Open Source projekty pro Big Data

Leo Galamboš

LG@HQ.EGOTHOR.ORG

Řešení pro velká data

Oblasti

1. ukládání dat
2. zpracování dat
3. analýza dat

(Dobrá zpráva)

OSS řešení nyní pokrývají všechny oblasti a jsou plně (pr)ověřeny v praxi.

(Špatné zprávy)

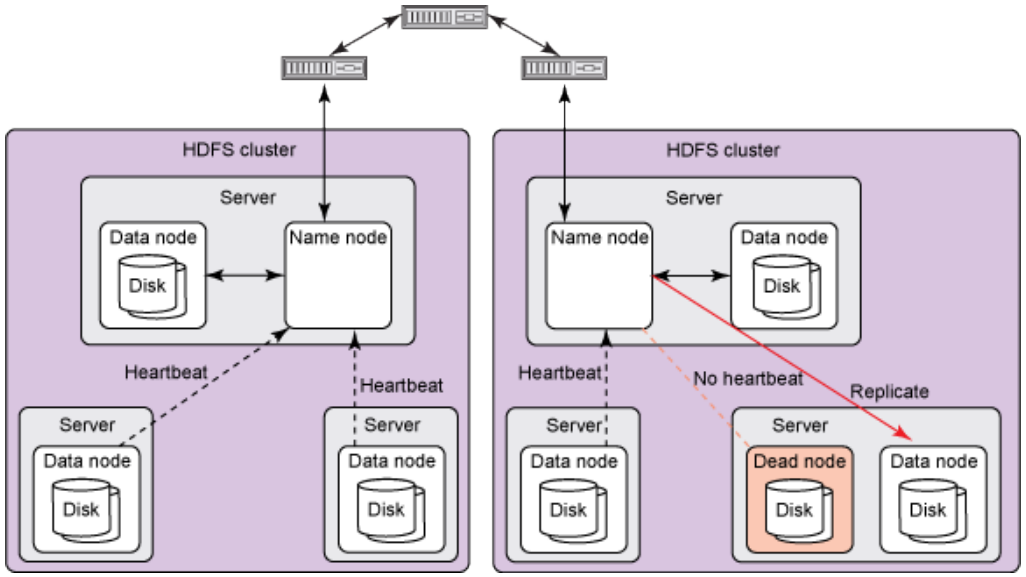
Vlastnosti implikují technická omezení a výkon. OSS má zpoždění oproti komerci.

Ukládání dat

- příklady využití
  - vyšetřování bezpečnostních incidentů
  - ukládání digitálních stop
  - úložiště pro data z otevřených zdrojů
  - dokumentové DB pro formuláře/systémy státní správy

- požadavek
  - ochrana proti výpadku
  - rychlost, bezpečnost
  - snadná údržba

OSS a velké souborové systémy



Distribuované FS pro řádově TB až PB

- využívá součet kapacit HDD dílčích uzlů
- automatické repliky, ochrana proti výpadku
- HDFS (Hadoop „GFS“)
- Lustre (toky řádově GB/s)

Ukládání (velkých) dat

- klasické RDBMS jsou universální
  - SQL, ACID a další komfort
  - nehodí se na velká data

současná „klasika“ Oracle, DB2, MSQL, apod.

- ukládání po sloupcích
  - C-store (Stonebraker)
  - výhodnější pro OLAP, text, vícerozměrná data

použitelné v bázích mýtného systému

- NoSQL („not only SQL“)
  - dokumentové, grafové, klíč-hodnota, …
  - obvyklá implementace stylem rozděl a panuj
    - drahá koordinace uzlů -> share-nothing architektura
    - omezená podpora ACID

použitelné na skutečné velká data a velké zátěže

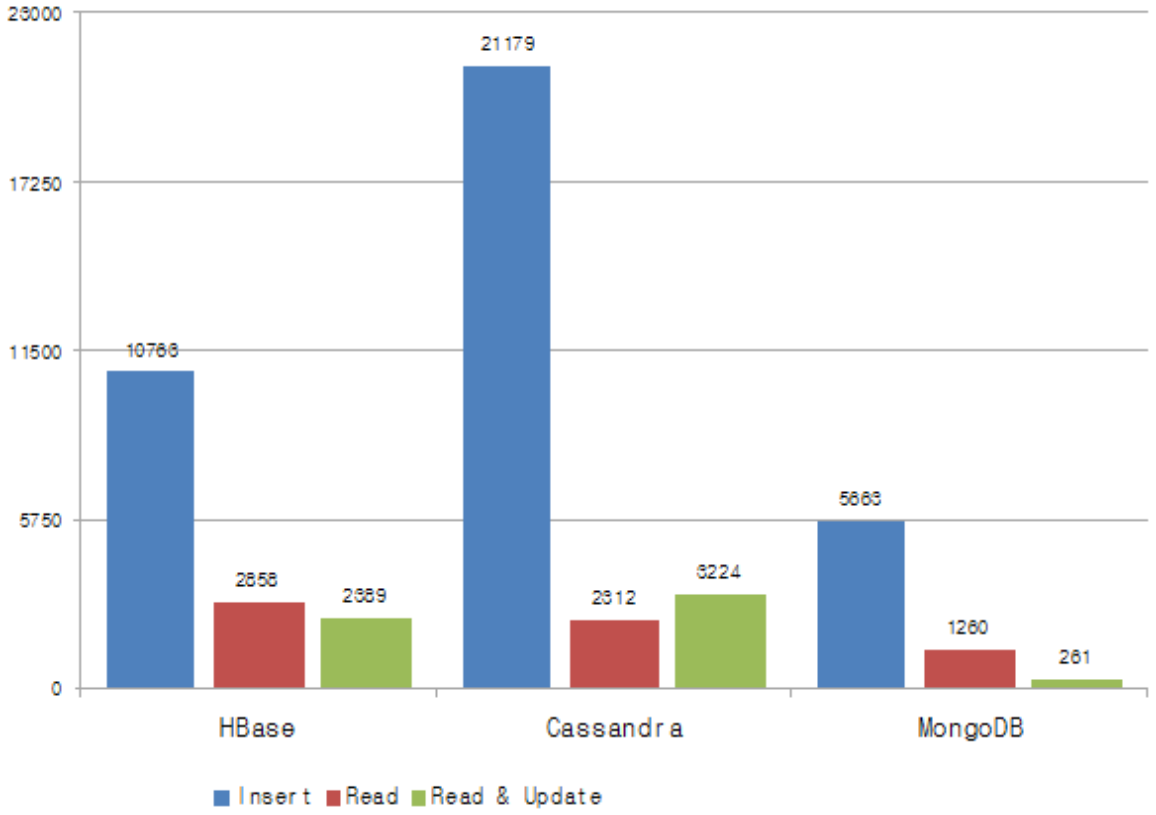
NoSQL databáze

- Cassandra (od Facebook-u)
  - sloupcová DB nad clusterem
- Dokumentové DB (typicky JSON styl)
  - MongoDB
  - CouchDB
    - Damien Katz (vývojář Lotus Notes)
- HBase
  - obdoba BigTable od Google

„volné“ formuláře

```
{
  „r.č.":8205172235,
  „foto“:„0x44ef5c1a…",
  „trestný_čin“:[
     "msg 1“,"msg 2“,"msg 3“
  ]
}
```

Yahoo Cloud Servicing Benchmark (YCSB)



Insert Only

- 50M x 1K

Read Only

- vyhledání klíče

Read & Update

- vyhledání a aktualizace klíče

Zpracování (proudů) dat

- příklady využití
  - okamžité zpracování událostí (zpráv)
    - mýtný systém
    - přenos událostí od informačního systému k DW
  - (weak) real-time decision support
    - detekce odcizených a přihlašovaných vozidel
  - integrace (vstupních) datových proudů
    - rychlé sloučení více databází veřejné správy

- požadavky
  - rychlost
  - spolehlivost

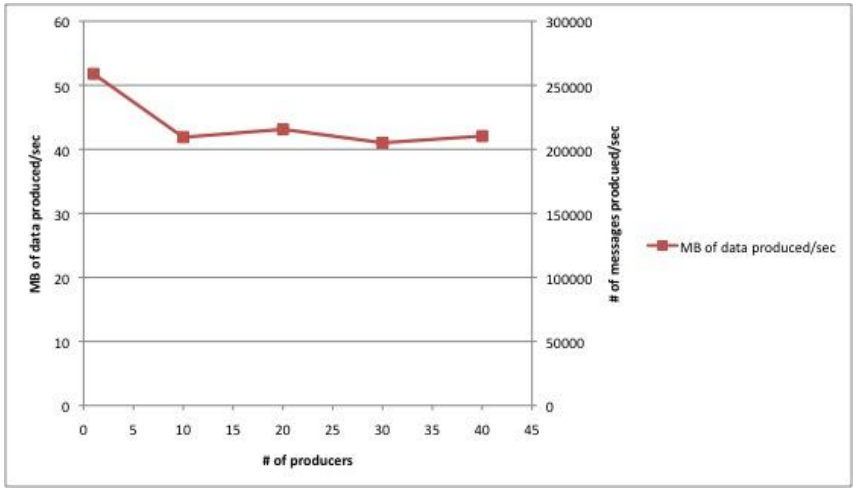
Kafka

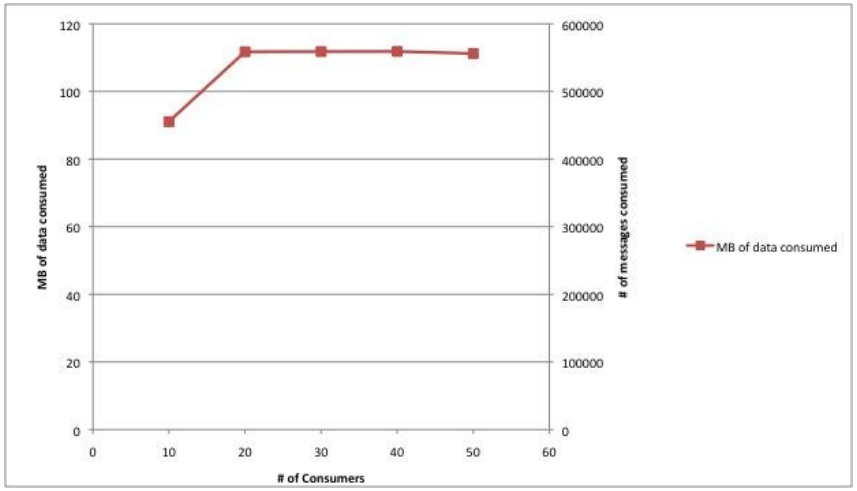
- Linkedin
- účel
  - sběr událostí od producentů
  - doručovaní zpráv odběratelům
  - zpráva má definované téma
- kde by našel využití
  - přenos události od mýtných bran
  - zpracování události od živého systému
  - jako doručovatel dat do DW nebo k dávkovému zpracování

Kafka - výkon

producent cca 40-50MB/s
výhoda: méně producentů

konzument cca 100MB/s
malý vliv #konzumentů

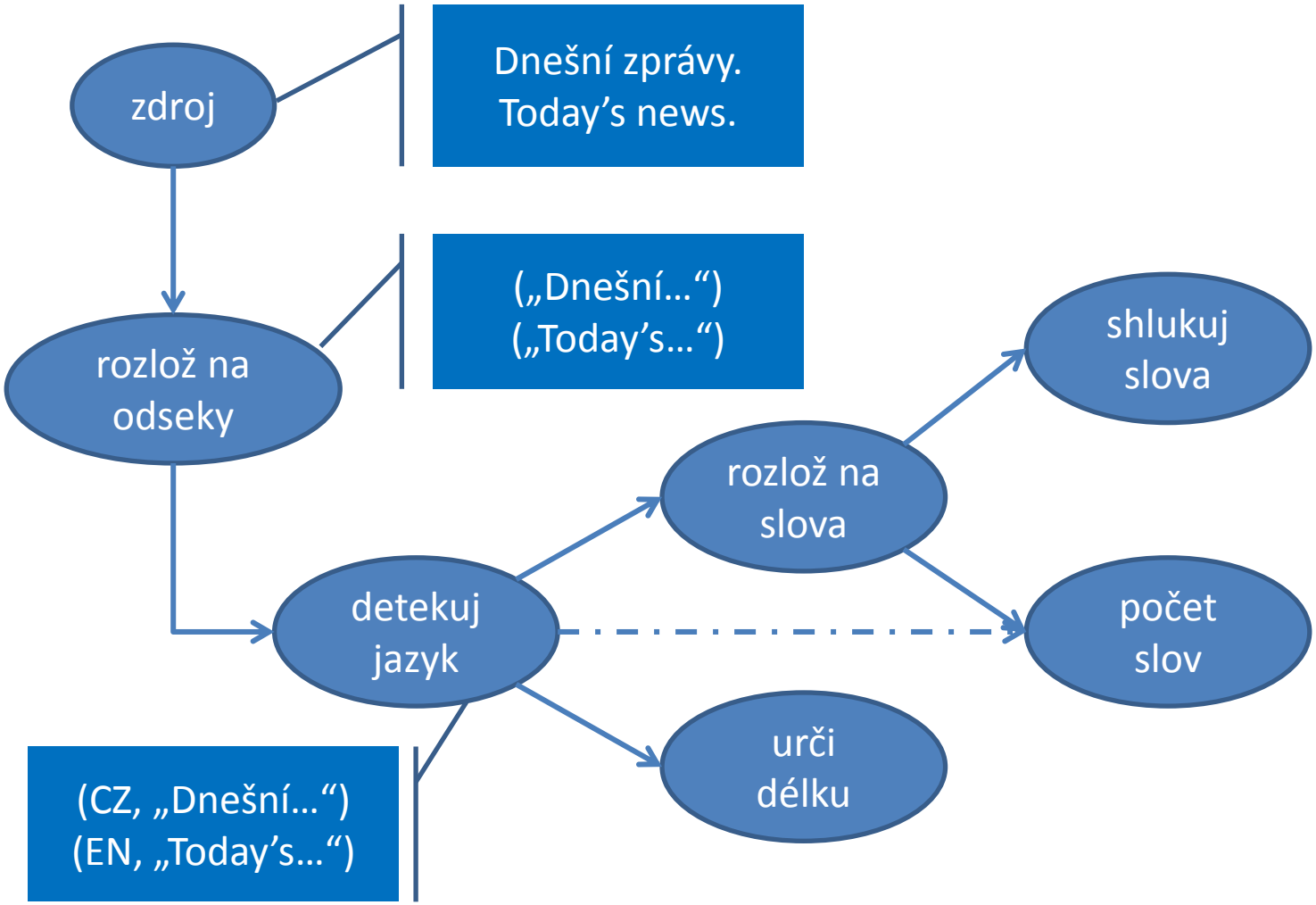




Storm

- Twitter
- výpočty v „reálném čase“, cca 100MB/s na jednom uzlu
- účel
  - výpočty nad proudem dat stylem rour v Unix-u
  - vstupem Kafka, RDBMS, JMS, …
- kde by našel využití
  - sumarizace událostí (6xklik = 1x6klik)
  - předzpracování toku (dokument -> slova -> <d,w,p>)
  - detekce ukradených a přihlašovaných vozidel
  - importy databází v jiných formátech (mezi úřady státní správy?)

Storm – ilustrační příklad



(Dávkové) zpracování dat

- příklady využití
  - klasifikace textů
    - „Washington“ versus „Washington“
  - indexování dat („malý“ Google)
  - analýza sociálních vztahů
    - kdo koho zná
  - mapování obchodních vztahů
    - detekce bílých koní, podezřelých organizací, apod.
- problematika
  - jsou OSS řešení dostatečné výkonná?

Dávkové zpracování dat

Sort Benchmark „Gray“ (100TB)

Hadoop	TritonSort
100TB/173min (0,578 TB/min)	100TB/138min (0,725 TB/min)
3452 uzlů (cca 3800 celkem) 2 QuadCore Xeon, 8 GB, 4 SATA	52 uzlů 2 QuadCore, 24 GB, 16 SATA
1 Gbps/uzel 40 uzlů/rack 8 Gbps uplink/rack	10 Gbps/uzel Cisco Nexus 5096 switch

- rozděl (map) a zpracuj (reduce)
- Apache Hadoop

Nedostatky Hadoop-u

- Share-nothing architektura, ale i nadále se single-master uzlem (výkon, stabilita)
- Spojení vícero datových množin je pomalé, neexistují indexy, data se často kopírují během zpracování (propustnost)
- Problém s řízením toku dat: optimalizace při využívání mezivýsledků (efektivita)
- Neexistuje centrální datová oblast, restriktivní programovací model (vývoj sw)
- Obtížná správa clusteru: vhodné nastavení počtu procesů typu mappers/reducers, paměťových limitů, … (management systému)

Plánovaní v Hadoop-u

- plánování úkolů trvá delší dobu
  - problematické pro malé úkoly a rychlé výsledky
- pevná kapacita „map“ a „reduce“ slotů
  - cluster pracuje neefektivně, když se úkoly nevejdou do volných slotů

- vrstva pro sdílení zdrojů
  - YARN – součást nového Hadoop
  - Corona – řešení pro FB adaptaci (proprietární)
  - Mesos – řešení od UC Berkeley

Corona (provoz od 3Q2012)

Parametr	Vylepšení
Zkrácení doby na znovupoužití volných zdrojů	-17%
Využití zdrojů (simulovaný cluster)	ze 70% na 95%
Přetěžování (unfairness) zdrojů	ze 14,3% na 3,6%

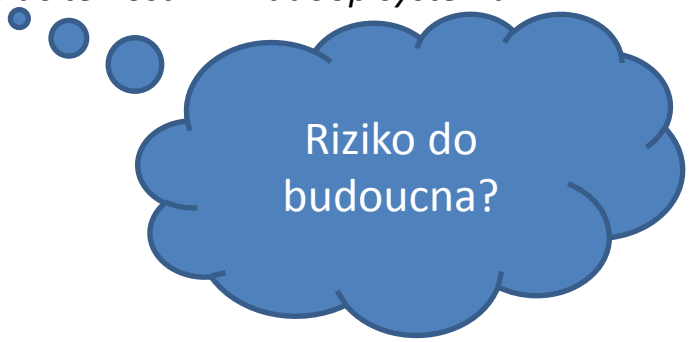
Facebook data warehouse

- 100PB (petabyte)
- 60.000 dotazů skrze Hive

Řešení efektivity clusteru?

Vylepšování současného stavu…

- Analýza dat Hadoopem
  - Pig – překlad jazyka Pig Latin na Map&Reduce programy
  - Hive – data warehouse
  - velká časová prodleva při zpracování (minuty až hodiny)
- Apache YARN – arbitr pro zdroje clusteru
  - Apache Mesos (UC Berkeley)
  - Corona (Facebook) – přímý ekvivalent, YARN nelze kvůli *neslučitelnosti FB Hadoop systému* použít



„…and now for something completely different…“

- Apache Drill = Google Dremel
  - pokládání ad-hoc dotazů nad velkými daty
  - latence v řádu jednotek sekund
  - výhodné pro BI, analýzy velkých dat apod.
  - OSS verze nebude ještě dlouho k dispozici, komerčně ji má jen Google

Čím analyzovat

- Hadoop Map&Reduce infrastruktura
  - nízko-úrovňové
- Apache Pig překládá z jazyka Pig Latin na MR programy
- Apache Hive = DW pro Hadoop
  - Shark (výpočty v RAM via Spark) – 100x rychlejší než Hive
- Statistická analýza via GNU R
  - mnoho balíčků (Comprehensive R Archive Network)
  - RHadoop, RHive, RHIPE (R & Hadoop Integrated Programming Environment)
- Apache Mahout
  - clustering, K-means, …
  - pattern mining, Bayesovské klasifikátory, apod.
- Apache Giraph (inkubátor), GoldenOrb, Gremlin
  - analýza grafové struktury

data warehouse

klasifikace textů

analýza sociálních vztahů

OSS pro velká data

Klady

- široké portfolio
  - zaštítěné velkými hráči
  - reimplementace komerčních řešení
- levný vývoj a údržba
  - nezačíná se na zelené louce
  - údržbu „hradí“ celá komunita
- přiměřený výkon

Zápory

- široké portfolio
  - dobře zvážit co chci teď
  - ještě lépe zvážit, zda zvolené půjde mým směrem – nutný silně kritický náhled
- malá efektivita oproti ad-hoc řešením
- hrozí rozštěpení API
  - Příklad: Facebook & Corona

Shrnutí

- OSS nabízí velkou škálu produktů
  - nasazení: Facebook, Twitter, Linkedin, …
- reimplementuje komerční řešení
  - GFS, Map&Reduce, BigQuery, …
- „OSS standard“ pro výpočty: Hadoop
  - není dokonalý
  - mnoho návazných platforem a produktů
  - nebezpečí: kompatibilita verzí
- OSS dává kvalitní prostředky, ne vždy ale rovnou finální a komplexní řešení