Apache Hadoop

Uvod, konfiguracija i primeri

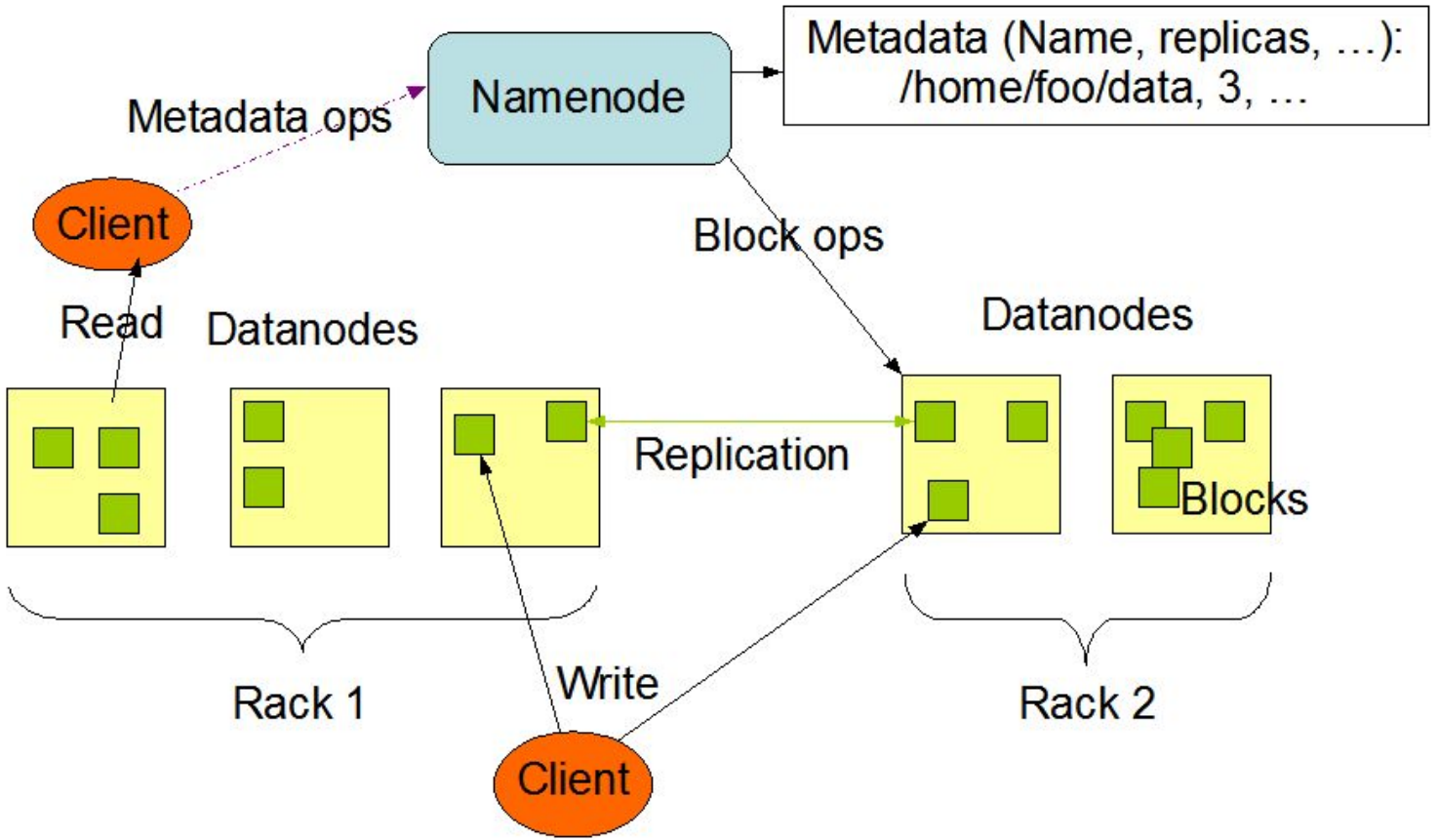
Hadoop - uvod

- *Hadoop* je distribuirani sistem za skladištenje i analizu velike količine podataka
- *Hadoop* radni okvir obuhvata sledeće osnovne module:
  - *Hadoop Common* - sadrži biblioteke i programe koje koriste ostali moduli,
  - *Hadoop Distributed File-System (HDFS)* - distribuirani fajl sistem na kom se skladište podaci,
  - *Hadoop Yet Another Resource Negotiator (YARN)* - platforma odgovorna za upravljanje resursima,
  - *Hadoop MapReduce* - implementacija *MapReduce* programskog modela za distribuiranu obradu velike količine podataka
- *Hadoop* - osobine:
  - projektovan za skladištenje i obradu velike količine podataka,
  - otpornost na parcijalne otkaze u distribuiranom sistemu,
  - lako skaliranje i fleksibilnost u obradi podataka,
  - radi na “običnom” hardveru.

HDFS - uvod

- Distribuirani sistem datoteka Hadoop (*eng. Hadoop Distributed File System,* HDFS)
- izuzetno velike datoteke
- optimizovane metode za čitanje podataka
- regularni hardver (eng. *commodity hardware*)
- veličina bloka 128MB (ili više)
- sastoji se od:
  - imenski čvorovi (eng. *namenodes*) - sadrže meta-podatke
  - čvorovi sa podacima (eng. *datanodes*) - sadrže blokove

HDFS Architecture



Izvor: https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

HDFS - konfiguracija

- Definisanje konfiguracije u *hdfs-site.xml*
- Podrazumevana konfiguracija HDFS klastera
  - https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml
- Neki često korišćeni parametri
  - dfs.name.dir
  - dfs.data.dir
  - dfs.hosts/dfs.hosts.exclude
  - dfs.blocksize
  - dfs.namenode.handler.count

HDFS - web interfejs

- Izvorni *web* interfejs
  - HDFS namenode
  - Default port 9870
  - Nudi pregled
    - stanja i dostupnosti datanode-ova
    - svih direktorijuma i datoteka u sistemu
- Alternativno - *Hue*
  - Poseban servis
  - *Web-based file browser*

HDFS CLI

- Format komande: hdfs dfs -{{operacija}} [-{{param1} -{{param2}} …]
- Osnovne komande
  - ls, cat
  - cp, mv, rm
  - mkdir, rmdir
- Komunikacija lokalne mašine i HDFS
  - put, get, copyToLocal, copyFromLocal, moveFromLocal
- Zauzeće prostora
  - df, du

HDFS - rad sa CLI

- Napraviti direktorijum /*test* na HDFS (mkdir)
- Uveriti se da je direktorijum kreiran (ls)
- Kreirati neprazan tekstualni fajl *test.txt* proizvoljnog sadržaja u lokalu
- Kopirati *test.txt* na HDFS u *test* direktorijum (put ili copyFromLocal)
- Prikazati sadržaj fajla *test.txt*
- Premestiti fajl *test.txt* u direktorijum *test/new*
- Preimenovati fajl *test.txt* u *new.txt*
- Preuzeti fajl *new.txt* na lokalni direktorijum (get ili copyToLocal)
- Prikazati veličinu svih fajlova u *test* direktorijumu (du)
- Obrisati kompletan direktorijum *test* na HDFS (rm -r -f)

MapReduce

- MapReduce - osnovni pojmovi
  - **posao** (*eng. job*)
    - osnovna jedinica obrade
    - sastoji se od ulaznih podataka, *MapReduce* programa i konfiguracije
  - **zadatak** (*eng. task*)
    - *Hadoop* deli svaki posao na zadatke
    - dva tipa zadatka: **zadatak *map*** i **zadatak *reduce***
      - implementiraju operacije *map* i *reduce*
  - **izvršni čvorovi** (*eng. execution nodes*)
    - čvorovi u distribuiranom sistemu koji kontrolišu izvršenje poslova
    - jedan čvor **upravljač poslovima** (*eng. jobtracker*)
      - upravlja izvršenjem svih poslova i raspoređuje zadatke na čvorove za upravljanje zadacima
    - više čvorova **upravljača zadacima** (*eng. tasktracker*)
      - izvršavaju prosleđene zadatke i šalju informacije o statusu izvršavanju (napretku)

MapReduce

- MapReduce - osnovni pojmovi
  - **paket** (*eng. input split / split*)
    - deo ulaznih podataka koji predstavlja osnovnu jedinicu obrade u zadacima
    - Hadoop kreira jedan zadatak za svaki paket koji je potrebno obraditi
      - radi paralelizacije obrade podataka
    - uvek je tačno definisane veličine (podrazumevano 128MB)
      - obično je jednaka veličini bloka u kojem su podaci uskladišteni (*HDFS*)
      - moguće je ručno podesiti veličinu paketa
      - premali paket - potrebno previše vremena za logistiku
      - preveliki paket - najsporiji računar prouzrokuje veliki zastoj u celokupnoj obradi
  - **slog** (*eng. record*)
    - deo paketa, pojedinačni zapis u paketu
    - nad svakim slogom u paketu se primenjuje *map* operacija

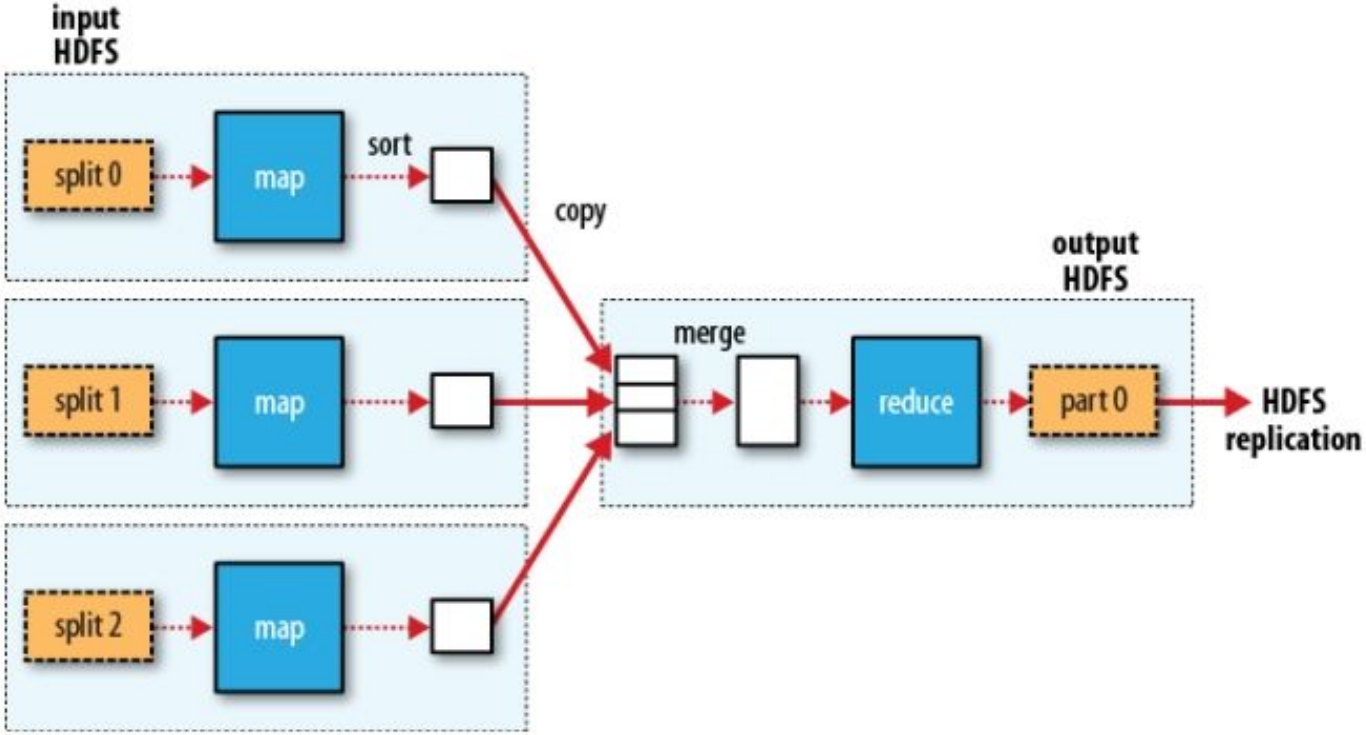
MapReduce

- MapReduce - osnovni pojmovi
  - **lokalizacija obrade podataka** (*eng. data locality optimization*)
    - *Hadoop* pokušava obraditi podatake u onim čvorovima u kojim su podaci skladišteni
      - ukoliko to nije moguće pribegava se izvršenju na drugim, dostupnim čvorovima
    - odnosi se na obradu u operaciji ***map***
      - koja preuzima podatke iz distribuiranog sistema datoteka
    - tri mogućnosti izvršavanja zadataka
      - **lokalno izvršavanje zadataka** (*eng. data-local task execution*)
        - izvršavanje na čvoru na kojem se nalaze podaci
      - **izvršavanje unutar *racka*** (*eng. rack-local task execution*)
        - izvršavanje na čvoru koji se fizički nalazi unutar smeštajne jedinice za računare (*rack*-a)
      - **globalno izvršavanje** (*eng. off-rack task execution*)
        - izvršavanje na čvoru koji se nalazi unutar klastera ali nije u okviru istog *rack*-a

MapReduce

- MapReduce - osnovni pojmovi
  - **lokalizacija obrade podataka**
    - ne odnosi se na obradu u zadatku *reduce*
      - pošto preuzima podatke koji su rezultat zadatka *map,* a koji su smešteni lokalno u čvorovima u kojima se taj zadatak i izvršava
        - ne smeštaju se u distribuirani sistem datoteka
      - pošto se podaci preuzimaju od više zadataka *map*
        - koji se ne moraju izvršavati na istom čvoru
  - **rezultati obrade podataka u okviru zadatka *reduce***
    - obično se smeštaju u distribuirani sistem datoteka
      - radi obezbeđenja pouzdanosti i redundanse podataka

MapReduce

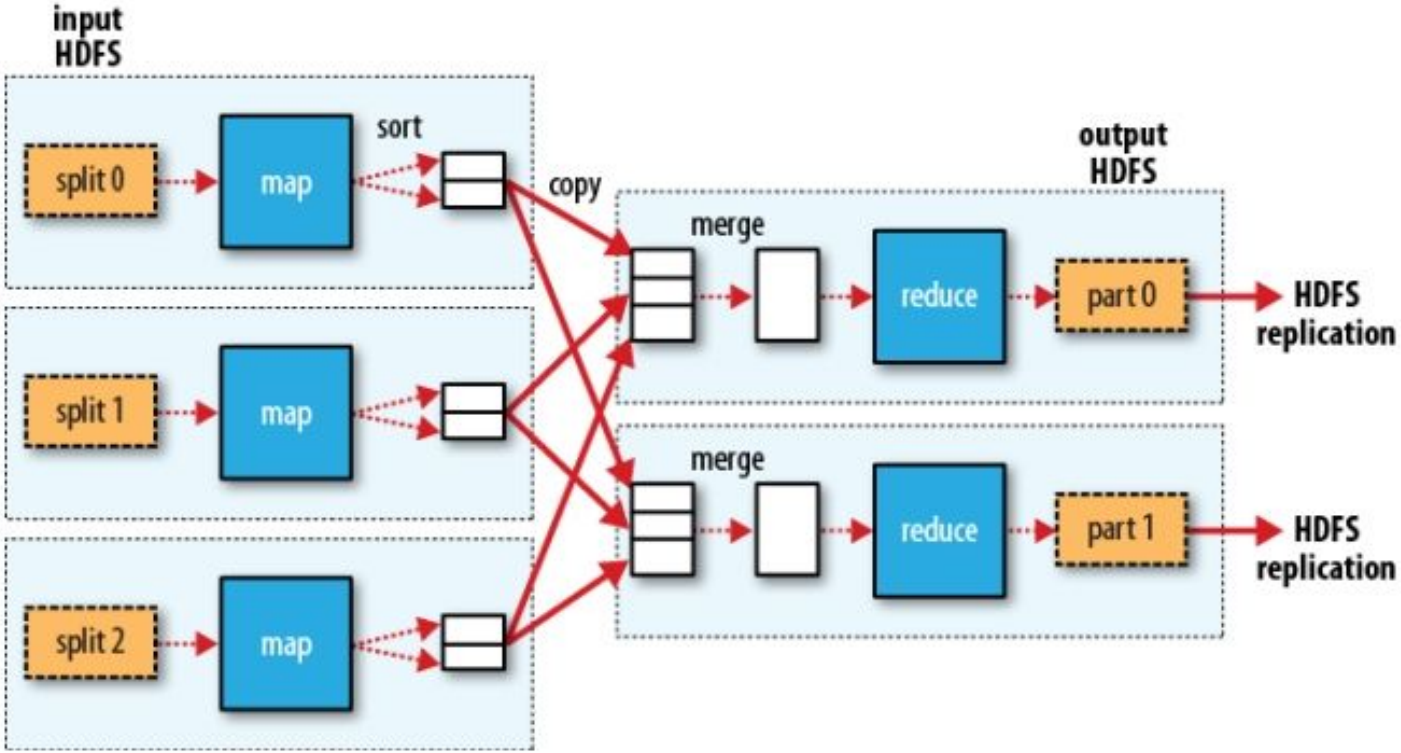


izvor: *Hadoop - The Definitive Guide, Tom White*

MapReduce

- MapReduce - osnovni pojmovi
  - broj zadataka *reduce* nije uslovljen veličinom ulaznog skupa podataka
    - već predstavlja parametar koji se posebno može podešavati
  - **particije** (*eng. partitions*)
    - skupovi podataka koji predstavljaju ulazne podatke za zadatak *reduce*
      - sačinjeni od izlaznih podataka iz zadatka *map*
    - može sadržavati više ključeva
      - ali svi slogovi koji pripadaju istom ključu moraju se naći u jednoj particiji
    - uobičajno se koristi ugrađena funkcija za particionisanje
      - zasnovana na izračunavanju *hash* funkcije nad ključem
      - moguće implementirati i koristiti korisnički definisanu funkciju za particionisanje
  - **raspodela podataka** (*eng. shuffle*)
    - obuhvata razmenu podataka između čvorova sa zadatkom *map* i zadatkom *reduce*
    - može biti izuzetno kompleksna i imati veliki uticaj na performanse celokupnog sistema

MapReduce



izvor: *Hadoop - The Definitive Guide, Tom White*

MapReduce

- Za pisanje map-reduce programa moguće je koristiti sledeće programske jezike
  - *Java* (izvorno)
  - *Scala*
  - *Python*
- Prilikom izvršavanja map-reduce programa pokreće je *Job* koji sadrži
  - *Map* task
  - *Reduce* task

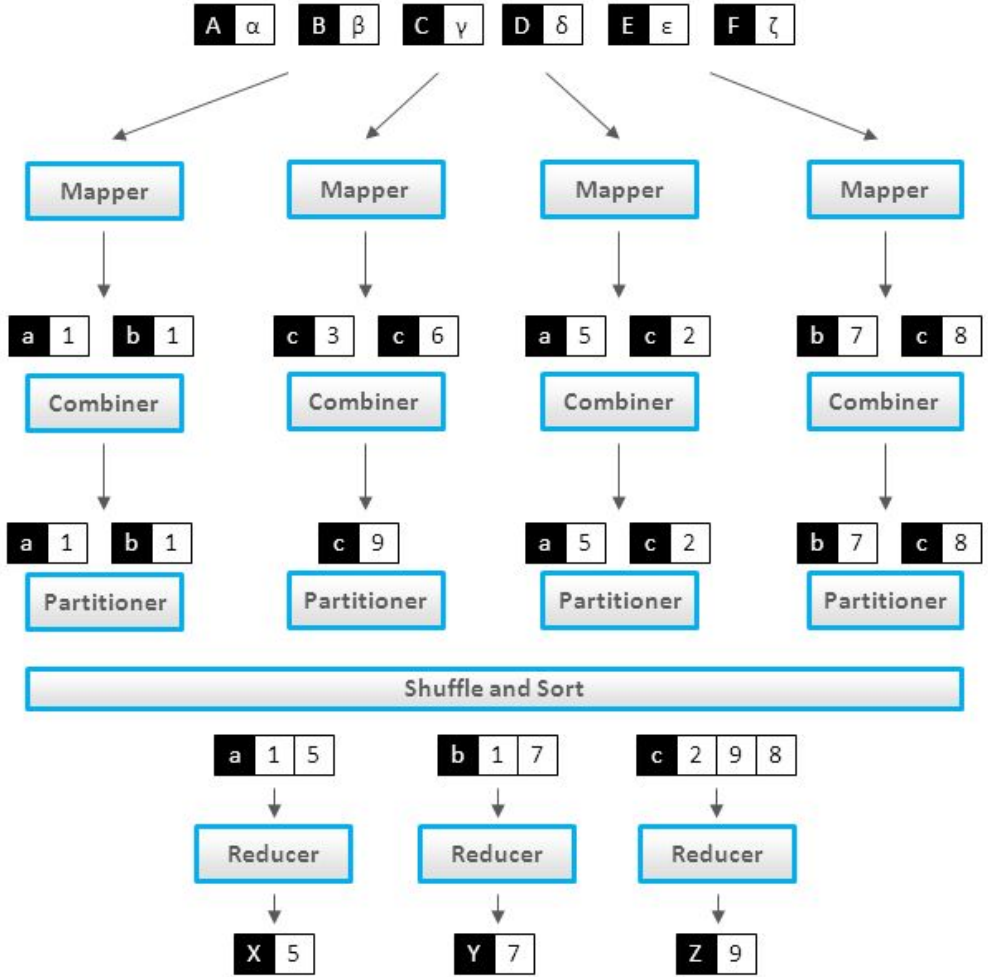
YARN

- ***Y**et **A**nother **R**esource **N**egotiator*
- Upravljač poslovima, zadacima, resursima
- Prisutan od Hadoop 2.0
- Rešio probleme sa *JobTracker*-om
- Master/slave arhitektura
- Podrazumevana konfiguracija
  - https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-common/yarn-default.xml

Raspodela i sortiranje podataka

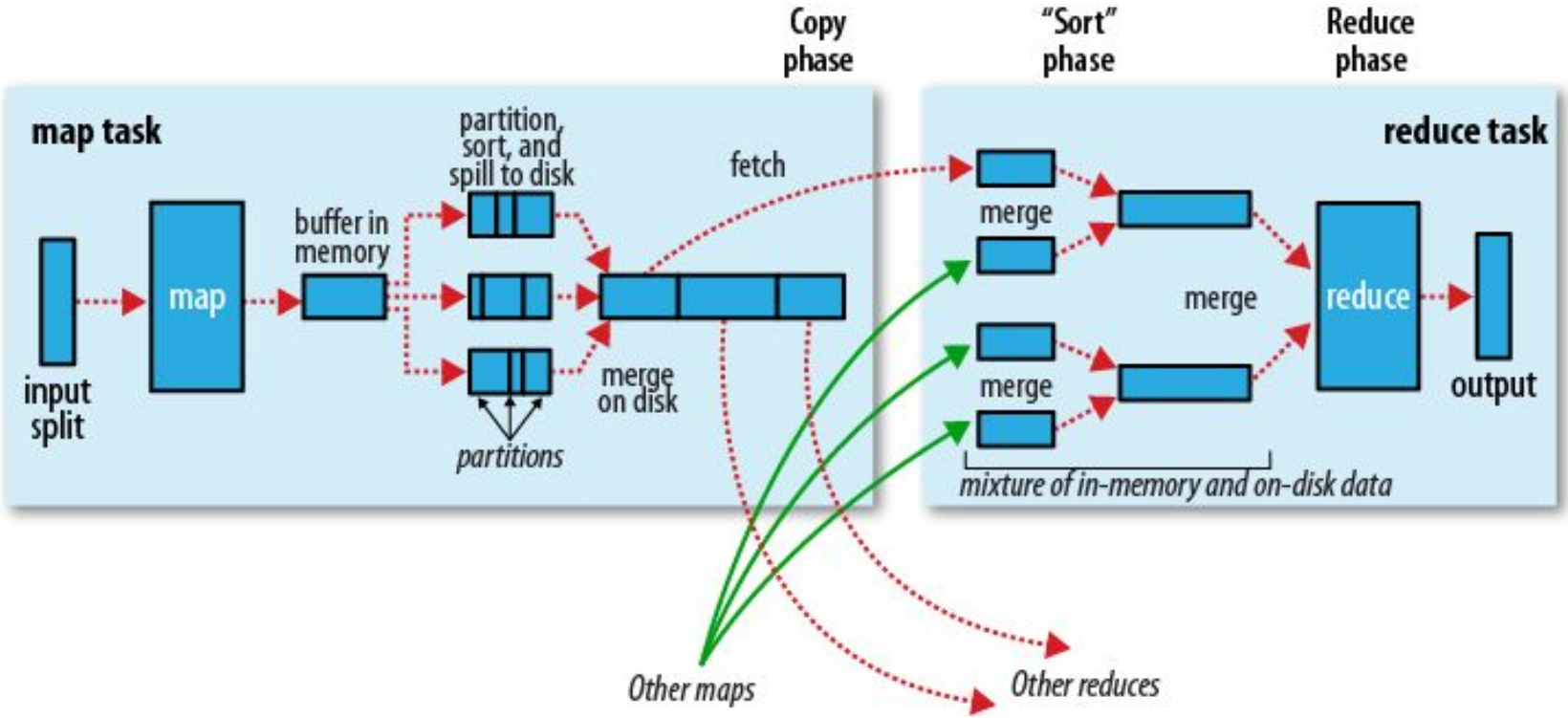
- Raspodela (*eng. shuffle*) i sortiranje (*eng. sort*) podataka
  - *MapReduce* okvir garantuje **da će ulaz u svaki *reducer* biti sortiran po ključu**
  - proces u okviru kojeg je implementirano sortiranje vrednosti koje predstavljaju izlaz iz koraka *map* i raspoređivanje tih vrednosti odgovarajućim koracima reduce**, naziva se raspodela podataka**
    - definisan radnim okvirom, a ne od strane korisnika
  - optimizacija *MapReduce* programa
    - da se što bolje iskoristi algoritam raspodele podataka

Raspodela i sortiranje podataka



izvor: *Mapreduce patterns, algorithms, and use cases, Ilya Katsov, https://highlyscalable.wordpress.com/2012/02/01/mapreduce-patterns/*

Raspodela podataka



izvor: *Hadoop - The Definitive Guide, Tom White*

Režimi pokretanja Hadoop-a

- *Standalone*
  - Lokalni fajl sistem (ne koristi HDFS)
- Pseudo-distribuirani
  - *HDFS* klaster na jednoj mašini (jedan čvor)
- Potpuno distribuirani
  - *HDFS* klaster na više mašina (više čvorova)

Distribuirani režim

- Emulacija distribuiranog ponašanja pomoću docker kontejnera
- Master/slave
  - Namenode - jedan kontejner
  - Datanode - svaki po jedan kontejner
  - Slično i za YARN master/slave
- BigData Europe
  - https://github.com/big-data-europe/docker-hadoop

Hadoop klaster

- Primer sadrži
  - HDFS sa jednim namenodom i 2 datanoda
  - YARN sa jednim master i jednim slave čvorom
- Sva podešavanja Hadoop klaster se vrše u **hadoop.env**
  - Prefiks ENV varijable označava kojem konfiguracionom fajlu pripada podešavanje

MapReduce - primeri

- Kreiranje foldera za pokretanje primera (u okviru YARN master kontejnera):
  - mkdir examples
  - cd examples
- Podešavanje JAVA classpath (u terminalu) kako bi pri kompajliranju *hadoop* biblioteke bile vidljive:
  - export CLASSPATH="$HADOOP_PREFIX/share/hadoop/mapreduce/hadoop-mapreduce-client-core-$HADOOP_VERSION.jar:$HADOOP_PREFIX/share/hadoop/mapreduce/hadoop-mapreduce-client-common-$HADOOP_VERSION.jar:$HADOOP_PREFIX/share/hadoop/common/hadoop-common-$HADOOP_VERSION.jar:~/examples/*:$HADOOP_PREFIX/lib/*"

MapReduce - primer 1

- U datoteci *sample.txt* dati su podaci koji o potrošnji električne energije jedne organizacije - datoteka sadrži mesečnu potrošnju kao i prosečnu potrošnju za celu godinu:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Avg
1979	23	23	2	43	24	25	26	26	26	26	25	26	25
1980	26	27	28	28	28	30	31	31	31	30	30	30	29
1981	31	32	32	32	33	34	35	36	36	34	34	34	34
1984	39	38	39	39	39	41	42	43	40	39	38	38	40
1985	38	39	39	39	39	41	41	41	00	40	39	39	45

MapReduce - primer 1

- U datoteci *ProcessUnits.java* dat je kod za *MapReduce* algoritam pomoću kog se vrši traženje svih godišnjih prosečnih vrednosti većih od 30.
  - MapReduceBase klasa
  - Mapper interfejs
  - Reducer interfejs
  - OutputCollector interfejs
  - Reporter interfejs
- Komplajliranje
  - javac -d . ProcessUnits.java
- Pakovanje u JAR
  - jar cfm ProcessUnits.jar Manifest.txt *.class

MapReduce - primer 1

- Kopirati sample.txt na HDFS
  - hdfs dfs -mkdir /units
  - hdfs dfs -put sample.txt /units/
  - hdfs dfs -ls /units
- Pokretanje
  - hadoop jar ProcessUnits.jar ProcessUnits /units /units_out
- Prikaz rezultata
  - hdfs dfs -cat /units_out/*
    - 1981 34
      1984 40
      1985 45
- Na kraju pobrisati foldere (kako bi se primer 1 mogao ponovo pokrenuti)
  - hdfs dfs -rm -r /units*
- Izazov - izračunavanje maksimuma za godinu

MapReduce - primer 2

- U datoteci *SalesJan2009.csv* dati su podaci o prodajama proizvoda:

Transaction_date	Product	Price	Payment_Type	Name	City	State	Country	Account_Created	Last_Login	Latitude	Longitude
1/2/09 6:17	Product1	1200	Mastercard	carolina	Basildon	England	United Kingdom	1/2/09 6:00	1/2/09 6:08	51.5	-1.1166667
1/2/09 4:53	Product1	1200	Visa	Betina	Parkville	MO	United States	1/2/09 4:42	1/2/09 7:49	39.195	-94.68194
1/2/09 13:08	Product1	1200	Mastercard	Federica e Andrea	Astoria	OR	United States	1/1/09 16:21	1/3/09 12:32	46.18806	-123.83
1/3/09 14:44	Product1	1200	Visa	Gouya	Echuca	Victoria	Australia	9/25/05 21:13	1/3/09 14:22	-36.1333333	144.75
1/4/09 12:56	Product2	3600	Visa	Gerd W	Cahaba Heights	AL	United States	11/15/08 15:47	1/4/09 12:45	33.52056	-86.8025
1/4/09 13:19	Product1	1200	Visa	LAURENCE	Mickleton	NJ	United States	9/24/08 15:19	1/4/09 13:04	39.79	-75.23806
1/2/09 20:09	Product1	1200	Mastercard	adam	Martin	TN	United States	1/2/09 17:43	1/4/09 20:01	36.34333	-88.85028
1/3/09 10:11	Product2	3600	Visa	Christiane	Delray Beach	FL	United States	1/3/09 9:27	1/10/09 9:46	26.46111	-80.07306

MapReduce - primer 2

- Zadatak: odrediti broj prodatih artikala po zemljama
- Java klase
  - SalesMapper.java
  - SalesCountryReducer.java
  - SalesCountryDriver.java
- Komplajliranje
  - javac -d . SalesMapper.java SalesCountryReducer.java SalesCountryDriver.java
- Pakovanje u JAR
  - jar cfm ProductSalePerCountry.jar Manifest.txt sales_country/*.class

MapReduce - primer 2

- Kopiranje podataka na HDFS
  - hdfs dfs -mkdir /sales
  - hdfs dfs -put SalesJan2009.csv /sales/
  - hdfs dfs -ls /sales/
- Pokretanje
  - hadoop jar ProductSalePerCountry.jar sales_country.SalesCountryDriver /sales /sales_out
- Prikaz rezultata
  - hdfs dfs -cat /sales_out/*

MapReduce - primer 3

- *wordcount* primer koristeći *Hadoop streaming*
- *Hadoop streaming* podrazumeva specifikaciju *MapReduce* programa pomoću *python* skripti
  - mapper.py
  - reducer.py
- Na početku *.py* fajlova obavezno
  - #!/usr/bin/python
  - Proveriti da li postoji *python* instaliran na YARN čvorovima
- Potrebno je omogućiti izvršivost skripti
  - chmod +x *.py
- Pre pokretanja odraditi kopiranje fajlova kao u *wordcount* primeru
- Pokretanje
  - mapred streaming -input /book -output /book_out -mapper mapper.py -reducer reducer.py
- hdfs dfs -cat /book_out/*

MapReduce - primer 4

- Podaci kao u primeru 2 - *SalesJan2009.csv*
- Zadatak: izračunati ukupan iznos po vrsti platne kartice
- mapper.py i reducer.py iskopirati u YARN master kontejner
- Pokrenuti pomoću *mapred streaming*

MapReduce - primer 5

- Podaci
  - Countries.dat - kod države, naziv države
  - Customers.dat - ime korisnika, tip korisnika, kod države
- Zadatak: odrediti broj korisnika određenog tipa po državi
- Šeme podataka ulaznih datoteka se razlikuju po strukturi (broju kolona)
- Zadatak zahteva spajanje (*join*) podataka
- Ključ kod reduce faze je (kod države, tip korisnika)

MapReduce - primer 6

- Relaciona algebra
  - Selekcija
    - *Mapper* emituje torke koje ispunjavaju uslov selekcije, *reducer* trivijalan
  - Projekcija
    - *Mapper* emituje projekciju torke, *reducer* trivijalan
  - Unija
    - *Mapper* trivijalan, *reducer* eliminiše duplikate
  - Presek
    - *Mapper* trivijalan, *reducer* emituje samo torke čiji se ključevi pojavljuju 2 puta
  - Razlika
    - *Mapper* emituje torku i njenu pripadnost, *reducer* emituje torke koje se pojavljuju samo jednom i to u prvoj relaciji
- Spajanje
  - Različite tehnike spajanja pomoću map-reduce
  - Više o ovoj temi: ovde

MapReduce - konfiguracija

- Definisanje konfiguracije u mapred-site.xml
- Predefinisana konfiguracija
  - https://hadoop.apache.org/docs/r2.7.2/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml
- Neki parametri
  - mapred.job.tracker
  - mapred.system.dir
  - mapred.local.dir
  - mapred.tasktracker.{map|reduce}.tasks.maximum

MapReduce - zadatak 1

- Srednja vrednost
- U zadatoj datoteci *sample.csv* druga kolona sadrži merenja neke posmatrane veličine
- Napisati map-reduce program kojim se određuje srednja vrednost tih merenja

$$\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

MapReduce - zadatak 2

- Varijansa
- U zadatoj datoteci *sample.csv* druga kolona sadrži merenja neke posmatrane veličine
- Napisati map-reduce program kojim se određuje srednja vrednost tih merenja

$$\mathrm{Var}(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

MapReduce - zadatak 3

- *Finding Friends*
- Za svakog korisnika društvene mreže poznat je skup njegovih prijatelje
  - Npr. A:B,C,D
- Napisati map-reduce program kojim se za svaki par prijatelja određuju njihovi zajednički prijatelji

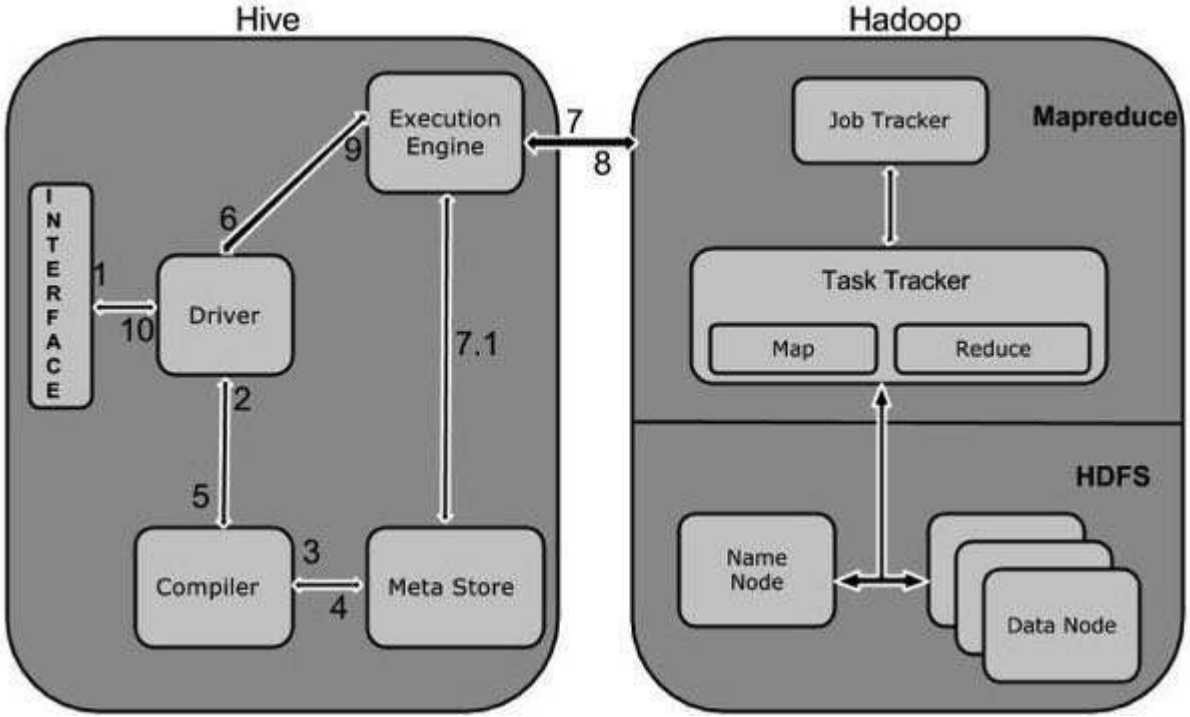
MapReduce - zadatak 4

- *Cross-Correlation*
- Napisati map-reduce program kojim se određuje koliko puta se svaki par reči u nekom tekstu pojavljuje zajedno u rečenici (liniji)
- Izvor besplatnih knjiga u tekstualnom formatu: http://www.gutenberg.org/catalog/

Apache Hive

- Deo Hadoop ekosistema
- Alat za procesiranje strukturiranih podataka nad Hadoop-om
- Nudi apstrakciju *map-reduce* operacija kroz kreiranje SQL-like skripti
- Ima shemu i namenjen je OLAP-u
- Korisnički zadate SQL naredbe transformiše u niz odgovarajućih *map-reduce* zadataka, skrivajući od korisnika konkretnu implementaciju

Hive - arhitektura



Hive - konfiguracija

- Definisanje konfiguracije u *hive-site.xml*
- Podrazumevana konfiguracija Hive servera
  - https://github.com/apache/hive/blob/master/data/conf/hive-site.xml
- Neki parametri
  - hadoop.tmp.dir
  - javax.jdo.option.ConnectionURL
  - hive.metastore.metadb.dir
  - hive.exec.mode.local.auto

Hive - pokretanje

- Kreirati direktorijume na HDFS za Hive metastore
  - hdfs dfs -mkdir /tmp
  - hdfs dfs -mkdir /user/hive/warehouse
  - hdfs dfs -chmod g+w /tmp
  - hdfs dfs -chmod g+w /user/hive/warehouse
- Pokretanje Hive CLI - hive (deprecated)
- Hive schema tool
  - $HIVE_HOME/schematool -initSchema -dbType <db_type>
- Pokretanje Hive servera
  - hiveserver2
- Hive klijent
  - beeline -u jdbc:hive2://$HS2_HOST:$HS2_PORT
  - Standalone: beeline -u jdbc:hive2://

Hive - osnove

- Apstrakcija nad HDFS datotekama
- Kreira se tabela sa datotekom kao sadržajem u pozadini
- Hive SQL umnogome podseća na standardni SQL RDBMS-ova
- Hive server parsira SQL komande i on njih kreira map-reduce poslove i zadatke
  - Koje posle izvršava nad HDFS datotekom
- Hive *local-mode*
  - `hive> SET mapreduce.framework.name=local;`
  - Korisno za male datoteke
  - Ako su ulazni podaci manji od granice (128MB) i manje od 4 zadataka
    - pokreće se *local mode*, odnosno jedan *reducer*

Hive - DDL

- Jezik za upravljanje shemom baze podataka
  - CREATE DATABASE/SCHEMA, TABLE, VIEW, FUNCTION, INDEX
  - DROP DATABASE/SCHEMA, TABLE, VIEW, INDEX
  - TRUNCATE TABLE
  - ALTER DATABASE/SCHEMA, TABLE, VIEW
  - MSCK REPAIR TABLE (or ALTER TABLE RECOVER PARTITIONS)
  - SHOW DATABASES/SCHEMAS, TABLES, TBLPROPERTIES, VIEWS, PARTITIONS, FUNCTIONS, INDEX[ES], COLUMNS, CREATE TABLE
  - DESCRIBE DATABASE/SCHEMA, table_name, view_name
- Primeri
  - CREATE TABLE pokes (foo INT, bar STRING);
  - SHOW TABLES '.*s';

Hive - DML

- Jezik za manipulaciju sadržajem tabela
  - LOAD
  - INSERT
    - into Hive tables from queries
    - into directories from queries
    - into Hive tables from SQL
  - UPDATE
  - DELETE
  - MERGE

Hive - QL

- **Select** naredba

```
[WITH CommonTableExpression (, CommonTableExpression)*]
SELECT [ALL | DISTINCT] select_expr, select_expr, ...
  FROM table_reference
  [WHERE where_condition]
  [GROUP BY col_list]
  [ORDER BY col_list]
  [CLUSTER BY col_list
    | [DISTRIBUTE BY col_list] [SORT BY col_list]
  ]
 [LIMIT [offset,] rows]
```

Hive - Primer 1

- Zakačiti se na *hive-server* docker kontejner
  - `docker exec -it hive-server bash`
- Pokrenuti hive klijent - *beeline*
  - `beeline -u jdbc:hive2://localhost:10000`
- Kreirati tabelu pokes
  - `CREATE TABLE pokes (foo INT, bar STRING);`
- Uvesti podatke u tabelu iz lokalnog fajla
  - `LOAD DATA LOCAL INPATH '/opt/hive/examples/files/kv1.txt' OVERWRITE INTO TABLE pokes;`
- Pokrenuti neki upit
  - `SELECT count(*) FROM pokes;`

Hive - Primer 2

- `CREATE TABLE invites (foo INT, bar STRING) PARTITIONED BY (ds STRING);`
- `DESCRIBE invites;`
- `LOAD DATA LOCAL INPATH '/opt/hive/examples/files/kv2.txt' OVERWRITE INTO TABLE invites PARTITION (ds='2008-08-15');`
- `LOAD DATA LOCAL INPATH '/opt/hive/examples/files/kv3.txt' OVERWRITE INTO TABLE invites PARTITION (ds='2008-08-08');`
- `SELECT a.foo FROM invites a WHERE a.ds='2008-08-15';`
- `INSERT OVERWRITE DIRECTORY '/tmp/hdfs_out' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT a.* FROM invites a WHERE a.ds='2008-08-15';`

Hive - Zadatak 1

- Kreirati tabelu *Emp* sa kolonama: *id*, *name*, *age*, *gender*
- Ručno napraviti testnu *data.csv* datoteku sa zadatim kolonama i bar 5 redova
- Datoteku *data.csv* kopirati na HDFS
- Učitati sadržaj datoteke *data.csv* u tabelu *Emp*
- Pomoću HQL upita, uveriti se da su podaci učitani
- Zaposlene starije od 30 godina smestiti u posebnu CSV datoteku sa delimiterom (;) na proizvoljnu HDFS lokaciju

Hive - Zadatak 2

- Preuzeti ovaj CSV fajl
- Kopirati ga na HDFS
- Kreirati tabelu *Zipcode*
- Učitati podatke iz datoteke u tabelu
- Prikazati sve podatke za kodove sa Floride
- Rezultate smestiti u novu tabelu *ZipcodeFL* (CTAS)

Primer - Movie ratings

- Skup podataka MovieLens 100k sadrži 100 hiljada stavki sa korisničkim ocenama filmova
- Kreira se tabela čiji je sadržaj biti skladišten na HDFS u tekstualnom fajlu
- U okviru Hive SQL naredbi moguće je uključiti i izvršavanje *python* skripti koje implementiraju neku kompleksniju obradu podataka
- Detaljnije o ovom primeru: ovde