Report First project Big Data

Luca De silvestris (486652), Luca Polidori (488434)
Group name: DesiDori

May 29, 2019

assignment

Si consideri il dataset Daily Historical Stock Prices, scaricabile dal sito del corso, che contiene l'andamento giornaliero di un'ampia selezione di azioni sulla borsa di New York (NYSE) e sul NASDAQ dal 1970 al 2018. Il dataset è formato da due file CSV. Ogni riga del primo ha i seguenti campi:

- ticker: simbolo dell'azione
- open: prezzo di apertura
- close: prezzo di chiusura
- adjclose: prezzo di chiusura "modificato" (potete trascurarlo)
- lowThe: prezzo minimo
- highThe: prezzo massimo
- volume: numero di transazioni
- date: data nel formato aaaa-mm-gg

Il secondo ha invece questi campi:

- ticker: simbolo dell'azione
- exchange: NYSE o NASDAQ
- name: nome dell'azienda
- sector: settore dell'azienda
- industry: industria di riferimento per l'azienda

Progettare e realizzare in: (a) MapReduce, (b) Hive e (c) Spark:
1. Un job che sia in grado di generare, in ordine, le dieci azioni la cui quotazione (prezzo di chiusura) è cresciuta maggiormente dal 1998 al 2018, indicando, per ogni azione: (a) il simbolo, (b) l'incremento percentuale, (c) il prezzo minimo raggiunto, (e) quello massimo e (f) il volume medio giornaliero in quell'intervallo temporale.
2. Un job che sia in grado di generare, per ciascun settore, il relativo "trend" nel

periodo 2004-2018 ovvero un elenco contenete, per ciascun anno nell'intervallo: (a) il volume complessivo del settore, (b) la percentuale di variazione annuale (differenza percentuale arrotondata tra la quotazione di fine anno e quella di inizio anno) e (c) la quotazione giornaliera media. N.B.: volume e quotazione di un settore si ottengono sommando i relativi valori di tutte le azioni del settore.
3. Un job in grado di generare coppie di aziende di settori diversi le cui azioni che, negli ultimi 3 anni, hanno avuto lo stesso trend in termini di variazione annuale indicando le aziende e il trend comune (es. Apple, Fiat, 2016:-1%, 2017:+3%, 2018:+5%).

First Job

Variables and choices

We have chosen to use the following variables to find the respective project specifications for the first job:

- **Incremento Percentuale**: is calculated by taking for each ticker the value of close variable with minimum date (*prz_ini_chiusura*), value of close with maximum date (*prz_fin_chiusura*) and applying the following formula:

  ((*prz_fin_chiusura* - *prz_iniz_chiusura*) / *prz_iniz_chiusura*)*100.

- **prezzo minimo**: calculated by taking the min (low) for each ticker.
- **prezzo massimo raggiunto**: calculated by taking the Max(high) for each ticker.
- **volume medio giornaliero**: represented by the variable *volume_avg_giornaliero* is calculated by taking for each ticker: *sum(volume) / numero_giorni*.

PseudoCode Map-Reduce and output

[**1**]**Map**

- [**1.1**] for each record of the csv "historical_stock_prices" it performs the filtering of the rows based on the date field such that 1998 ¡= anno ¡= 2018;
- [**1.2**] return ticker, close, low, high, volume, data.

[**2**]**Reduce**

- [**2.1**]For each resource of the map output group the actions for ticker
- [**2.2**] Calculates for each ticker the value of: incremento_percentuale, prez_min_raggiunto, prez_max_raggiunto, volume_avg_giornaliero.
- [**2.3**]sort by the percentage increment field and take the first 10 actions
- [**2.4**]return ticker, incrementPercentuale, prezzoMin, prezzoMax, volumeAvg

```
SAB     2629529.50579%  1.3654999733    319600.0        1608166.87361
PJT     296300.000522%  0.00999999977648        11102.5 71484.4493298
EAF     267757.134589%  0.00200000009499        24.3640003204   1245139.03846
UVE     226900.012703%  0.019999999553  45.9000015259   224226.442874
ORGS    217415.926128%  0.00313999992795        19.9200000763   8570.9623431
PUB     179900.004023%  0.00899999961257        138.0   34449.4007092
MNST    163340.387616%  0.0305979158729 70.2200012207   7347898.8208
RMP     121081.601901%  0.0299999993294 79.8187332153   120487.047671
CCD     111250.004778%  0.0149999996647 25.9799995422   105416.892662
KE      99400.0003166%  0.0149999996647 22.4500007629   73249.8461538
```

Figure 1: output Job 1 Map Reduce

PseudoCode Hive and output

1 create table "prices" and upload data from the csv "*historical_stock_prices.csv*

2 for filtering year from 1998 to 2018

3 final section of the fields: ticker, incremento percentuale, prezzo minimo, prezzo massimo

4 grouping of the fields by ticker (by using *Group by*) and by sorting according to "incrementoPercentuale" (by using *Order by*)

```
ticker  crescita                valore min              valore max      volume Medio

SCON    1.7797499999988145E8    1.47000002861023        2070000.0       14982.826199071686
CODA    1.354499999999979E8         0.0140000004321337  1736700.0       11013.228124040528
TVIX    1.3474999999776104E8    29.6299991607666        1542500.0       982725.8870967742
TOPS    1.1897999999994032E8    0.689999997615814       1276200.0       988077.4390243902
CTIC    8.002499999978507E7         1.70000004768372    819000.0        135820.1916495551
DRYS    5.723199999981829E7         0.980000019073486   799680.0        2577652.561247216
YRCW    4.754999999903049E7         4.55999994277954    483525.0        285654.3903110377
SAB     3.1879999999561325E7    1.36549997329712        319600.0        1608166.873605948
ABIO    3.1814999999858554E7    0.449999988079071       351540.0        71239.26764635027
MYND    2.6249999999546666E7    1.14900004863739        309300.0        81206.73291925466
```

Figure 2: output Job 1 Hive

PseudoCode Spark and output

1 csv loading and DataFrame "init" creation containing all the csv columns and adding the "year" column by processing the date field with filtering of the years¿ = 1998;

2 create dataFrames "prezzominimo","prezzomassimo" and "volumeGiornalieroMedio" starting from "init" dataframe and group by the ticker and adding respectively "prezzoMin", "prezzoMax" and "volumeMedio" columns;

3 join between dataframe and maximum, price by ticker and with the addition of the "incremento percentuale" column;

4 realization of the final dataframe with the addition of the column concerning the average daily volume and sorting by percentage increase.

ticker	prezzoMax	prezzoMin	crescita	avg_daily_volume
PJT	9937.8896484375	0.00999999977648258	9.93787987056665E7	71484.44932975872
CODA	9975	0.0140000004321337	7.12498978007482E7	11013.228124040528
DTW	98.4899978637695	0.00100000004749745	9848899.318574598	10360.927419354839
ORG	98.4899978637695	0.00100000004749745	9848899.318574598	1897.2380952380952
AJXA	91	0.00100000004749745	9099899.567773225	1957.512676056338
CTAA	90	0.00100000004749745	8999899.57252297	26025.496788008564
CTIC	99240	1.70000004768372	5837546.895081929	135820.1916495551
HMNY	995	0.0199999995529652	4974900.11119991	4884203.156146179
MAMS	9706.5	0.270000010728836	3594899.857147541	12696.681071737252
TWLO	9.60000038146973	0.000300000014249235	3199899.9751647376	1312462.7000726217

Figure 3: output Job 1 Spark

Second Job

Variables and choices

We have chosen to use the following variables to find the respective project specifications for the second job:

- **volume_complessivo_settore**: calculated for each sector and for each year by adding up all the volumes of the ticker;
- **quotazione_giornaliera_media**:calculated for each sector, for each year as the average of the sum of the relative values of that sector;
- **percentuale_variazione_annuale**: calculated for each sector, for each year by taking the sum of the close values with minimum date and same sector (*prz_ini_chiusura*), the sum of the close values having maximum date and same sector (*prz_fin_chiusura*)and applying the following formula:

$$((prz\_fin\_chiusura - prz\_iniz\_chiusura) / prz\_iniz\_chiusura)*100$$

PseudoCode Map-Reduce and output

[**1**]**Map**

- **[1.1]** for each record of the csv "historical_stock_prices" it performs the filtering of the rows based on the date field such that 2004 ¡= anno ¡= 2018;
- **[1.2]** return record, close, volume, data;
- **[1.3]** for each row of the csv historical_stocks take the fields for ticker and sector;
- **[1.4]** maps the records obtained based on the ticker;
- **[1.5]** return sector, ticker, data, close, volume.

[**2**]**Reduce**

- **[2.1]**group tickers by sector and by year;

- [**2.2**]calculates the "volume_complessivo","percentuale_di_variazione_annuale", "quotazione_giornaliera_media";
- [**2.3**]return sector, anno, volume_settore, percentuale_di_variazione, quotazione_giornaliera_media.

```
BASIC INDUSTRIES        2004    30767395827     0.228150473063  2865.87369398
BASIC INDUSTRIES        2005    37457588379     0.0633341791619 4367.6822369
BASIC INDUSTRIES        2006    50413342778     0.294700074157  7110.07082055
BASIC INDUSTRIES        2007    67640775192     0.185667471534  9211.13392303
BASIC INDUSTRIES        2008    104336790359    -0.050131389643 7124.09400267
BASIC INDUSTRIES        2009    113161759706    0.0348287578952 4727.91920472
BASIC INDUSTRIES        2010    96267427694     0.217900355607  6126.05228338
BASIC INDUSTRIES        2011    93277620675     -0.586007636167 8535.55187455
BASIC INDUSTRIES        2012    79648935208     -0.687885248968 8694.81899382
BASIC INDUSTRIES        2013    81167036326     0.103226361955  28486.4249861
BASIC INDUSTRIES        2014    82010502666     -0.719021325857 24380.2074047
BASIC INDUSTRIES        2015    95592658398     -0.481011719633 9538.71135927
BASIC INDUSTRIES        2016    120096921114    0.138293577492  7721.85529589
BASIC INDUSTRIES        2017    100051884333    0.152790107615  9101.40840867
BASIC INDUSTRIES        2018    61239875499     -0.0307952148952        9883.13711755
CAPITAL GOODS   2004    42642420682     0.101273663677  6235.49223848
CAPITAL GOODS   2005    42828857449     -0.0557418400368        7148.94421579
CAPITAL GOODS   2006    51749654684     0.071397974715  7704.19065998
CAPITAL GOODS   2007    65271537325     0.0588558732532 9028.14733633
CAPITAL GOODS   2008    90216887556     -0.483138325411 7290.78474163
CAPITAL GOODS   2009    96803739926     0.287272416465  5181.69883919
```

Figure 4: output Job 2 Map Reduce

PseudoCode Hive and output

1 create table "prices" and "stock" and uploading data of the relative csv "*historical_stock_prices.csv*" and "*historical_stocks.csv*";

2 create table of joins between prices and stock tables with "ticker" as join condition with field selection: ticker, close, volume, data and sector and filtering year since 2004;

3 create table "sumOfVolume" for exclusive calculation of the "volume_complessivo_settore" variable by grouping the table by sector and year;

4 create table "DateMinMax" containing for each sector, ticker, year the "mindate" and the "maxdate";

5 create tables "minClose" e "maxClose" containing for each sector and year the sum("close") variable for each ticker with the condition "date"== "mindate" (for minClose table) and "date"=="maxdate" (for maxClose table);

6 create table "percentualeVariazione" having the fields sector, year and "variazione_annuale" sorted by sector and by year;

7 create table "quotazione_giornaliera_media" which takes the average of the sum of the closing values for each sector and year

8 final query is the selection of the required fields FROM tabbles "quotazione_giornaliera_media", "percentualeVariazione" and "sumOfVolume" with the conditions for the sector and the year.

SECTOR	ANNO	VOLUME COMPLESSIVO	VARIAZIONE ANNUALE	QUOTAZIONE GIORNALIERA MEDIA
BASIC INDUSTRIES	2004	3.0767395827E10	22.82	2865.873693978266
BASIC INDUSTRIES	2005	3.7457588379E10	6.33	4367.682236903125
BASIC INDUSTRIES	2006	5.0413342778E10	29.47	7110.070820546957
BASIC INDUSTRIES	2007	6.7640775192E10	18.57	9211.133923027857
BASIC INDUSTRIES	2008	1.04336790359E11	-5.01	7124.094002667505
BASIC INDUSTRIES	2009	1.13161759706E11	3.48	4727.919204718714
BASIC INDUSTRIES	2010	9.6267427694E10	21.79	6126.05228337582
BASIC INDUSTRIES	2011	9.3277620675E10	-58.6	8535.55187455352
BASIC INDUSTRIES	2012	7.9648935208E10	-68.79	8694.818993824214
BASIC INDUSTRIES	2013	8.1167036326E10	10.32	28486.424986144062
BASIC INDUSTRIES	2014	8.2010502666E10	-71.9	24380.207404688266
BASIC INDUSTRIES	2015	9.5592658398E10	-48.1	9538.711359273686
BASIC INDUSTRIES	2016	1.20096921114E11	13.83	7721.855295893219
BASIC INDUSTRIES	2017	1.00051884333E11	15.28	9101.40840866675
BASIC INDUSTRIES	2018	6.1239875499E10	-3.08	9883.137117548868
CAPITAL GOODS	2004	4.2642420682E10	10.13	6235.492238480657
CAPITAL GOODS	2005	4.2828857449E10	-5.57	7148.944215789201
CAPITAL GOODS	2006	5.1749654684E10	7.14	7704.190659975626
CAPITAL GOODS	2007	6.5271537325E10	5.89	9028.147336332922
CAPITAL GOODS	2008	9.0216887556E10	-48.31	7290.784741628547
CAPITAL GOODS	2009	9.6803739926E10	28.73	5181.6988391883315
CAPITAL GOODS	2010	9.0763630698E10	21.22	6858.207316732714
CAPITAL GOODS	2011	8.993998717E10	-12.43	7994.033017847347
CAPITAL GOODS	2012	7.6821481295E10	17.32	8345.535848169304

Figure 5: output Job 2 hive

PseudoCode Spark and output

1 upload csv and create dataframe doc1 and doc2 containing all the columns of the respective csv;

2 creation of join dataframes with addition of the "year" column and filtering YEAR¿ = 2004;

3 create dataframe "volumeComplessivo" for calculate "volume_complessivo_settore" grouping by sector and year;

4 create dataframe "DateMinMax" containing for each sector, ticker, year the minimum_date and the maximum_date;

5 create dataframe containing the sum of the closing prices grouping by sector, year and date;

6 create dataframe "a" e "b" containing for each sector and year the sum("close") of each ticker of a given sector with date == minimum_date (for a) or date == maximum_date (for b);

7 create dataframe for "percentualeVariazione" variable with the fields sector, year and "percentuale_variazione_annuale" sorted by sector and by year;

8 create final dataframe containing sector,year,"percentuale_variazione_annuale", "volume_complessivo_settore" and "quotazione_media_giornaliera" sorted by sector and by year by merging the relevant columns of the previous data frames created.

anno	sector	volume	quotazione_giornaliera
2004	BASIC INDUSTRIES	3.0767395827E10	19.657589234397317
2004	CAPITAL GOODS	4.2642420682E10	27.49989576648802
2004	CONSUMER DURABLES	1.0057518399E10	22.704145573729257
2004	CONSUMER NON-DURA...	2.804226368E10	29.727524987153874
2004	CONSUMER SERVICES	9.8440715492E10	39.10409861501838
2004	ENERGY	4.71715531E10	43.74027176198643
2004	FINANCE	4.4979537816E10	138.72628717180496
2004	HEALTH CARE	6.3110406513E10	433.81778886413997
2004	MISCELLANEOUS	1.8427942979E10	21.313258798557065
2004	N/A	3.1733968642E10	19.96436814167221
2004	PUBLIC UTILITIES	2.8589711906E10	42.244283983295176
2004	TECHNOLOGY	2.23368677466E11	69.70110563959625
2004	TRANSPORTATION	1.0146574E10	3875.09863624273
2005	BASIC INDUSTRIES	3.7457588379E10	28.28286369872514
2005	CAPITAL GOODS	4.2828857449E10	30.625832013784827
2005	CONSUMER DURABLES	1.0096843252E10	23.613072315948816
2005	CONSUMER NON-DURA...	3.4156900575E10	26.16217261672889
2005	CONSUMER SERVICES	9.7636202423E10	56.80403797896612
2005	ENERGY	6.666579755E10	66.29484903152019
2005	FINANCE	4.9407020954E10	138.83052695829232

Figure 6: output Job 2 Spark

Third Job

Variables and choices

We have chosen to use the following variables to find the respective project specifications for the third job:

- **percentuale_variazione_annuale** calculated for each sector, for each year and for each name by taking the value of close with minimum date (*prz_ini_chiusura*), value of close with maximum date (*prz_fin_chiusura*) and applying the following formula:

  ((*prz_fin_chiusura* - *prz_iniz_chiusura*) / *prz_iniz_chiusura*)*100

PseudoCode Map-Reduce and output

[**1**]**Map**

- **[1.1]** for each record of the csv "historical_stock_prices" it performs the filtering of the rows based on the date field such that 2016 ¡= anno ¡= 2018;
- **[1.2]** return ticker, close e date;
- **[1.3]** for each record of the csv historical_stocks retrieve the fields related to the ticker, name and sector;
- **[1.4]** combines records based on the ticker;
- **[1.5]** return name, date, sector, close;

[**2**]**Reduce**

- **[2.1]**groups ticker by name, year
- **[2.2]**for each year and for each company name calculates the "percentuale_variazione _annuale" rounded to the int part.
- **[2.3]**for each company, emit the trend that "percentuale_variazione_annuale", name,sector for the last tree years.

[3]**Map**

- **[3.1]** load reduce's output.

[4]**Reduce**

- **[4.1]**check company trends;
- **[4.2]**issue company pairs with the same trend, such that the company belongs to different sectors.

```
LINCOLN EDUCATIONAL SERVICES CORPORATION        STURM, RUGER & COMPANY, INC.    2016: -14%      2017: 4%        2018: 8%
LEAR CORPORATION        PENNYMAC FINANCIAL SERVICES, INC.       2016: 10%       2017: 33%       2018: -8%
HERSHEY COMPANY (THE)   SPECTRUM BRANDS HOLDINGS, INC.  2016: 18%       2017: 9%        2018: -11%
CALERES, INC.   GENTEX CORPORATION      2016: 25%       2017: 3%        2018: 11%
TERRENO REALTY CORPORATION      DUNKIN&#39; BRANDS GROUP, INC.  2016: 28%       2017: 23%       2018: 9%
SIGNATURE BANK  COMPASS MINERALS INTERNATIONAL, INC.    2016: 3%        2017: -8%       2018: -15%
MIDDLESEX WATER COMPANY INTERNATIONAL BANCSHARES CORPORATION    2016: 64%       2017: -3%       2018: 18%
```

Figure 7: output Job 3 Map Reduce

PseudoCode Hive and output

1 create table "prices" and "stock" and uploading data of the relative csv "*historical_stock_prices.csv*" and "*historical_stocks.csv*";

2 create table of joins between prices and stock tables with "ticker" as join condition with field selection: ticker, close, volume, data and sector and filtering year since 2016;

3 create dataframe "DateMinMax" containing for each sector,name, ticker, year the minimum_date and the maximum_date;

4 create tables "minClose" e "maxClose" containing for each sector,name and year the sum("close") variable for each ticker with the condition "date"== "mindate" (for minClose table) and "date"=="maxdate" (for maxClose table);

5 create table "percentuale" with the fields name, sector, year e "percentuale_variazione_annuale" sorting by name,sector and year;

6 create table "finalTable" and take name1,name2,year,"percentuale_variazione_annuale" FROM 2 tables "percentuale" renamed n1 and n2 with conditions: n1.name!=n2.name, n1.sector!=n2.sector, n1.anno==n2.anno e n1." percentuale_variazione_annuale" = n2."percentuale_variazione_annuale"

7 final query is the selection of the required fields FROM tree tables "finalTable" with conditions for name and years and sort by "name1" and "name2".

NOME 1	NOME 2	ANNO	PERC.	ANNO	PERC.	ANNO	PERC.
ADAMS NATURAL RESOURCES FUND. INC.	TALLGRASS ENERGY PARTNERS. LP	2016	15.0	2017	-3.0	2018	-3.0
AMDOCS LIMITED	COHEN & STEERS CLOSED-END OPPORTUNITY FUND. INC.	2016	7.0	2017	13.0	2018	-2.0
AMDOCS LIMITED	FLEXSHARES REAL ASSETS ALLOCATION INDEX FUND	2016	7.0	2017	13.0	2018	-2.0
AMERICAN AXLE & MANUFACTURING HOLDINGS.	CAPITALA FINANCE CORP.	2016	4.0	2017	-13.0	2018	3.0
AMERICAN EXPRESS COMPANY	ISHARES EXPONENTIAL TECHNOLOGIES ETF	2016	10.0	2017	32.0	2018	7.0
CALERES. INC.	GENTEX CORPORATION	2016	25.0	2017	3.0	2018	11.0
CANADIAN IMPERIAL BANK OF COMMERCE	POWERSHARES DWA BASIC MATERIALS MOMENTUM PORTFOLIO	2016	24.0	2017	18.0	2018	-4.0
CAPITALA FINANCE CORP. AMERICAN AXLE	MANUFACTURING HOLDINGS. INC.	2016	4.0	2017	-13.0	2018	3.0
CHINA MOBILE (HONG KONG) LTD.	MFS GOVERNMENT MARKETS INCOME TRUST	2016	-5.0	2017	-4.0	2018	-8.0
COHEN & STEERS CLOSED-END OPPORTUNITY	AMDOCS LIMITED	2016	7.0	2017	13.0	2018	-2.0
COMPASS MINERALS INTERNATIONAL. INC.	SIGNATURE BANK	2016	3.0	2017	-8.0	2018	-15.0
COMSTOCK RESOURCES. INC.	KAYNE ANDERSON ENERGY DEVELOPMENT COMPANY	2016	10.0	2017	-12.0	2018	4.0
DENNY&#39	WINMARK CORPORATION	2016	35.0	2017	4.0	2018	14.0
BRANDS GROUP. INC.	TERRENO REALTY CORPORATION	2016	28.0	2017	23.0	2018	9.0
EQUUS TOTAL RETURN. INC.	SALISBURY BANCORP. INC.	2016	16.0	2017	19.0	2018	-13.0
ERIE INDEMNITY COMPANY	POWERSHARES S&P SMALLCAP UTILITIES PORTFOLIO	2016	19.0	2017	9.0	2018	7.0
ESSA BANCORP. INC.	GUGGENHEIM CREDIT ALLOCATION FUND	2016	16.0	2017	-2.0	2018	4.0
GENTEX CORPORATION	CALERES. INC.	2016	25.0	2017	3.0	2018	11.0
GLOBAL INDEMNITY LIMITED	FIRST TRUST INDXX GLOBAL NATURAL RESOURCES INCOME ETF	2016	20.0	2017	7.0	2018	-1.0
GUGGENHEIM CREDIT ALLOCATION FUND	ESSA BANCORP. INC.	2016	16.0	2017	-2.0	2018	4.0
HAVERTY FURNITURE COMPANIES. INC.	KAYNE ANDERSON MLP INVESTMENT COMPANY	2016	13.0	2017	-5.0	2018	-2.0
HERSHEY COMPANY (THE) SPECTRUM BRANDS	HOLDINGS. INC.	2016	18.0	2017	9.0	2018	-11.0
INTERNATIONAL BANCSHARES CORPORATION	MIDDLESEX WATER COMPANY	2016	64.0	2017	-3.0	2018	18.0

Figure 8: output Job 3 Hive

PseudoCode Spark and output

1 upload csv and create dataframe doc1 and doc2 containing all the columns of the respective csv;

2 creation of join dataframes with addition of the "year" column and filtering YEAR¿ = 2016;

3 create dataframe "DateMinMax" containing for each sector, ticker, year the minimum_date and the maximum_date;

4 create dataframe "a" e "b" containing for each sector and year the sum("close") of each ticker of a given sector with date == minimum_date (for a) or date == maximum_date (for b);

5 create dataframe for "percentualeVariazione" variable with the fields sector,name, year and "percentuale_variazione_annuale" sorted by sector and by year;

6 create final dataframe like cross-join of 2 dataframe "percentualeVariazione" with conditions: different name, different sector, same year and same "percentuale_variazione_annuale".

name	name	anno	perc.	anno	perc.	anno	perc.
ADAMS NATURAL	TALLGRASS	2016	15.0	2017	-3.0	2018	-3.0
AMDOCS LIMITED	COHEN	2016	7.0	2017	13.0	2018	-2.0
AMDOCS LIMITED	FLEXSHARES	2016	7.0	2017	13.0	2018	-2.0
AMERICAN AXLE	CAPITALA FIN	2016	4.0	2017	-13.0	2018	3.0
AMERICAN EXPRES	ISHARES EXP	2016	10.0	2017	32.0	2018	7.0
CALERES. INC.	GENTEX CORP	2016	25.0	2017	3.0	2018	11.0
CANADIAN IMPE	POWERSHARES DWA	2016	24.0	2017	10.0	2018	-4.0
CAPITALA FIN	AMERICAN AXLE	2016	4.0	2017	-13.0	2018	3.0
CHINA MOBILE	MFS GOVERNMENT	2016	-5.0	2017	-4.0	2018	-8.0

Figure 9: output Job 3 Spark

Execution times and statistics

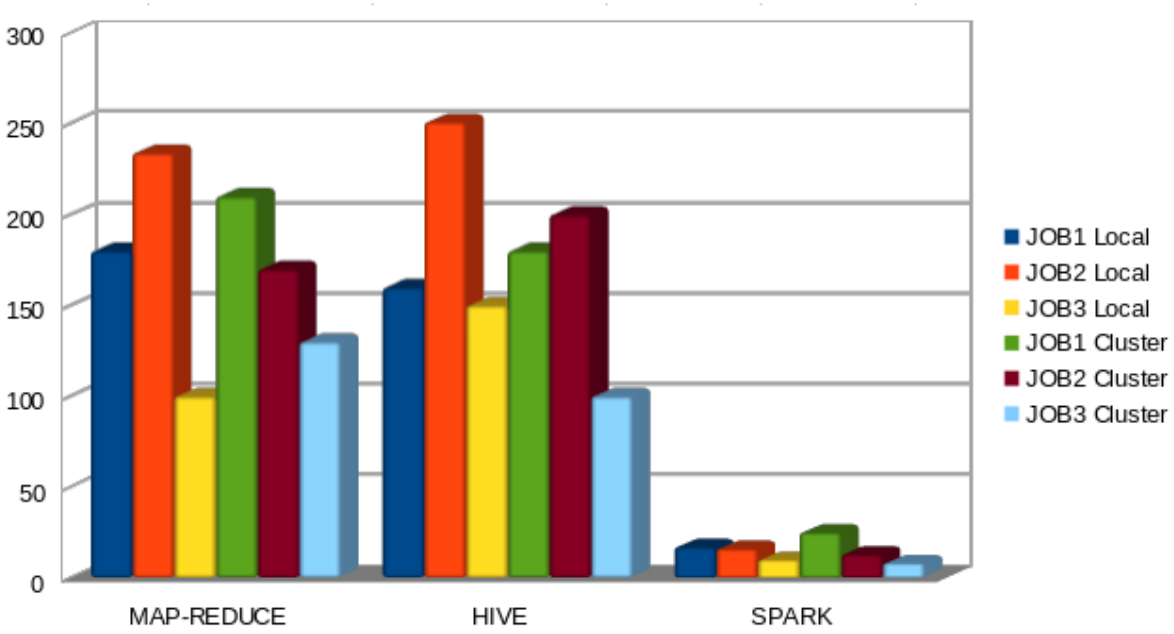
To testing the timing of various jobs and various technologies have been used plus csv. In particular, the csv "historical_stock_prices.csv" was also halved and reduced to a third, thus creating copies with input for testing variable variables and allocating them for delivery.

Map reduce vs. Hive vs. Spark

xxxxxxxxxxxxxxx	JOB1 Local	JOB2 Local	JOB3 Local	JOB1 Cluster	JOB2 Cluster	JOB3 Cluster
MAP-REDUCE	180	234	100	210	170	130
HIVE	160	251	150	180	200	100
SPARK	17	16	10	25	13	8

Figure 10: execution time in seconds

The various times reported for the Hive and Spark technologies are also formed by the loading time of the CSV and in the case of the second and third exercise also by the time of Join of the two CSVs, which in particular worsens the execution time of Hive. The big difference in execution times between Spark and the two other technologies can already be analyzed. This difference resolved even more evident going to make the "bar Charts" to compare the time as shown below.



As mentioned, the CSVs have been divided and partitioned to take note of the execution times on several inputs. The following are the times and barcharts for each csv and for each job of all technologies.

Spark execution time

CSV TYPE	JOB1	JOB2	JOB3
CSV ORIGINAL	17	16	10
1/2 CSV	5	8	4
1/3 CSV	2	5	3

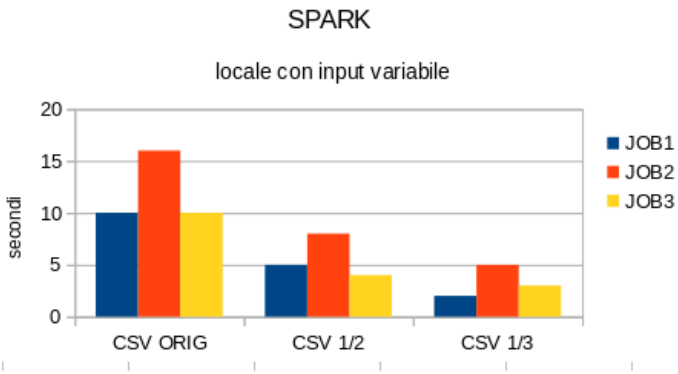


Figure 11: Time in second for spark

Hive execution time

CSV TYPE	JOB1	JOB2	JOB3
CSV ORIGINAL	160	251	150
1/2 CSV	110	115	88
1/3 CSV	63	80	62

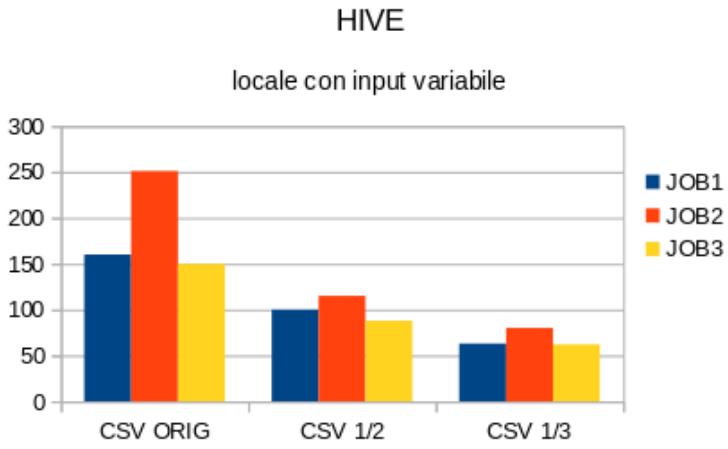


Figure 12: Time in second for Hive

Map Reduce execution time

CSV TYPE	JOB1	JOB2	JOB3
CSV ORIGINAL	180	234	100
1/2 CSV	124	73	35
1/3 CSV	73	35	23

Figure 13: Time in second for Map Reduce