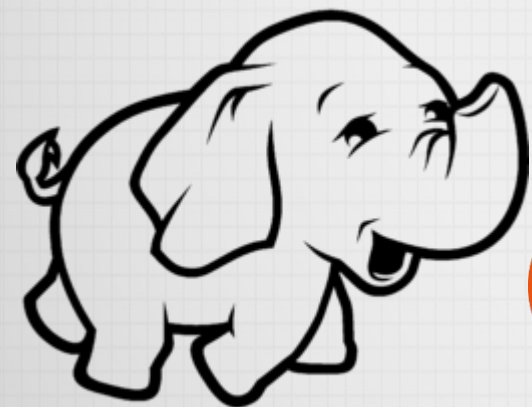


BIG DATA

GULYÁS MÁTÉ

SPARK, STORM



dmlab

2014.



APACHE SPARK

MIÉRT LASSÚ A HADOOP

1. Minden diszkre perzisztálódik
2. Sok esetben több iteráció kell, ami a növekvő overhead miatt lassú
3. Core MR API egyszerű, nehezen tesztelhető, könnyű hibázni
4. Sokat kell kódolni a legegyszerűbb taszkokhoz is
5. Boilerplate, boilerplate, boilerplate...

APACHE SPARK

Fast in-memory cluster computing framework. Batch and stream processing.

Gyors.

Nagyon Gyors. (100x gyorsabb egy atlagos Hadoop clusternél)

Twitternél kezdték fejleszteni.

DISTRIBUTED – MIN FUT?

LOCAL MODE

MESOS

YARN

EC2 FELETT IS FUTHAT (SCRIPT)

APACHE MESOS

KLASZTER ERŐFORRÁS KEZELŐ

Olyan mint a YARN. Elosztott rendszerek operációs rendszere.

- Zookeepert használ koordinálásra
- Linux container sandboxingra
- WebUI-on megtekinthető a klaszter állapota

RESILIENT DISTRIBUTED DATASETS (RDDS)

Központ fogalom Sparkban.

Elosztott, hibatűrő adathalmaz. Párhuzamosan feldolgozható.

Két módon hozhatóak létre:

- Létező, egyszerű kollekciók párhuzamosításából
`sc.parallelize(Collections)`
- Elosztott tárhelyen lévő forrás beolvasásával
P1: `sc.textFile("/path/to/my/file")`

Támogat: HDFS, Cassandra, HBase, Amazon S3, text files, SequenceFiles, any Hadoop InputFormat

RESILIENT DISTRIBUTED DATASETS (RDDS)

KÉTFÉLE OPERÁCIÓT TÁMOGAT

- **TRANSFORMATION** - MAP, ...

Új RDD-t hoz létre.

- **ACTION** - REDUCE, ...

Kiszámít egy értéket az elosztott adathalmazon, majd visszaadja az értékét.

Van olyan ami elosztott adathalmazt ad vissza.

(reduceByKey)

SPARK STREAMING

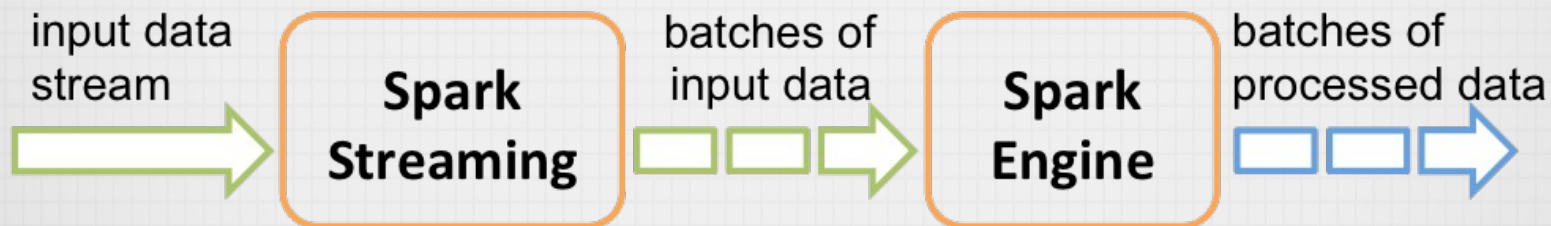
Spark API kiterjesztése, skálázható, hibatűrő streaming feldolgozás kiépítésére.

ADAT INGRESS

- KAFKA
- FLUME
- HDFS / S3
- KINESIS
- TWITTER

ADAT EGRESS


- DATABASE
- HDFS
- DASHBOARD



SPARK STREAMING

WEB UI:

http://localhost:4040/

Stages Storage Environment Executors Streaming NetworkWordCount application UI

Spark Stages

Total Duration: 14 min
Scheduling Mode: FIFO
Active Stages: 1
Completed Stages: 904
Failed Stages: 0

Active Stages (1)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Shuffle Read	Shuffle Write
0	runJob at ReceiverTracker.scala:275	+details (kill) 2014/10/28 10:42:51	14 min	0/1			

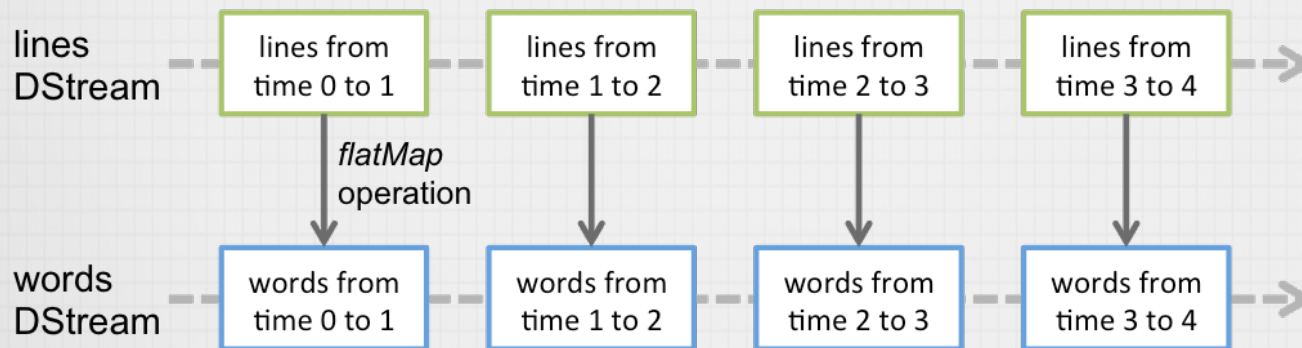
Completed Stages (904)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Shuffle Read	Shuffle Write
3407	take at DStream.scala:608	+details 2014/10/28 10:57:02	2 ms	1/1			
3405	take at DStream.scala:608	+details 2014/10/28 10:57:02	2 ms	1/1			
3403	take at DStream.scala:608	+details 2014/10/28 10:57:01	1 ms	1/1			
3401	take at DStream.scala:608	+details 2014/10/28 10:57:01	2 ms	1/1			
3399	take at DStream.scala:608	+details 2014/10/28 10:57:00	1 ms	1/1			
3397	take at DStream.scala:608	+details 2014/10/28 10:57:00	2 ms	1/1			
3395	take at DStream.scala:608	+details 2014/10/28 10:56:59	1 ms	1/1			
3393	take at DStream.scala:608	+details 2014/10/28 10:56:59	2 ms	1/1			
3391	take at DStream.scala:608	+details 2014/10/28 10:56:58	2 ms	1/1			
3389	take at DStream.scala:608	+details 2014/10/28 10:56:58	2 ms	1/1			
3387	take at DStream.scala:608	+details 2014/10/28 10:56:57	2 ms	1/1			
3385	take at DStream.scala:608	+details 2014/10/28 10:56:57	2 ms	1/1			
3383	take at DStream.scala:608	+details 2014/10/28 10:56:56	1 ms	1/1			
3381	take at DStream.scala:608	+details 2014/10/28 10:56:56	1 ms	1/1			
3379	take at DStream.scala:608	+details 2014/10/28 10:56:55	1 ms	1/1			
3377	take at DStream.scala:608	+details 2014/10/28 10:56:55	1 ms	1/1			
3375	take at DStream.scala:608	+details 2014/10/28 10:56:54	2 ms	1/1			

DSTREAMS

DISCRETIZED STREAMS (DSTREAMS)

- Folyamatos data stream absztrakció
- Diszkrét RDD-k, amelyek időszeletet reprezentálnak
- Spark engine generálja őket





APACHE STORM

APACHE STORM

Elosztott, skálázható, valós idejű, hibatűrő, garantált célba érkezést biztosító számítási rendszer. Stream alapú, eventeket kezel.

Eredetileg Twitterről, Clojure nyelven.

Hadoop = batch processing, Storm = realtime processing

Real-time alkalmazást nehéz írni. Storm sokat könnyít ezen. Kevesebbet kell kódolni, primitíveket ad RT rendszerek építéséhez.

MŰKÖDÉS

Hadoopban MR jobokat futtatunk, Stormban **topology**kat futtatunk.

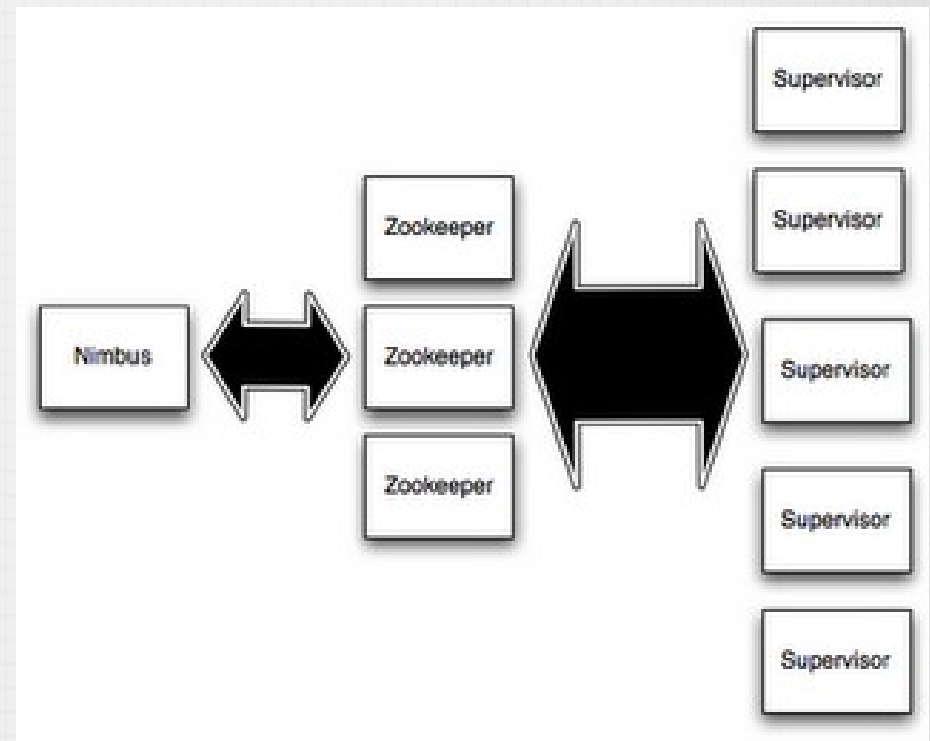
MR egyszer véget ér, egy topology végtelenítve is futhat (általában nem ér véget).

Master - Worker topológia

Master: Nimbus

Worker: Supervisor

Koordináció: Zookeeper



MŰKÖDÉS

NIMBUS (DAEMON) ~ JOBTACKER

EZ A MASTER.

- Code disztribúció
- Taszk kiosztás
- Hiba monitorozás

ZOOKEEPER

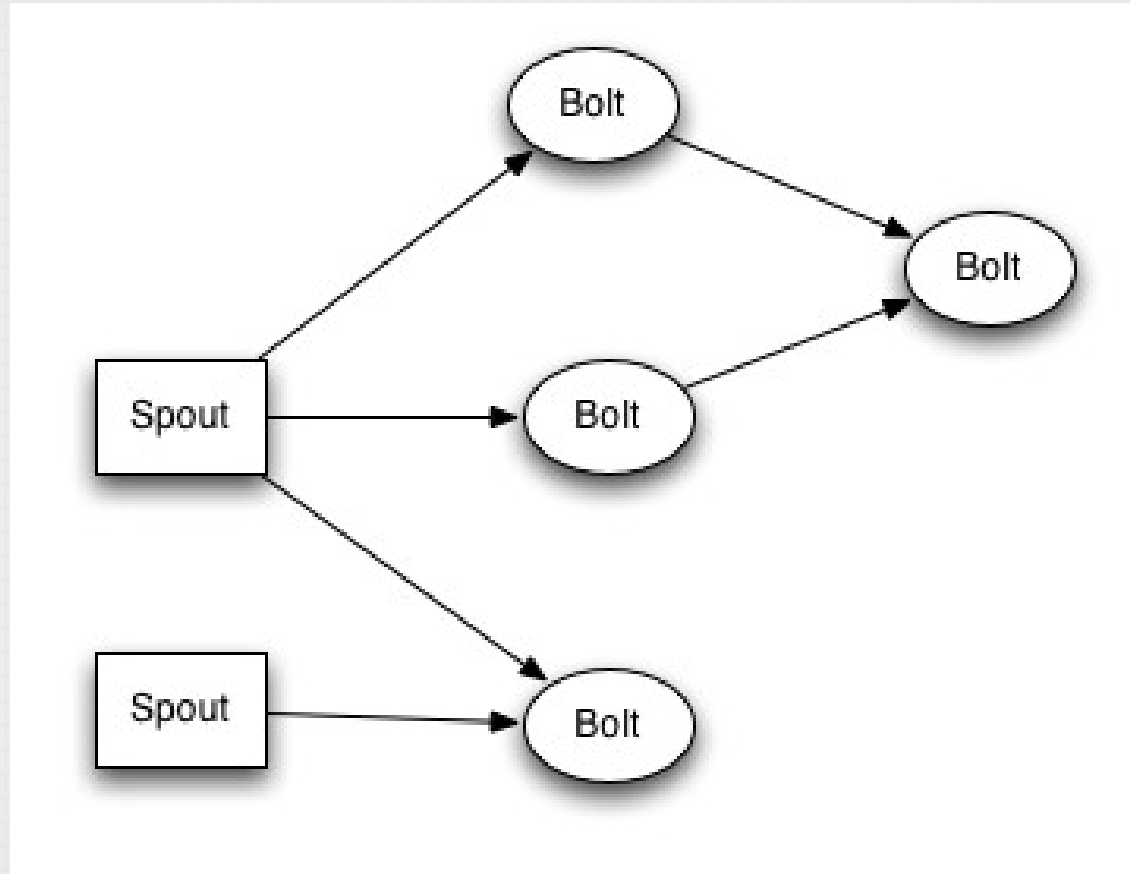
KOORDINÁCIÓ.

SUPERVISOR (DAEMON)

WORKER NODE-OK.

Szintén daemon fut rajtuk.

KOMPONENSEK



KOMPONENSEK

SPOUT (IRichSpout)

Tekinthető forrásnak. Valahonnan kap adatot (Twitter API, MQ, etc...), majd azt streambe ereszti.

BOLT (IRichBolt)

Adat feldolgozási lépés.

Egy vagy több streamből kapja, és opcionálisan új streamet adnak. Filter, DB művelet, Streaming join, Functions, write, etc...

STREAM

Amikor a streambe kerül egy tuple, akkor minden Bolt megkapja aki fel van iratkozva az adott streamre.

KOMPONENSEK

Minden komponens parallel fut. A párhuzamosíthatóságra a klaszter mérete, kihasználtsága van befolyással.

Üzenetvesztés, node kiesés, timeout vagy failure esetén is újraütemezi és újra végrehajtja.

Storm biztosítja, hogy üzenet nem veszik el.

Párhuzamosíthatóság mértéke megadható.

LocalCluster teszteléshez. Ez a Local Mode, Remote mode, létező klaszterrel.