komunikuje pomocí standardu SQL. MySQL je oblíbená zejména díky své výkonnosti, snadné implementovatelnosti a také zejména pro to, že se jedná o volně šiřitelný software, což je také hlavní důvod, proč má tak vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace GNU/Linux, Apache, MySQL a PHP, jako základní software webového serveru („technologie LAMP“).

3.1.4 ClickStream

ClickStream je "snímání" části počítačové obrazovky, na kterou uživatel klikne během surfování po internetu. Jakmile uživatel klikne kdekoliv na webovou stránku, informace o tom, kde klikl, jsou zaznamenány do logu, který je odeslán na webový server, nebo na router[6]. Clickstream je většinou používán při sledování aktivity na webu, nebo při průzkumu trhu.

3.1.5 MapReduce

MapReduce je programovací model pro zpracování velkých množin dat pomocí paralelního zpracování a současně také knihovna v jazyce C++, která jej implementuje.[7]

Princip fungování

1. Mějme cluster databázových nebo jiných serverů.

2. Jeden ze serverů (říkejme mu master, ale v některých modelech to může být libovolný uzel z clusteru) přijme požadavek *Map/Reduce* od klienta.

3. Uzel master rozešle funkci *Map* (nebo více zřetězených funkcí) všem ostatním uzlům clusteru, ty provedou kód této funkce a vrátí masteru výsledky, které mohou být i duplicitní (více stejných výsledků z několika uzlů — to je žádoucí kvůli odolnosti proti výpadkům). Master může také zpracovat funkci *Map* a také to v některých implementacích dělá.

4. V okamžiku, kdy má master dostatek výsledků z fáze *Map* od ostatních uzlů (a sám od sebe), nebo vyprší časový limit pro odpověď od těchto uzlů, provede master nad navrácenou množinou dat funkci *Reduce*. Fáze *Reduce* odstraní duplicitní data a provede operace, které je možné provést jen v případě, že máme kompletní množinu výsledků ze všech uzlů.

5. Na konci fáze *Reduce* je možné navrátit výsledek klientovi, který si tuto operaci zadal.

3.1.6 Apache Pig

Apache Pig je platforma pro tvorbu programů, které běží na technologii Apache Hadoop. Jazyk pro tuto platformu se jmenuje Pig latin a pomocí tohoto jazyka uživatel může vykonávat práce v MapReduce, Apache Tez, nebo Apache Spark. Pig Latin abstrahuje programování z Java MapReduce idiom do zápisu, který umožňuje MapReduce vysokoúrovňové programování podobně jako SQL na RDBMS.[8]

4. Zvolený postup při řešení zadaných úkol

V této kapitole budou popsány jednotlivé úkoly a bude zde podrobně rozepsán postup při jejich následném řešení.

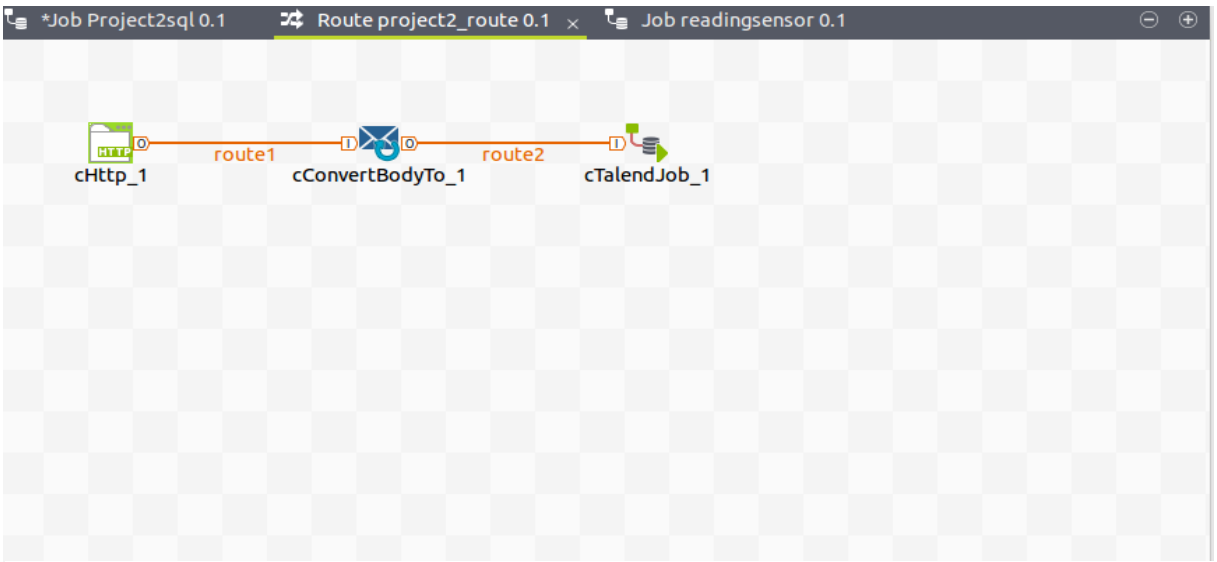
4.1 Zpracování a integrace dat získaných ze senzorů

4.1.1 Základní popis úkolu

Jedním z úkolů, který mi byl zadán, bylo zpracování dat generovaných senzorem použitím softwaru Talend. Senzor odesílá data na určitou IP adresu. Data byla generována v podobě Stringu, který byl ve tvaru SenzorX[Value]. Kdy X reprezentuje číslo senzoru a value reprezentuje hodnotu, kterou senzor snímá. Mým úkolem bylo odesílaná data odchytit, rozdělit do čitelnějšího tvaru, který vypadal následovně: název senzoru / hodnota a takto rozdělená data nahrát do HDFS.

4.1.2 Řešení úkolu

Jelikož generovaná data senzoru byla odesílána na určitou IP adresu pomocí metody HTTP post, prvním krokem, kterým jsem musel začít, bylo odchycení dat. Pro datový přenos a jeho odchycení slouží v softwaru Talend tzv. mediation route joby. Oproti klasickým integračním jobům, ve kterých se data upravují do určitého formátu a nahrávají na uložiště, je tento typ jobu určen pro datovou integraci v reálném čase využitím Apache Camel frameworku. Komponentou, která se chová jako server a umožňuje zachycovat HTTP zprávy je komponenta cHttp. Tuto komponentu jsem musel nastavit tak, aby se chovala jako server a ne jako client, a aby naslouchala na adrese a portu, na kterou jsou data odesílána. Dalším krokem bylo zvolení komponenty cSetbody, která přečte obsah těla HTTP zprávy a pošle ho do komponenty cJob, která spustí klasický integrační job, ve kterém se data upravují do určitého formátu.



Obrázek 2: Výsledný mediation route job

Nyní, když jsem data odchytil, bylo možné je již pomocí komponent v klasickém integračním jobu upravovat do požadovaného formátu. Do integračního jobu data přicházejí ve formě jednoho String řetězce a rozdělit je do určitého formátu umožňuje komponenta s názvem tExtractRegexField, kdy tato komponenta umožňuje napsat regulární výraz, jehož pomocí můžeme jeden řetězec rozdělit např. do více sloupců. Jak již jsem uvedl, v mém případě jsem měl rozdělit řetězec na sloupce Sensor_Name a Value. Jelikož obě hodnoty reprezentují celá čísla, výstupní sloupce budou typu Integer a Sensor_Name bude navíc klíč, protože jméno senzoru je pro každý senzor unikátní.

K rozdělení Stringu jsem použil tento regulární výraz:

([0-9]+).([0-9]+)

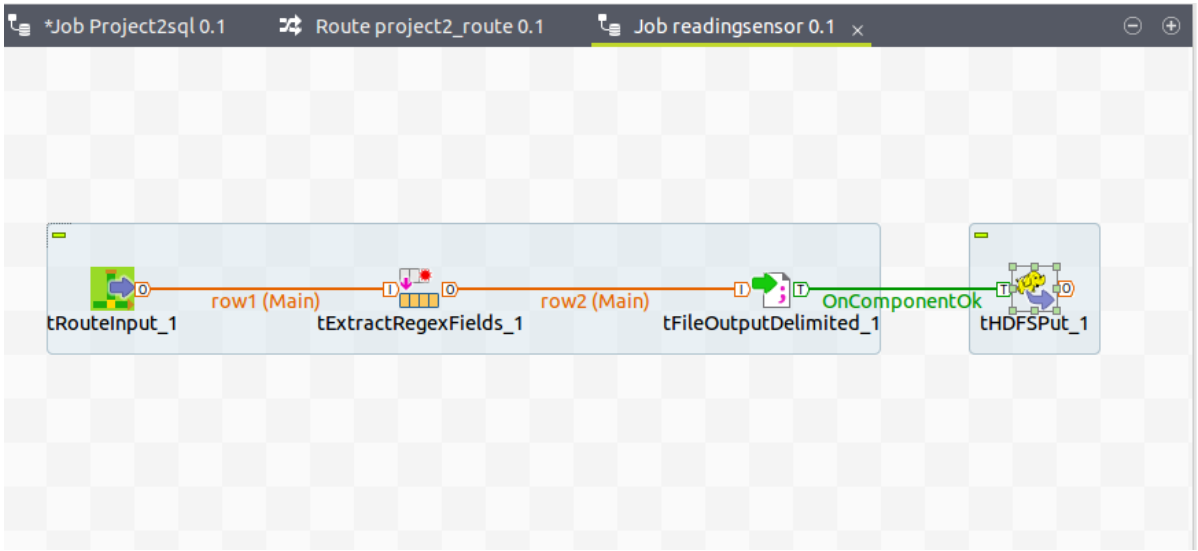
Kulaté závorky reprezentují definovaný sloupec (Sensor_name, Value), [0-9] reprezentuje číslo v rozmezí 0-9, + za číselným rozmezím říká, že čísel může být více a . znamená, že mezi dalším číslem, které má být zachyceno se mohou nacházet jiné nedefinované znaky.

Po rozdělení dat do sloupců bylo zapotřebí data nahrát do HDFS. Připojení k HDFS a metodu PUT, která nahrává data na HDFS obstarává komponenta tHDFSPut, kdy v nastavení komponenty stačí vyplnit IP adresu serveru, hostname, heslo a cestu k místu kde se mají data na serveru uložit.

Po návrhu jobu bylo zapotřebí celé schéma otestovat na testovacích datech, které generoval skript napsaný v jazyce python. Po otestování funkčnosti bylo schéma připraveno k prezentaci a reálnému nasazení v praxi.

4.1.3 Výsledek řešeného úkolu

Mnou navržené schéma, které je uvedeno na obrázku č.2 a níže v obrázku č.3, jsem předvedl manažerovi a předal svému tutorovi. Tento úkol byl součástí většího projektu v rámci offeringu služeb, při kterém byl také využit.



Obrázek 3: Výsledný integrační job

4.2 Zpracování a integrace dat získaných ze senzorů s využitím MySQL

4.2.1 Popis úkolu

Mým úkolem bylo zachytit a upravit data generovaná senzorem. Senzor generoval údaje o teplotě, tlaku vzduchu, hluku a vlhkosti vzduchu. Senzor opět vše odesílal v podobě jednoho dlouhého Stringu na určitou IP adresu pomocí HTTP post metody.

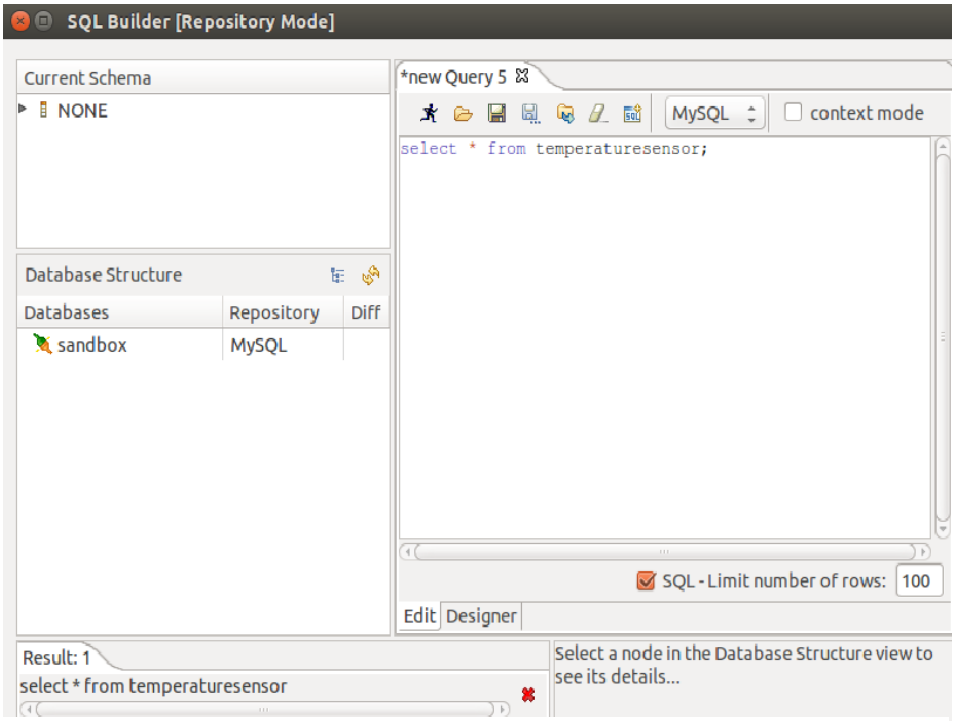
Dalším požadavkem kromě zachycení a upravení dat také bylo, aby kompletně všechna data včetně historických byla ukládána na HDFS a poslední známá teplota brána jako aktuální byla nahrána do databáze typu MySQL.

4.2.2 Řešení úkolu

K zachycení dat jsem použil úplně stejný postup jako při předešlém úkolu, jelikož se jednalo o úplně totožný případ, tak stačilo pouze změnit IP adresu a port, na kterém má komponenta cHttp naslouchat. Průběh integračního jobu už je odlišný. Data jsem opět rozdělil do sloupců pomocí komponenty tExtractRegexFields. Regulární výraz, který jsem musel použít pro rozdělení dat mi dělal potíže, jelikož s použitím regulárních výrazu k rozdělení Stringu jsem doposud neměl žádné zkušenosti. Po rozdělení dat jsem použil komponentu tMap, která slouží k mapování dat a umožňuje rozdělit jeden datový vstup na více výstupů.

Datový vstup, jsem tedy pomocí komponenty tMap rozdělil na dva výstupy a to na výstup HDFS_output, který bude poté nahrán na HDFS a Database_Output, který jak už z názvu vyplývá, bude nahrán do databáze MySQL. Komponenta tMap umožňuje mapování automaticky, což velice ulehčuje celý proces. Po mapování už jen bylo zapotřebí pro datový proud, který vede k HDFS nejprve nahrát data do souboru a ten poté nahrát na HDFS, jelikož HDFS neumožňuje přidávání dat po částech z toho důvodu, že metoda append, není na HDFS podporována.

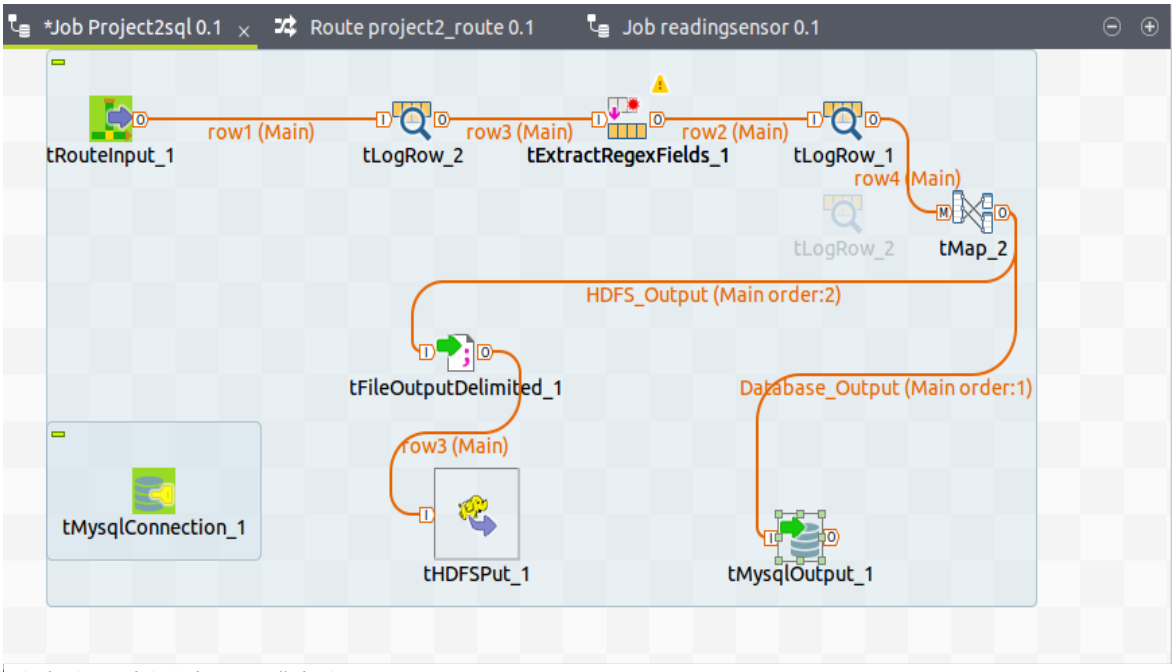
Požadavkem, který jsem musel splnit, bylo, aby data nahraná do databáze byla aktuální, tedy aby hodnota teploty byla "poslední známá". Toho jsem dosáhl tak, že komponenta tMysqlOutput, která nahrává data do databáze, obsahuje nastavení operace, která se má při zavolání metody, která reprezentuje tuto komponentu provést. Aby byla data vkládána a hodnoty aktualizovány zvolil jsem metodu insert or update, která má zá následek to, že se data pokusí přidat a pokud se v databázi již nachází položka se stejným primárním klíčem, tak se tato položka aktualizuje novou hodnotou. S databází lze komunikovat přímo v softwaru Talend a to přes rozhraní, které je uvedeno níže na obrázku č.4.



Obrázek 4: Rozhraní pro komunikaci s databází MySQL

4.2.3 Výsledek zpracovaného úkolu

Mnou navržené a testované schéma zobrazeno na obrázku č.5, bylo předáno tutorovi a prezentováno kolegům a poté dále využito v rámci většího projektu, který se týkal offeringu služeb.



Obrázek 5: Výsledný integrační job

4.3 Zpracování dat získaných pomocí clickstreamu

4.3.1 Popis

Mým úkolem bylo zpracovat a upravit clickstreamová data, které se týkala zájmu o produkt v různých evropských státech. Pro zpracovaná data bylo zapotřebí vytvořit výstupy pro analytické nástroje Google Chart a Tableu.

4.3.2 Řešení úkolu

Logovací soubor, který obsahoval informace z Clickstreamu byl již nahraný na HDFS, takže prvním krokem, který jsem musel udělat, bylo stáhnout data z tohoto souboru do softwaru Talend. Toho jsem dosáhl použitím komponenty tHDFSInput, kde k získání dat stačí vyplnit údaje nutné k připojení na HDFS (IP adresa, hostname, heslo, cesta k souboru).

V HDFS se nacházely také údaje o státech, ze kterých uživatelé jsou, produktech a samotných uživatelích. Tyto údaje bylo také nutné nahrát pomocí komponent tHDFSInput a pomocí komponenty tMap je spojit a vytvořit z nich výstup. Samotný logovací soubor obsahoval informace o datu zachycení, URL na které se kliklo, SWID, které reprezentuje uživatele, IP adresu, ze které se kliklo a State což reprezentuje stát, ze kterého IP adresa pochází.

První údaje, které byly zapotřebí propojit s údaji z logovacího souboru, byly údaje o produktu. Každý produkt je reprezentován svým URL, který je také klíčem pro každou hodnotu. Pomocí již výše zmíněné komponenty tMap jsem pomocí funkce join (kterou tato komponenta umožňuje) na sloupec URL data spojil do jednoho výstupu. Následující postup byl obdobný, zde jsem pomocí funkce join spojil dosavadní výstup s daty o státu, a to pomocí sloupce State, která reprezentuje ID každého státu. A jako poslední spojení pomocí metody join jsem spojil výstup s údaji o uživatelích pomocí sloupce SWID, které jak již se popsal výše, reprezentuje každého uživatele.

Závěrečné výstupy jsem rozdělil na dva datové proudy. Kdy jeden proud obsahoval sloupce IP, State, Cathegory, kdy z těchto sloupců se následné pomocí komponenty tAggregateRow (která jak již z názvu vypovídá, umožňuje použití agregačních funkcí) vypočítal pomocí funkce count počet IP adres z konkrétního státu, které projevily zájem o určitou kategorii produktu. Výsledek jsem setřídil abecedně podle názvu státu pomocí komponenty tSortRow a nahrál na HDFS pomocí komponenty tHDFSPut.

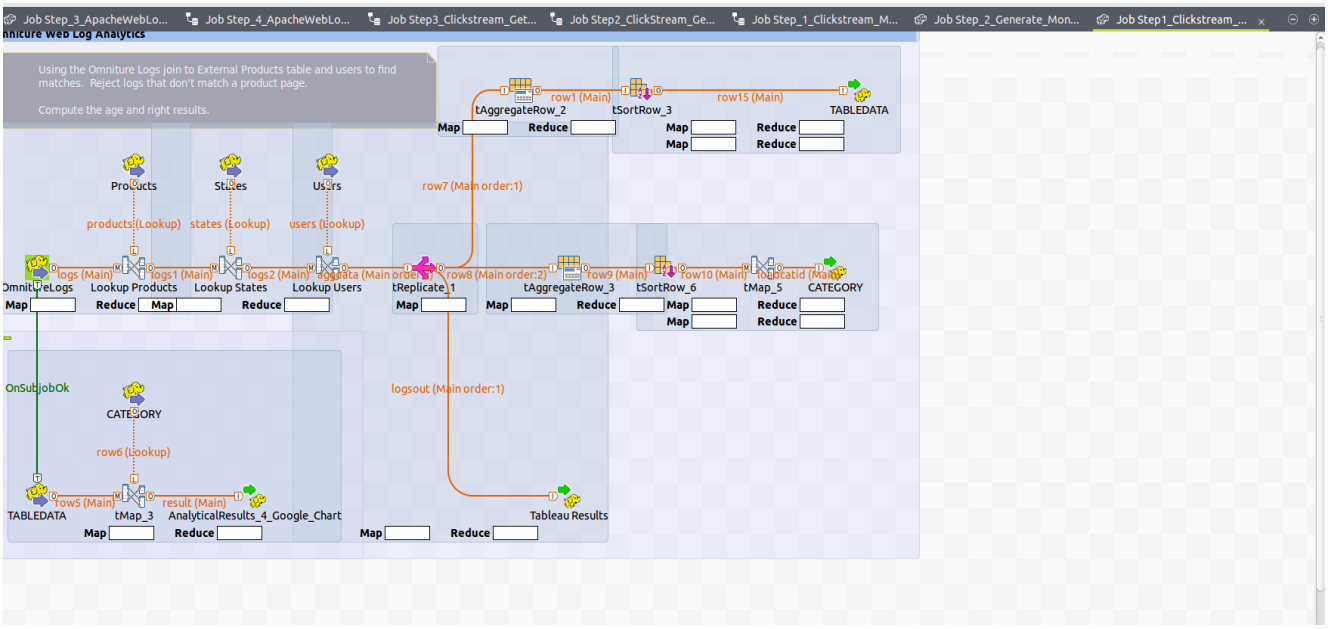
Pro datový proud IP, State a Cathegory jsem také vytvořil výstup, kde jsem opět využil komponentu tAggregateRow k výpočtu celkového počtu kliknutí na určitou kategorii výrobku. Data jsem opět setřídil vzestupně podle vypočítaného počtu návštěvníků a nahrál do HDFS. Kromě těchto výstupů bylo zapotřebí udělat ještě jeden výstup, který obsahoval všechny informace z WebLogu a následně spojených tabulek. Tento výstup byl přímo nahrán do HDFS pro pozdější analýzu v Tableau (software pro analýzu dat).

Po provedení úprav, která musela být provedena před nahráním dat do HDFS, bylo zapotřebí udělat ještě jeden výstup, který sloužil k vizualizaci dat přes Google Chart. Toho jsem dosáhl tím, že komponenta tHDFSInput se spustí až tehdy, budou-li předchozí komponenty pro úpravu dat a jejich nahrání na HDFS úspěšně dokončeny. K vytvoření závěrečného výstupního souboru pro Google Chart bylo nutné nahrát oba soubory, kde byl vypočítán zájem o kategorii produktu podle státu a celkový zájem o kategorii. A poté data z těchto dvou souborů spojit do jednoho výstupního souboru.

Ke spojení dat jsem použil opět komponentu tMap, kde jsem pomocí joinu na sloupec Category spojil data z obou souborů dohromady, tudíž výsledný výstup, který jsem poté nahrál do HDFS obsahoval informace o kategorii, státu, počtu návštěvníků dané kategorie z každého státu a celkovém počtu návštěvníků dané kategorie.

4.3.3 Výsledek zpracovaného úkolu

Výsledné schéma, které je zobrazeno na obrázku č.6, jsem otestoval a výsledek prezentoval a předal svému tutorovi. Výsledné soubory vytvořené pomocí tohoto návrhu byla dále využita při pozdější analýze dat.



Obrázek 6: Výsledné schéma pro zpracování clickstreamových dat

4.4 Zpracování WebLogu pomocí Apache Pig a Map Reduce

4.4.1 Popis úkolu

V tomto případě bylo mým úkolem demonstrovat rozdíly Big Data metody při agregování velkého množství WebLogových dat. Mezi metody, které jsem měl za úkol použít, patřily: metoda

MapReduce a jazyk Pig Latin. Tyto metody jsem měl použít k analýze a spočítání počtu IP adres z Apache WebLog souboru.

4.4.2 Řešení úkolu

WebLog soubor, který obsahoval všechna data, již byl nahraný v systému HDFS, ke kterému jsem měl přístup. Nejdříve jsem začal s metodou Pig. Pokud chceme v softwaru Talend použít metodu Pig stačí vybrat v našem integračním jobu komponenty typu Pig. Prvním krokem, kterým jsem musel začít, bylo načítání dat. Toho jsem dosáhl pomocí komponenty tPigLoad. Mezi daty, které byly nahrány z Apache WebLogu bylo mnoho chybových zpráv, ty se poznaly tak, že ve sloupci Code (který byl jeden z mnoha sloupců, které weblog obsahoval) byla hodnota 404. Tyto data bylo potřeba odstranit a toho jsem dosáhl pomocí komponenty tPigFilterRow.

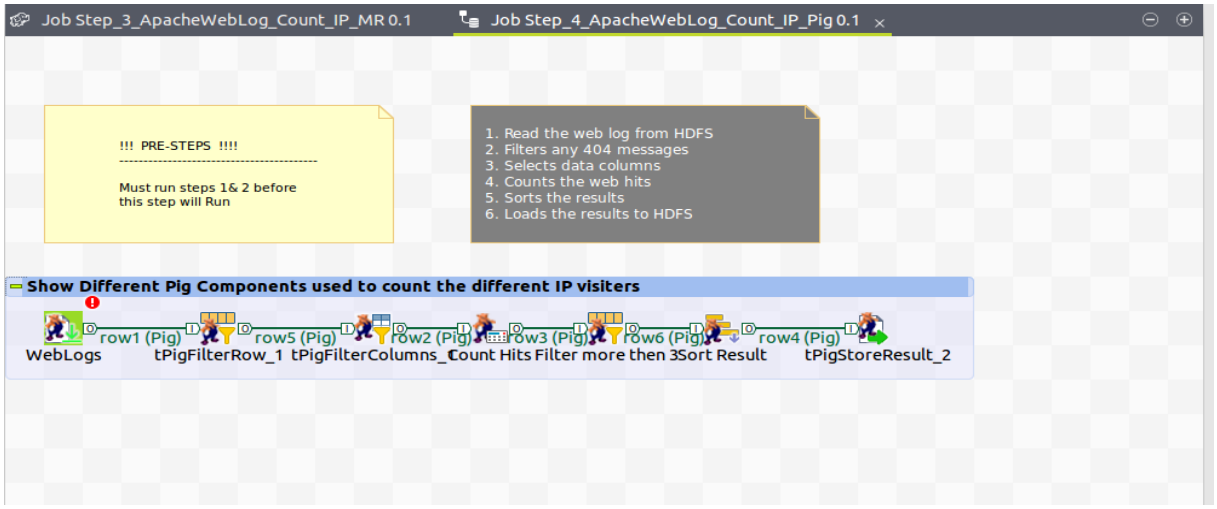
Dalším krokem bylo vyfiltrovat jen sloupce, které potřebujeme, a to jen sloupec Host. Filtrování sloupců a změnu schématu datového proudu umožňuje komponenta tPigFilterColumns, kde přes grafické rozhraní přidám do výstupu pouze sloupce, se kterým chci dále pracovat. Dalším krokem bylo již samotné spočítání počtu připojení na web pro každou IP adresu. Agregační funkce umí komponenta tPigAggregate, kde pomocí funkce count zjistíme počet připojení každé adresy.

Následujícím krokem bylo vybrat jen ty adresy, které se připojily více než třikrát, K filtraci řádků jsem opět použil komponentu tPigFilterRow, kde jsem nastavil, že hodnoty ve sloupci, které reprezentuje počet připojení, který jsem vypočítal v předchozí komponentě, musí být greather than (větší než) tři. Předposledním krokem, který bylo zapotřebí provést, bylo data setřídit. Toho jsem dosáhl pomocí tPigSort, kde jsem nastavil, že data mají být setříděny sestupně podle počtu připojení. Výsledné data jsem poté nahrál zpět na HDFS pomocí komponenty tPigStore. Job pro klasické MapReduce je naprosto totožný až na ten rozdíl, že se při něm nepoužívají komponenty typu Pig, ale klasické integrační komponenty.

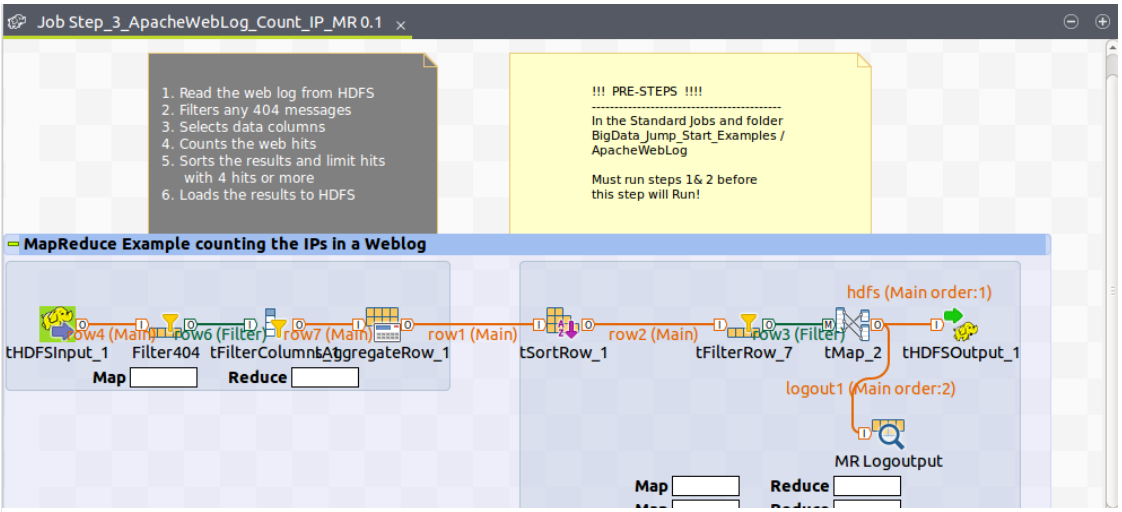
Pokud bych měl zhodnotit rozdíl v práci s klasickým MapReduce a jazykem Pig Latin v softwaru Talend, tak jediný rozdíl je v rychlosti výpočtů, která je značně vyšší u Pig Latin než u klasického MapReduce.

4.4.3 Výsledek zpracovaného úkolu

Výsledné schémata znázorněné níže na obrázcích č.7 a č.8 a mnou zaznamenané poznatky byly popsány mému tutorovi a kolegům. Poznatky byly použity kolegy při budoucí práci se softwarem Talend.



Obrázek 7: Výsledné schéma zpracované pomocí Pig komponentů



Obrázek 8: Výsledné schéma zpracované pomoci MapReduce

5. Použité a chybějící znalosti

5.1 Uplatněné teoretické a praktické znalosti

Během průběhu praxe ve firmě Tieto jsem pracoval s daty a zpracovával je. Při zpracovávání dat mi byly užitečné znalosti, které jsem získal v databázových předmětech, a to především v úvodu do databázových systémů a technologie databázových systémů. Také jsem zužitkoval znalosti týkající se programovacího jazyku Java. Jelikož software, který jsem při zpracování a integraci dat používal, generuje metody v jazyce Java automaticky, ale někdy bylo potřeba, některé metody upravit ručně a k tomuto mi posloužily znalosti získané v předmětu programovací jazyky 1. V některých úlohách jsem pro simulaci senzorů, které generují data, používal skripty napsané v jazyce Python. Skript bylo nutné často upravovat. Při těchto úpravách mi pomohly znalosti, které jsem získal v předmětu skriptovací jazyky a jejich aplikace. Při své práci jsem také ve většině případů pracoval s operačním systémem linux, konkrétně s distribucí Ubuntu a to jak v grafickém prostředí, tak s příkazovým řádkem. Při těchto činnostech mi byly užitečné znalosti z předmětu správa operačních systémů.

5.2 Scházející znalosti

Během průběhu mé praxe byla pro mne největší překážkou absolutní neznalost softwaru a principů používaných pro integraci a zpracování velkých dat. Také s technologiemi jako je Apache Hadoop, Apache Spark či Data Streaming jsem se neměl možnost setkat, jelikož před začátkem praxe jsem se setkal pouze s vytvářením reportů což je jiné odvětví činností Business Intelligence, musel jsem se prakticky všechny technologie a principy učit za chodu.

Velkým pomocníkem pro mne byly, jak jsem již zmiňoval znalosti z databázových předmětů. Díky základům, které mi tyto předměty daly, např. určování cizích klíčů a upravování dat tak, aby splňovaly zákony normálních forem, bylo jednodušší se všechny principy učit a jejich osvojení už poté nebyl tak velký problém.

Závěr

Po dobu vykonávání odborné praxe ve firmě Tieto, jsem zpracovával data generována senzory, clickstreamová data a data z WebLogu. Vše pomocí softwaru Talend určeného pro datovou integraci. Během mého působení v Tietu jsem se také setkal a osvojil technologie jako Apache Hadoop, databáze typu MySQL a již zmíněný software Talend. Vykonání odborné praxe mi umožnilo získat spoustu praktických znalostí a zkušeností z oblasti Business Intelligence. Velký přínos vidím také v tom, že jsem si vyzkoušel práci v mezinárodní společnosti, která se řadí mezi jedny z největších zaměstnavatelů v oblasti IT. Během působení v Tietu, které trvalo téměř 7 měsíců, jsem měl možnost seznámit se podrobněji s chodem společnosti a chodem oddělení, ve kterém jsem působil. Business Intelligence byl již delší dobu obor, o který jsem se zajímal a chtěl jsem poznat, jak fungují věci v praxi a tato praxe mi to umožnila.

Celkově z mé strany hodnotím praxi velmi kladně, jelikož jsem se naučil spoustu nových věcí, naučil jsem se pracovat pro mě s doposud neznámými technologiemi a zjistil jsem, jaké to je pracovat ve velké mezinárodní firmě.

Použitá literatura

[1] Informace o Tieto. Tieto -IT, výzkum a vývoj a poradenství. [online]. 2015 [cit. 2017-04-14]. Dostupné z: http://www.tieto.cz/tieto-o-nas

[2] Historie společnosti Tieto. Historie -Tieto -Czech Republic [online]. 2015 [cit. 2017-04-14]. Dostupné z: http://www.tieto.cz/tieto-o-nas/historie-tieto-czech-republic

[3] HDFS. [online]. [cit. 2017-04-14].
Dostupné z: https://hadoop.apache.org/docsr1.2.1/hdfs_design.html

[5] MySQL. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-14]. Dostupné z: https://cs.wikipedia.org/wiki/MySQL

[6] Clickstream analysis. *Searchcrm.techtarget.com* [online]. [cit. 2017-04-14].
Dostupné z: http://searchcrm.techtarget.com/definition/clickstream-analysis

[7] MapReduce. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-14]. Dostupné z: https://cs.wikipedia.org/wiki/MapReduce

[8]Apache Pig. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-14]. Dostupné z https://en.wikipedia.org/wiki/Pig_(programming_tool)