DĚLAT
DOBRÝ SOFTWARE
NÁS BAVÍ

PROFINIT

B0M33BDT – 2. přednáška

Marek Sušický

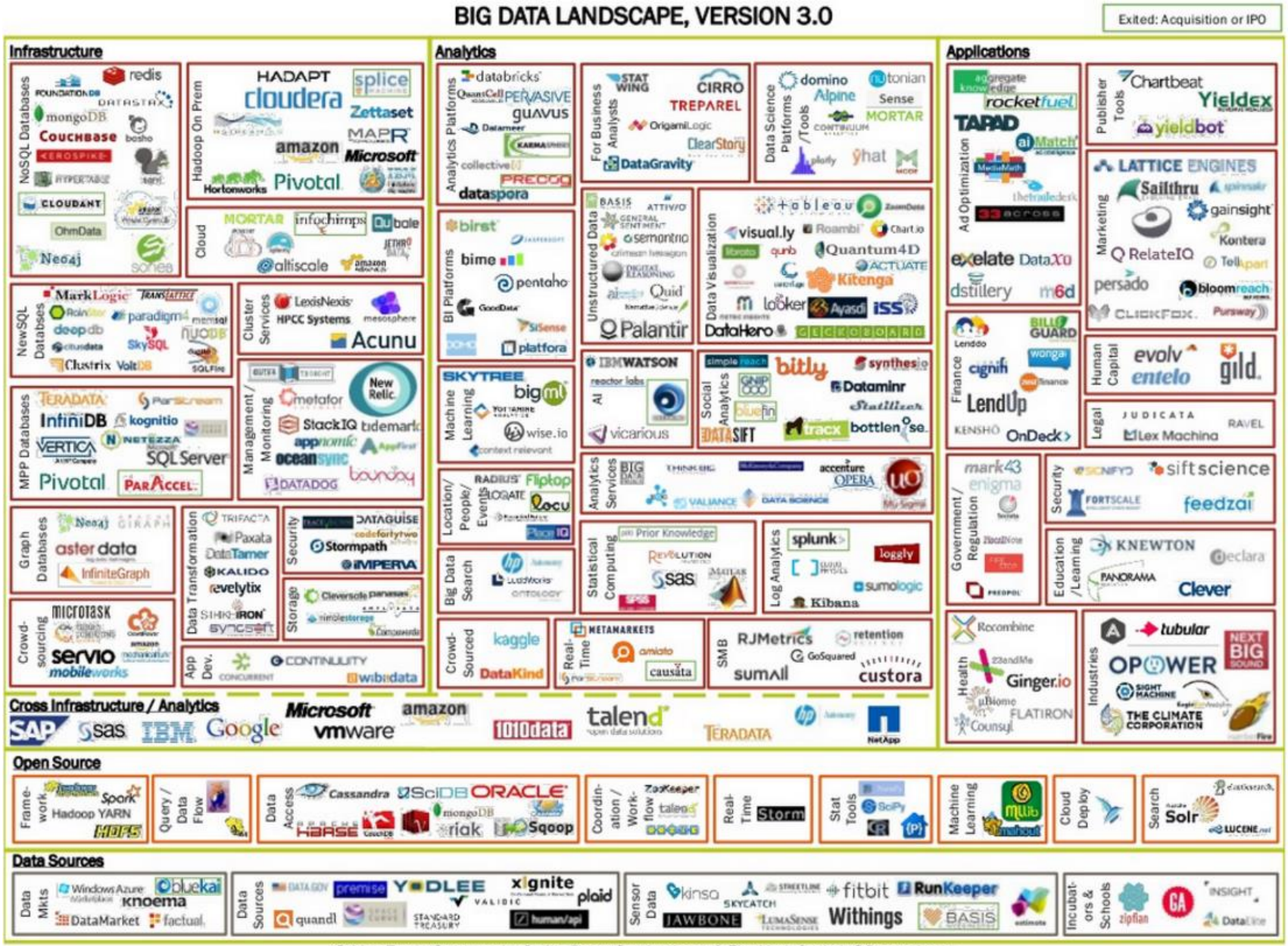
3. 10. 2018

Osnova

- Big data a Hadoop
- Na jakém hardware + sizing
- Jak vypadá cluster - architektura
- HDFS
- Distribuce
- Komponenty
- YARN, správa zdrojů

PROFINIT

Big data neznamená Hadoop



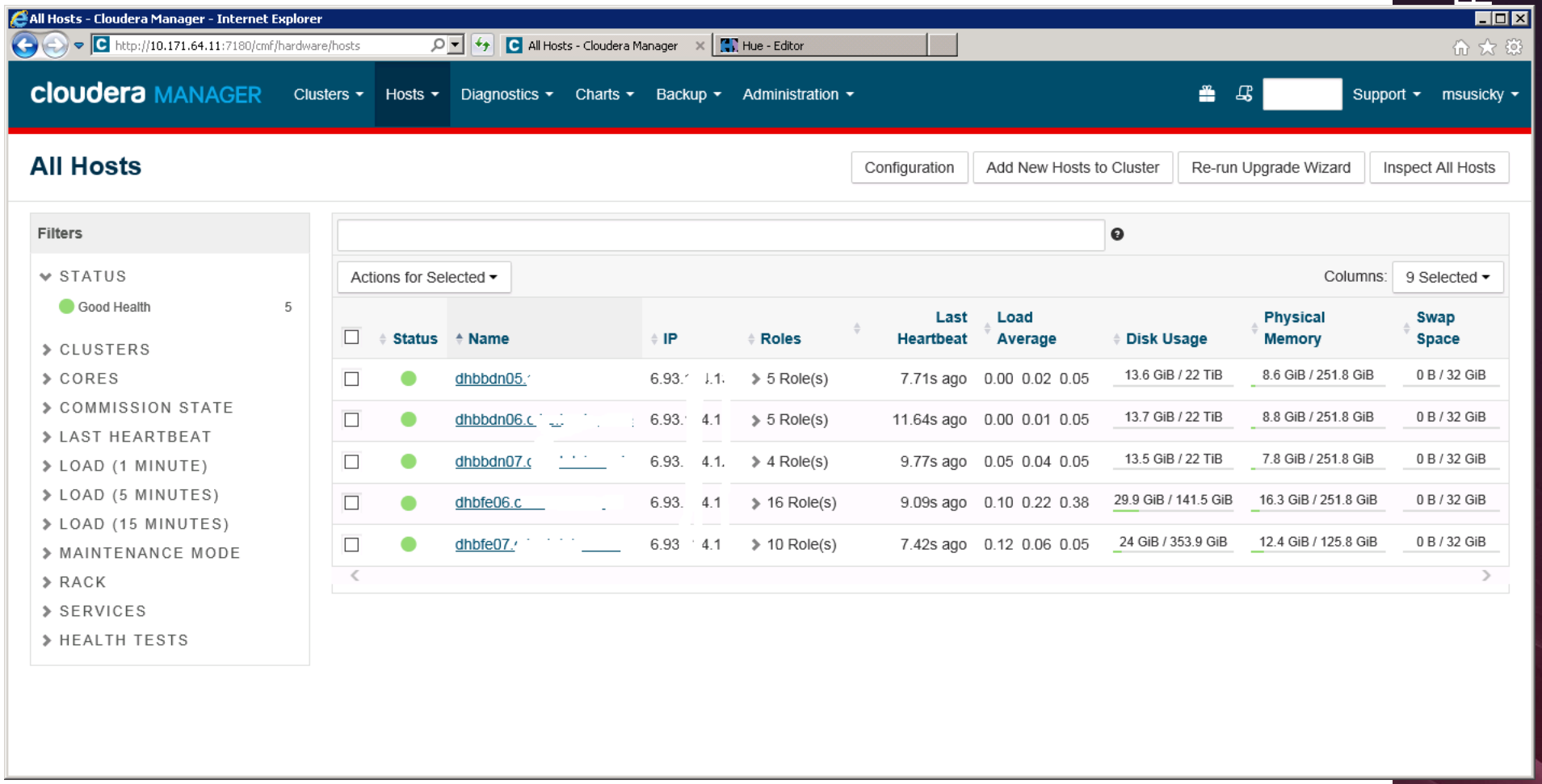
PROFINIT

Apache Hadoop

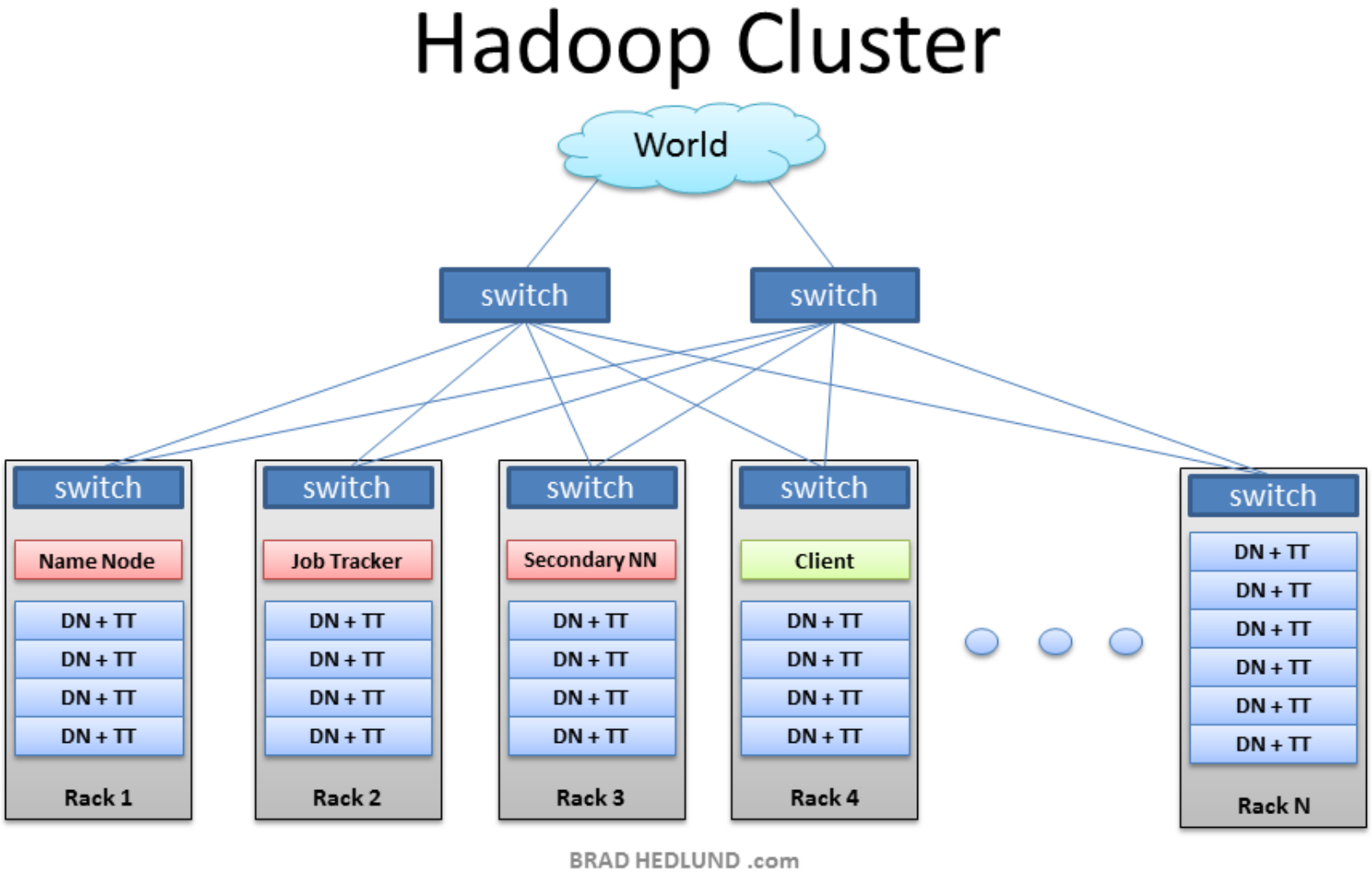
- Wikipedia:
  - Apache Hadoop (pronunciation: /həˈduːp/) is an open-source software framework for **distributed storage** and **distributed processing** of **very large data sets** on computer clusters ***built from commodity hardware***. All the modules in Hadoop are designed with a **fundamental assumption that hardware failures are common** and should be automatically handled by the framework.
- Commodity hardware
  - stroje za statisíce CZK (ale ne desítky mil.)
    - 2-4 CPU, každé CPU 10-16 jader
    - 256-512 GB RAM, min. 128GB
    - 10-20 2-4TB HDD
  - rozhodně ne to, co jako server prodává Alza ☺

PROFINIT

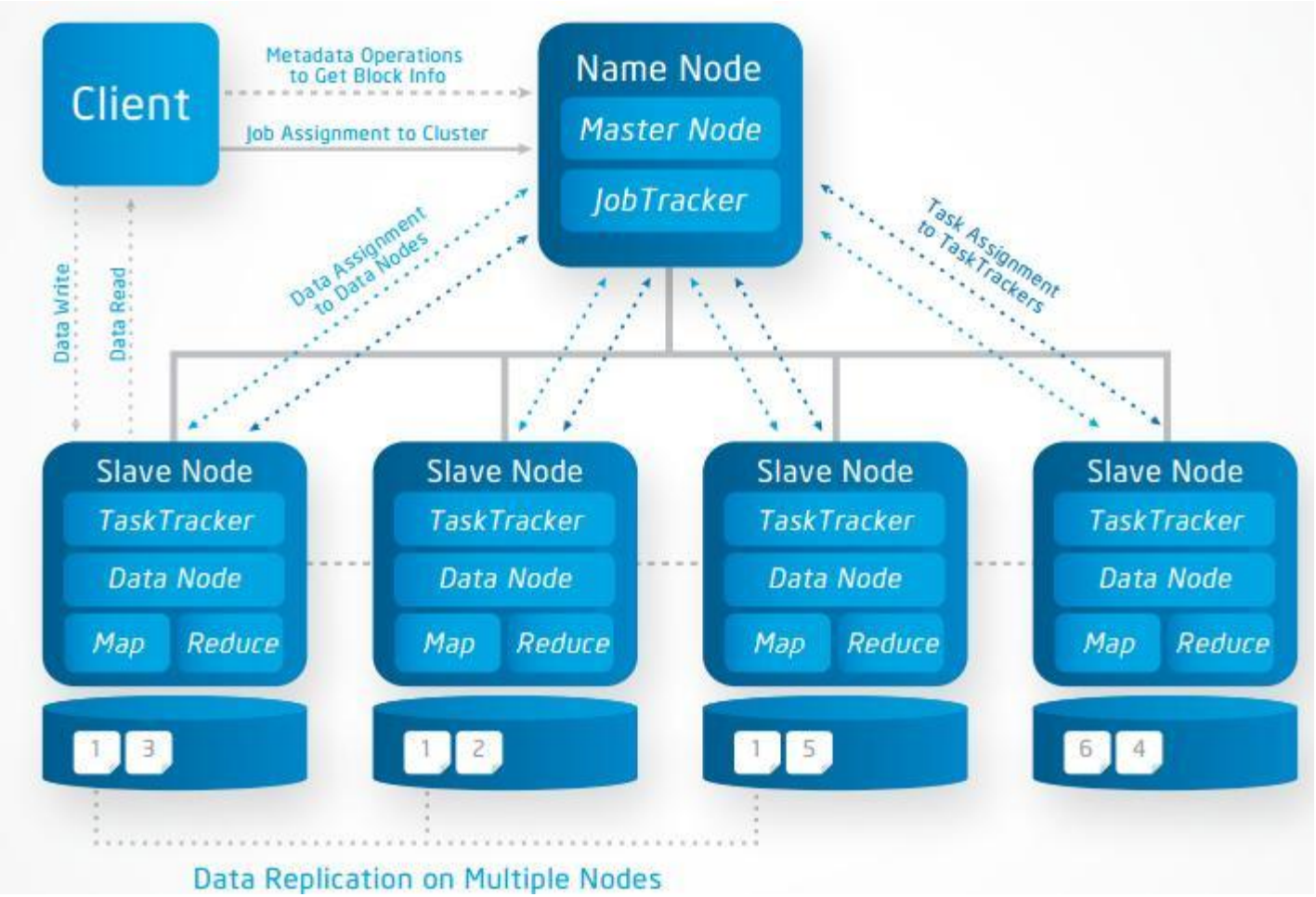
Jak vypadá levný HW



Hadoop – architektura I



Hadoop – architektura II



PROFINIT

Sizing

- Jak postavit Hadoop
  - Jak si ho objednat
  - HDD parametry
    - Přenosová rychlost
  - RAID
    - 0, 1, 1+0, 5, 6, (2,3,4,7)
  - Síťová rychlost
  - SAN/NAS
  - Paměť
  - CPU jádra
  - Obecná doporučení

PROFINIT

Sizing

- Kalkulace HW požadavků
  - Počet nodů
  - Počet disků
  - HW parametry (CPU, RAM, RAID…) typů nodů

PROFINIT

Rychlosti čtení dat

- RAM
  - DDR4 cca 15 GB/s
- Síť 10 Gbit
  - 1.25 GB/s
- SSD disk
  - 200-700 MB/s
  - existují „Enterprise level“, které vydrží (garance 5 let)
  - malé kapacity (max 1TB) a hodně drahé
- HDD 7.2k
  - latence cca 4ms
  - sekvenční čtení 50-100 MB/s
  - velké kapacity (běžně 4TB-8TB) a relativně levné
  - ➔ Hadoop typicky pracuje s úložištěm

PROFINIT

Rychlosti čtení dat – HDD

- **Sekvenční čtení** – cca 100 MB/s za jednom disku
- Random access
  - velikost bloku ext4 bývá 4kB
  - latence, než disk najde blok cca 4ms
  - max. rychlost čistě náhodného čtení 1/0.004*4096 = 1 MB/s

PROFINIT

Omezující faktory

- Příklad: 10 nodů, každý node 12 * 2 TB HDD
  - Rychlost čtení v rámci nodu: 12*100 MB/s = 1.2 GB/s
  - Rychlost čtení v rámci clusteru: 12 GB/s

- Omezení:
  - rychlost RAM – 10x větší
  - CPU – čtení nezatěžuje
  - síť – na hraně pro jeden node!
  - sběrnice – pozor na počet disků, musí zvládnout

PROFINIT

Sizing

- Ukázkový příklad k zamyšlení
  - 15GB/5 min
  - Historie 30 dní
  - SLA – 5 sekund pro 85% dotazů
    - 10s pro 100% dotazů
    - Při nesplnění vysoké pokuty

PROFINIT

Principy

- Ukládání velkého množství dat
  - mnoho serverů = nodů [desítky až tisíce]
  - každý node mnoho disků [10-20]
- Zamezení ztráty dat – výpadek nodu
  - replikace (typicky tři kopie každého souboru)
  - 2 repliky ve stejném racku, třetí replika mimo
- Rychlost čtení
  - data jsou rozložena v celém clusteru – 1 soubor nemusí být celý v jednom nodu!
  - data jsou replikována – lze paralelně číst na několika nodech bez nutnosti přenosu dat přes síť
  - velké soubory – **výhody sekvenčního čtení**
- Distribuce výpočtů
  - mnoho nodů, přiřazování výpočetního výkonu

PROFINIT

Principy Hadoop – syntéza

- Maximálně využívat sekvenční čtení
- Pracovat s velkými soubory, které se čtou sekvenčně
  - ušetří se čas na synchronizaci/orchestraci v rámci distribuovaného systému
- Čím méně dat se načte, tím rychleji se načtou – využití **komprese**
- CPU se při čtení fláká, paměť je násobně rychlejší, tj. typicky dekomprese bude rychlejší než IO operace
- Maximum výpočtů provádět na nodu, kde probíhá čtení dat, přes síť posílat jen co nejvíce agregovaná data
  - síťové rozhraní nebude mít dostatečnou kapacitu

- **Zásadní omezení – I/O operace**
  - **Tip:** při odhadu, jak dlouho co bude trvat stačí prakticky vždy počítat jen s IO a počtem paralelních čtení/uživatelů (ale neplatí samozřejmě pro ML aplikace)

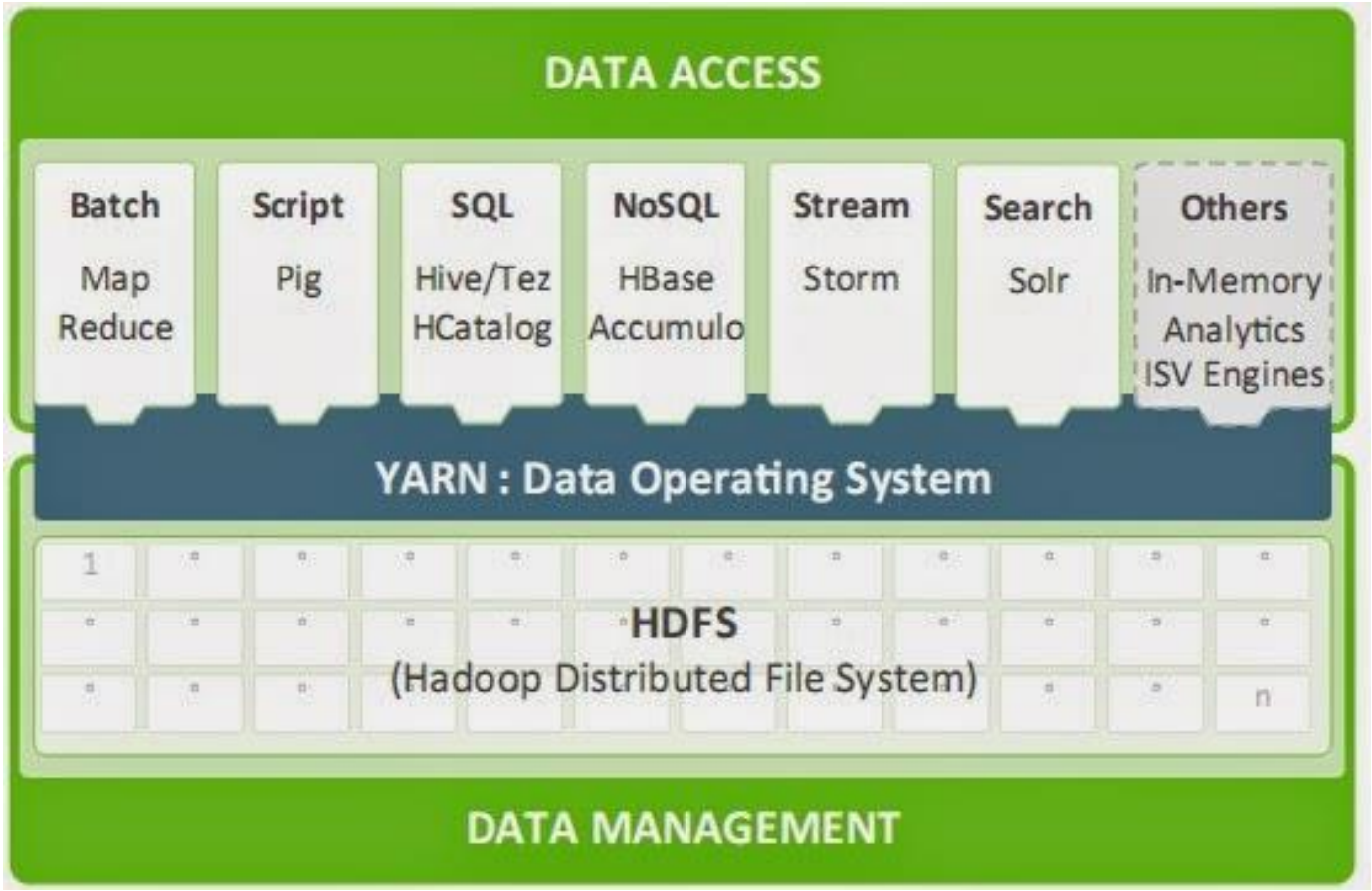
PROFINIT

HDFS

HDFS

- HDFS
  - NameNode, DataNode
  - Replikace
  - Operace na souborovém systému
  - Bloky, velikost bloku

HDFS- architektura

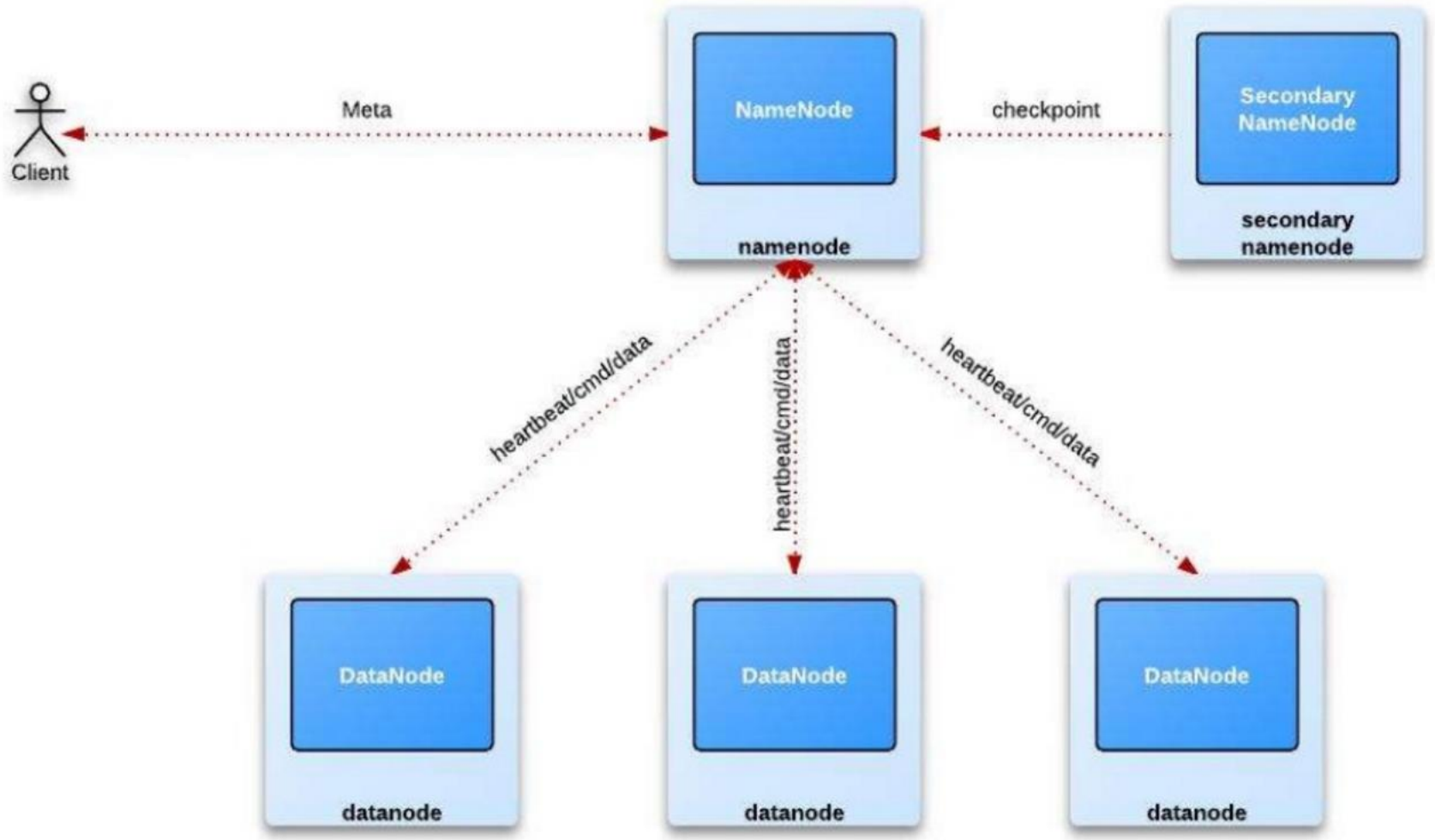


PROFINIT

HDFS

- Hadoop Distributed Filesystem
- Dobrý pro
  - Velké soubory
  - Streamovaný přístup
- Špatný pro
  - Spoustu malých souborů
  - Náhodný přístup
  - Nízkolatenční přístup
- Master-slave design
  - Master – NameNode
  - Slave – DataNode
  - SecondaryNameNode

HDFS



PROFINIT

HDFS

- HDFS soubory jsou rozděleny do bloků
  - Default 64MB/128MB, ale lze změnit
  - Dobré pro velké soubory
  - Děsné pro malé…
- Replikace
  - Jeden blok může být v nejméně xxx DataNodes
  - Fault tolerant
  - Default hodnota 3

PROFINIT

NameNode

- Metadata filesystému
  - Kde jsou data
  - V paměti
  - 1GB pro každý milion bloků
- Ve spojení s
  - Klienty
  - DataNody
  - SecondaryNamenody
    - Checkpointing
      - Editlogs a fsimage

PROFINIT

DataNode

- Ukládá Databloky
- Získává bloky od klientů
- Získává bloky od ostatních DataNodů
  - Replikace
- Dostává příkaz delete od NameNode

HDFS filesystem

- put
- get
- copyFromLocal
- ls
- Rights
  - Chmod
  - Chown
  - Chgrp

- Další zde
  - https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html

- Vyzkoušíme na cvičení

PROFINIT

Distribuce

Distribuce

PROFINIT

Hotové distribuce

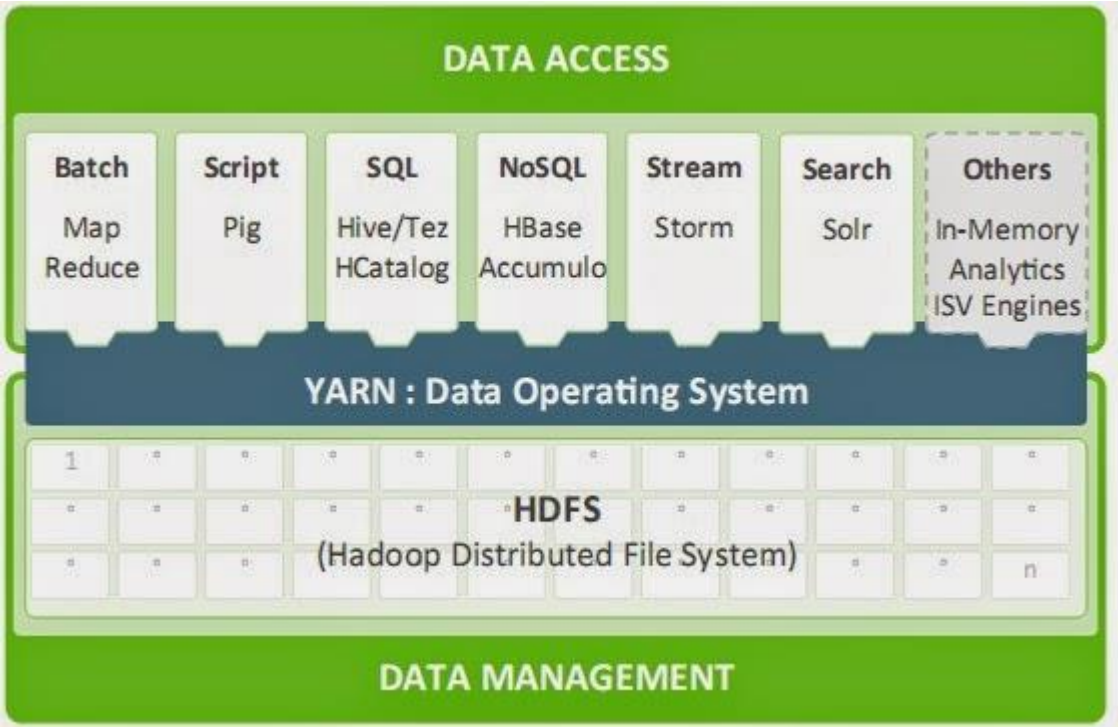
- Řeší peklo závislostí – jeden update může vyvolat řetězovou vlnu
- Nabízejí komerční podporu
- Rychlejší reakce ?
- Co je zadarmo, je špatné ?!
- Proč znovu vymýšlet kolo

- „Musí se k tomu dospět“

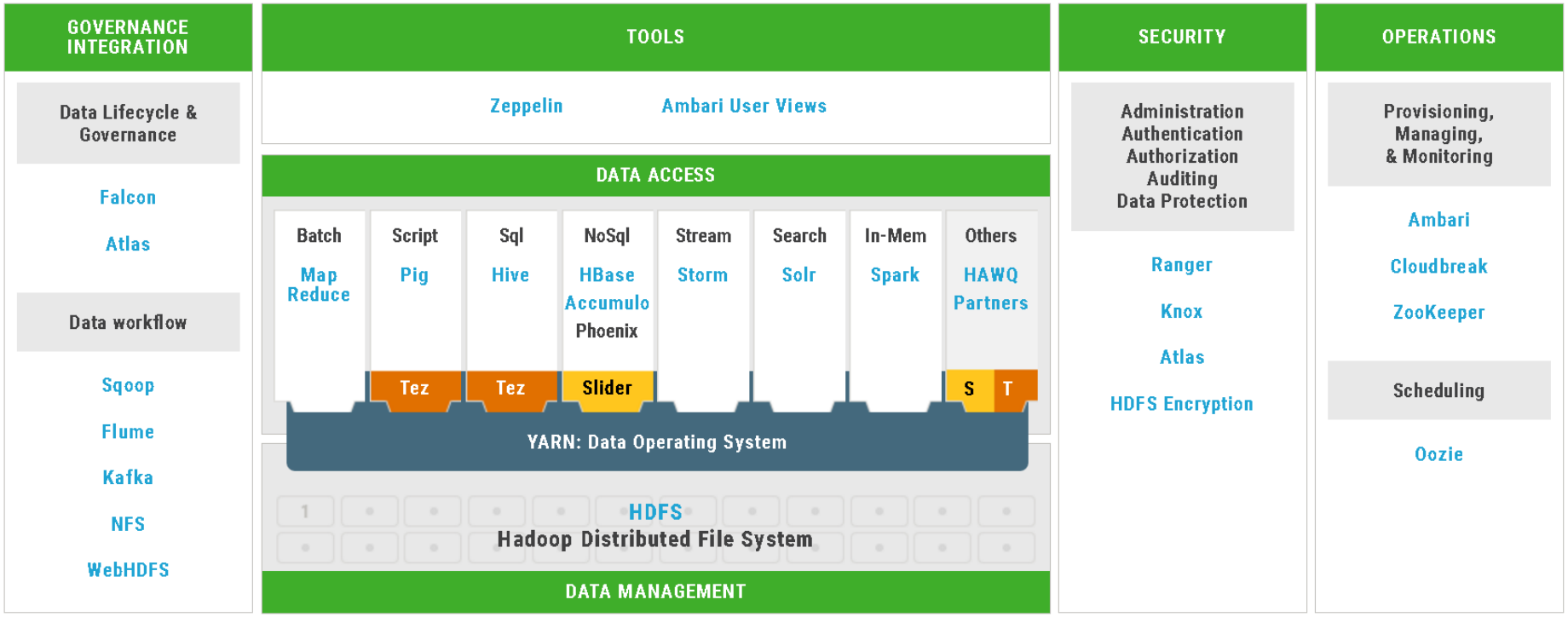
PROFINIT

Komponenty

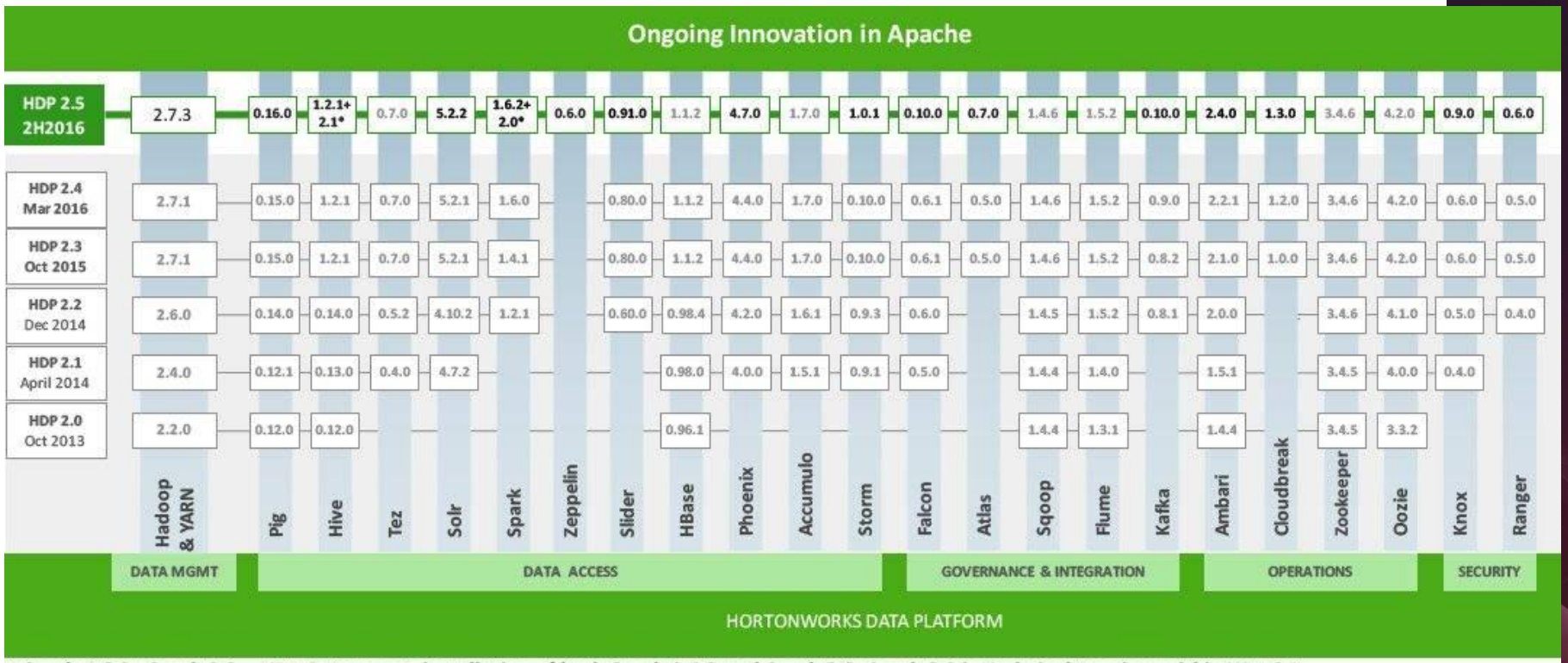
Zvěřinec - zjednodušeně



PROFINIT


GOVERNANCE INTEGRATION
Data Lifecycle & Governance
Falcon
Atlas
Data workflow
Sqoop
Flume
Kafka
NFS
WebHDFS
TOOLS
Zeppelin
Ambari User Views
DATA ACCESS
Batch
Map Reduce
Script
Pig
Sql
Hive
NoSql
HBase
Accumulo
Phoenix
Stream
Storm
Search
Solr
In-Mem
Spark
Others
HAWQ
Partners
Tez
Tez
Slider
S
T
YARN: Data Operating System
HDFS
Hadoop Distributed File System
DATA MANAGEMENT
SECURITY
Administration
Authentication
Authorization
Auditing
Data Protection
Ranger
Knox
Atlas
HDFS Encryption
OPERATIONS
Provisioning, Managing, & Monitoring
Ambari
Cloudbreak
ZooKeeper
Scheduling
Oozie

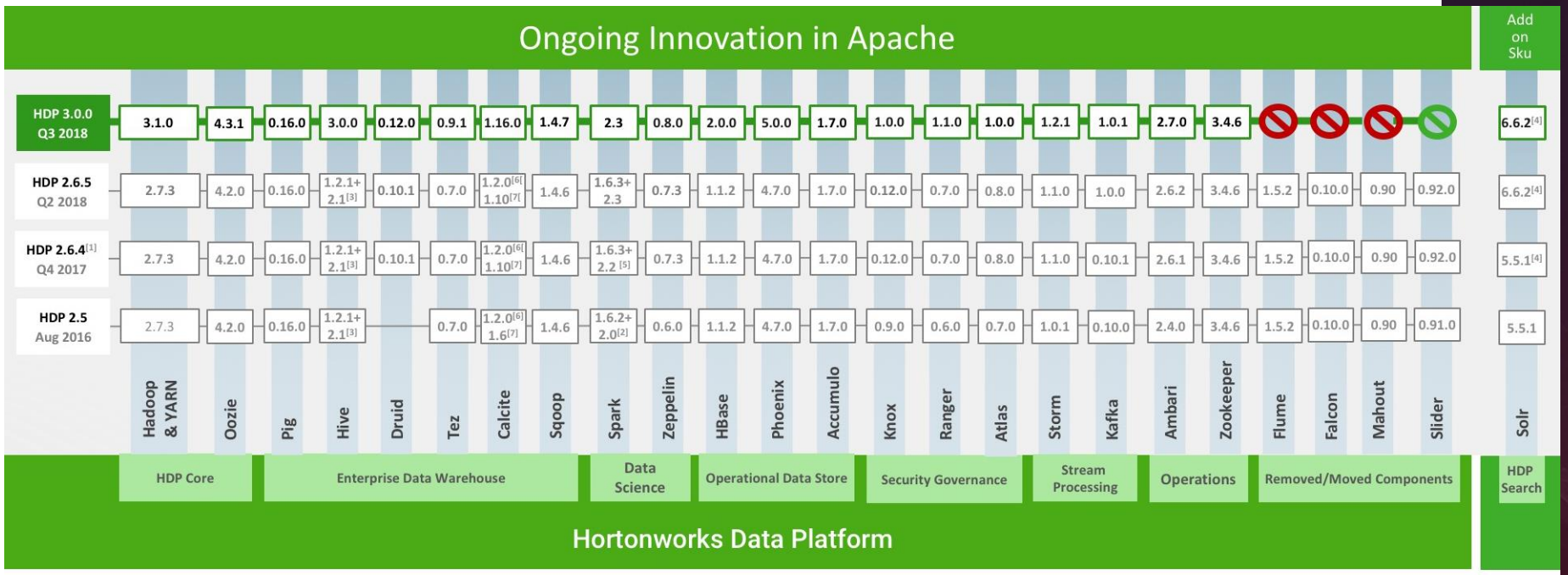
Verze mezi releasy



** Spark 1.6.2+ Spark 2.0 – HDP 2.5 support installation of both Spark 1.6.2 and Spark 2.0. Spark 2.0 is Technical Preview within HDP 2.5.*
Hive 1.2.1+ Hive 2.1 – Hive 2.1 is Technical Preview within HDP 2.5.

PROFINIT

Verze mezi releasy



[1] HDP 2.6 – Shows current Apache branches being used. Final component version subject to change based on Apache release process.
[2] Spark 1.6.3+ Spark 2.1 – HDP 2.6 supports both Spark 1.6.3 and Spark 2.1 as GA.
[3] Hive 2.1 is GA within HDP 2.6.
[4] Apache Solr is available as an add-on product HDP Search.
[5] Spark 2.2 is GA

PROFINIT

Koloběh technologií

- Nadšení
  - Našel jsem skvělou technologii ! Vyřeší naše problémy.
- Realita – o několik hodin později
  - Kde je dokumentace?
- Vystřízlivění – podle povahy o několik hodin, až jednotek dní později
  - Ono to asi vážně nefunguje.
- Zklamání – o několik zoufalých dnů později
  - Nefungují ani příklady na webu ! Kdo probůh tohle vyvíjí ?
  - Porozhlédneme se tedy jinde…

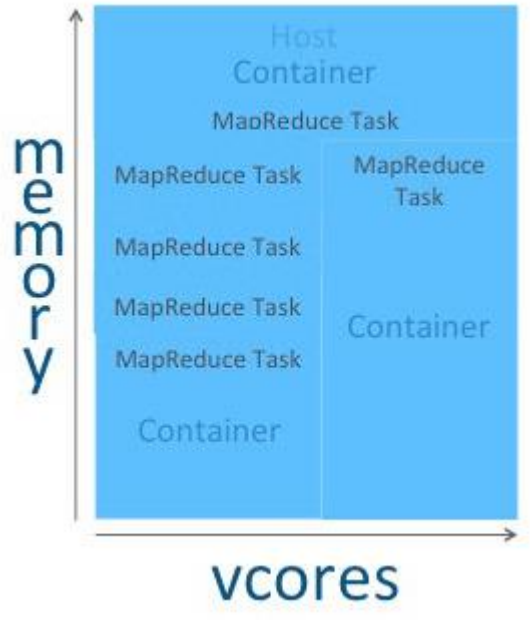
YARN

YARN

- **Y**et **A**nother **R**esource **N**egotiator
- tj. plánovač a alokátor zdrojů
  - paměť
  - CPU
  - počet vláken
  - síť...
- Většinou se využívá transparentně, uživatel o něm moc neví, záleží ale hodně na konfiguraci
- Ne všechny aplikace YARN využívají, např. Impala má vlastní plánovač
  - každý alokátor by tak měl mít výhradní zdroje...

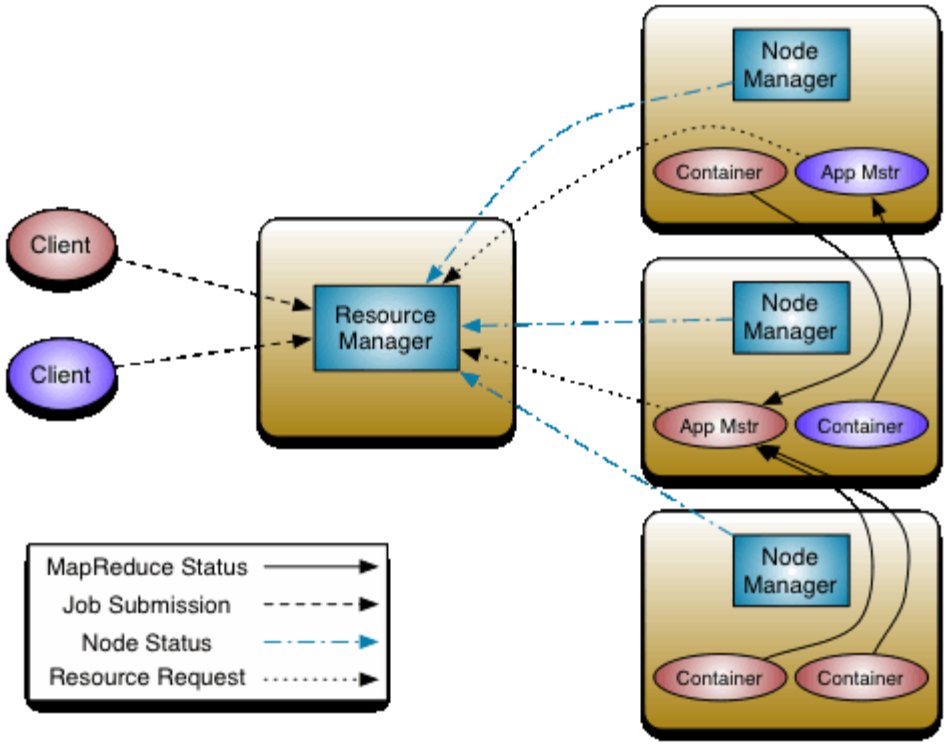
YARN

- Application – klientská aplikace
- Container – zdroje přiřazené aplikaci na konkrétním nodu
- Resource Manager – globální správce zdrojů pro cluster
- Node Manager – podřízený správce zdrojů na nodu



PROFINIT

YARN



PROFINIT

Díky za pozornost

PROFINIT

Profinit, s.r.o.
Tychonova 2, 160 00 Praha 6

Telefon
+ 420 224 316 016

Web
www.profinit.eu

LinkedIn
linkedin.com/company/profinit

Twitter
twitter.com/Profinit_EU