

Apache Hadoop

Værktøjskasse til
Big Data



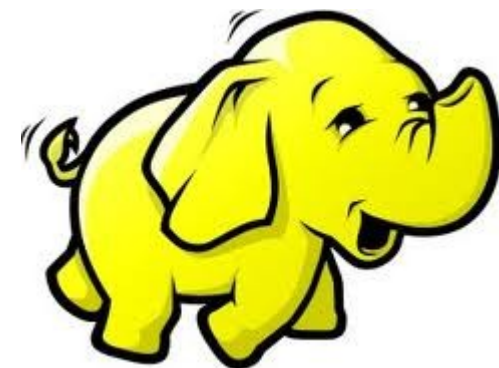
epSos.de

Kenneth Geisshirt
DKUUG - 21. august 2012

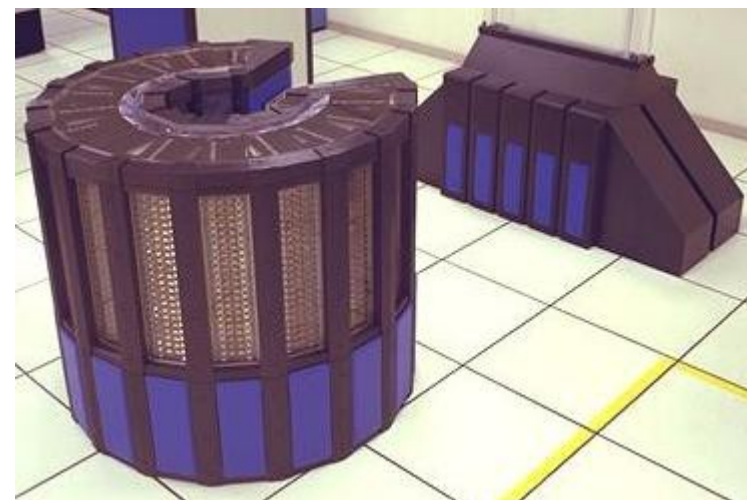


Agenda

- Hvad er Big Data?
- Hadoop (version 1.0.x)
 - Komponenter og arkitektur
 - Parallellisering med Map/Reduce
 - Installation
 - Et lille M/R-job
- Applikationer
 - Pig, HBase, Hive



- High Performance Computing
 - CPU-orienteret
 - Linux-klynger meget populære men special-netværk (InfiniBand)
 - NFS som centralt filsystem
 - Køsystemer: SGE, Torque
 - Parallellisering med MPI
 - Sprog: Fortran-77/9x, C, C++



- Big Data og data science handler om data
 - Komplexiteten vokser
 - Mængden vokser
- Eksempel: Sloan Digital Sky Survey (DR9)
 - Næsten 1 mia. objekter!
 - 1,5 mio. spektra af galakser
 - 100 TB
- Eksempel: Bilka (2008-tal)
 - 700 brugere
 - 10 TB i data warehouse

- Statistik, statistikk og statistikk
 - Programmeringssproget R er populært
- Machine learning
 - Support-vector machines, klassifisering, unsupervised learning
- Databaser – og data warehouses
 - Tradisjonelle relationsdatabaser og NoSQL

```
# SSmicmen
# Michaelis-Menten enzyme reaction
#      ka      kb
# E + S <--> ES --> E + P
#      ka'
#
# Vm = kb*[E](0)
# K  = (kb+ka')/ka

# two experimental data sets - time vs [P]
time <- c(0.02, 0.02, 0.06, 0.06, 0.11, 0.11, 0.22, 0.22, 0.56, 0.56, 1.10, 1.10)
concP <- c(76, 47, 97, 107, 123, 139, 159, 152, 191, 201, 207, 200)

# fit data
fn <- function(p) sum((concP - (p[1] * time)/(p[2] + time))^2)
out <- nlm(fn, p = c(200, 0.1), hessian = TRUE)
df <- data.frame(x=time, y=concP)
fit <- nls(concP ~ SSmicmen(time, Vm, K), df)

# plot data and fit
postscript("file.ps", horizontal=FALSE, height=5, pointsize=10)
plot(time, concP)
lines(time, predict(fit, data.frame(time)))
```

NoSQL

- Tåbeligt navn
- Mange varianter
 - Nøgle/værdi-par, søjle-orienteret, dokumenter, grafer
- Ofte distribueret og fejl-tolerant
- Ikke altid fuld ACID
- Ikke altid SQL som søgesprog

Hyldevarer

- Hyldevarer er i dag stor kapacitet
 - 2 TB harddisk koster 800 kr
 - 8 GB ram koster 325 kr
 - Quad core i7 koster 2500 kr
- Billigt og let at bygge “supercomputere”
- Strøm og køling er en faktor
- Open Source Software har hele stakken
 - Kerne, web-server, udviklingsmiljø, database, filserver

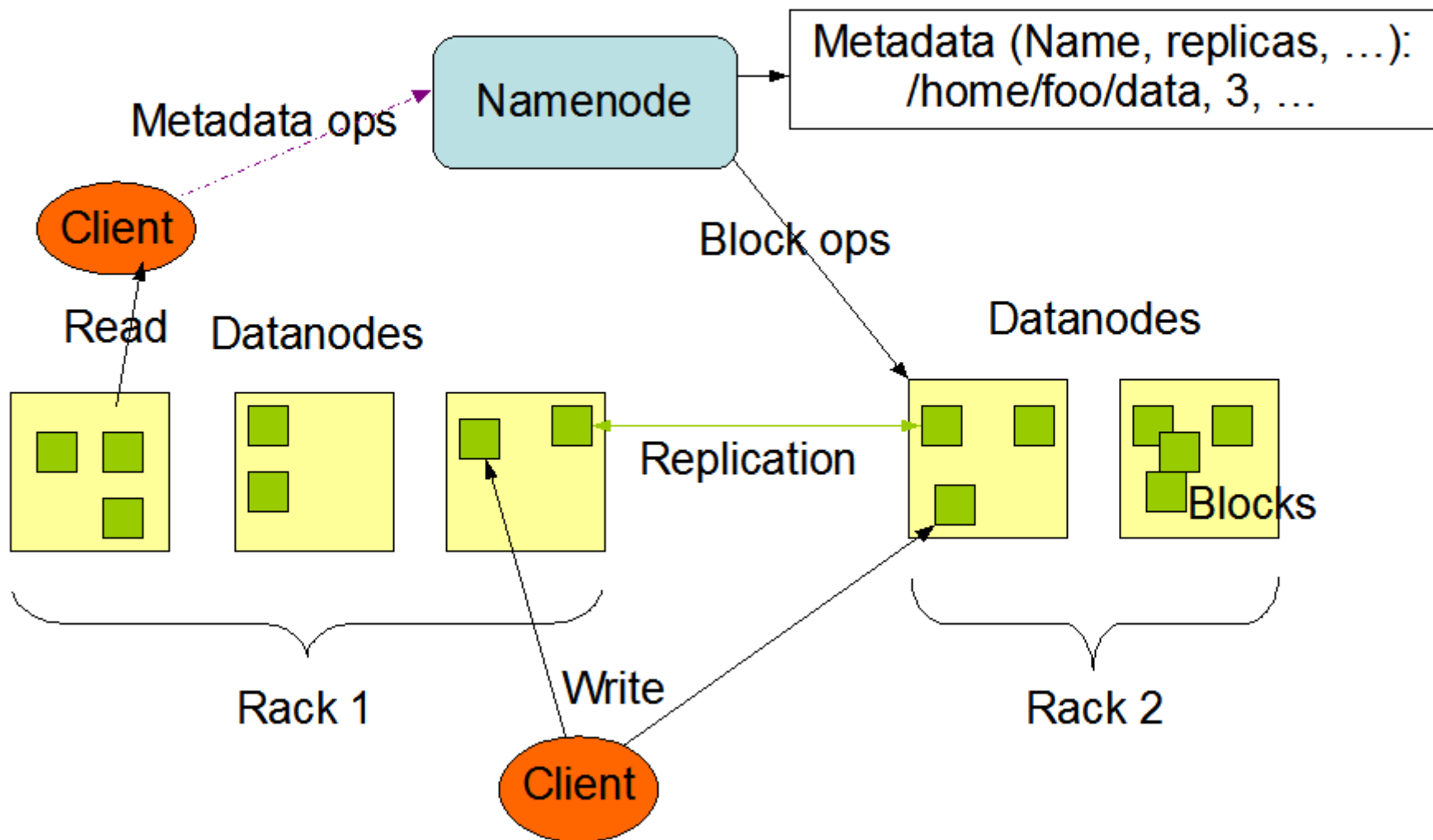
- Platform til analyse af store datamængder
 - Parallelprogrammering vha. Map/Reduce
 - Fejl-tolerent
- Open Source Software
 - Primært udviklet i Java
 - Apache varetager juridiske og organisatoriske interesser
- Inspireret af Googles BigTable og MapReduce
- I brug mange steder
 - Yahoo! har 10.000+ core klynge
 - Facebook har 30 PB i deres klynge

Komponenter

- NameNode
 - Holder styr på filers metadata
 - JobTracker
 - Styrer brugernes job
 - TaskTracker
 - Jobs brydes ned i mindre opgaver
 - DataNode
 - Distribuerer filer udover klyngen
 - Blokke er pr. default 64 MB
 - Blokke replikeres (typisk til 3 maskiner)
-
- ```
graph LR; Master[Master] --> NameNode; Master --> JobTracker; MapReduce[Map/Reduce] --> JobTracker; MapReduce --> TaskTracker;
```

# Arkitektur

## HDFS Architecture



# Map/Reduce

$$\sum_{k=1}^N x_k = \sum_{j=0}^{n-1} \sum_{i=1}^{N/n} x_{j+n \cdot i}$$

- Map
    - Data spredes over  $n$  maskiner
    - Delsummer udregnes parallelt
  - Reduce
    - Endelig sum udregnes
- 
- HDFS spreder data i 64 MB blokke på  $n$  servere
  - Jobtracker holder styr på jobbet
  - Jobbet brydes ned i  $n$  mindre opgaver (tasks)
    - Tasktracker holder styr på opgaverne
  - Hver opgave arbejder (helst) på lokal datablok

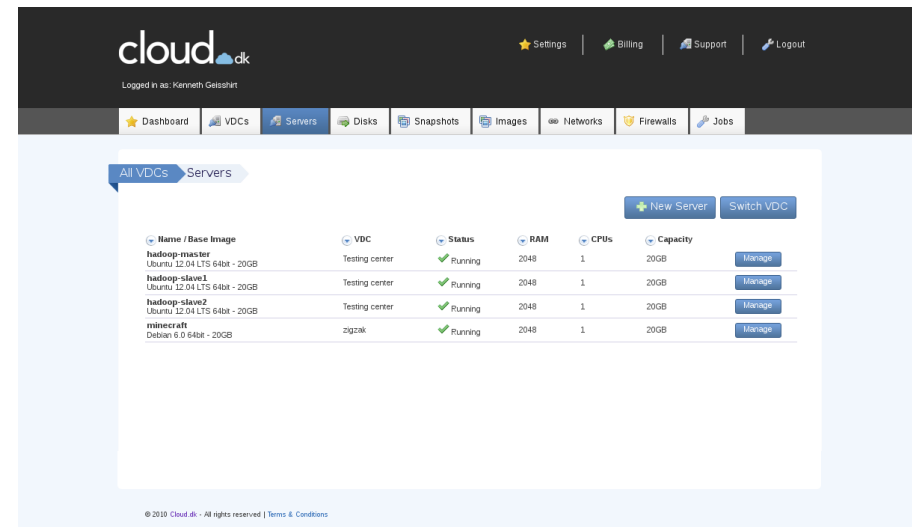
- Implementation af klassen  
`org.apache.hadoop.fs.FileSystem`
- HDFS – meget udbredt (default)
- Local
- HDTP – read only HTTP-adgang til HDFS
- HSFTP – read only HTTPS-adgang til HDFS
- HAR – pakkede og komprimerede filer
- Og en del flere

# Installation

- God guide:
  - <http://www.michael-noll.com/tutorials/running-hadoop-on>
  - Klynge med 1 maskine til lege
- Linux er default
  - Ubuntu server er glimrende
  - Hadoop Ubuntu Packagers
    - 4 medlemmer
    - Tre udgaver: development, testing og stable

# Min test-klynge

- Servere findes hos cloud.dk – tak til dem!
- Ubuntu Linux 12.04 (server) 64 bit – Precise Pangolin
- Master: 77.66.109.234
- 2 slaver: 77.66.109.235, 77.66.109.236
- 1 CPU, 2 GB ram, 20 GB harddisk



The screenshot shows the cloud.dk management interface. The top navigation bar includes links for Settings, Billing, Support, and Logout. Below this is a secondary navigation bar with icons for Dashboard, VDCs, Servers, Disks, Snapshots, Images, Networks, Firewalls, and Jobs. The main content area is titled 'All VDCs > Servers' and features a 'New Server' button and a 'Switch VDC' button. A table lists the following servers:

| Name / Base Image                              | VDC            | Status  | RAM  | CPUs | Capacity | Manage |
|------------------------------------------------|----------------|---------|------|------|----------|--------|
| hadoop-master<br>Ubuntu 12.04 LTS 64bit - 20GB | Testing center | Running | 2048 | 1    | 20GB     | Manage |
| hadoop-slave1<br>Ubuntu 12.04 LTS 64bit - 20GB | Testing center | Running | 2048 | 1    | 20GB     | Manage |
| hadoop-slave2<br>Ubuntu 12.04 LTS 64bit - 20GB | Testing center | Running | 2048 | 1    | 20GB     | Manage |
| minecraft<br>Debian 6.0 64bit - 20GB           | zigzak         | Running | 2048 | 1    | 20GB     | Manage |

© 2010 Cloud.dk - All rights reserved | Terms & Conditions

- Ubuntu er holdt op med at understøtte SUN/Oracle Java
  - Se <http://www.duinsoft.nl/packages.php>
- Tilføj deb `http://www.duinsoft.nl/pkg`  
`debs all til /etc/apt/sources.list`
- Tilføj rette signatur:
  - `apt-key adv --keyserver`  
`keys.gnupg.net --recv-keys 5CB26B26`
- Installation er nu klar
  - `apt-get update ; apt-get install`  
`update-sun-jre`



- En anden mulighed er at bruge webupd8team PPA
- <http://www.webupd8.org/2012/01/install-oracle-java-jdk->

```
add-apt-repository ppa:webupd8team/java
```

```
apt-get update
```

```
mkdir -p /usr/lib/mozilla/plugins
```

```
apt-get install oracle-jdk7-installer
```

- Tak til Jarl Friis for at videregive denne opskrift

# Hadoop et al

- Ubuntu PPA:
  - <https://launchpad.net/~hadoop-ubuntu/+archive/stable>
- `apt-add-repository ppa:hadoop-ubuntu/stable`
- `apt-get update`
- `apt-get install hadoop pig hive hbase`
- Gentag for alle maskiner i klyngen

## Særlig bruger

- Opret brugeren hduser: `adduser --ingroup hadoop hduser`
- Tilføj `JAVA_HOME` til `.bashrc`
- Tilføj Hadoops placering til `PATH` i `.bashrc`

- Ændre servernavn i /etc/hostname
- /etc/hosts – på alle tre servere

```
77.66.109.234 master master.localdomain
```

```
77.66.109.235 slave-1 slave-1.localdomain
```

```
77.66.109.236 slave-2 slave-2.localdomain
```

- Generér SSH-nøgle og distribuer den til slaverne

```
ssh-keygen -t dsa -P ""
```

```
cat .ssh/id_dsa.pub >> .ssh/authorized_keys
```

```
scp -r .ssh slave-1: ; scp -r .ssh slave-2:
```

# Lokalt filsystem

- HDFS' blokke er placeret lokalt

```
mkdir -p /app/hadoop/tmp
```

```
chown hduser.hadoop /app/hadoop/tmp
```

- Du skal overveje harddisk-teknologi og kapacitet nøje
- Linux-filsystemers parametre er nok også vigtige

# Konfiguration

- `/etc/hadoop/conf` indeholder konfiguration
  - `masters` angiver master, mens `slaves` angiver alle nodes
  - `core-site.xml` anviser hvor HDFS må placere filer
  - `hdfs-site.xml` styrer hvor mange replika, vi ønsker
  - `mapred-site.xml` angiver hvilken server, som styrer Map/Reduce-jobbene
- Javas placering anvises i `hadoop-env.sh`
- Konfigurationen skal være synkroniseret mellem master og slaver

# Op på ørene vi er kørende

- Kun på master (som hduser):
  - Formatering af HDFS: `hadoop namenode -format`
  - Start datanode/HDFS: `start-dfs.sh`
  - Start jobtracker o.lign: `start-mapred.sh`
- Slaverne vil blive startet op automatisk
- Nedlukning med `stop-dfs.sh` og `stop-mapred.sh`

# Web-interface

- <http://master:50070> - Filerne
- <http://master:50030> - Map/Reduce
- <http://master:50060> - Tasks



# Test-data

- *On time flight data* fra Bureau of Transportation Statistics
  - [http://www.transtats.bts.gov/Fields.asp?Table\\_ID=236](http://www.transtats.bts.gov/Fields.asp?Table_ID=236)
  - Alle flyafgang i USA: ~110 værdier for hver afgang
- Januar-juni 2012:
  - ~1,3 GB og ~3 mio. rækker
  - ~20 blokke til tre servere eller 6-7 blokke/server
- Klargøring:

```
grep -v ^2 *.csv | tr -d "\"" >
On_Time_Performance_2012_H1.csv
```

- Kopiering til HDFS:

```
hadoop dfs -copyFromLocal
On_Time_Performance_2012_H1.csv /
```

# Map/Reduce

- Skrives i Java med underklasser
  - `SELECT Origin, COUNT(*) GROUP BY Origin`
  - Map:
    - linje for linje splittes ved komma og lufthavn findes
    - Returnerer `<lufthavn, 1>`
  - Reduce:
    - Tæller lufthavn op
  - Bruger også en “combiner” som er en lokal reducer
  - Oversættelse:

```
javac -classpath /usr/lib/hadoop/hadoop-
core-1.0.2.jar -d ontime_classes
OnTime.java
```

```
jar -cvf ontime.jar ontime_classes
```

# Kør Map/Reduce job

- Output skal have en folder (må ikke eksistere)

```
hadoop dfs -rmr /output
```

- Selve kørslen sker med

```
hadoop jar ontime.jar dk.dkuug.OnTime
/On_Time_Performance_2012_H1.csv /output
```

- Læs resultatet med

```
hadoop dfs -cat /output/part-00000
```

- HBase er en database
- Søjle/kolonne-orienteret
- Distribueret gennem HDFS
- Når data er skrevet i Hbase, kan de ikke
  - Ændres
  - slettes

# Grisene kommer

- Pig er et dataflow-sprog til dataanalyser
  - Omskriver automatisk til Map/Reduce jobs
  - LOAD kan indlæse en CSV-fil
  - Mulighed for JOIN, GROUP, ORDER
- Skema til *On Time* data:

```
head -1
```

```
On_Time_On_Time_Performance_2012_1.csv
| tr -d "\"" | tr ", " "\n" | while
read f ; do echo -n "$f:chararray, " ;
done
```

- Desværre får jeg “out-of-memory”



- Hive er et data warehouse
  - Data er gemt i HDFS
  - SQL-lignende syntask
  - Automatisk Map/Reduce
- Lettest at bruge fra kommando-linje

- Hadoops hjemmeside:
  - <http://hadoop.apache.org/>
- Googles oprindelige artikel:
  - <http://research.google.com/archive/bigtable.html>
- Gode guides til opsætning:
  - <http://www.michael-noll.com/tutorials/>
- Online-kursus (lige nu):
  - <http://ampcamp.berkeley.edu/>

- *Hadoop: The Definitive Guide, 2nd edition*. T. White. O'Reilly Media, 2010.
- *Hbase: The Definitive Guide*. L. George. O'Reilly Media, 2011.
- *Programming Pig*. A. Gates. O'Reilly Media, 2011.
- *Data-Intensive Text Processing with MapReduce*. J. Lin & C. Dyer. Morgan&Claypool, 2010.



# Konklusioner

- Opsætning er ikke trivielt
- Hadoop er laaaangsomt
- HDFS er noget bøvet at arbejde med
  - Filsystemer i Java?
- Interessant indgangsvinkel på (automatisk) parallellisering