

PTS Vaja 5

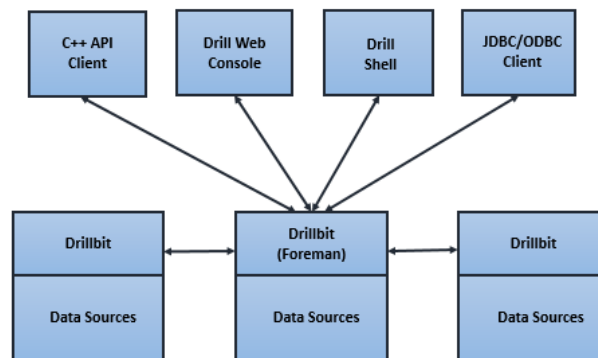
Za vajo 5 bomo uporabili naslednje tehnologije in orodja:

- Apache Drill 1.19

Opomba: Za uspešen zagon Drilla je potrebno, da v Docker kontejnerju zažene HDFS in HBase.

1. Apache Drill

Apache Drill je odprtokodni porazdeljeni SQL povpraševalni mehanizem (angl. query engine), ki omogoča analizo velepodatkov. Za razliko od ostalih povpraševalnih mehanizmov (Hive, Presto itn.), Drill ne pričakuje vnaprej določeno podatkovno shemo, ampak temelji na JSON shemi podobni MongoDB-u. Kot ena izmed bolj pomembnih lastnosti Drilla se lahko izpostavi tudi pisanje povpraševanj z uporabo "čiste" SQL sintakse, kar pomeni, da ni potrebe po uporabi marsikaterih spremenjenih sintaks, ki temeljijo na SQL-u. Apache Drill se lahko namesti v dveh oblikah: vgrajeno (angl. embedded) in porazdeljeno (angl. distributed). Glede na to da delamo v psevdo-porazdeljenem okolju z le enim HBase procesom, nam je dovolj tudi le en Drill process, in ga bomo zato namestili v vgrajeni obliki.



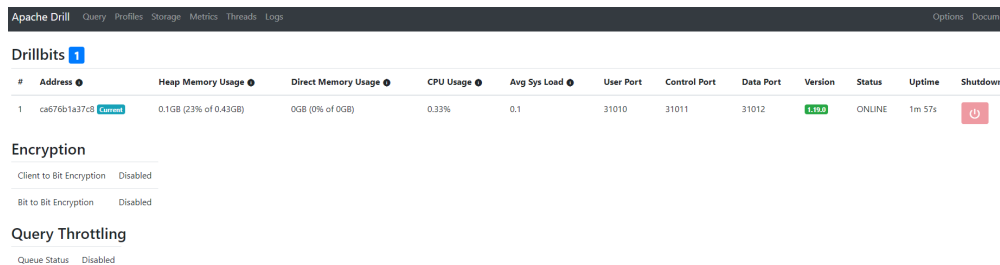
Slika 1: Arhitektura Apache Drill mehanizma.²
² <https://drill.apache.org/docs/drill-query-execution/>

Namestitev Drill-a v našem sistemu je izjemno preprosta in zelo podobna namestitvi ostalih orodij v Hadoop ekosistemu. Za izvedbo poizvedb Apache Drill uporablja orodje *SQLLine* temeljeno na Javi. Ko smo v Drill korenski mapi (*/opt/drill*), proces lahko zaženemo z ukazom:

```
bin/sqlline -u jdbc:drill:zk=local
```

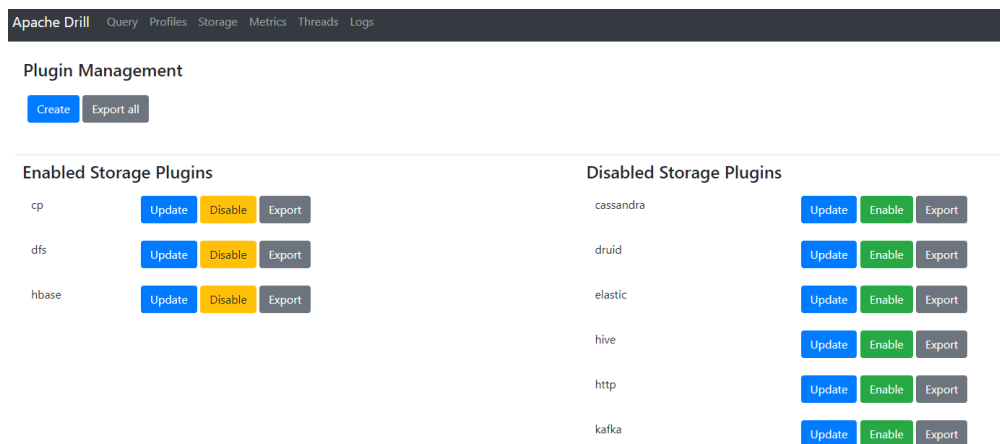
Zažene se terminal, v katerem lahko izvajamo SQL povpraševanja. V seznamu aktivnih Java procesov pa se vmes pojavi proces *SqlLine*.

Ko smo zagnali Apache Drill, dodatno konfiguracijo lahko izvedemo z uporabo spletne strani, ki nam bo dostopna na *localhost:8047*, kot prikazano na sliki 2.



Slika 2: Spletna stran za konfiguracijo Drill-a.

Pod opcijo *Storage* v meniju lahko vidimo seznam trenutno aktivnih in možnih vtičnikov za povezavo z različnimi viri podatkov (slika 3). Privzeto sta v Drill-u omogočena le *cp* in *dfs* vtičnika, oz. *cp*, ki kaže na JAR datoteke znotraj Drill poti do razredov, in *dfs*, ki kaže na lokalni datotečni sistem.



Slika 3: Seznam dostopnih vtičnikov za Drill.

Mi bomo povezali Drill na ustvarjeno HBase bazo. Zato moramo najprej na spletni strani omogočiti HBase vtičnik. Nato se ta pojavi v seznamu omogočenih vtičnikov, kot prikazano na sliki 3.

Izvedba SQL povpraševanj nad HBase bazo z uporabo Apache Drill-a

Ko želimo pisati SQL povpraševanja v Drill-u, najprej moramo določiti kateri vtičnik želimo uporabljati z ukazom *USE*. Mi bomo uporabljali HBase vtičnik, in to določimo z ukazom:

```
USE hbase;
```

Zdaj pa lahko preverimo, če smo se uspešno povezali na HBase, tako da poskusimo dostopiti do zapisov v HBase tabeli *offers*, ki smo jo predhodno ustvarili. **Opomba:** bodite previdni pri uporabi narekovaj, saj orodje interpretira edino narekovaje (**Alt Gr + 7** pri UK razporedi tipkovnice). Zaženemo ukaz:

```
select * from `pts:offers`;
```

Rezultat tega ukaza so vrstice iz tabele v HBase bazi, kot prikazano na sliki 4, vendar so posamezne vrednosti v poljih kodirane.

row_key	item	price	rest_id	rest
1	{ "item_id": "MQ==", "item_name": "TWJLudCBTXYVjZQ==" }	{ "price": "H54wMA==" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
10	{ "item_id": "MTA=", "item_name": "VGfUzG9vcmkO9p0aQ==" }	{ "price": "HjYvMDA==" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
100	{ "item_id": "MTAX", "item_name": "Q3Jycnk=" }	{ "price": "NzkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
101	{ "item_id": "MTAX", "item_name": "VmIuzGF5Bz8==" }	{ "price": "NzkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
102	{ "item_id": "MTAY", "item_name": "S29ybWELSBdaGlja2VuIHRpa2ht" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
103	{ "item_id": "MTAZ", "item_name": "Q2hpY2t1biB1bGUatVjYSXBWYXRoZWbe=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
104	{ "item_id": "MTA0", "item_name": "Q2hpY2t1biB1bGUatVjYSXBWYXRoZWbe=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
105	{ "item_id": "MTA1", "item_name": "Q2hpY2t1biB1bGUatVjYSXBWYXRoZWbe=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
106	{ "item_id": "MTA2", "item_name": "TMWoaGklSBHYW1t" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
107	{ "item_id": "MTA3", "item_name": "VGfUzG9vcmkO2hpY2t1biAOTWFpbik=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
108	{ "item_id": "MTA4", "item_name": "OmhlbmEgLlSBdaGlja2Vu" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
109	{ "item_id": "MTA5", "item_name": "S29ybWELSBdaGlja2VuIFRpap2ht" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
11	{ "item_id": "MTF=", "item_name": "UGXhaw4gtmfbhg==" }	{ "price": "HjYvMDA==" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
110	{ "item_id": "MTF=", "item_name": "Q2hpY2t1biB1bGUatVjYSABWTFpbik=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
111	{ "item_id": "MTF=", "item_name": "Q2hpY2t1biB1bGUatVjYSABWTFpbik=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
112	{ "item_id": "MTF=", "item_name": "RHVwaWFGYSAIENoalnNrZM4=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
113	{ "item_id": "MTF=", "item_name": "OmhlbmEgLlSBdaGlja2VuIFRpap2ht" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
114	{ "item_id": "MTF=", "item_name": "UGf0p0t1hiC9tRGFYe-" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
115	{ "item_id": "MTF=", "item_name": "Q2hpY2t1biB1bGUatVjYSABWTFpbik=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
116	{ "item_id": "MTF=", "item_name": "OmhlbmEgLlSBHYW1t" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
117	{ "item_id": "MTF3", "item_name": "RGhhbnNhYAtIENoalnNrZk4vGVRla2E=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	
118	{ "item_id": "MTF4", "item_name": "RGhhbnNhYAtIENoalnNrZk4=" }	{ "price": "ODkLjAw" }	{ "rest_id": "MQ==", "rest_name": "RlwGgGlhyMxv" }	

Slika 4: Dostop do podatkov v HBase tabeli.

Namreč, Drill nam vrne rezultate v binarni obliki, oz. kot bajtna polja (angl. byte array), ki jih potem najprej moramo pretvoriti v UTF-8 obliko z izvajanjem ukaza. Za to uporabljamo funkcijo `CONVERT_FROM`, kjer kot prvi argument podamo naziv stolpca, ki ga želimo pretvoriti v obliko določeno v drugem argumentu. **Pomembno:** HBase stolpcu znotraj stolpčne družine pristopamo z uporabo t.i. "dot" notacije. Ta postopek dostopa v globino do posameznih stolpcev se imenuje "drill down".

Kot primer, če želimo izpisati cene 10 izdelkov po restavracijah, bomo izvedli naslednji ukaz:

```
SELECT CONVERT_FROM(row_key, 'UTF8') as offer_id,
       CONVERT_FROM(`pts:offers`.rest.rest_name, 'UTF8') as name,
       CONVERT_FROM(`pts:offers`.item.item_name, 'UTF8') as item,
       CONVERT_FROM(`pts:offers`.o.price, 'UTF8') as price
FROM `pts:offers` LIMIT 10;
```

```

apache drill (hbase)> SELECT CONVERT_FROM(row_key, 'UTF8') as offer_id,
CONVERT_FROM(`pts:offers`.rest.rest_name, 'UTF8') as name, CONVERT_FROM
`pts:offers`.item.item_name, 'UTF8') as item, CONVERT_FROM(`pts:offers`.
o.price, 'UTF8') as price FROM `pts:offers` LIMIT 10;

```

offer_id	name	item	price
1	El Diablo	Mint Sauce	5.00
10	El Diablo	Tandoori Roti	26.00
100	El Diablo	Curry	795.00
101	El Diablo	Vindaloo	795.00
102	El Diablo	Korma - Chicken	895.00
103	El Diablo	Chicken Tikka Masala	895.00
104	El Diablo	Pathia - Chicken Tikka	895.00
105	El Diablo	Chicken Tikka Jalfrezi	895.00
106	El Diablo	Methi - Lamb	895.00
107	El Diablo	Tandoori Chicken (Main)	895.00

```

10 rows selected (0.313 seconds)

```

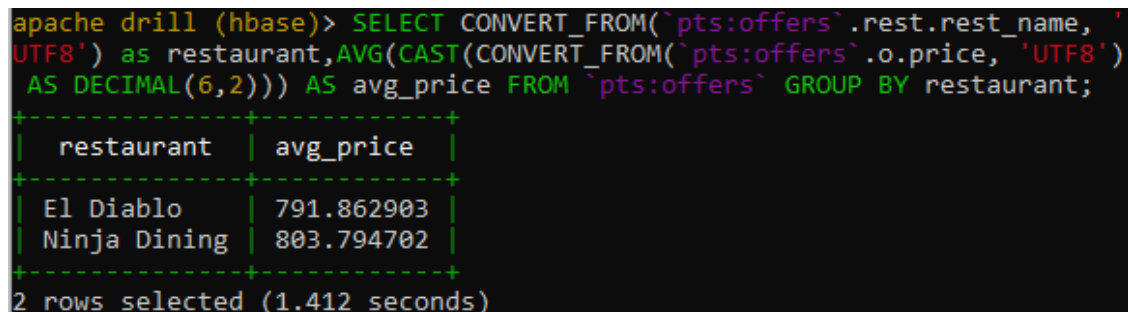
Slika 5: Prikaz podatkov vrnjenih iz HBase-a spremenjenih v UTF-8 obliko.

Vse možnosti izvedbe SQL povpraševanj nad podatki v različnih virih lahko pogledate v uradni dokumentaciji Drill-a, ki je dostopna na <https://drill.apache.org/docs/>. Preverite tudi dostopne podatkovne tipe in načine njihove pretvorbe (CAST, CONVERT_FROM, CONVERT_TO funkcije) ter ostale SQL funkcije.

Razen osnovnega dostopa do podatkov v HBase-u, Apache Drill nam omogoča tudi uporabo različnih SQL funkcij nad podatki. Kot primer, lahko napišemo SQL povpraševanje, ki nam vrne povprečno ceno izdelkov po posameznih restavracijah (seveda, upoštevajoč pretvorbo tipov podatkov):

```
SELECT CONVERT_FROM(`pts:offers`.rest.rest_name, 'UTF8') as restaurant,
       AVG(CAST(CONVERT_FROM(`pts:offers`.o.price, 'UTF8') AS DECIMAL(6,2))) AS avg_price
FROM `pts:offers`
GROUP BY restaurant;
```

Za to povpraševanje lahko uporabimo SQL AVG() funkcijo, ki združi posamezne cene za vsako restavracijo. Poskusite napisati SQL povpraševanje, ki bo vrnilo najcenejša izdelka (skupaj z njuno ceno) v posamezni restavraciji.



```
apache drill (hbase)> SELECT CONVERT_FROM(`pts:offers`.rest.rest_name, '
UTF8') as restaurant,AVG(CAST(CONVERT_FROM(`pts:offers`.o.price, 'UTF8')
AS DECIMAL(6,2))) AS avg_price FROM `pts:offers` GROUP BY restaurant;
+-----+-----+
| restaurant | avg_price |
+-----+-----+
| El Diablo   | 791.862903 |
| Ninja Dining | 803.794702 |
+-----+-----+
2 rows selected (1.412 seconds)
```

Slika 6: Prikaz povprečne cene izdelkov v restavracijah.

Opomba: Apache Drill terminal lahko zaprete z ukazom *!quit*.