

Cloud Computing

Cap 1. Introdudere. Bazele MapReduce

October 29, 2022

- 1 Clouds
- 2 Noțiunea de cloud
- 3 Istoric
- 4 Caracteristici
- 5 Costuri
- 6 Paradigma MapReduce

- 1 Clouds
- 2 Noțiunea de cloud
- 3 Istoric
- 4 Caracteristici
- 5 Costuri
- 6 Paradigma MapReduce

Modă sau necesitate?

- Gartner, 2009: Cifra de afaceri a cloud computing va depăși \$150 miliarde în 2013. Va reprezenta 19 % din cheltuielile IT în 2015.
- IDC, 2009: Cheltuielile IT cu serviciile cloud se vor tripla în următorii 5 ani, ajungând la \$42 miliarde.
- Forrester, 2010: Volumul operațiilor cu cloud computing va crește de la \$41 miliarde în 2010 la \$241 miliarde în 2020.

Furnizori

- AWS: Amazon Web Services
 - EC2: Elastic Compute Cloud
 - S3: Simple Storage Service
 - EBS: Elastic Block Storage
- Microsoft Azure
- Google Compute Engine
- Oracle, VMWare, Yahoo, Cloudera, Gigaspaces, Datastax
- alții

Categorii de clouds

- public clouds vs. private clouds
- public clouds oferă servicii clienților de tip pay-per-use:
 - Amazon S3 (Simple Storage Service): key-value store, stochează date de orice tip, plata se face pe GB pe luna de stocare
 - Amazon EC2 (Elastic Compute Cloud): upload, rularea oricărei imagini de sistem de operare (SO), plata pe CPU pe ora de rulare
 - Google Application Engine / Compute Engine: dezvoltare de aplicații ce vor fi rulate pe Application Engine, urmate de upload-ul lor și rulare

Clienții economisesc timp și bani

- Dave Power, consultant Eli Lilly: “Cu AWS, un nou server poate fi up-and-running în trei minute, (Eli Lilly avea nevoie de șapte săptămâni pentru a instala intern un server) iar în cluster Linux de 64 de noduri poate fi online în 7 minute (comparat cu o durată internă de trei luni)”
- Ingo Elfering, VP GlaxoSmithKline: “Cu serviciile online, costurile operaționale IT pot fi reduse cu aproximativ 30%”
- Jim Schwartz, CIO, Sybase: “Un cloud privat în datacenter ne-a economisit aproape \$2 milioane anual din 2006, deoarece compania poate pune în comun puterea de procesare și resursele de stocare”
- sute de startup-uri pot folosi efectiv resurse largi de calcul fără a face investiții masive în hardware

- 1 Clouds
- 2 Noțiunea de cloud
- 3 Istoric
- 4 Caracteristici
- 5 Costuri
- 6 Paradigma MapReduce

Cloud (nor de calcul)

- cluster?
- supercomputer?
- datastore?
- nici una din cele de mai sus
- toate cele de mai sus

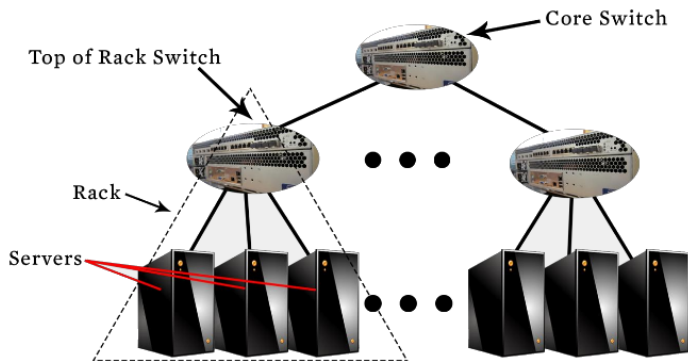


- **cloud:** capacitate mare de stocare + putere de procesare lângă date

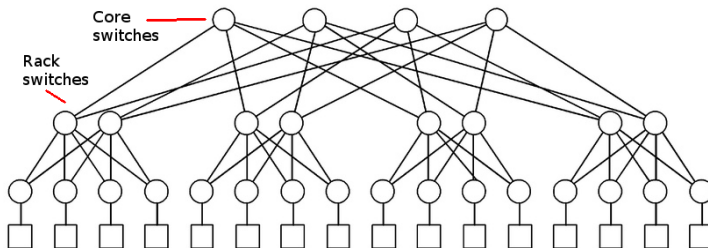
Ce este un nor de calcul (cloud)

- un datacenter (single-site cloud) constă în
 - noduri de calcul grupate în rack-uri
 - switch-uri, ce conectează rack-urile
 - topologie de rețea ierarhică, de tip arbore (hyper-arbore)
 - noduri de stocare (storage backend), conectate în rețea
 - nod front-end pentru preluarea job-urilor și servirea clienților
 - servicii software
- un cloud distribuit geografic constă în
 - mai multe datacenters
 - fiecare site poate avea structură și servicii diferite

Exemplu de topologie



Exemplu de hyper-arbore binar

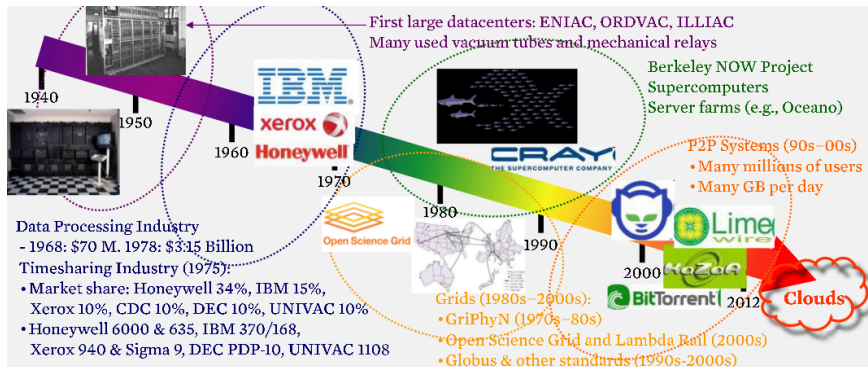


- 1 Clouds
- 2 Noțiunea de cloud
- 3 Istoric**
- 4 Caracteristici
- 5 Costuri
- 6 Paradigma MapReduce

Istoric - sisteme batch

- 1940 - 1960: primele datacenters
- 1960 - 1980: time-sharing (sisteme batch)
- 1985: primele PC-uri (non-distributed)
- 1980 - 2012: grids, clusters
- 1995 - 2008: peer-to-peer systems
- 2004 - now: clouds, datacenters

Timeline



Legea lui Moore

- observare: numărul de tranzistori într-un circuit integrat se dublează la fiecare doi ani (Gordon E. Moore, co-founder, Intel, 1975)
- perioade de dublare:
 - storage: 12 luni
 - lăţime de bandă: 9 luni
 - putere de calcul CPU: 18 luni
- bandwidth: 56kbps în 1985, de ordinul Tbps în prezent
- capacitate de stocare: 10MB HDD PC XT în 1988, TBs în prezent
- producţie de date: simulări de molecule simple în 1990, LHC produce în ordinul PBytes (10^{15}) / an

Profeții

- în 1965, Fernando Corbato (MIT), designer al sistemului MULTICS - GE, Honeywell (precursorul UNIX) a previzionat apariția unui serviciu de calcul disponibil ca utilitate, precum compania de apă sau de energie electrică
- legea lui Corbato: “The number of lines of code a programmer can write in a fixed period of time is the same, independent of the language used.” :-P

- 1 Clouds
- 2 Noțiunea de cloud
- 3 Istoric
- 4 Caracteristici**
- 5 Costuri
- 6 Paradigma MapReduce

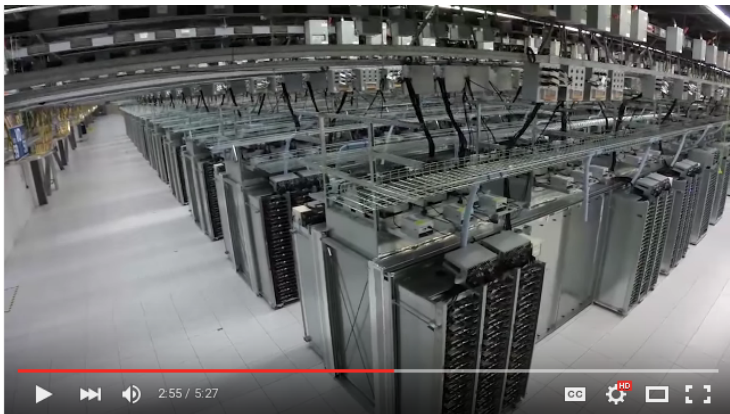
Caracteristicile cloud-ului

- scară mare, masiv
 - datacenters de mii de noduri
- acces on-demand
 - oricine îl poate accesa
 - pay-per-use
- data intensive
 - stocare de la MB la TB (10^{12}), PB și EB (informal 10^{18} , un giga de GB)
 - log-uri, web data
 - spre comparație, Wikipedia se comprimă în ordinul GBytes
- paradigme noi de programare
 - MapReduce - Hadoop, NoSQL - Cassandra - MongoDB
 - ușurință în programare, accesibilitate
 - open-source
- combinații ale acestor tehnologii dau naștere unor noi probleme nerezolvate (încă) de calcul distribuit (de tip cloud)

Exemple - număr de noduri

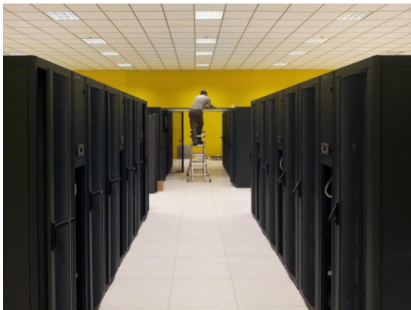
- Facebook: 30K în 2009, 60K în 2010, 180K în 2012
- Microsoft: 150K; creștere de 10K / lună; 80K rulează Bing
- Yahoo!: 100K în 2009; împărțire pe clustere, 4000 noduri / cluster
- AWS EC2: 40K noduri în 2009, 8 cores / nod
- eBay: 50K mașini, 2012
- HP: 380K noduri în 2012, în 180 datacenters
- Google: 1 Mio (estimare, 2013); 45K noduri / datacenter (2005)

Inside a Google data center

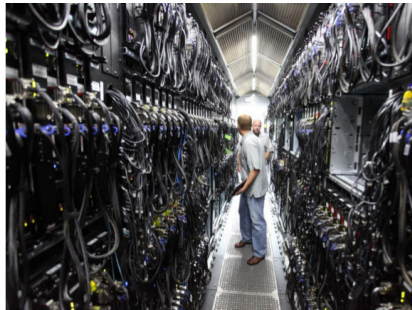


<https://www.youtube.com/watch?v=XZmGGAbHqa0>

Servers



Front view



Rear view

Power Usage Efficiency



- $WUE = \frac{\text{Water Usage}}{\text{IT Equipment Energy}}$
(l / kWh)
- $PUE = \frac{\text{Total Facility: Power}}{\text{IT Equipment Power}}$
(supraunitar, cât mai aproape de 1)
- off-site
- on-site

Răcire



- admisie de aer
- umidificarea cu apă purificată
- ventilarea sistemului cu aer rece
- evacuarea aerului cald

Acces la cerere

- on-demand: ca și închirierea unei mașini vs. cumpărarea uneia
 - AWS EC2: de la câțiva cenți la câțiva \$ pe oră
 - AWS S3: de la câțiva cenți la câțiva \$ pe GB / lună
- HaaS: Hardware as a Service
 - acces la mașinile fizice, pentru a face orice cu ele (propriul cluster)
 - implică riscuri de securitate (cine are acces fizic la ele)
- IaaS: Infrastructure as a Service
 - acces la o arhitectură de calcul flexibilă (virtualizare)
 - AWS EC2, AWS S3, Eucalyptus, Microsoft Azure

Acces la cerere

- PaaS: Platform as a Service
 - acces flexibil la arhitectura de calcul și cea de stocare, cuplată cu o platformă software
 - Google Application Engine (Python, Java, Go)
- SaaS: Software as a Service
 - acces la servicii software, la cerere
 - Google Docs, MS Office on demand, Dropbox

Calcul data-intensive

- computation-intensive
 - exemple: MPI-based, HPC¹, grids
 - supercomputers (NCSA's Blue Waters²)
- data-intensive
 - date stocate în datacenters
 - folosirea nodurilor de procesare din imediata vecinătate a datelor
- în data-intensive computing, accentul se mută de la calcul la date: încărcarea CPU nu mai este cea mai importantă metrică, locul ei fiind luat de factorul I/O (atât disk-ul cât și rețeaua)

¹High Performance Computing

²https://en.wikipedia.org/wiki/Blue_Waters

Paradigme de programare pentru Cloud

- de obicei ușor de scris și de exprimat programe cu grad ridicat de paralelism
- Google: MapReduce, Sawzall (procedural)
- Amazon: Elastic MapReduce service (pay-per-use)
- Google MapReduce: indexarea se face cu un chain de 24 job-uri MR; 200K joburi procesând 50PB / lună (2006)
- Yahoo!: Hadoop + Pig; WebMap este un chain de 100 job-uri MR; 280 TB, 2500 noduri
- Facebook: Hadoop + Hive; 300 TB, adaugă 2 TB / zi (2008); 3K jobs ce procesează 55 TB / zi
- NoSQL: MySQL este standard industrial de-facto, dar Cassandra este de 2400 ori mai rapid

- 1 Clouds
- 2 Noțiunea de cloud
- 3 Istoric
- 4 Caracteristici
- 5 Costuri**
- 6 Paradigma MapReduce

Categorii de clouds

- pot fi fie publici fie privați
- private clouds - doar angajaților
- public clouds - oricui plătește
- pentru un nou serviciu / companie: e mai rentabilă folosirea unui cloud public, sau alcătuirea propriului cloud?

Outsourcing sau ownership?

- problema: o organizație medie vrea să ruleze un serviciu pentru câteva luni
 - 128 servere (1024 cores)
 - 524 TB
- outsourcing (via AWS, cost lunar)
 - costul S3: \$0.12 / GB / lună
 - pentru 524 TB: $0.12 * 524 * 1000 = \$62K$
 - costul EC2: \$0.10 / CPU / oră
 - costul total pe lună: storage + CPU =
 $62K + 0.10 * 1024 * 24 * 30 = 136K$
- ownership (cost lunar)
 - storage: \$349K / nr. luni
 - total: \$1555K / nr. luni + 7.5K (1 sysadm la 100 noduri)

Outsourcing sau ownership?

- analiza costurilor: mai preferabil ownership-ul dacă
 - \$349K / nr. luni < \$62K (storage)
 - \$1555K / nr. luni < \$136K (total)
- prag
 - nr. luni > 6 (storage)
 - nr. luni > 12 (total)
- startup-urile folosesc outsourcing
- deținătorii de datacentere publice monetizează cu precădere închirierea spațiului de stocare

- 1 Clouds
- 2 Noțiunea de cloud
- 3 Istoric
- 4 Caracteristici
- 5 Costuri
- 6 Paradigma MapReduce**

Operațiile map și reduce

- termeni împrumutați din programarea funcțională

```
>>> map(lambda x: x * x, [1, 2, 3, 4])
```

```
[1, 4, 9, 16]
```

```
>>> reduce(lambda a, b: a + b, [1, 4, 9, 6])
```

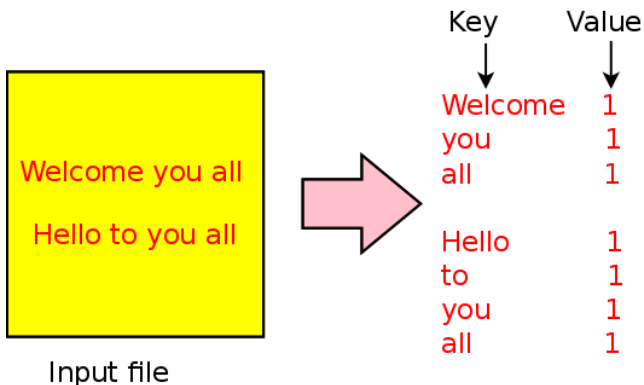
```
20
```

- aplicație exemplu:

pentru un set de date imens (ex. Wikipedia sau o bibliotecă de ordinul TB), se dorește să se determine frecvența de apariție a fiecărui cuvânt, pentru fiecare document

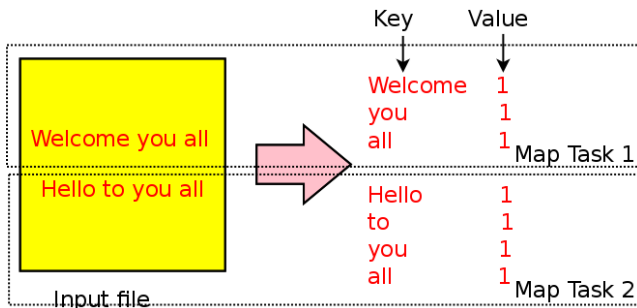
Map

- se procesează individual fiecare înregistrare și se generează perechi (key, value) intermediare



Map

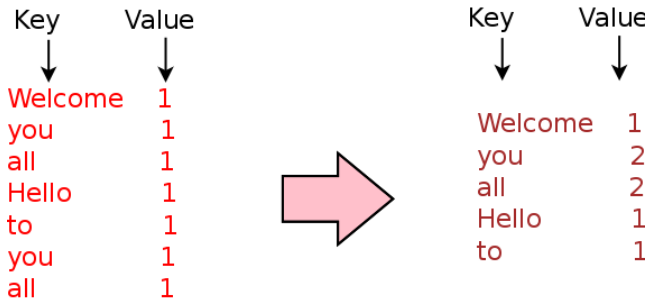
- se procesează **în paralel** fiecare înregistrare și se generează perechi (key, value) intermediare



- de regulă numărul de task-uri map este de un ordin de mărime mai mare decât numărul de noduri disponibile

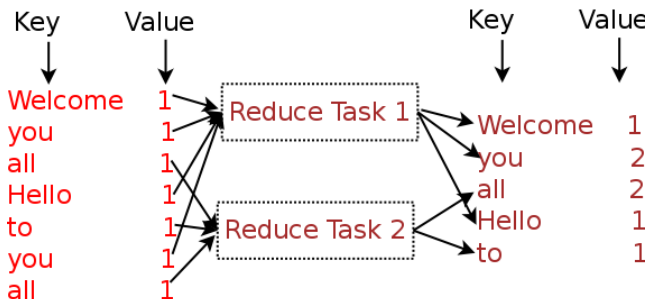
Reduce

- toate valorile asociate cu aceeași cheie se procesează împreună
< key, (listă de valori) >



Reduce

- fiecare cheie este asignată unui singur reducer
- procesarea paralelă se face prin partiționarea cheilor



- default, se realizează hash partitioning
node nr. = $\text{hash}(\text{key}) \bmod \text{nodes}$

Codul mapper-ului în Hadoop

```
public static class TokenizerMapper
    extends Mapper<Object, Text, Text, IntWritable>{

private final static IntWritable one = new IntWritable(1);
private Text word = new Text();

public void map(Object key, Text value, Context context
    ) throws IOException, InterruptedException {
    StringTokenizer itr = new
        StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
    }
}
```

Codul reducer-ului în Hadoop

```
public static class IntSumReducer
    extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
        Context context
        ) throws IOException,
            InterruptedException {

        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```


Codul driver-ului în Hadoop

```
public static void main(String[] args) throws Exception {  
    Configuration conf = new Configuration();  
    Job job = Job.getInstance(conf, "word count");  
    job.setJarByClass(WordCount.class);  
    job.setMapperClass(TokenizerMapper.class);  
    job.setCombinerClass(IntSumReducer.class);  
    job.setReducerClass(IntSumReducer.class);  
    job.setOutputKeyClass(Text.class);  
    job.setOutputValueClass(IntWritable.class);  
    FileInputFormat.addInputPath(job, new Path(args[0]));  
    FileOutputFormat.setOutputPath(job, new Path(args[1]));  
    System.exit(job.waitForCompletion(true) ? 0 : 1);  
}
```

Exemplul Hadoop pentru WordCount

- sursa:

```
https://hadoop.apache.org/docs/current/  
hadoop-mapreduce-client/hadoop-mapreduce-client-core/  
MapReduceTutorial.html#Source_Code
```

- combiner-ul are rolul de a reduce la nivelul unui nod valorile emise de mapper (reduce în mapper)
- va avea intrare și ieșire identice
- opțional

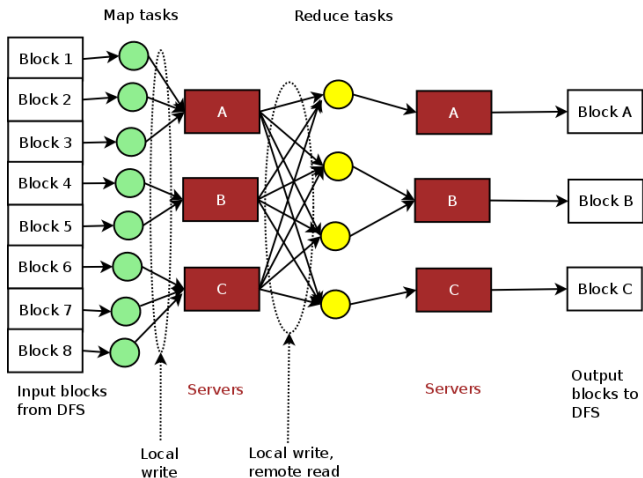
Programarea MapReduce

- extern: dpdv. al utilizatorului
 - program tip map, program tip reduce (de regulă scurte)
 - submit job, preluare rezultate
 - nu necesită cunoștințe de programare paralelă
- intern: pentru planificator
 - paralelizare mapper
 - transfer date de la mapper la reducer
 - paralelizare reducer
 - implementare storage pentru mapper input/output, reducer input/output
 - variantă: nici un reducer nu poate porni înainte ca toți mapper-ii să se fi terminat (barrier)
 - în practică reducer-ul poate porni când are un număr suficient de perechi, înainte de terminarea mapper-ilor

MapReduce internals (1)

- paralelizare map
 - fiecare task e independent de celelalte
 - toate perechile generate de mapper care au aceeași cheie sunt trimise spre același reducer
- transferul datelor de la mapper la reducer
 - toate perechile generate de mapper care au aceeași cheie sunt trimise spre același reducer
 - folosirea unei funcții de partiționare (implicit este funcția de hash-ing)
- paralelizare reduce
 - fiecare task e independent de celelalte
- storage pentru mappers și reducers
 - map input: din DFS (Distributed File System)
 - map output: filesystem local
 - reduce input: de pe discuri remote multiple; folosește local filesystems și rețeaua de interconectare
 - reduce output: DFS
- DFS poate fi: HDFS (Hadoop DFS), GFS (Google filesystem)

MapReduce internals (2)



- Resource Manager asignează mapper-ele și reducer-ele către servere