Sistemes distribuïts d'alta capacitat

: Dissenyat per a Google, per a poder donar resposta a l'alta demanda de requirements del seu motor de cerca.

: Dissenyat per a poder donar resposta a l'anàlisis de dades en entorns de BigData.

HADOOP

- La programació de sistemes distribuïts tradicionals és complexe:
  - L'intercanvi de dades necessita de sincronització
  - Cal tenir en compte les restriccions en l'amplada de banda
  - Les dependències temporals són complicades de gestionar
  - Es fa difícil gestionar les fallades parcials del sistema.

L'emmagatzemament de dades

- Típicament les dades en un sistema distribuït es guarden en una unitat de dades en xarxa.
- En temps de computació, les dades són copiades als nodes de computació.
- Això funciona bé fins a una quantitat relativament limitada de dades.

Problema: Els sistemes moderns treballen amb una quantitat de dades cada dia més gran (**BIG DATA!!**)

- Com proporcionar les dades als processadors ha esdevingut l'autèntic coll d'ampolla dels sistemes distribuïts actuals.
- Cal una nova aproximació a la distribució del sistema.

Requeriments de la nova aproximació (I)

- Robustesa a fallades parcials del sistema.
  - No es pot permetre que el sistema caigui completament degut a fallades parcials.
- Recuperació de les dades.
  - Si un component del sistema falla, la seva càrrega l'ha d'assumir una altra unitat del sistema
  - Mai s'ha de permetre la pèrdua de dades com a conseqüència d'una fallada.
- Recuperació d'un component
  - si un component falla i després es recupera, ha de ser capaç d'ajuntar-se de nou al sistema, sense que sigui necessari re-engegar el sistema sencer.

Requeriments de la nova aproximació (II)

- Consistència
  - La fallada de components del sistema durant l'execució d'un treball no ha d'afectar a la sortida d'aquest treball.
- Escalabilitat
  - El fet d'afegir càrrega al sistema no ha de provocar una fallada en el sistema, sinó que ha d'haver-hi un descens en el rendiment global dels treballs individuals.
  - El fet d'incrementar el nombre de recursos hauria de repercutir en un augment de la capacitat de càrrega del sistema.

HADOOP

- Hadoop es basa en el treball fet per Google als inicis dels 2000.
- Adopta una nova aproximació al problema de computació distribuïda, complint tots els requiriments desitjats de escalabilitat, robustesa, fiabilitat...
- Idea bàsica: Distribuir les dades tan bon punt siguin guardades en el sistema.
- No es necessita transferir dades per la xarxa per a començar a processar en cada node.

Conceptes Clau

- Les aplicacions són escrites a alt nivell, sense haver de preocupar-se per temes de baix nivell com la xarxa, dependències, sincronitzacions...
- Els nodes es comuniquen els uns amb els altres tan poc com sigui possible
- Les dades es reparteixen ”a priori”entre les màquines.
- La computació passa allà on les dades estiguin emmagatzemades, sempre que sigui possible.
- Les dades estan replicades vàries vegades en el sistema per augmentar la disponibilitat i fiabilitat.

Components del Hadoop

- Components bàsics:
  - Hadoop Distributed File System (HDFS).
  - MapReduce Software Framework.
- L'ecosistema del Hadoop, però conté d'altres components (no els veurem)
  - Pig, Hive, HBase, Flume, Oozie, Sqoop, Mahout...
- Un cluster Hadoop es basa en un conjunt de màquines fent anar HDFS i MapReduce.
- Cada màquina individual s'anomena node.
- Un cluster pot tenir des d'un sol node fins a milers. (Com més millor!)

Cinc cèntims de Hadoop

- Quan les dades es carreguen en el sistema, es divideixen en blocs (64/128MB)
- Les tasques de mapping treballen en porcions petites de dades (1 bloc)
- El programa Màster assigna la feina als nodes de tal forma que la tasca de mapping sempre treballi amb un bloc de dades que es trobi en el seu mateix node. Molts nodes treballaran en paral·lel, cadascú amb el seu bloc de dades.

Tractament de fallades.

- Si un node falla, el Màster detecta la fallada i reassigna la feina a un altre node del sistema.
- Tornar a engegar una tasca no implica comunicació amb els nodes que estiguin treballant amb altres porcions de dades
- Si un node caigut torna a engegar-se, s'afegeix automàticament al sistema i se li assignen noves tasques.
- Si un node no s'està executant amb un rendiment prou òptim, el màster pot executar una altra instància de la mateixa tasca. Els resultats del primer que acabi són els que s'usaran.

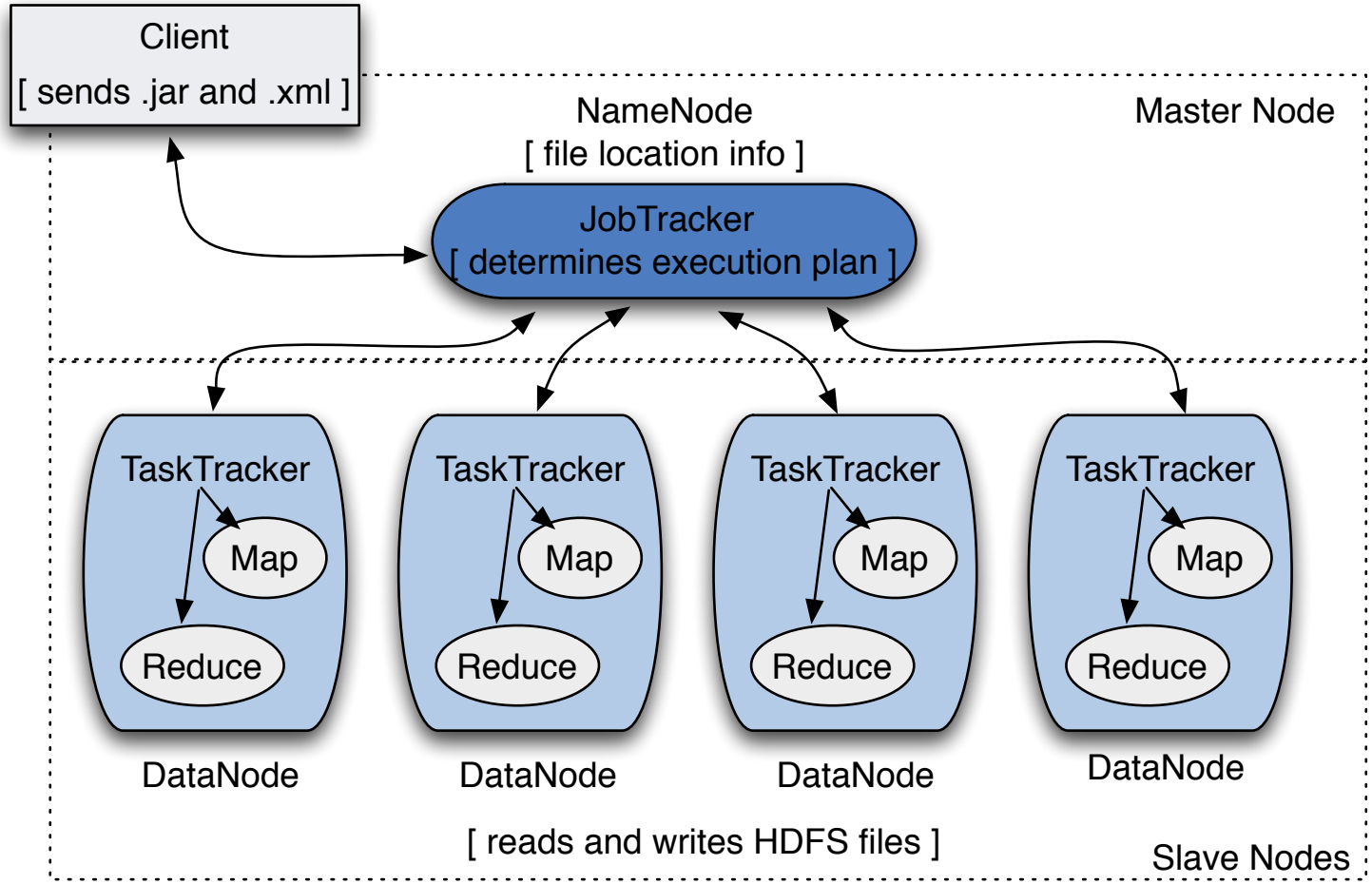
Instal·lació d'un Cluster Hadoop

- Normalment ho fa l'administrador del sistema
- La forma més senzilla és usant la distribució de Cloudera que inclou Apache Hadoop (CDH)

Serveis funcionant en el Cluster

- **NameNode:** Manté el Metadata del HDFS.
- **NameNode Secundari:** Fa funcions de manteniment pel NameNode. No és un backup.
- **DataNode:** Encarregat d'emmagatzemar blocs de dades HDFS.
- **JobTracker:** Gestiona treballs de MapReduce, distribueix tasques individuals...
- **TaskTracker:** Responsable d'instanciar i monitorar tasques de Map i Reduce individuals.

Anatomia del Hadoop Cluster



Com funciona el HDFS

- HDFS és un sistema de fitxers escrit en JAVA
- Es situa a sobre de qualsevol sistema de fitxers natiu.
- Ens dóna emmagatzemament redundant per a grans quantitats de dades, fent servir qualsevol tipus d'ordinador.
- HDFS funciona millor amb un nombre moderat de fitxers grans (Milions de fitxers de 100MB o més)
- HDFS està optimitzat per a fitxers llargs i lectures consecutives de dades (millor que accés aleatori).

Com es guarden els fitxer en HDFS

- Es divideixen en blocs.
- Les dades es distribueixen a diverses màquines en el moment en que es carreguen.
- Diferents blocs del mateix fitxer es guardaran a diferents màquines.
- Els blocs es repliquen en diverses màquines (DataNodes). Per defecte es guarda en 3 màquines
- El NameNode es l'encarregat de mantenir la informació sobre quins blocs formen un fitxer i en quines màquines es troben els blocs (Metadata)

Com posar i treure dades en el HDFS

- API de Hadoop (línia de comandes):
  - Per a treballar amb el HDFS: `hadoop fs`
  - Còpia de local a HDFS:
    `hadoop fs -copyFromLocal local_dir /hdfs_dir`
  - Còpia de HDFS a local:
    `hadoop fs -copyToLocal /hdfs_dir local_dir`

Exemple de funcionament complet

Metadata:

`/user/eloi/foo` → blocs: 1, 2, 4

`/user/eloi/bar` → blocs: 3, 5

Node	Blocs			
N1	1	2		
N2	1	5		
N3	3	5	2	
N4	1	4	5	3
N5	4	2		
N6	3	4		

- NameNode guarda la Metadata dels fitxers desats al HDFS
- DataNodes guarden els blocs. Cada bloc es replica tres cops.

Exemple de funcionament complet

Metadata:

`/user/eloi/foo` → blocs: 1, 2, 4

`/user/eloi/bar` → blocs: 3, 5

Node	Blocs			
N1	1	2		
N2	1	5		
N3	3	5	2	
N4	1	4	5	3
N5	4	2		
N6	3	4		

- Quan una aplicació client vol llegir un fitxer (bar p.ex):
- Comunica amb el NameNode per saber quins blocs formen el fitxer i en quin DataNode es troben
- Llavors es comunica directament amb els DataNodes (N2, N3) per a llegir les dades.

Com funciona el MapReduce

- Mètode per a distribuir tasques a diferents nodes
- Cada node processa dades guardades en aquell node (sempre que sigui possible)
- Consisteix en dues fases: Map i Reduce.

Map vs Reduce

A la funció Map, processarem les dades i realitzarem les operacions més costoses. A la funció Reduce, ajuntarem la sortida de la funció Map, realitzant operacions “trivials”.

Característiques del MapReduce

- Paral·lelització i distribució automàtica
- Tolerant a fallades.
- Eines per a la monitorització del procés.
- Ofereix abstracció per a programadors. (JAVA o Python).
- MapReduce amaga al desenvolupador totes les tasques que passen per sota. Només cal definir les funcions Map i Reduce.

Comptar paraules(I)

- Comptar les paraules d'un text és especialment complexe si tenim un gran volum de dades.
- Tasca seqüencial, però altament paral·lelitzable
- Si distribuïm les dades, podem fer-ho sense problemes
- Hadoop és una molt bona eina per a fer-ho

Comptar paraules (II)

1. Primer cal descomposar les dades (Hadoop ho fa automàticament)
2. Ara, enumerarem totes les paraules (sense sumar-les) al Map, processant la entrada línia a línia
3. Hadoop ordenarà la sortida per clau i la transmetrà al procés de Reduce
4. A la funció Reduce sumarem totes les dades amb una mateixa clau

Comptar paraules (III)

Codi Font: WordCount