Distribuovaný kolektor záznamů o IP tocích: Experimenty s big data platformami

CESNET technical report 6/2015

Martin Žádník, Pavel Krobot, Lukáš Kekely, Viktor Puš, Jan Kořenek

Přijato 30. 6. 2015

Abstrakt

Monitorování síťového provozu vyžaduje sběr informací exportovaných z důležitých uzlů sítě. Vzhledem ke značnému růstu množství sbíraných informací je nutné hledat možnosti distribuce úlohy sběru a následného dotazování nad uloženými daty. V nedávné době byly představeny platformy a databázové systémy, které dovolují pracovat s velkými objemy dat. Vybrané platformy byly testovány právě na úloze distribuovaného sběru a dotazování na daty získávanými ze sítě (data o IP tocích). Tato technická zpráva popisuje samotné experimenty a podává shrnutí výsledků těchto experimentů.

Klíčová slova: distribuce, kolektor, data, toky

1 Úvod

Počítačové sítě (ať už domácí sítě, podnikové sítě či sítě poskytovatelů a operátorů) se staly nedílnou součástí fungování společnosti. Vzhledem k vysokým požadavkům na spolehlivost těchto sítí a jejich ochrany jsou z klíčových míst sítí sbírána data o síťovém provozu. Toto množství dat se zvyšuje s rostoucím množstvím aplikací provozovaných přes síť, s nárůstem provozu z mobilních a jiných inteligentních zařízení. Rovněž samotné útoky způsobují velký nárůst dat a kladou tak vysoké nároky na robustnost sběrného bodu při ukládání dat a rovněž vysoké nároky na výkon při dohledávání událostí v uložených datech. Je tedy důležité dosáhnout potřebný výkon, kapacitu a škálovatelnost sběrného bodu (dále jen kolektoru). Technologie používané v současných kolektorech převážně nedovolují distribuovat úlohu kolektoru na více uzlů. Klasické databázové přístupy dosáhly svých limitů a zpracování velkého objemu síťových dat vyžaduje nové přístupy. Zejména nelze používat přístupy založené na běžných relačních databázích, především kvůli velkému objemu příchozích dat, kdy není prostor na vytváření indexu, dynamičnosti dat (data nemají pevnou a předem danou strukturu) a v neposlední řadě kvůli stylu práce s daty, kdy uložená data se již nemění a není třeba zajišťovat atomicitu transakcí. Pro interaktivní práci s daty je nutné se zaměřit na mechanismy, které umožní data zpracovávat dostatečně efektivně, v závislosti na jejich aktuální velikosti a dostupných výpočetních a úložných zdrojích. Je potřeba tedy hledat mechanismy, které se dokáží přizpůsobovat dostupným kapacitám a dynamicky navyšovat, příp. snižovat jejich využití podle potřeb uživatele. Slibným přístupem se jeví model MapReduce, tedy minimalizace přenosu dat mezi uzly a distribuce výpočtu na uzly

uchovávající dat. Z tohoto důvodu hledáme platformu, která by se stala základem distribuovaného kolektoru, který bude schopen pojmout velký objem dat a především provádět velmi rychlé dotazy nad uloženými daty.

Za účelem získání znalostí o vlastnostech platforem určených pro distribuované zpracování velkého množství dat jsme provedli analýzu několika veřejně dostupných možností. Slibnou platformou se jeví Apache Hadoop [4]. Apache Hadoop je volně dostupný framework, který realizuje spolehlivé distribuované výpočty na rozsáhlých datech s využitím počítačového clusteru v paradigmatu MapReduce. Apache Hadoop byla již jako platforma pro kolektor použita a výsledky publikovány v [3]. Naším cílem je ovšem vytvořit kompletně distribuovaný kolektor, včetně příjmu dat, jejich okamžitého zpracování a v neposlední řadě uložení. Z tohoto důvodu plánujeme ověřit některé technologie pro zpracování velkého množství dat, i ty které již byly publikovány, neboť idealizované prostředí již publikovaných experimentů často ovlivní výsledky. Nad Apache Hadoop jsou dále dostupná rozšíření, která zjednodušují práci s daty, například rozšíření Hive implementující SQL-like rozhraní a Pig implementující rozhraní v podobě funkcionálního programování. Další kandidáty pro distribuované uchování a dotazování tvoří volně dostupné distribuované databázové platformy. V této práci jsou experimenty provedeny s databázovými systémy ElasticSearch a Vertica.

ElasticSearch je moderní fulltextový vyhledávač a bezeschémová databáze, založená na projektu Apache Lucene. Tato cloudová technologie umožňuje dynamicky přidávat a ubírat uzly clusteru podle aktuální potřeby (nárůst dat, zvýšený nebo naopak snížený počet dotazů). Tato vlastnost je implementována především schopností autodetekce okolních uzlů na stejné síti protokolem multicast, automatického rozkládání databází/indexů a jejich částí na jednotlivé uzly a automatickou replikací nebo přesuny dat podle aktuální konfigurace clusteru (např. při selhání jednoho nebo více uzlů). Těmito technikami ElasticSearch dosahuje vysoké dostupnosti, horizontální škálovatelnosti a minimalizaci režie spojené se správou clusteru.

Vertica Analytic Database je databázová platforma určená k managementu velkých dat. Hlavní koncepty jsou sloupcově orientované ukládání dat, což urychluje dotazy, které vyžadují pouze některé sloupce tabulky. Sloupcové ukládání urychluje dotazy, které nepracují se všemy položkami (tj. sloupci) jednotlivých záznamů. Z disku pak nemusí být oproti řádkovému uspořádání čtena data, jenž nejsou požadována. Dalším důležitým aspektem je kódování a komprese dat (je proveden automatický výběr vhodného kódování (RLE, delta, float compression)), distribuce a replikace dat mezi uzly v rámci clusteru, což zajišťuje škálovatelnost a spolehlivost.

2 Podklady

2.1 Dotazy

Pro účely experimentů s distribuovaným zpracováním dat byly navrženy 4 dotazy, které se snaží svou strukturou zachytit typické dotazy, používané při analýze dat ze síťového provozu. Jedná se o následující 4 dotazy:

1. Výpočet celkového počtu toků, sumy počtu paketů a sumy bytů v nich obsažených ze všech záznamů o tocích. Výsledkem jsou tedy tři hodnoty – celkový počet toků, paketů a bytů.
2. Získání celkového počtu všech záznamů s cílovým portem 53. Výsledkem tohoto dotazu je pak jediná hodnota.
3. Další dotaz vybírá jen zvolená pole (časová značka příchodu záznamu, protokol, zdrojová a cílová IP adresa, zdrojový a cílový port, počet paketů a počet bytů) pro záznamy o IP tocích přenášené spolehlivým protokolem TCP na portu 53. Výstupem dotazu jsou jednotlivé řádky obsahující požadované záznamy.
4. Poslední ze sady dotazů pro každou zdrojovou adresu počítá sumu paketů, bytů a celkový počet záznamů přenesených z této adresy pomocí protokolu TCP. Výsledkem jsou jednotlivé agregované záznamy, které jsou seřazeny dle počtu záznamů a předávané na výstup po řádcích.

2.2 Datová sada

Pro experimentování byla použita datová sada s anonymizovanými daty o IP tocích z jednoho dne reálného provozu. Datová sada byla rozdělena na soubory s přírůstkem dat odpovídajícím jedné hodině provozu (tj. 0/24, 1/24, 2/24, ..., 24/24 z celkového množství dat). Celkem testovací data obsahovala zhruba 880 mil. záznamů. Tento počet záznamů odpovídá provozu na 10 Gb/s páteřní lince, přes kterou bylo v průběhu 24 hodin přeneseno 30 mld. paketů a 27 terabytů dat. Pro účely experimentů jsou používány velmi jednoduché záznamy. Každý ze záznamů obsahuje pouze následující základní položky:

1. zdrojová IP adresa,
2. cílová IP adresa,
3. zdrojové číslo portu,
4. cílové číslo portu,
5. číslo protokolu,
6. startovní časová značka,
7. koncová časová značka,
8. počet paketů,
9. počet bytů,
10. TCP příznaky.

Výsledná datová sada je zapsána textově ve formátu CSV. Velikost datové sady v tomto formátu je 86 GB. Pro účely některých experimentů byla datová sada uložena i v binárním formátu. V tomto formátu zabírá datová sada 56 GB.

3 Experimenty

Experimenty byly prováděny s výše popsanou datovou sadou. Bohužel nebylo možné zajistit konzistentní prostředí pro různé platformy. Z tohoto důvodu je u každého experimentu dobře popsána konfigurace daného hardware. Z důvodu přehlednosti jsou některé experimenty, které neobsahují zajímavá data vypuštěny.

3.1 Platformy založené na hadoop

V této sekci jsou popsány experimenty s platformami, založenými na systému Hadoop. Zkoumanými přístupy byly vlastní implementace dotazů v jazyce Java, které byly následně spouštěny nad daty ve formátu CSV a nad binárními daty. Dále jsme použili dvě rozšíření pro Hadoop, Hive a Pig, poskytující rozhraní pro dotazování nad uloženými daty. Tyto experimenty se kromě testů zaměřených na porovnání výkonnosti vůči tradičnímu, vysoce optimalizovanému kolektoru NfDump [1], zaměřovaly zejména na optimalizaci dotazování nad daty o IP tocích (odstranění latence apod.) a na zjištění možností paralelního zpracování více dotazů.

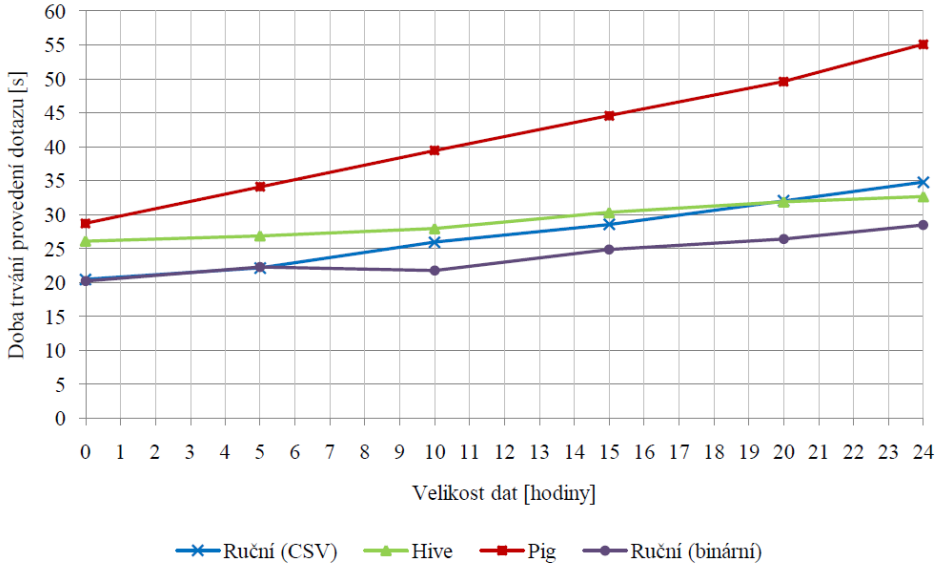
Pro tyto experimenty byl použit specializovaný Hadoop cluster, poskytnutý VO MetaCentrum [2]. Cluster se sestává z 27 strojů, vybavených šestnáctijádrovými procesory (hyperthreading, Intel® Xeon® CPU E5-2630 v3 @ 2.40GHz), 128 GB operační paměti na každém z nich a s úložištěm o celkové velikosti 1.02 PB. Vzhledem k povolené 4-násobné replikaci je celková kapacita 261 TB. V experimentech, cílených na optimalizaci systémů pro dotazování, byl zkoumán vliv změny různých konfiguračních parametrů na dobu odezvy dotazů. Sledované parametry a jejich hodnoty budou uvedeny v následujícím textu u popisů jednotlivých experimentů.

Při experimentech nebyly z důvodu úspory času při dlouho trvajících testech spouštěny dotazy nad daty z každé hodiny, ale pouze nad vybranými datovými celky, jenž odpovídaly daným počtům hodin. Nad těmito celky byl každý ze 4 dotazů spuštěn třikrát. U každého experimentu byla měřena doba trvání dotazu od jeho spuštění po získání úplného výsledku. Výsledná doba trvání dotazu se pak spočítala jako průměr trvání jednotlivých dotazů. V následujícím textu budou v jednotlivých částech uvedeny různé parametry, s jejichž hodnotami bylo experimentováno. Výsledky těchto experimentů jsou demonstrovány na grafech časů provádění dotazu č. 2 jednak z důvodu omezení rozsahu, jednak proto, že výsledky dalších dotazů neukazují žádné další významné informace.

První graf zobrazuje výsledky měření Hadoop včetně svých rozšíření. Graf zachycuje trvání zpracování druhého dotazu, tj. dotazu využívající pouze filtraci. Ve výsledcích měření (především na hodnotách pro 24 hodinovou sadu) je možné pozorovat při srovnání se

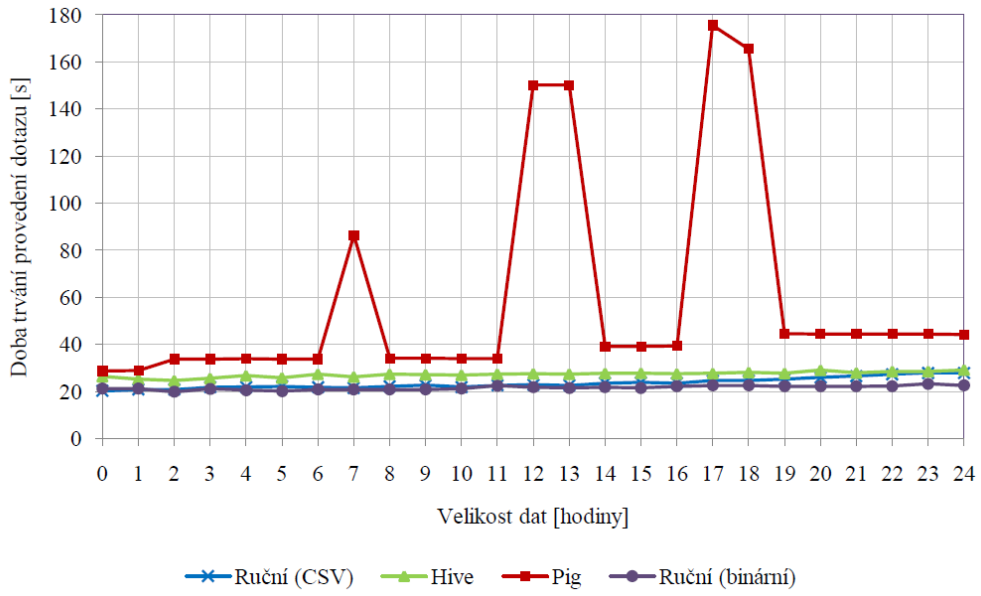
základními experimenty, výrazné zrychlení při zpracování větších objemů dat za použití většího clusteru oproti předchozím měřením, které probíhali na clusteru 10 strojů s nižší konfigurací [5]. Rovněž je možné pozorovat, že se zvyšujícím se množství dat má pouze minimální vliv na dobu běhu dotazu, tj. doba trvání dotazu mírně roste.

Velmi důležité je zaměřit pozornost na výsledky v oblasti první hodiny. Výsledky ukazují, že i nad téměř prázdnou datovou sadou jsou výsledky vráceny až po dvaceti vteřinách. Tato prodleva souvisí s režií Hadoop distribuce dotazů a sběru výsledků a cílem je tuto režii optimalizovat. Pokusy o optimalizaci jsou zachyceny v dalších experimentech. Výše popsaný experiment byl proveden s následujícími parametry. Heartbeat interval pro výměnu zpráv mezi jednotlivými uzly byl nastaven na 3 sekundy a replikační faktor nastaven na hodnotu 4. V rámci tohoto experimentu byl spouštěn vždy pouze jediný dotaz samostatně v daný čas (tj. nedocházelo k paralelnímu spouštění úloh).



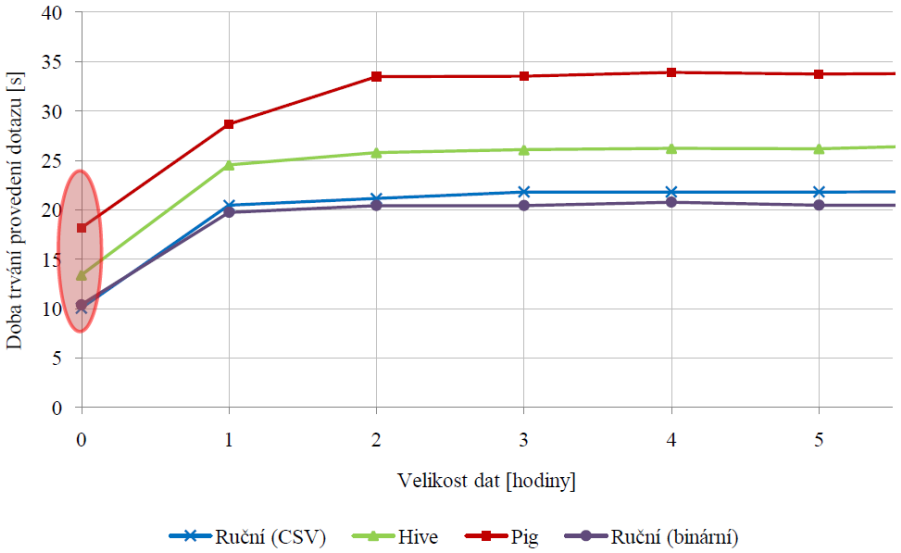
Obrázek 1. Dotaz č. 2 s výchozím nastavením parametrů Hadoopu.

První z parametrů, na který jsme se při optimalizaci konfigurace zaměřili, byl interval výměny zpráv jednotlivých uzlů, tzv. heartbeat interval. Cílem ladění tohoto parametru bylo snížení počáteční latence při zpracování dotazů menšího až středního rozsahu. Obrázek Obrázek 2 zobrazuje graf naměřených hodnot při nastavení heartbeat intervalu na 1 sekundu. Z výsledků je vidět, že změna tohoto parametru nepřinesla žádné zlepšení v délce odezvy provádění dotazu. Naopak přinesla spíše zhoršení spolehlivosti, jenž lze pozorovat u platformy Pig, kde některé dotazy trvají neúměrně dlouhou dobu.



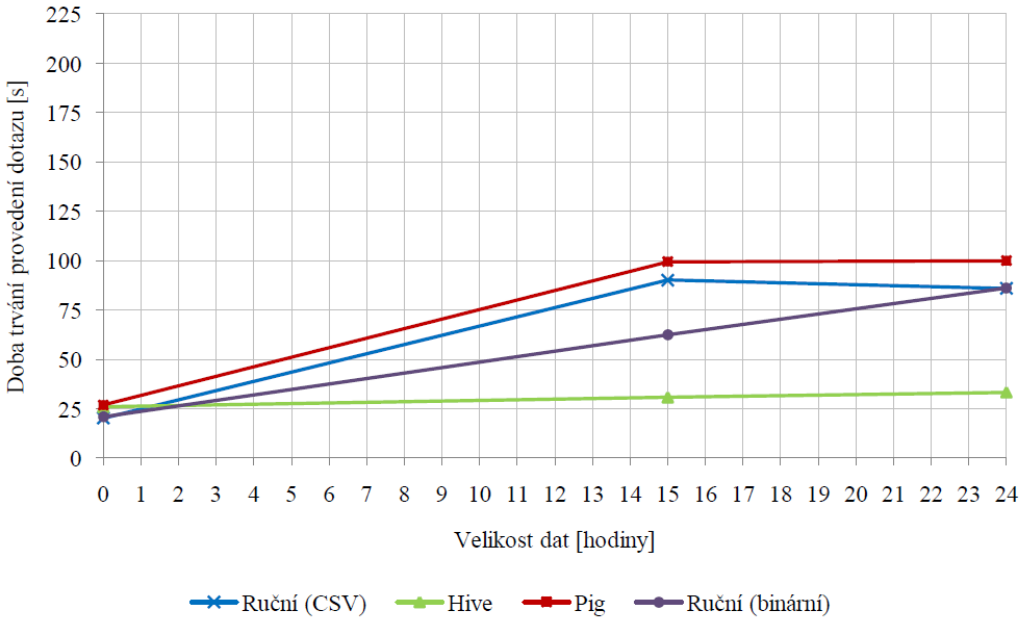
Obrázek 2. Dotaz č. 2 s nastavením heartbeat intervalu na 1 sekundu.

Pro dotazy pracující s menšími objemy dat byl v dalším experimentu testován tzv. über mód, který je poskytován Hadoopem v novější verzi (hadoop-0.23, resp. YARN Hadoop2) právě pro úlohy menšího rozsahu, jenž je možné provést lokálně na jednom z pracovních uzlů. Ve výchozím stavu je tento mód vypnutý. Obrázek 3 zobrazuje výsledky testování se spuštěným über módem. Z grafu je zobrazena pouze část s několika prvními testy, na které bylo toto testování zaměřeno nejvíce. Je zde vidět zlepšení při dotazu na nejmenší množství dat, nicméně tento dotaz pracuje pouze s malým souborem o deseti tocích, který je v experimentech zařazen spíše pro odhad režie při minimální výpočetní zátěži. Z principu činnosti über módu lze odhadnout, že přínos pro zpracování požadavku lze očekávat pouze při úlohách pracujících s daty ne většími, než je velikost HDFS bloku, protože právě taková data se mohou vyskytovat na jednom výpočetním uzlu. Přestože über mód přináší zrychlení pro velmi malé dotazy, je počáteční prodleva stále dlouhá a pro vyvíjený kolektor nevhodná. Dalším negativem při použití über módu, byl zvýšený výskyt chyb při zpracování dotazů prostřednictvím rozšíření Hive.

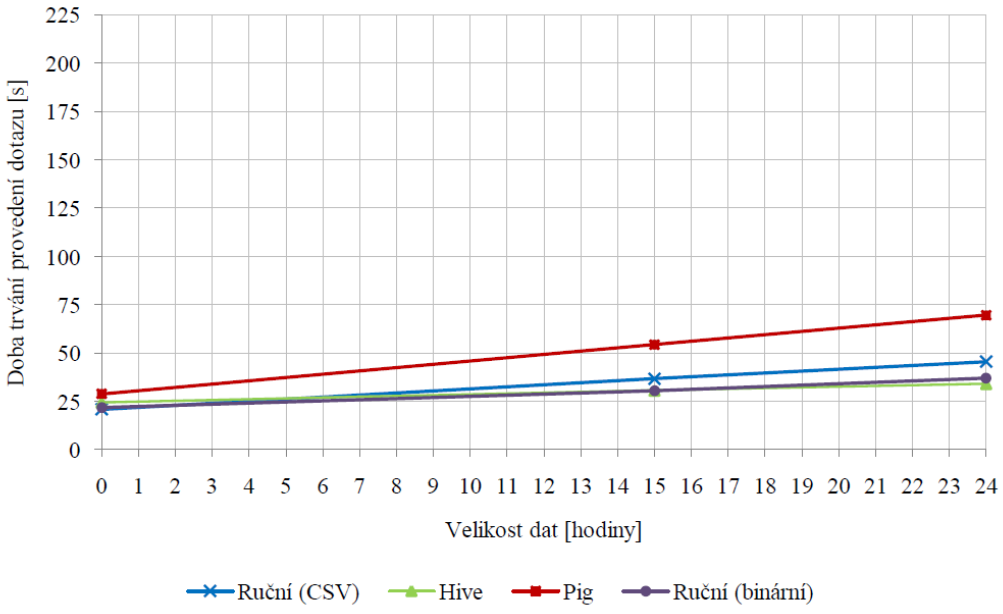


Obrázek 3. Dotaz č. 2 se spuštěným über módem (prvních 5 hodin dat).

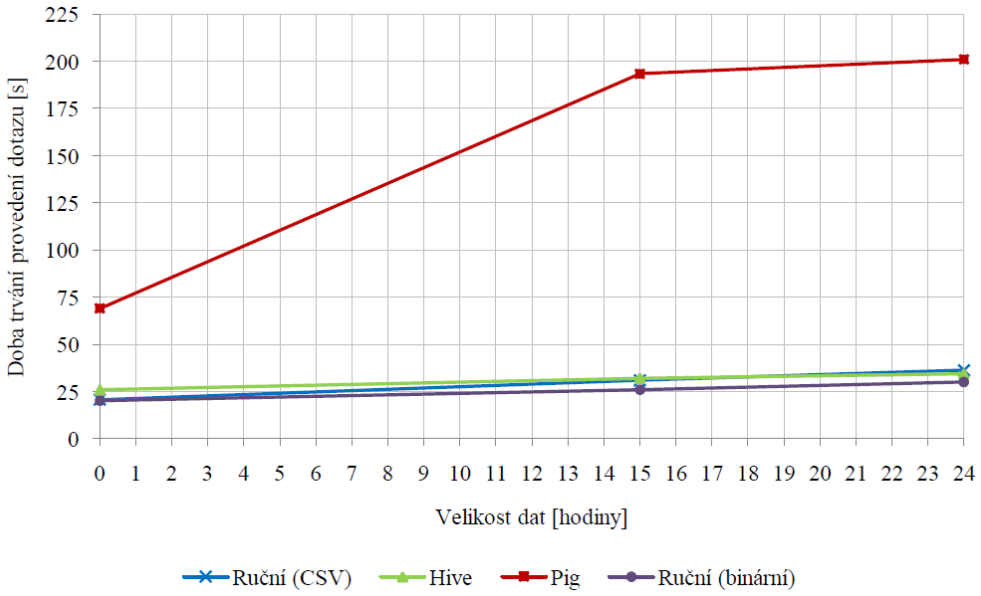
Dalším sledovaným parametrem byl replikační faktor. Jeho změna může mít vliv jednak na rychlost výpočtu, kdy přítomnost více kopií dat v systému umožňuje jemnější rozdělení dílčích úloh a lepší rozložení zátěže mezi uzly. Dále replikační faktor ovlivňuje také spolehlivost výpočtu a zálohování dat, kdy je při výpadku jednoho uzlu zajištěna přítomnost jiné kopie dat v systému. Kromě výchozí hodnoty replikačního faktoru, jenž byl roven 4 kopiím každého bloku dat, jsou na následujících grafech zobrazeny výsledky z pokusů pro hodnoty replikačního faktoru 1, tj. bez replikace, 2 a 6. Z výsledků je možné pozorovat výrazné zpomalení výpočtu při zrušení replikace (replikační faktor 1) a o něco mírnější zhoršení při replikačním faktoru 2. Je tomu tak kvůli neoptimálnímu rozložení zátěže, kdy může docházet k nerovnoměrnému vytížení uzlů, majících uloženu větší část aktuálně požadovaných dat a ke zvýšeným přenosům dat mezi uzly, které nemají k dispozici momentálně potřebná data. Dalším negativem těchto dvou nastavení je zhoršení zálohování dat, jenž však nebylo hlavním předmětem těchto experimentů. Ve srovnání s experimenty s výchozím nastavením Hadoopu bylo dosaženo s replikačním faktorem 6 obdobných výsledků z hlediska délky provádění dotazů. Vzhledem k tomu, že toto nastavení nepřináší zkrácení odezvy, lze považovat další zvyšování replikačního faktoru za nevhodné, jelikož s sebou přináší prodloužení doby ukládání dat a nárůst prostoru potřebného pro uložení dat. Zlepšení zálohovacích vlastností oproti replikačnímu faktoru 4 již není zásadní.



Obrázek 4. Dotaz č. 2 s nastavením replikačního faktoru na hodnotu 1 (tj. bez replikace).

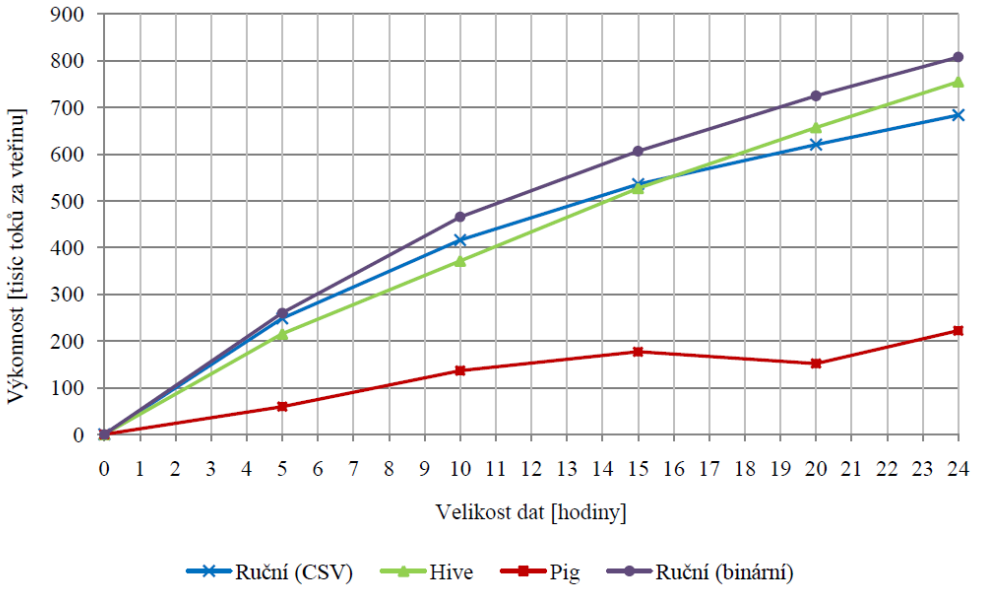


Obrázek 5. Dotaz č. 2 s nastavením replikačního faktoru na hodnotu 2.



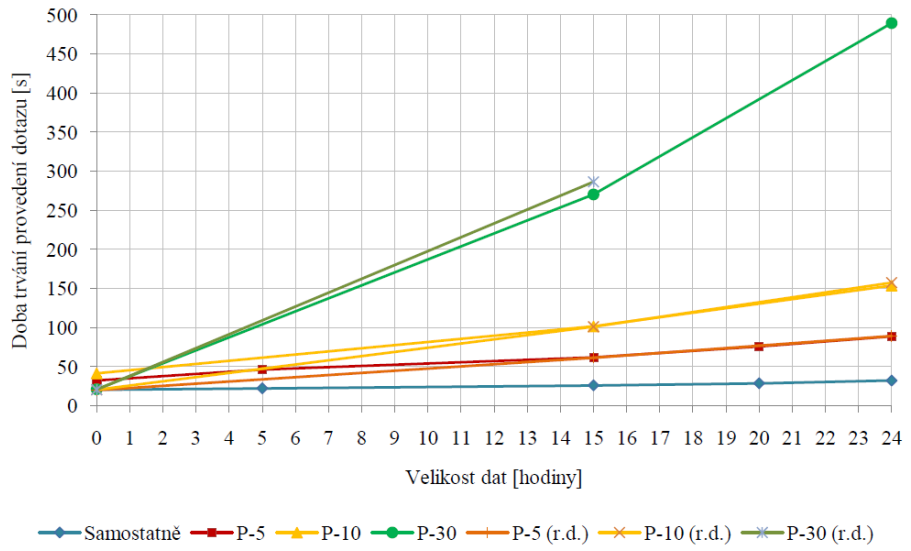
Obrázek 6. Dotaz č. 2 s nastavením replikačního faktoru na hodnotu 6.

Vzhledem k tomu, že Hadoop řešení využívá více uzlů, nabízí se otázka, jaká je výkonnost/efektivita jednoho uzlu v takovém řešení. Výkonnost systému je v následujících výsledcích vyjádřena počtem toků zpracovaných za vteřinu jedním uzlem. Výkonnost jednoho uzlu je vypočtena podílem množství zpracovaných toků vůči počtu vteřin doby zpracování a počtu zpracovávajících uzlů. Tento přepočet výkonnosti alesoň částečně dovoluje porovnávat výsledky naměřené na různě velkých clusterech, se kterými jsme postupně v průběhu experimentů pracovali. Výkonnost systémů je zachycena na obrázku Obrázek 7. V grafu je možné sledovat růst výkonu s větším množstvím zpracovávaných dat, kdy se podíl režie na celkové době provádění dotazu snižuje s růstem doby skutečného zpracování dat.

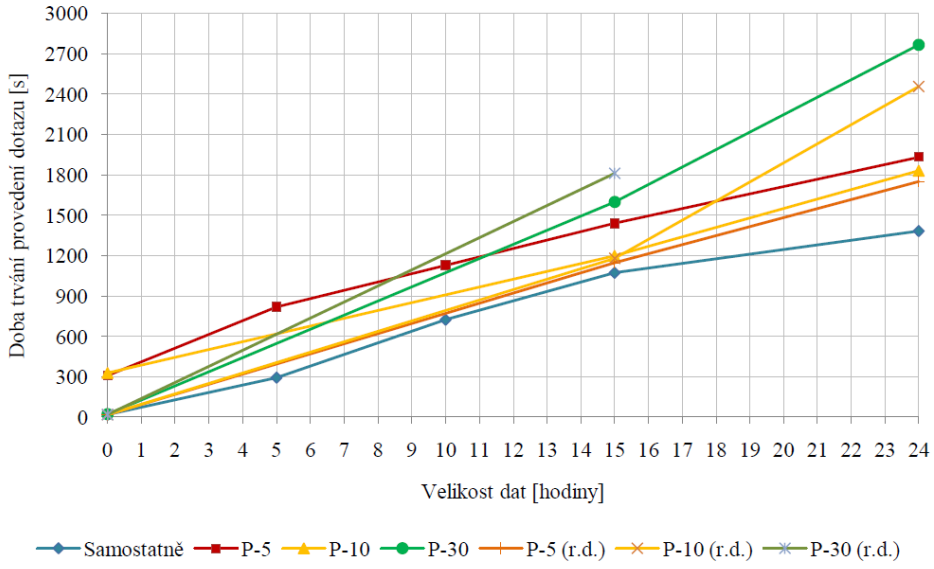


Obrázek 7. Výkonnost přepočtena na jeden uzel pro Hadoop experimenty.

Významným zkoumaným aspektem systémů byla také jejich schopnost zpracovávat více dotazů současně. Obrázek Obrázek 8 ukazuje vliv spuštění více paralelních úloh. Obrázek Obrázek 9 pak zobrazuje výsledky stejného typu naměřené pro dotaz č. 4. Ty jsou zde uvedeny, protože oproti ostatním experimentům poskytují vůči dotazu č. 2 další dodatečnou informaci v podobě projevu většího přenosu dat na dobu trvání dotazů při zpracování více úloh současně. Na těchto grafech jsou zobrazeny výsledky pro spuštění jediné úlohy (referenční hodnota), pro vícenásobné spuštění dané úlohy 5x, 10x a 30x nad stejným datovým souborem (v grafech značeno P-5, P-10 a P-30) a stejné počty úloh pro rozdílné datové soubory, kdy byla pro každou spuštěnou úlohu vytvořena zvláštní kopie zdrojového datového souboru (hodnoty označeny navíc r.d.), aby nebylo možné datovou sadu cachovat v diskové cache).



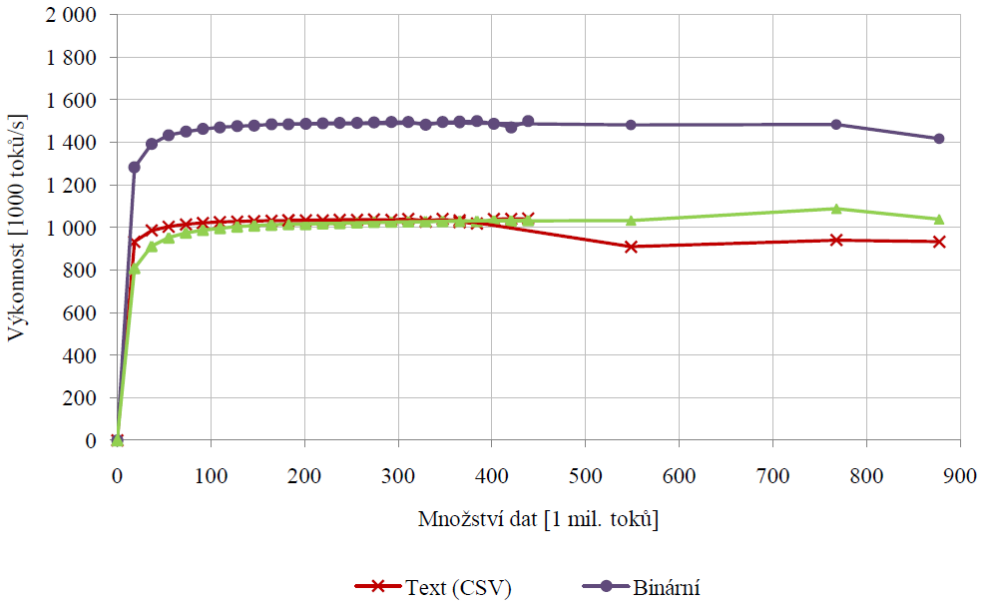
Obrázek 8. Dotaz č. 2 při paralelním spuštění úloh.



Obrázek 9. Dotaz č. 4 při paralelním spuštění úloh.

Z výsledků je patrné, že i při zvyšujícím se počtu paralelně běžících dotazů, se doba běhu jednoho dotazu sice zvýší, ale pouze o zlomek celkového trvání dotazu. U dotazů, které pracují paralelně s různými daty, se doba trvání prodloužila pouze nepatrně oproti dotazům, které pracují se shodnými daty. Tento výsledek ukazuje, že je možné zvýšit výkonnost/efektivitu Hadoop řešení zvýšením zátěže celého clusteru, tak aby režie představovala pouze nepatrnou část celého výpočtu. V takovém případě výkonnost jednoho uzlu v clusteru začíná dosahovat 2-3 mil. toků/s a začíná se blížit nfdump řešení, které dosahuje přibližně 3-4 mil. toků/s.

Poslední zkoumanou vlastností systému Hadoop je rychlost ukládání dat do distribuovaného souborového systému HDFS. Obrázek Obrázek 10 zobrazuje průběh doby trvání ukládání jednotlivých datových souborů, použitých pro výše uvedené experimenty. Z grafu lze pozorovat, že se tato doba relativně rychle ustálí na hodnotě přibližně 1,5 mil. záznamů za sekundu pro data uložená v binárním formátu a zhruba 1 mil. záznamů pro textová data či Hive formát.



Obrázek 10. Rychlost ukládání dat do HDFS.

Ukládání dat do HDFS probíhalo kopírováním dat z lokálního disku jednoho stroje. Z experimentů s ukládáním dat do HDFS je vidět, že na počtu ukládaných toků záleží jen nepatrně. Celková propustnost je dána objemem dat zapisovaných do HDFS. Z tohoto důvodu trvá déle uložit textovou reprezentaci než binarní. Pozn.: Vzhledem ke kopírování dat z disku bylo spíše dosaženo limitu propustnosti samotného lokálního disku spíše než limitu HDFS. Nicméně pro účely sběru dat o IP tocích je dosažená propusnost dostatečná.

3.2 NfDist

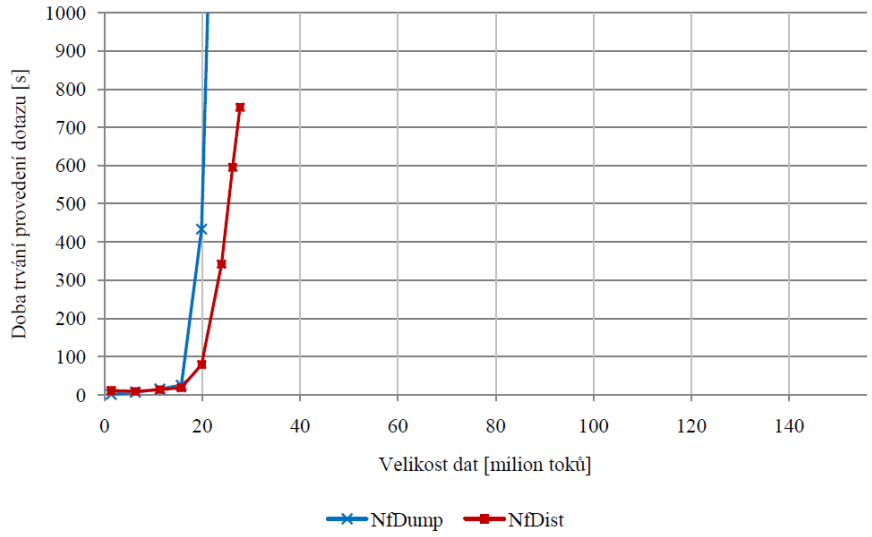
Další z testovaných platforem je NfDist [6]. NfDist umožňuje nástroji Nfdump, pracovat v distribuovaném prostředí. Souborové operace jsou zde realizovány prostřednictvím souborového systému Hadoopu HDFS, úlohy jsou pak na pracovních uzlech spouštěny pomocí Apache ZooKeeper. Experimenty byly prováděny na clusteru VO MetaCentra o 7 virtuálních strojích (1x frontend, 1+1(záloha) master uzel, 4x worker uzel). Tyto stroje byly vybaveny dvoujádrovými procesory (Intel E5-2620 @ 2000 MHz) a 2,95 GB paměti (4,0 GB swap).

Při těchto experimentech byla kvůli omezenému datovému úložišti použita menší datová sada a jiné dotazy, které jsou však typově podobně zaměřené. Datová sada sestávala z 5,5 GB (nfdump formát) dat o přibližně 155 milionech záznamů. V samotných experimentech byly použity následující dotazy:

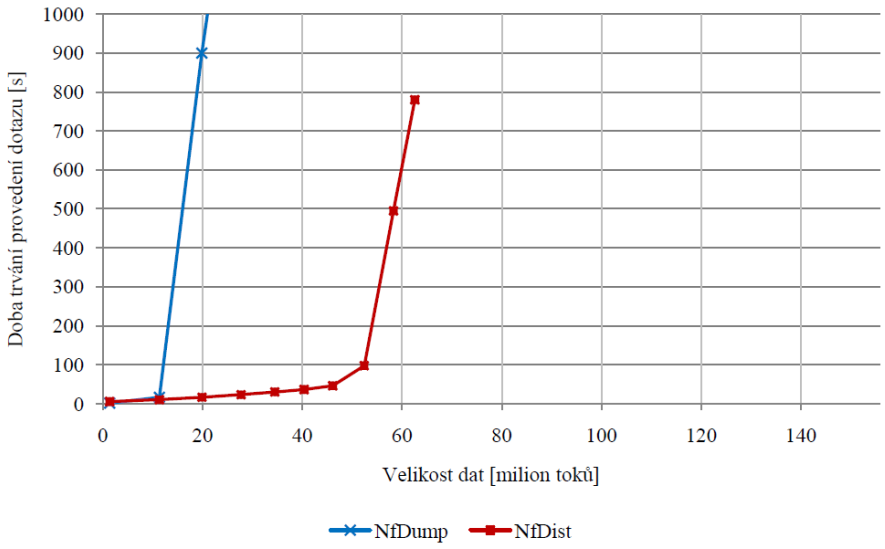
1. Agregace dle flow (protokol, zdrojová/cílová adresa, zdrojový/cílový port), výpis prvního řádku.
2. Stejný dotaz jako v předchozím případě s obousměrnými toky.
3. Agregace dle IP protokolu.
4. Statistiky top-10 dle IP adres.
5. Statistiky top-10 dle IP adres s filtrem na ICMP toky.
6. Statistiky top-10 toků s největším objemem bytů.
7. Výpis všech TCP toků
8. Výpis všech ICMP toků

Maximální délka provádění dotazů byla shora omezena na 1000 vteřin za účelem zkrácení doby celého experimentování, neboť zpracování dotazu trvající déle než 1000 vteřin je pro vyvíjený kolektor neakceptovatelné i pro dotazy nad rozsáhlými daty. Z uvedených dotazů jsou dále uvedeny pouze ty výsledky, které oproti ostatním ukazují významné informace.

Na prvních dvou grafech, zobrazujících výsledky provádění dotazů č.1 a 6 lze pozorovat, že i s pomocí distribuovaného zpracování je relativně brzy dosažen limit 1000 vteřin pro omezení doby provádění dotazů. To znamená, že už při práci s měnšími objemy dat (do 5GB) je u těchto druhů dotazů systém NfDist pomalý.

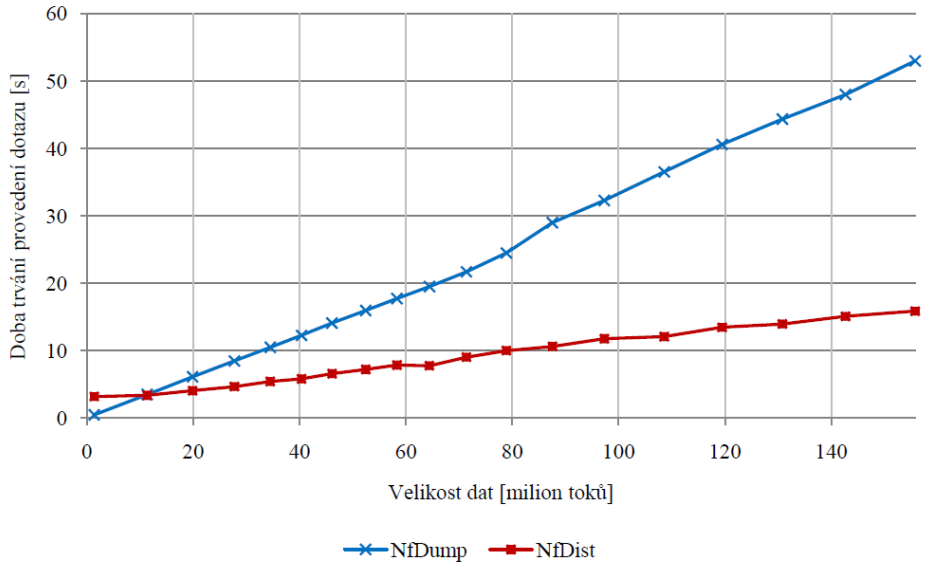


Obrázek 11. Dotaz č. 1: agregace dle pětice protokol, zdrojová/cílová adresa, zdrojový/cílový port.



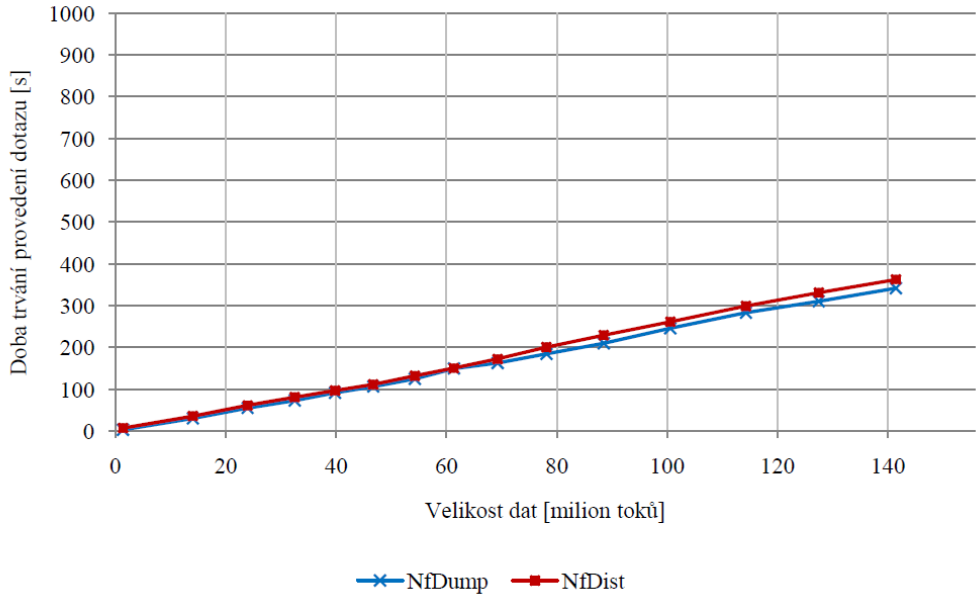
Obrázek 12. Dotaz č. 6: statistiky top-10 toků s největším objemem bytů.

Dotaz č. 3 provádí nad daty jednodušší agregaci než u předešlých dotazů. Na obrázku Obrázek 13 jsou zobrazeny výsledky doby provádění tohoto dotazu. Je na něm možné sledovat výrazné zkrácení odezvy při této jednodušší agregaci.



Obrázek 13. Dotaz č. 3: agregace dle IP protokolu.

Při přenosech a získávání větších množství dat, dosahuje NfDist výsledků srovnatelných s lokálně spuštěným nástrojem Nfdump. Výsledky takovéhoto dotazu jsou zobrazeny na obrázku Obrázek 14, jež zachycuje dobu provádění dotazu č. 7.



Obrázek 14. Dotaz č. 7: výpis všech TCP toků.

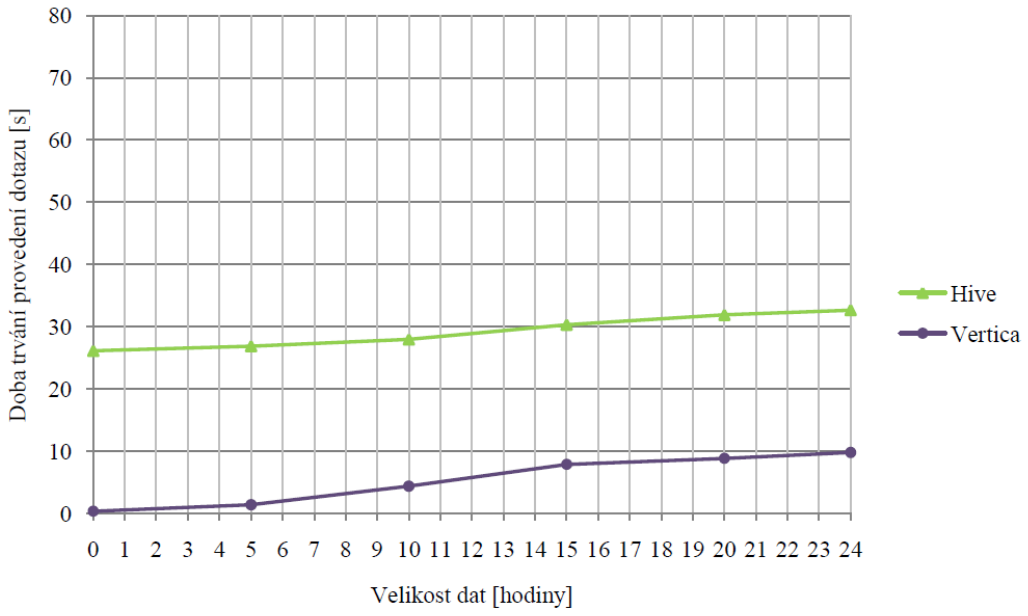
Celkově tedy platforma Nfdist nepřinesla žádné výrazné zlepšení z pohledu výkonnosti, výraznější zkrácení doby dotazování lze pozorovat pouze u některých typů dotazů. Přínosem je zejména spolehlivost ukládaných dat, která je však zajištěna distribuovaným úložištěm HDFS. Výkonnost jednoho uzlu pak odpovídá přibližně platformě Hadoop.

3.3 Vertica

Experimenty se systémem Vertica probíhaly na clusteru o 3 virtuálních strojích. Počet strojů byl omezen licencí platformy, kerá je při použití do 3 zařízení zdarma. Jednotlivé stroje v sestavě byly vybaveny dvoujádrovými procesory (Intel E5-2670 @ 2600 MHz) a 4 GB pamětmi (6 GB swap). Dotazy a datové soubory byly v případě těchto testů shodné s experimenty s platformou Hadoop. Jazykem pro tvorbu dotazů byl v případě Verticy jazyk SQL.

V následujících grafech je znázorněna doba odezvy při dotazování ve srovnání s nástavbou Hadoopu Hive, která v předchozích experimentech vykazovala nejlepší výkonnost v poměru k univerzálnosti a snadnosti použití dotazovacího subsystému. Srovnání jsou opět uvedena na reprezentativních dotazech č. 2 a 4. Na posledních dvou grafech je pak uvedena výkonnost systému Vertica opět v počtu zpracovaných záznamů za vteřinu jedním uzlem a rychlost ukládání dat. Je zde důležité zdůraznit, že výsledky platformy Hive byly získány na clusteru s 24 výpočetními uzly, kdežto databázový systém Vertica měl k dispozici pouze 3 stroje.

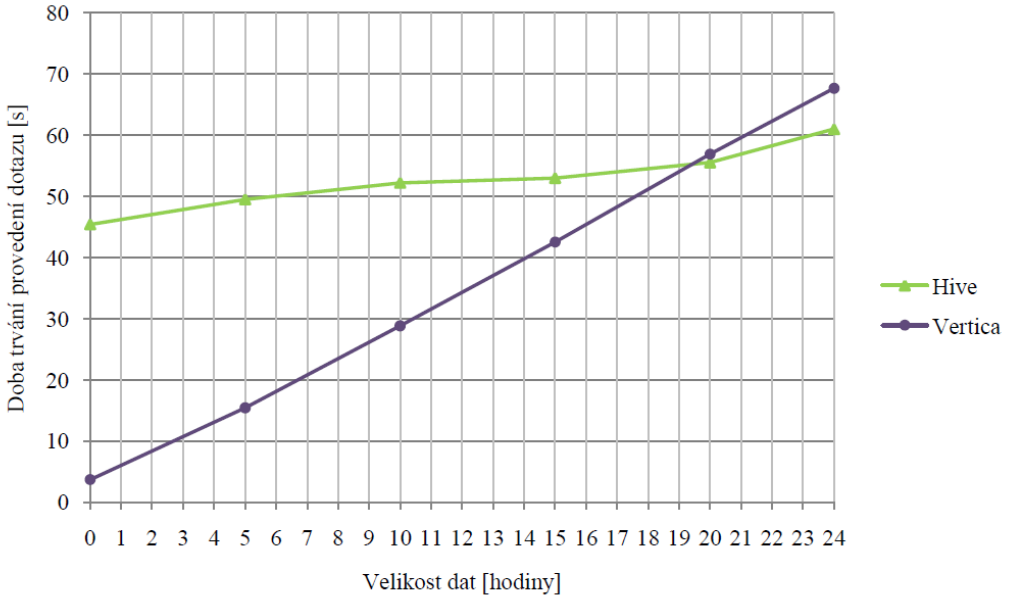
Na prvním grafu, srovnávajícím dobu provádění dotazu č. 2 platformami Hive a Vertica, lze sledovat výrazné zkrácení doby dotazování i při použití značně menšího clusteru systémem Vertica. Důvodem pro toto zrychlení je jednak lepší práce z daty, jak již bylo naznačeno v úvodu (sloupcový přístup, komprese, atd.), jednak lépe vyřešená komunikace a s ní spojená režie výpočtu.



Obrázek 15. Srovnání Apache Hive a systému Vertica - dotaz č.2.

Obrázek Obrázek 16 ukazuje stejné srovnání pro dotaz č.4. Při zpracování tohoto dotazu je přenášeno výrazně více dat, což se také promítlo do výsledné dobby provádění. Zde se časy provádění dotazů setkávají při práci s celým objemem testovacích dat. Lze však předpokládat,

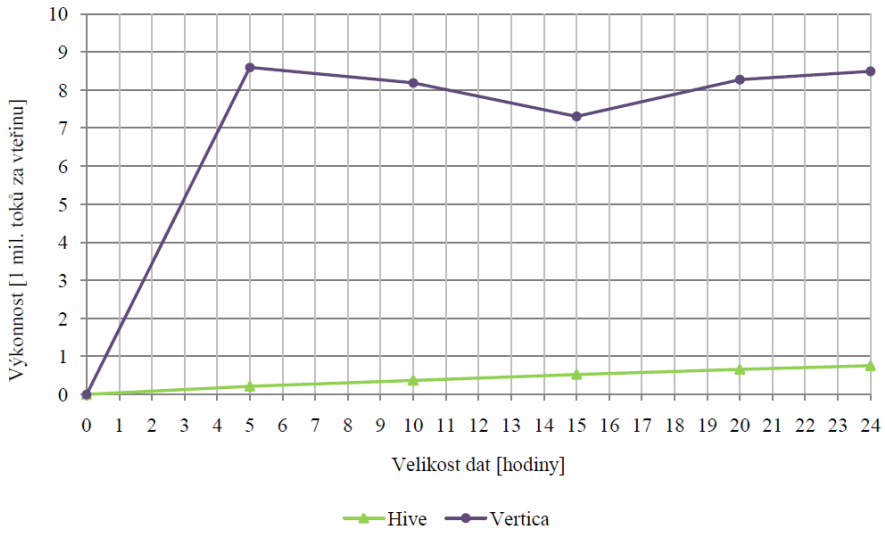
že prudší růst doby zpracování u systému Vertica je zde způsobem právě menším počtem výpočetních uzlů.



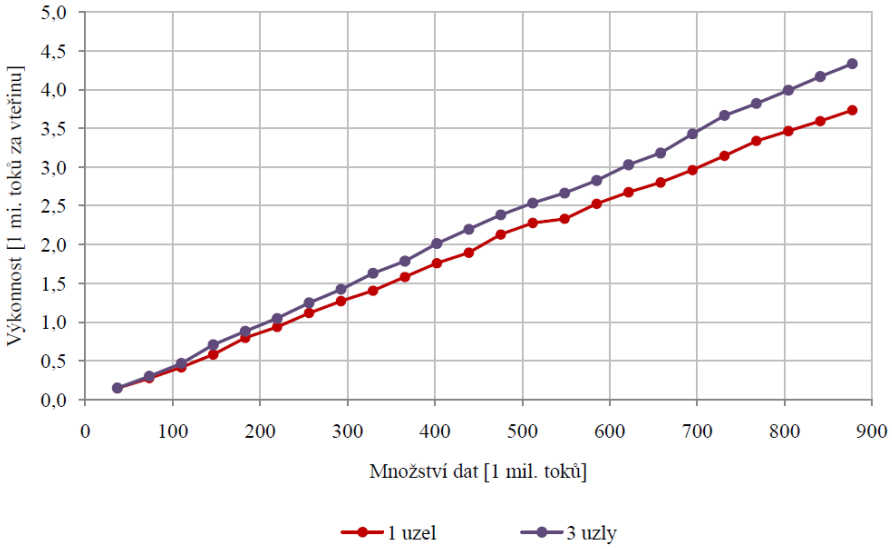
Obrázek 16. Srovnání Apache Hive a systému Vertica - dotaz č.4.

Výkonnost databázového systému je dobrá a pro námi zamýšlený kolektor dostačující. Ve srovnání s platformami založenými na Hadoopu je několikanásobně větší. Rychlost ukládání je naopak něco horší, nicméně pro ukládání dat v reálném čase stále vyhovující. Navíc se zdá, že přidávání uzlů nemá na dobu ukládání dat zásadní vliv.

Celkově se databázový systém Vertica společnosti HP jeví jako vhodné řešení pro námi vyvíjený kolektor. Jeho výkonnost při dotazování nad uloženými daty je velmi dobrá, jazyk SQL použitý pro dotazování poskytuje potřebné dotazovací rozhraní a doba ukládání dat postačuje pro ukládání v reálném čase. Bohužel je zde však zásadním problémem lincencování systému a příliš vysoká cena, která byla společností HP pro naše řešení nabídnuta.



Obrázek 17. Výkonnost dotazování systému Vertica.



Obrázek 18. Rychlost ukládání dat systému Vertica.

3.4 Elastic Search

Elastic Search je volně dostupný nástroj pro analýzu rozsáhlých dat. Pracuje distribuovaně a je zaměřen zejména na full-textové vyhledávání. Tato platforma se nejvíce zaměřuje na snížení doby dotazování nad uloženými daty. Za tímto účelem vytváří při ukládání rozsáhlý index, který v našem případě představoval až čtyřnásobek dat. Spolehlivost je zde zajištěna prostřednictvím replikace ukládaných dat, nicméně jejím použití v našem případě neúnosně stoupá doba ukládání dat, jak bude možné vidět v tabulce níže.

Experimenty se systémem Elastic Search byly provedeny na dvou clusterech označovanými čísly 1 a 2. Oba clustery pracovaly s 9 uzly. Stroje v clusteru 1 byly vybaveny čtyřjádrovými procesory (Intel(R) Xeon(R) CPU E3-1280 V2 @ 3.60GHz), 8GB pamětmmi a oproti

clusteru 2 pomalejšími disky. Cluster 2 obsahoval počítače se čtyřjádrovými procesory (Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz) a 32GB pamětmi. Právě množství paměti se při práci s daty jevilo jako podstatným aspektem, kdy kvůli nedostatku paměti rozsáhlejší dotazy trvaly výrazně delší dobu, nebo dokonce selhávaly.

Pro testovánní byla použita stejná datová sada jako při předchozích experimentech (tj. kromě Nfdist), nicméně dotazy již nebyly spouštěny nad jednotlivými datovými úseky s postupným hodinovým přírůstkem, ale přímo na celé datové sadě (odpovídá hodnotě "24" na ose x v předchozích grafech). Z tohoto důvodu jsou také výsledky zobrazeny ve formě tabulky. Uvedeny jsou hodnoty pro dotazy, odpovídající 4 původním dotazům, použitým při experimentech s Hadoopem.

Číslo dotazu	Číslo clusteru	Doba provedení dotazu
1	1	16,00
1	2	23,00
2	1	1,00
2	2	0,93
3	1	1,31
3	2	1,54
4	1	382,00
4	2	229,00

Tabulka 1. Rychlost dotazování

Počet replik	Číslo clusteru	Doba uložení
1	1	10h 34min
1	2	netestováno
2	1	9h 4min
2	2	7h 30min

Tabulka 2. Rychlost ukládání dat

Práce v systému Elastic Search umožňuje velmi rychlé provádění dotazů i nad rozsáhlými daty. To je umožněno díky velkému indexu, který však zabírá několikanásobek velikosti samotných dat (v našem případě až čtyřnásobek). Problémem je u této platformy také zajištění spolehlivosti, kdy při použití replikace neúnosně stoupá doba ukládání dat, kvůli níž by nebylo možné takto data ukládat do systému v reálném čase

4 Závěr

V rámci této práce byly provedeny rozsáhlé experimenty se stávajícími platformami pro distribuované zpracování dat. Tyto experimenty částečně ukázaly potenciál distribuovaného zpracování větších objemů dat. Současně také přinesly důležité poznatky o existujících platformách, určených ke zpracování masivních objemů dat s využitím počítačového clusteru. Na základě těchto poznatků se však vybraná řešení jeví spíše jako nevhodná. Důvodem je zejména režie výpočtu, která je v kontrastu dotazů malého až středního rozsahu pro použití

pro dotazování ve vyvíjeném kolektoru neúnosná. Další významnou nevýhodou je relativně malá výkonnost, která zřídka přesahuje hodnotu 500 000 toků za vteřinu na jeden uzel. Pozitivem těchto systémů je řešení zálohy dat a spolehlivosti výpočtů prostřednictvím replikace datových bloků, nicméně toto jediné významné pozitivum nepřesahuje svým přínosem nedostatečnou výkonnost testovaných řešení. Další nevýhodou těchto systémů je jazyk JAVA, ve kterém jsou tyto systémy napsány. Chybějící rozhraní pro jazyk C/C++, jenž je nativní pro zdrojové kódy rozšiřovaného kolektoru IPFIXcol představují významnou překážku při propojení těchto dvou platforem.

Poděkování

Tato práce byla podpořena Technologickou agenturou České republiky v rámci grantu č. TA04010062 – Technologie pro zpracování a analýzu síťových dat velkého rozsahu.

References

[1] NfDump. http://nfdump.sourceforge.net/

[2] http://www.metacentrum.cz/cs/hadoop/

[3] Yeonhee Lee, Youngseok Lee. Toward scalable internet traffic measurement and analysis with Hadoop, ACM SIGCOMM, 2013.

[4] Apache Hadoop. http://hadoop.apache.org/

[5] Martin Žádník, Pavel Krobot, Lukáš Kekely, Viktor Puš, Jan Kořenek: Distribuovaný kolektor záznamů o IP tocích: návrh a první experiment, technická zpráva, CESNET 2014.

[6] Vytautas Krakauskas, NfDist. https://github.com/vytautas/nfdist/

[7] ElasticSearch. http://www.elasticsearch.org/

[8] Apache Kafka. http://kafka.apache.org/

[9] Apache Storm. https://storm.apache.org/

[10] StreamMine3G. https://streammine3g.inf.tu-dresden.de/trac