Software Distribuït - T10 - Sistemes distribuïts d'alta capacitat

Eloi Puertas i Prats

Universitat de Barcelona
Grau en Enginyeria Informàtica

30 de maig de 2018

SPARK: Motivació

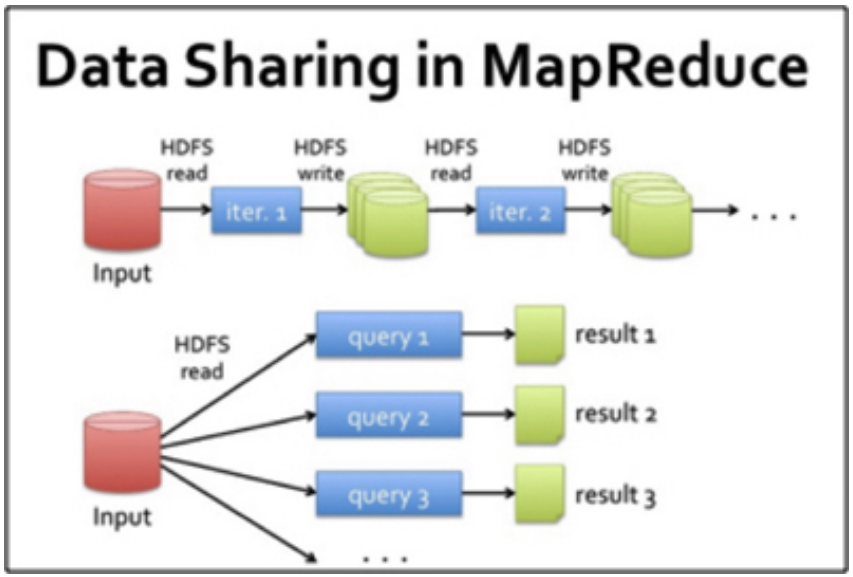
- Dissenyar un model de programació distribuïda que suporti una gamma més àmplia d'aplicacions que MapReduce, mantenint el seu sistema automàtic de tolerància a falles.
- MapReduce és ineficient per a programes de múltiples passades que no requereixen de massa intercanvi de dades entre múltples operacions paral·leles.
- Aquest tipus de programes són força comuns en tasques d'anàlisi de dades massives (BigDATA):
  - Algorismes Iteratius, incloent la majoria d'alogorismes d'Aprenentatge Automàtic (Machine Learning) i algorismes de grafs.
  - Mineria de Dades interactiva, on un usuari carrega dades en memòria mitjançant un cluster i la consulta repetidament.
  - Aplicacions d'Streaming que mantenen un estat agregat al llarg del temps.

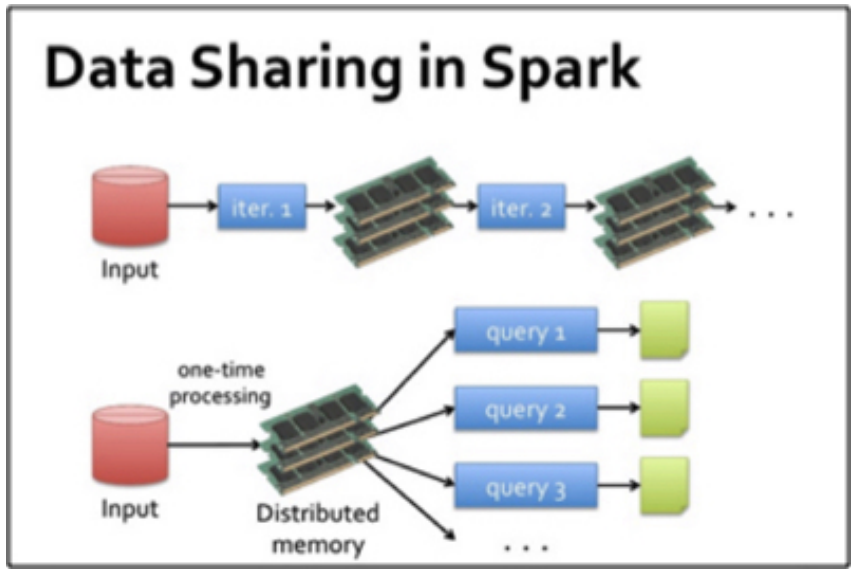
Característiques

- Mapreduce es basa en fer córrer una sèrie de diferents tasques on cada una d'elles llegeix d'un disc físic i torna a escriure en ell.
  - El fet de llegir i escriure les dades de forma replicada a cada pas té un cost significatiu.
- Spark ofereix una abstracció anomenada *Resilent Distributed Datasets* (RDD).
  - RDD es guarda en memòria entre les diferents peticions sense necessitar de replicació.
  - En cas de fallada el que fa és *reconstruir* les dades perdudes fent servir *traçabilitat*.
  - Cada RDD *recorda* com va ser construït a partir dels altres datasets (a partir de maps, join, groupby) per tal de reconstruir-se.
- Està implementat usant SCALA. A més a més, de base, té APIs per python i Java

UNIVERSITAT DE BARCELONA

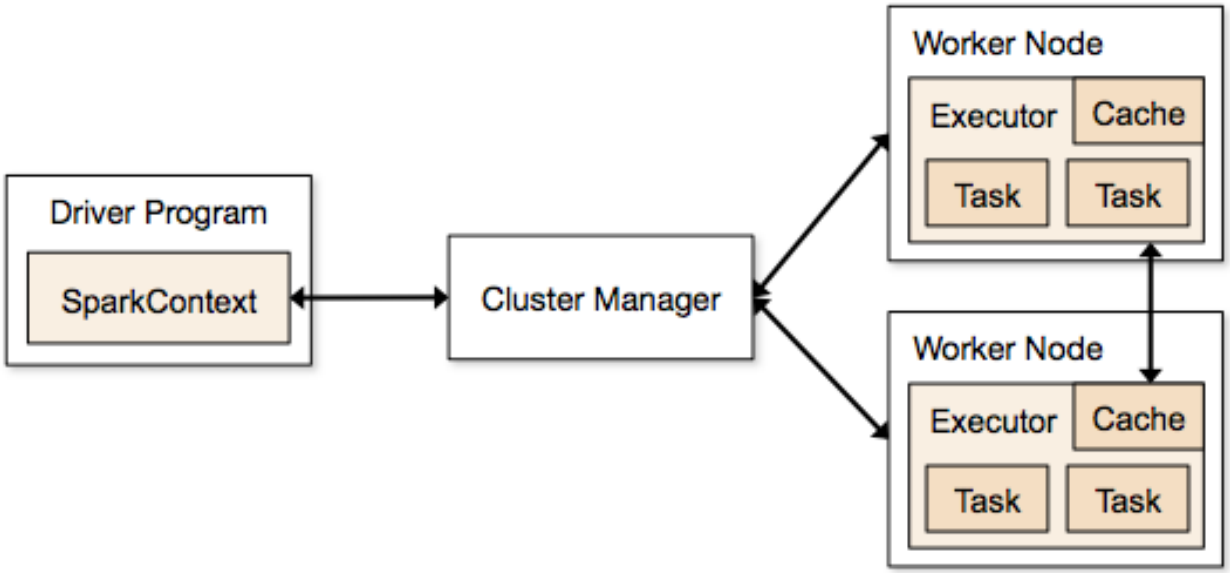
SPARK VS Hadoop





SPARK: Driver i Workers

- Un codi Spark fa córrer dos tipus de programes diferents:
  - Un driver i varis workers.
- El driver s'executa en un node passarela.
- Els workers s'executen en nodes del cluster o en threds locals
- RDDs es distribueixen al llarg dels workers



SPARK: Context

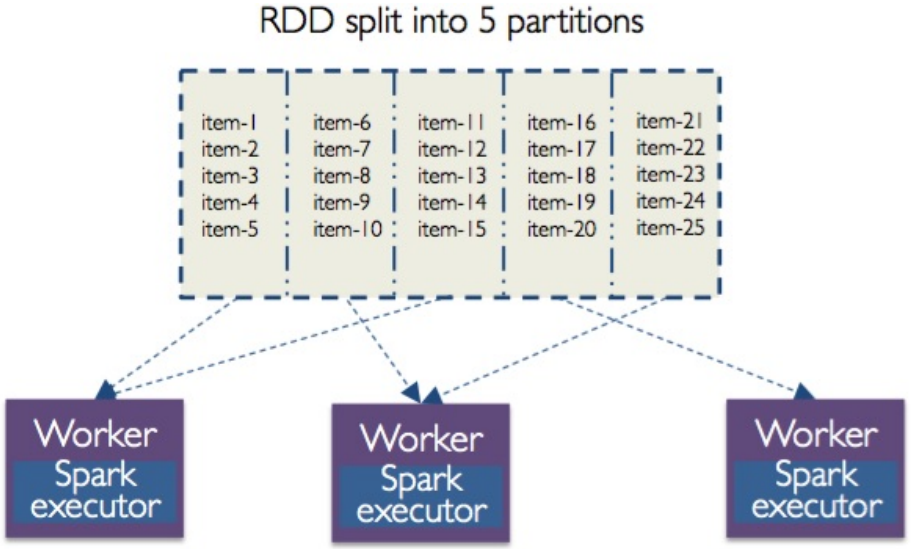
- Un programa Spark abans de res crea un objecte SparkContext:
  - Li diu a Spark com i a on accedir al cluster de nodes.
- S'usa la variable sc (SparkContext) per a crar els RDDs.
- El paràmetre *master* del SparkContext determina quin tipus i mida de cluster s'està usant.

Resilient Distributed Datasets (RDD) (I)

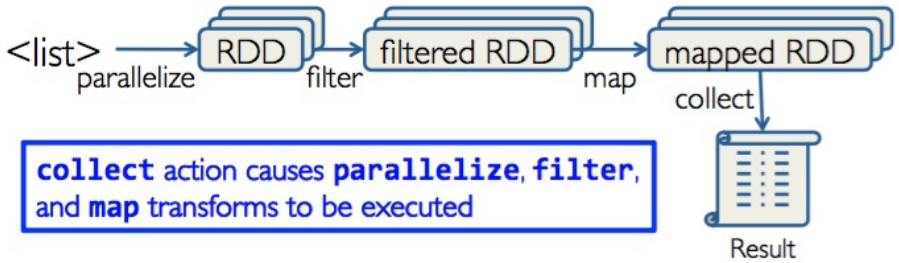
- És la principal abstracció en Spark
  - És immutable un cop construïda.
  - Traça tota l'activitat que se'n fa per tal de recomputar eficientment la possible pèrdua de dades.
  - Permet operacions en paral·lel en col·leccions d'elements.
- Les RDDs es poden construir a partir de:
  - Paral·lelitzar una col·lecció que ja existeixi.
  - Transformant una RDD que ja existeixi.
  - A partir de fitxers en un sistems distribuït com HDFS o qualsevol altre.

Resilient Distributed Datasets (RDD) (II)

- El programador especifica el nombre de particions que pot tenir una RDD.
  - En general: Més particions = Més paral·lelisme (Cal tenir en compte però el nombre de nodes vs threads disponibles)



- Es permeten dos tipus d'operacions: Transformacions i Accions



Resilient Distributed Datasets (RDD): Transformacions

- Crear nous datasets a partir d'un altre que ja existeixi.
- Fa servir avaluació mandrosa: els resultats no es computen de seguida, sino que es guarda el conjunt de transformacions que s'han d'aplicar al dataset per tal de:
  - Optimitzar els càlculs necessaris.
  - Recuperar-se de caigudes i de workers lents.
- Penseu en les Transformacions com a receptes per a obtenir un resultat.

Transformation	Description
`map(`*func*`)`	return a new distributed dataset formed by passing each element of the source through a function *func*
`filter(`*func*`)`	return a new dataset formed by selecting those elements of the source on which *func* returns true
`distinct([`*numTasks*`]))`	return a new dataset that contains the distinct elements of the source dataset
`flatMap(`*func*`)`	similar to map, but each input item can be mapped to 0 or more output items (so *func* should return a Seq rather than a single item)

UNIVERSITAT DE BARCELONA

Resilient Distributed Datasets (RDD): Accions

- Fa que Spark executi tot el conjunt de transformacions a un conjunt de dades.
- És la forma per tal d'enviar resultats fora de l'Spartk.

Action	Description
`reduce(func)`	aggregate dataset's elements using function *func*. *func* takes two arguments and returns one, and is commutative and associative so that it can be computed correctly in parallel
`take(n)`	return an array with the first *n* elements
`collect()`	return all the elements as an array WARNING: make sure will fit in driver program
`takeOrdered(n, key=func)`	return n elements ordered in ascending order or as specified by the optional key function

Resilient Distributed Datasets (RDD): Caching

- Es pot fer que un RDD sigui persistent usant els mètodes `persist()` o `cache()` sobre ell.
- La cache és tolerant a falles. Si qualsevol partició d'una RDD es perd, automàticament és recalculada usant la cadena de transformacions que la va crear originalment.

Més informació sobre RDD

UNIVERSITAT DE BARCELONA

Cicle de vida d'un programa Spark

1. Crear RDDs a partir de dades externes o paral·lelitzar una col·lecció en el programa driver.
2. Fer transformacions mandroses sobre les dades produint nous RDDs.
3. Marcar els RDDs com a caché per a reutilitzar les dades.
4. Realitzar accions que executaran computacions en paral·el i acabaran per a produir els resultats finals.

Variables Compartides

- Spark crea automàticament tancaments (closures) (Funcions amb els seus propis entorns de variables) quan són :
  - Funcions que corren en RDDs en els workers (ex. funcions map)
  - Qualsevol variable global utilitzada per aquests workers
  - No es requereix comunicació entre workers.
- Problemes:
  - Els canvis en variables globals als workers no s'envien als workers.
  - Per exemple, si vols comptar certs elements en els workers i obtenir el nombre total dels elements a tots els workers.

Variables Compartides

- Solució:
  - Variables de Redifusió (Broadcasting)
    - Serveixen per enviar eficientment grans quantitats de dades de només lectura des del driver a tots els workers.
    - Es guarda en els workers per usar-se en una o més operacions.
  - Variables Acumuladores (tipus: integers, double, long, float)
    - Poden agregar valors des dels workers i enviar-se al driver.
    - Només el driver té accés al valor de l'acumulador.
    - És a dir, pels workers, l'acumulador només és d'escriptura.
    - Els acumuladors poden ser usats tant en accions com en transformacions, però recordeu que només les accions garanteixen la seva execució.

Comptar paraules

Exemple 1: WordCount.py

Exemple 2: findWords.py