



Dr. Hegedűs Péter, Dr. Ferenc Rudolf

Nagyméretű adatbázisok

Jelen tananyag a Szegedi Tudományegyetemen készült az Európai Unió támogatásával.

Projekt azonosító: EFOP-3.4.3-16-2016-00014

Apache Hadoop és HDFS

Összefoglalás

Ez a kombinált olvasó és video lecke bemutatja hogyan használható az Apache Hadoop HDFS elosztott fájlrendszer a gyakorlatban. Docker virtualizáció segítségével bemutatjuk, hogyan lehet egy HDFS klasztert felállítani, majd azon fájlműveleteket végezni. Mindezt egy videó leckével is demonstráljuk, a könnyebb követhetőség érdekében.

A lecke fejezetei:

- 1. fejezet: **Hadoop klaszter létrehozása és futtatása Docker segítségével (olvasó)**
- 2. fejezet: **HDFS használata, alapvető parancsok (videó)**

Téma típusa: **gyakorlati**

Olvasási idő: **30 perc**

1. fejezet

Apache Hadoop klaszter indítása

Ahogy azt az előadásban is láthattuk, az Apache Hadoop keretrendszer és a HDFS elosztott fájlrendszer üzemeltetéséhez több szoftverkomponensre is szükségünk van: `NameNode`, `ResourceManager`, `DataNode`, `HistoryNode`, stb. Ebben a fejezetben bemutatjuk, hogy lehet telepítés nélkül, viszonylag egyszerű módon egy ilyen klasztert tetszőleges gépen beüzemelni. Ehhez a Docker [1] konténer szoftvert és előre elkészített konténer fájlokat fogunk használni. A Docker rendszer bemutatása nem tartozik jelen anyag keretébe, de az alábbi forrásokból további információk szerezhetők róla:

- <https://www.youtube.com/watch?v=fqMOX6JjhGo>
- <https://www.youtube.com/watch?v=zj6WbK9zFpI>
- <https://docs.docker.com/get-started/>

Docker elérhető az összes fontosabb operációs rendszerre, a következők feltételezik, hogy rendelkezésre áll egy telepített Docker rendszer a gépen, ami a hivatalos dokumentáció [1] alapján könnyen kivitelezhető. A klaszterhez szükséges egyes node-ok képfájljai GitHub-on elérhetők [2], a példa során innen fogjuk letölteni őket.

Képfájlok letöltése és Hadoop klaszter indítása

Első lépésként a Big Data Europe GitHub repozitóriumból töltsük le a megfelelő képfájlokat. Ehhez feltétel, hogy a Git [3] verziókövető elérhető legyen gépünkön. Amennyiben nem az, először telepítsük. Ezután a következő paranccsal letölthetjük gépünkre a megfelelő Docker image fájlokat:

```
$ git clone https://github.com/big-data-europe/docker-hadoop.git
```

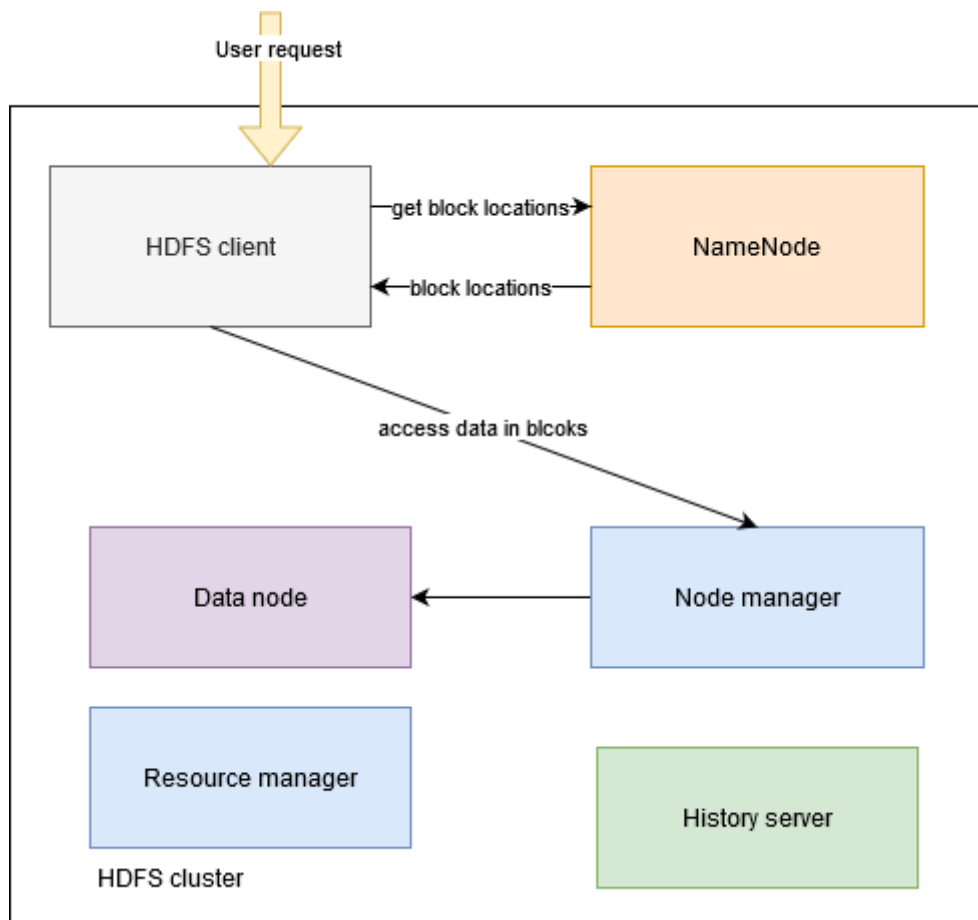
A parancs végrehajtása után letöltődnek a szükséges image-ek a `docker-hadoop` mappába a saját gépünkre. A klaszter elemeit a `docker-compose.yml` fájl írja le, amennyiben szükséges a klaszter működésének módosítása, úgy ezt a fájlt kell átszerkeszteni. A példánk során mi a módosítatlan konfigurációs fájlt fogjuk használni, ami a következő komponenseket definiálja:

- `namenode` - a Hadoop NameNode csomópont a megfelelő port beállításokkal, ami a ~~master~~ primary node szerepét tölti be
- `datanode` - adattároló csomópont, egy darab (a klaszter egyetlen adat node-ot tartalmaz)
- `resourcemanager` - erőforrás kezelésért felelős node (Apache Yarn)
- `nodemanager1` - Hadoop Node Manager csomópont
- `historyserver` - Hadoop history server node

A klaszter elindításához az alábbi Docker parancsot kell kiadnunk:

```
$ docker-compose up -d
```

A parancs hatására a megfelelő Docker image-ek letöltődnek a Docker-Hub [4] központi image tárhelyről, lefordulnak és előállnak a szükséges állományok, majd a fenti szolgáltatások elindulnak. A futó szolgáltatásokat és a Hadoop klaszter elemeit az alábbi ábra szemléltet.



Hadoop klaszter beállításai

Először ellenőrizzük le, hogy a fenti parancs sikeresen végrehajtott. Adjuk ki az alábbi Docker parancsot:

```
$ docker ps
```

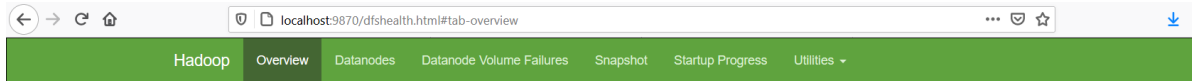
Ez a parancs az éppen futó Docker konténereket listázza ki, így a fenti ábrán is látható öt komponens mindegyikét látnunk kell a listában:

```

λ docker ps
CONTAINER ID        IMAGE                                     COMMAND                  CREATED            STATUS
PORTS
f9f64f9e7eb8      bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8  "/entrypoint.sh /run..."  14 hours ago      Up 14 hours (healthy)
0.0.0.0:9000->9000/tcp, 0.0.0.0:9870->9870/tcp    namenode
9beae689e0fd      bde2020/hadoop-resource-manager:2.0.0-hadoop3.2.1-java8  "/entrypoint.sh /run..."  14 hours ago      Up 14 hours (healthy)
8088/tcp                                               resource-manager
daa75a13e848      bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8  "/entrypoint.sh /run..."  14 hours ago      Up 14 hours (healthy)
8188/tcp                                               historyserver
8dbf9a06a4fe      bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8  "/entrypoint.sh /run..."  14 hours ago      Up 14 hours (healthy)
8042/tcp                                               nodemanager
e4782339acd9      bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8  "/entrypoint.sh /run..."  14 hours ago      Up 14 hours (healthy)
9864/tcp                                               datanode

```

Ha minden rendben van, a NameNode komponenst böngészőből is elérjük a <http://localhost:9870> címen. A következő kép a NameNode webes felületét mutatja:



Overview 'namenode:9000' (active)

Started:	Thu Aug 06 21:18:36 +0200 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 17:56:00 +0200 2019 by rohlitsharmaks from branch-3.2.1
Cluster ID:	CID-d07764c7-56a9-4602-b776-a65083b5eea9
Block Pool ID:	BP-1855640961-172.22.0.6-1596736551543

Summary

Security is off.

Safemode is off.

64 files and directories, 31 blocks (31 replicated blocks, 0 erasure coded block groups) = 95 total filesystem object(s).

Heap Memory used 76.12 MB of 149.5 MB Heap Memory. Max Heap Memory is 443 MB.

Non Heap Memory used 62.9 MB of 64.63 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	58.42 GB
Configured Remote Capacity:	0 B
DFS Used:	1.46 MB (0%)

Előfordulhat, hogy a NameNode indulás után ún. safe mode-ba kerül, ekkor a fenti áttekintő oldalon a `safemode is on` felirat jelenik meg. Ekkor az adat csomópontok nem írhatók, a klasztert nem tudjuk teljes körűen használni. Ebben az esetben a következő lépéseket kell végrehajtanunk.

1. Indítsunk egy bash terminált a futó NameNode konténeren belül:

```
$ docker exec -it namenode bash
```

2. Hagyjuk el a safe mode állapotot a következő parancs kiadásával

```
root@f9f64f9e7eb8:/# hdfs dfsadmin -safemode leave
```

3. Töröljünk és állítsunk helyre minden hibás blokkot, amit a safe mode miatti read only mód okozott

```
root@f9f64f9e7eb8:/# hdfs fsck / -delete
```

Ezután töltsük be a NameNode webes felületét, már a fenti `safemode is off` feliratot kell látnunk.

Hadoop klaszter tesztelése

Amennyiben fut a Hadoop klaszterünk, a NameNode konténerben futtatott bash terminál segítségével Hadoop parancsokat adhatunk ki a `hdfs` parancssori kliens segítségével. Ezt az eszközt használhatjuk arra, hogy fájlműveleteket végezzünk a HDFS-en (lásd 2. fejezet). Ehhez lépünk ismét be a futó NameNode dokcer konténerbe és indítsunk egy bash terminált:

```
$ docker exec -t namenode bash
```

Hozzunk létre pár egyszerű szöveges fájlt és másoljuk fel őket HDFS-re:

```
root@f9f64f9e7eb8:/# mkdir input
root@f9f64f9e7eb8:/# echo "Hello world Bye world" >input/file01
root@f9f64f9e7eb8:/# echo "Hello Hadoop Goodbye Hadoop" >input/file02
```

Miután a NameNode konténeren belül létrehoztuk a fájlokat, hozzunk létre egy könyvtárat HDFS-en és másoljuk fel ezt a két fájlt:

```
root@f9f64f9e7eb8:/# hadoop fs -mkdir -p input
root@f9f64f9e7eb8:/# hdfs dfs -put ./input/* input
```

Most már a HDFS-en is elérhető a két fájl, tetszőleges Hadoop kompatibilis eszközzel feldolgozható (pl. MapReduce, Spark, stb.). Listázzuk ki az input könyvtár tartalmát és nézzük meg a fájl tartalmát:

```
root@f9f64f9e7eb8:/# hadoop fs -ls /input
Found 2 items
-rw-r--r--  3 root supergroup      21 2020-08-07 07:56 /input/file01
-rw-r--r--  3 root supergroup      27 2020-08-07 07:56 /input/file02
```

```
root@f9f64f9e7eb8:/# hadoop fs -cat /input/file01
2020-08-07 11:34:03,619 INFO sasl.SaslDataTransferClient: SASL encryption trust
check: localHostTrusted = false, remoteHostTrusted = false
Hello world Bye world
```


A HDFS klaszter leállításához az alábbi Docker parancsot kell használni:

```
$ docker-compose down
```

2. fejezet

Apache Hadoop File System (HDFS)

Az alábbi videó lecke a HDFS használatát, a rajta végezhető műveleteket demonstrálja:

 [video/3g_BigData-hadoop-SPOC/hdfs-demo.mp4](#)

✓ További feladatok

1. Módosítsuk a Hadoop lkaszter konfigurációt úgy, hogy egy data node helyett három induljon!
Segítség: <https://github.com/big-data-europe/docker-hadoop/issues/40> ★★ ★
2. Hozz létre egy `feladat` mappát a HDFS fájlrendszerben, és másolj fel egy tetszőleges csv fájlt bele, ezután listázd ki a fájlt parancssorból és nézd meg a NameNode webes felületén keresztül is a browser utility segítségével! Ezután pedig töltsd le a csv fájlt HDFS-ről a saját gépre! ★

Referenciák

[1] <https://www.docker.com/>

[2] <https://github.com/big-data-europe/docker-hadoop>

[3] <https://git-scm.com/>

[4] <https://hub.docker.com/>

[5] https://www.tutorialspoint.com/hadoop/hadoop_command_reference.htm