

Random Sampling for Group-By Queries

Implemented algorithm that dynamically computes a random sampling for a single aggregation, multiple group by using [Apache Flink](#) and [Apache Kafka](#). The algorithm implementation was based on the following [paper](#).

Our approach parses a stream of twice and performs sampling for group by queries. Given that the stream must be parsed twice, we can use this implementation **only on bounded streams**. During the first pass we compute average, standard deviation, γ_i statistics for each stratum (group by) and total γ (sum of all γ_i values). In the second pass, we can perform sampling using all statistics precomputed in first pass.

Between stages we use Apache Kafka to produce-consume data. The initial stream of data can be parsed from .csv files.

Implementation Analysis

CSV_Parser

Description

Class is responsible to parse data from a .csv file and appends them to a certain Kafka topic

Arguments

Available arguments are listed below

Required	Description
-csv-path	full path to the .csv we wish
-topic	Kafka topic to write data parsed from .csv

Optional	Description	Default
-ip	give ip for our server to work	localhost:9092
-header-exists	Ignore first line when there is header option (0 when there is no header, 1 when there is header to ignore in csv)	1(header exists)

FirstAlgorithmPass

Description

This is the implementation of the first required job for our algorithm. In this job we parse data for the first time (bounded stream) and compute required aggregation such as average, count for each stratum (each stratum is formed by each distinct a group by attribute). We also compute values γ_i for each stratum and γ (sum of γ_i for all stratum) which are required in the second pass of the algorithm

Arguments

Available arguments are listed below

Required	Description
-all-attributes	All fields contained in the parsed .csv file(all csv fields) comma seperated
-keys	all keys to create stratum from group bys comma seperated(from attributes)
-aggr	field for aggregation from attributes(Currently only one is supported)

Optional	Description	Default
-p	Parallelism in execution environment	4
-input-topic	Kafka topic that includes our main stream	input-topic-job1
-output-topic	Kafka topic to export the aggregations required in the second pass	output-topic-job1
-consumer-group	Kafka consumer group	KafkaCsvProducer
-ip	Give ip for our server to work	localhost:9092
-windows-time	User-defined time for windows	30

NOTE: User must ensure that there is adequate time in the time window for all entries to be processed

IMPORTANT: During this pass kafka reads our stream as defined from -input-topic and writes the result(aggregation values) to a new topic as defined from output-topic-job1. However, to support dynamic group by, we perform some modification to initial stream and create another topic with name _<-input-topic>.(e.x if -input-topic= “new-topic”, our custom stream is named _new-topic).As input in the second algorithm pass we use the topic containing the initial stream(!). This detail is important, as the created topic has the default properties from Kafka and --partition and --replication value is always 1. So, in order to increase parallelism and partition data evenly the user must define previously topics with custom properties

SecondAlgorithmPass

Description

In this second pass of the algorithm, we parse once again the initial data stream. Also we use aggregation data extracted from first pass and combined we execute our demanded algorithm

Arguments

Available arguments are listed below

Optional	Description	Default
-p	Parallelism in execution environment	4
-input-topic	Kafka topic that includes our main stream	input-topic-job1
-output-topic	Kafka topic to export the final results after job2 executes	output-topic-job2
-aggr-topic	Kafka topic to access the aggregations required in the second pass	output-topic-job2
-consumer-group	Kafka consumer group	KafkaCsvProducer
-ip	Give ip for our server to work	localhost:9092
-windows-time	User-defined time for windows	30
-M	Variable demanded in the algorithm	20

NOTE: User must ensure that there is adequate time in the time window for all entries to be processed

Execution Example on Linux

As first step activate zookeeper and kafka servers

Window 1

- Start zookeeper server

```
$ /<Kafka_path>/bin/zookeeper-server-start.sh /<Kafka_path>/config/zookeeper.properties
```

Window 2

- Start Kafka server

```
$ /<Kafka_path>/bin/kafka-server-start.sh /<Kafka_path>/config/server.properties
```

Now kafka servers are up and running waiting for new topics to be created.

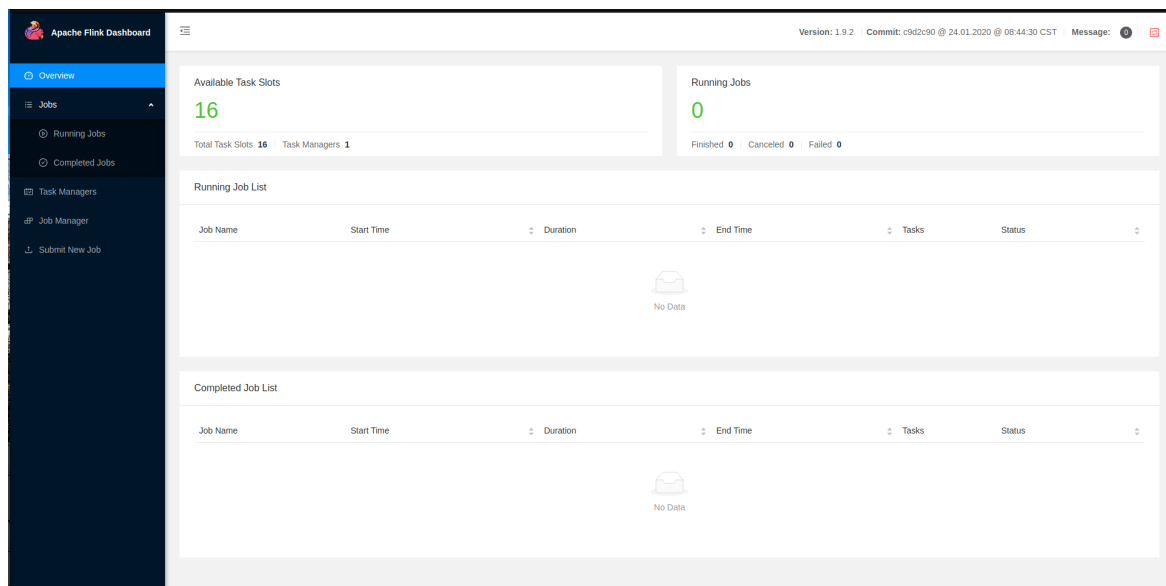
Window 3

- Start flink cluster

```
$ /<flink_path>/bin/start-cluster.sh
```

To confirm that flink servers started successfully open any browser and type <http://localhost:8081/>

It should look something like that



Window 4

Open a new terminal in the project directory folder `/ECE622/` and execute the following command to compile and build project

```
$ mvn clean package
```

BUILD SUCCESS message should show up. This commands generates a .jar files which will be used to execute our code from command line. Jar file can be found in the directory

```
/ECE622/target/ECE622-1.0-SNAPSHOT.jar
```

The execution should start from CSV_Parser to parse data from a .csv file and append them to a kafka topic

```
$ /<flink_path>/bin/flink run -c utils.CSV_Parser <project_path>/ECE622/target/ECE622
```

In our example, we use population.csv file located in `/MyDocs/` directory. See sections above to check all available arguments for CSV_Parser.

Procedure finished message as shown below demonstrates that parser completed

```
skalogerakis@skl ➔ /home/skalogerakis/flink-1.9.2-bin-scala_2.12/flink-1.9.2/bin/flink run -c utils.CSV_Parser /home/skalogerakis/TUC_Projects/TUC_Advanced_Database_Systems/ECE622/target/ECE622-1.0-SNAPSHOT.jar -c sv-path /home/skalogerakis/TUC_Projects/TUC_Advanced_Database_Systems/MyDocs/population.csv -topic input-topics -p 4
Starting execution of program
Feb 27, 2020 12:45:05 PM utils.CSV_Parser LogConfig
CONFIG: Configuration done.
Invalid headerExists option. Proceed with the default option(Header exists)
Ignore first header line
Procedure finished
```

NOTE: In order to avoid unexpected behaviour, wait for steps to complete and generate output

In order to check that kafka received our data we can use the following command

```
$ <kafka_path>/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic <t
```

We should see all the data contained in that specific topic. In our execution `population.csv` contains 70080 entries(excluding header) and as we can see all entries were parsed correctly.

```
Terminal
File Edit View Search Terminal Help
2013,8,"Nou Barris",48,"la Guineueta","Female", ">=95",13
2013,8,"Nou Barris",49,"Canyelles","Female", ">=95",3
2013,8,"Nou Barris",50,"les Roquetes","Female", ">=95",12
2013,8,"Nou Barris",51,"Verdun","Female", ">=95",15
2013,8,"Nou Barris",52,"la Prosperitat","Female", ">=95",23
2013,8,"Nou Barris",53,"la Trinitat Nova","Female", ">=95",17
2013,8,"Nou Barris",54,"Torre Baró","Female", ">=95",4
2013,8,"Nou Barris",55,"Ciutat Meridiana","Female", ">=95",4
2013,8,"Nou Barris",56,"Vallbona","Female", ">=95",5
2013,9,"Sant Andreu",57,"la Trinitat Vella","Female", ">=95",9
2013,9,"Sant Andreu",58,"Baró de Viver","Female", ">=95",3
2013,9,"Sant Andreu",59,"el Bon Pastor","Female", ">=95",8
2013,9,"Sant Andreu",60,"Sant Andreu","Female", ">=95",103
2013,9,"Sant Andreu",61,"la Sagrera","Female", ">=95",45
2013,9,"Sant Andreu",62,"el Congrés i els Indians","Female", ">=95",35
2013,9,"Sant Andreu",63,"Navas","Female", ">=95",39
2013,10,"Sant Martí",64,"el Camp de l'Arpa del Clot","Female", ">=95",91
2013,10,"Sant Martí",65,"el Clot","Female", ">=95",36
2013,10,"Sant Martí",66,"el Parc i la Llacuna del Poblenou","Female", ">=95",18
2013,10,"Sant Martí",67,"la Vila Olímpica del Poblenou","Female", ">=95",9
2013,10,"Sant Martí",68,"el Poblenou","Female", ">=95",48
2013,10,"Sant Martí",69,"Diagonal Mar i el Front Marítim del Poblenou","Female", ">=95",21
2013,10,"Sant Martí",70,"el Besòs i el Maresme","Female", ">=95",25
2013,10,"Sant Martí",71,"Provençals del Poblenou","Female", ">=95",31
2013,10,"Sant Martí",72,"Sant Martí de Provençals","Female", ">=95",38
2013,10,"Sant Martí",73,"la Verneda i la Pau","Female", ">=95",47
^CProcessed a total of 70080 messages
skalogerakis@skl
```

Next step is to execute the first algorithm pass using the following command

```
$ /<flink_path>/bin/flink run -c tuc.FirstAlgorithmPass /<project_path>/ECE622/target
```

In our example we use default topic input and output values. See sections above to check all available arguments for FirstAlgorithmPass.

As expected from our default topics we can see data in the topics **output_topic_job1(aggregation values)** and **_input-topic-job1(Initial custom stream)** after execution

```
Terminal
File Edit View Search Terminal Help
skalogerakis@skl /home/skalogerakis/kafka_2.12-2.4.0/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic
output-topic-job1 --from-beginning
ETxample,2;229.62222222222223;107.79574463645766;5760.0;0.46944822497247596;7.6495376103380455
Horta-Guinardó,7;79.28702651515151;72.47781597814203;10560.0;0.9141194866765717;7.6495376103380455
Nou Barris,8;66.21570512820513;53.75210440259919;12480.0;0.8117727402966677;7.6495376103380455
Sant Andreu,9;109.17380952380952;103.68816501442659;6720.0;0.9497531089799831;7.6495376103380455
Ciutat Vella,1;131.665625;120.94349001369487;38400.0;0.9185654191342263;7.6495376103380455
Gràcia,6;125.81;110.6078232615879;4800.0;0.8791655930497408;7.6495376103380455
Sarrià-Sant Gervasi,5;127.63298611111111;82.71484455125753;5760.0;0.6480679256320931;7.6495376103380455
Sants-Montjuïc,3;118.11888028833334;93.15440284032597;7680.0;0.7886495594609767;7.6495376103380455
Les Corts,4;141.68229166666666;94.70396913407731;2880.0;0.6684248823197017;7.6495376103380455
Sant Martí,10;121.72447916666667;73.22587646524781;9600.0;0.601570669815609;7.6495376103380455
```

Topic output-topic-job1

```
File Edit View Search Terminal Help
Nou Barris,8;4.0;2013,8,54,Torre Baró,Female,>=95,4,2013,Nou Barris,54,Torre Baró,Female,>=95,4
Nou Barris,8;4.0;2013,8,55,Ciutat Meridiana,Female,>=95,4,2013,Nou Barris,55,Ciutat Meridiana,Female,>=95,4
Nou Barris,8;5.0;2013,8,56,Vallbona,Female,>=95,5,2013,Nou Barris,56,Vallbona,Female,>=95,5
Sant Andreu,9;9.0;2013,9,57,la Trinitat Vella,Female,>=95,9,2013,Sant Andreu,57,la Trinitat Vella,Female,>=95,9
Sant Andreu,9;3.0;2013,9,58,Baró de Viver,Female,>=95,3,2013,Sant Andreu,58,Baró de Viver,Female,>=95,3
Sant Andreu,9;8.0;2013,9,59,el Bon Pastor,Female,>=95,8,2013,Sant Andreu,59,el Bon Pastor,Female,>=95,8
Sant Andreu,9;103.0;2013,9,60,Sant Andreu,Female,>=95,103,2013,Sant Andreu,60,Sant Andreu,Female,>=95,103
Sant Andreu,9;45.0;2013,9,61,la Sagrera,Female,>=95,45,2013,Sant Andreu,61,la Sagrera,Female,>=95,45
Sant Andreu,9;35.0;2013,9,62,el Congrés i els Indians,Female,>=95,35,2013,Sant Andreu,62,el Congrés i els Indians,Female,>=95,35
Sant Andreu,9;39.0;2013,9,63,Navas,Female,>=95,39,2013,Sant Andreu,63,Navas,Female,>=95,39
Sant Martí,10;91.0;2013,10,64,el Camp de l'Arpa del Clot,Female,>=95,91,2013,Sant Martí,64,el Camp de l'Arpa del Clot,Female,>=95,91
Sant Martí,10;36.0;2013,10,65,el Clot,Female,>=95,36,2013,Sant Martí,65,el Clot,Female,>=95,36
Sant Martí,10;18.0;2013,10,66,el Parc i la Llacuna del Poblenou,Female,>=95,18,2013,Sant Martí,66,el Parc i la Llacuna del Poblenou,Female,>=95,18
Sant Martí,10;9.0;2013,10,67,la Vila Olímpica del Poblenou,Female,>=95,9,2013,Sant Martí,67,la Vila Olímpica del Poblenou,Female,>=95,9
Sant Martí,10;48.0;2013,10,68,el Poblenou,Female,>=95,48,2013,Sant Martí,68,el Poblenou,Female,>=95,48
Sant Martí,10;21.0;2013,10,69,Diagonal Mar i el Front Marítim del Poblenou,Female,>=95,21,2013,Sant Martí,69,Diagonal Mar i el Front Marítim del Poblenou,Female,>=95,21
Sant Martí,10;25.0;2013,10,70,el Besòs i el Maresme,Female,>=95,25,2013,Sant Martí,70,el Besòs i el Maresme,Female,>=95,25
Sant Martí,10;31.0;2013,10,71,Provençals del Poblenou,Female,>=95,31,2013,Sant Martí,71,Provençals del Poblenou,Female,>=95,31
Sant Martí,10;38.0;2013,10,72,Sant Martí de Provençals,Female,>=95,38,2013,Sant Martí,72,Sant Martí de Provençals,Female,>=95,38
Sant Martí,10;47.0;2013,10,73,la Verneda i la Pau,Female,>=95,47,2013,Sant Martí,73,la Verneda i la Pau,Female,>=95,47
^CProcessed a total of 70080 messages
```

Topic _input-topic-job1

Final step is to execute the second algorithm pass using the following command

```
$ <flink_path>/bin/flink run -c tuc.SecondAlgorithmPass <project_path>/ECE622-1.0-SNA
```

In our example we use default topic input and output values. See sections above to check all available arguments for FirstAlgorithmPass.

We can see final output results in topic **output-topic-job2**

```
skalogerakis@skali:~/TUC_Projects/TUC_Advanced_Database_Systems/mydocs/Scripts$ /home/skalogerakis/kafka_2.12-2.4.0/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic output-topic-job2 --from-beginning
Example,2,2017,2,5,el Fort Plenc,Male,0-4,124,2017,Example,5,el Fort Plenc,Male,0-4,124
Nou Barris,8,2017,8,44,Vilapicina i la Torre Llobeta,Male,0-4,85,2017,Nou Barris,44,Vilapicina i la Torre Llobeta,Male,0-4,85
Nou Barris,8,2017,8,45,Porta,Male,0-4,101,2017,Nou Barris,45,Porta,Male,0-4,101
Les Corts,4,2017,4,19,les Corts,Male,0-4,173,2017,Les Corts,19,les Corts,Male,0-4,173
Les Corts,4,2017,4,20,la Maternitat i Sant Ramon,Male,0-4,85,2017,Les Corts,20,la Maternitat i Sant Ramon,Male,0-4,85
Ciutat Vella,1,2017,1,1,el Raval,Male,0-4,224,2017,Ciutat Vella,1,el Raval,Male,0-4,224
Ciutat Vella,1,2017,1,2,el Barri Gòtic,Male,0-4,50,2017,Ciutat Vella,2,el Barri Gòtic,Male,0-4,50
Sant Martí,10,2017,10,64,el Camp de l'Arpa del Clot,Male,0-4,151,2017,Sant Martí,64,el Camp de l'Arpa del Clot,Male,0-4,151
Sant Martí,10,2017,10,65,el Clot,Male,0-4,92,2017,Sant Martí,65,el Clot,Male,0-4,92
Sants-Montjuïc,3,2017,3,11,el Poble Sec,Male,0-4,180,2017,Sants-Montjuïc,11,el Poble Sec,Male,0-4,180
Sants-Montjuïc,3,2017,3,12,la Marina del Prat Vermell,Male,0-4,10,2017,Sants-Montjuïc,12,la Marina del Prat Vermell,Male,0-4,10
Horta-Guinardó,7,2017,7,33,el Baix Guinardó,Male,0-4,117,2017,Horta-Guinardó,33,el Baix Guinardó,Male,0-4,117
Horta-Guinardó,7,2017,7,34,Can Baró,Male,0-4,27,2017,Horta-Guinardó,34,Can Baró,Male,0-4,27
Sant Andreu,9,2017,9,57,la Trinitat Vella,Male,0-4,61,2017,Sant Andreu,57,la Trinitat Vella,Male,0-4,61
Sant Andreu,9,2017,9,58,Baró de Viver,Male,0-4,9,2017,Sant Andreu,58,Baró de Viver,Male,0-4,9
Sarrià-Sant Gervasi,5,2017,5,22,Vallvidrera, el Tibidabo i les Planes,Male,0-4,16,2017,Sarrià-Sant Gervasi,22,Vallvidrera, el Tibidabo i les Planes,Male,0-4,16
Sarrià-Sant Gervasi,5,2017,5,23,Sarrià,Male,0-4,119,2017,Sarrià-Sant Gervasi,23,Sarrià,Male,0-4,119
Gràcia,6,2017,6,28,Vallcarca i els Penitents,Male,0-4,66,2017,Gràcia,28,Vallcarca i els Penitents,Male,0-4,66
Gràcia,6,2017,6,29,el Coll,Male,0-4,37,2017,Gràcia,29,el Coll,Male,0-4,37
```

We can also verify our result from the web UI of flink. In the image below we notice our two jobs running successfully

The screenshot shows the Apache Flink Dashboard with the following details:

- Overview:** 8 Available Task Slots, 2 Running Jobs.
- Running Job List:**

Job Name	Start Time	Duration	End Time	Tasks	Status
SecondAlgorithmPass	2020-02-27 14:13:16	18s	-	28 / 28	RUNNING
Streaming FirstAlgorithmPass	2020-02-27 14:12:01	1m 33s	-	20 / 20	RUNNING

Each separate job preserves metrics and statistics such as Records received/sent that match our desired behaviour

The screenshot shows the details for the **Streaming FirstAlgorithmPass** job, which is in a **RUNNING** state. The job ID is 6020e4e43701a69eccc4b930b9a737. The start time is 2020-02-27 14:12:01 and the duration is 3m 52s.

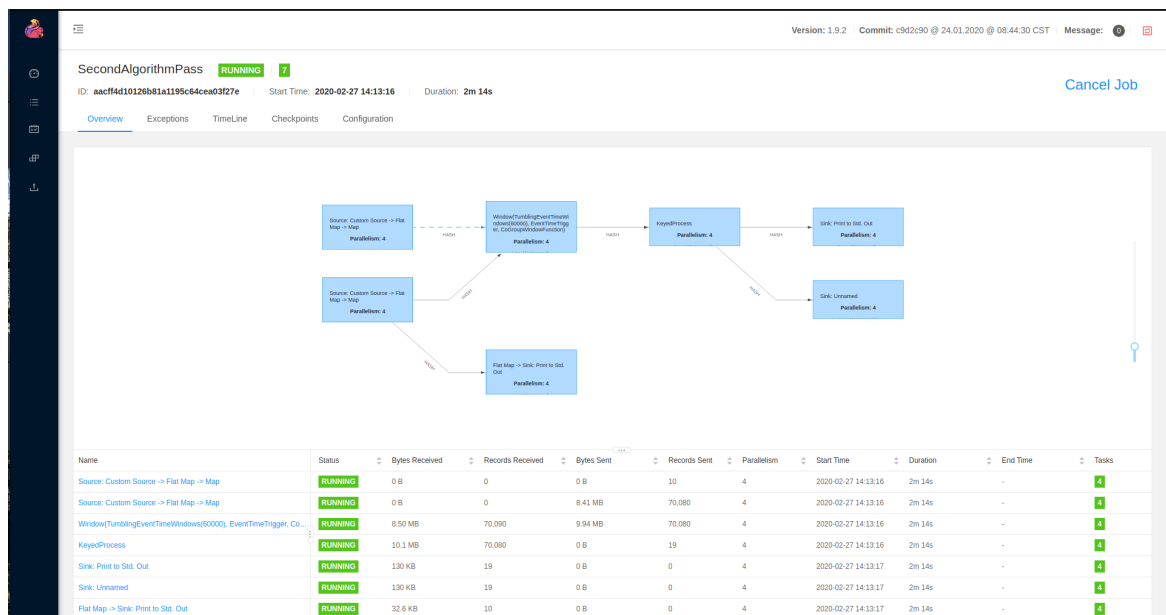
Task Graph:

```
graph LR
    Source[Source: Custom Source -> Flat Map -> Sink: Unnamed] --> P1[Parallelism: 4]
    P1 --> P2[Parallelism: 4]
    P1 --> P3[Parallelism: 4]
    P2 --> P4[Parallelism: 4]
    P3 --> P4
    P4 --> Sink[Sink: Unnamed]
```

Metrics Table:

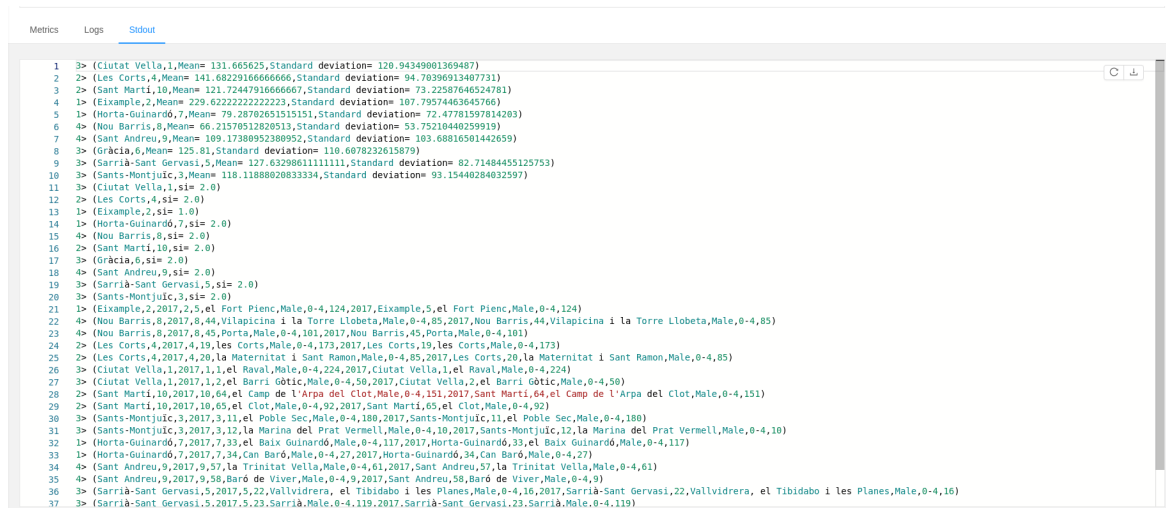
Name	Status	Bytes Received	Records Received	Bytes Sent	Records Sent	Parallelism	Start Time	Duration	End Time	Tasks
Source: Custom Source -> Flat Map -> Sink: Unnamed	RUNNING	0 B	0	0.88 MB	0	4	2020-02-27 14:12:01	3m 51s	-	4
Window(TumblingEventTimeWindows(60000), EventTimeTrigger, min)	RUNNING	8.95 KB	70,080	0 B	20	4	2020-02-27 14:12:01	3m 51s	-	4
Window(TumblingEventTimeWindows(60000), EventTimeTrigger, Su...	RUNNING	221 KB	10	0 B	1	4	2020-02-27 14:12:01	3m 51s	-	4
Window(TumblingEventTimeWindows(60000), EventTimeTrigger, Co...	RUNNING	441 KB	11	0 B	0	4	2020-02-27 14:12:01	3m 51s	-	4
Flat Map -> Sink: Print to Std. Out	RUNNING	221 KB	10	0 B	0	4	2020-02-27 14:12:01	3m 51s	-	4

FirstAlgorithmPass

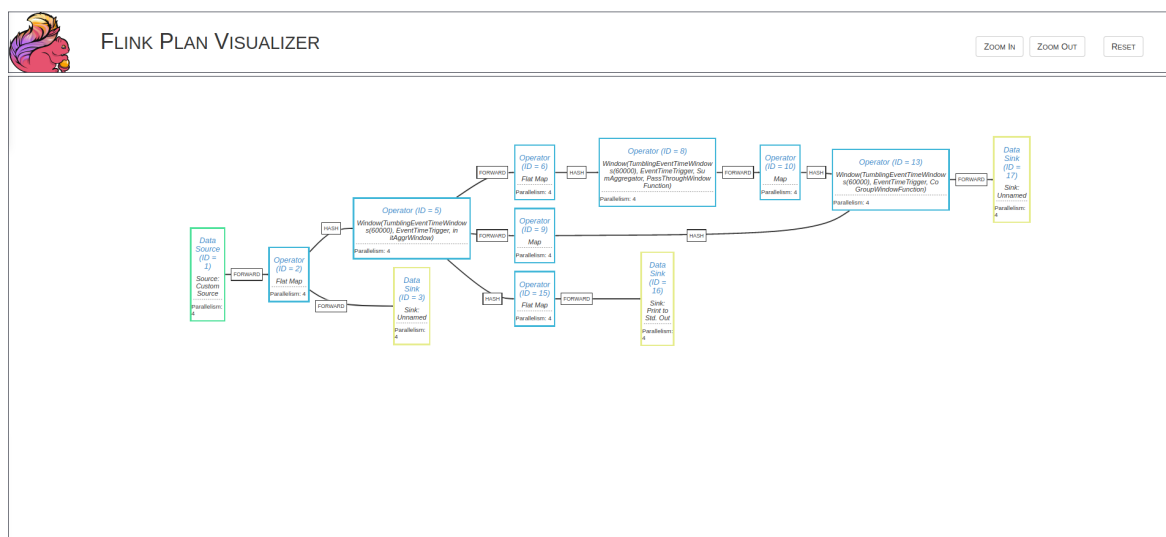


SecondAlgorithmPass

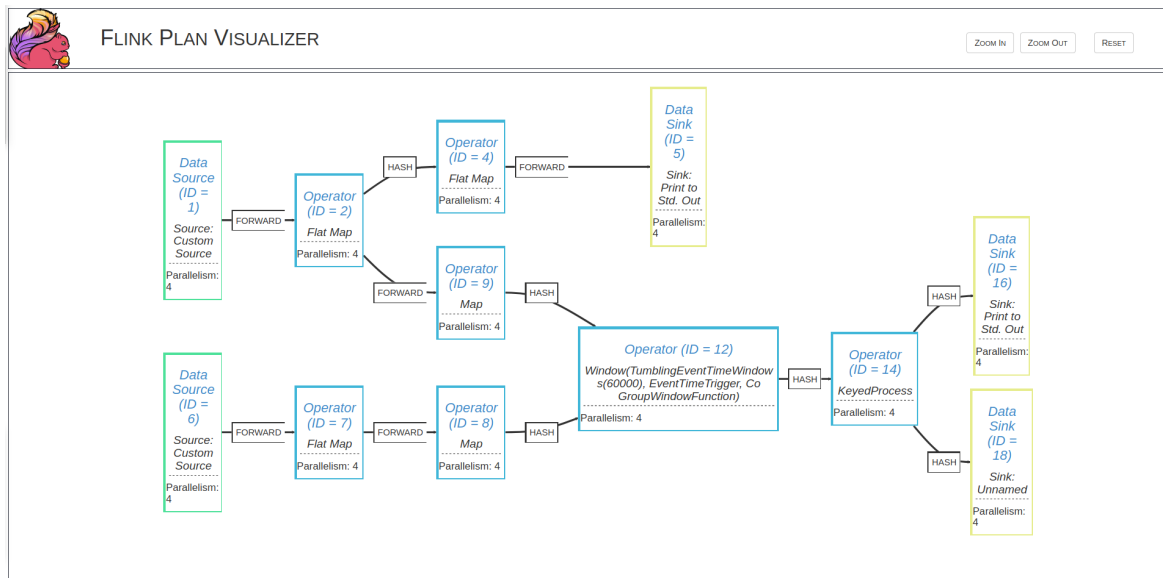
In addition, task manager preserves Stdout logs producing output we wish for both separate jobs in one place



We also demonstrate for both jobs the plan visualizer<https://flink.apache.org/visualizer/>



FirstAlgorithmPass



SecondAlgorithmPass

NOTE: In our examples we didn't create kafka topic. Instead Kafka created topics when a topic that didn't exist encountered. In this case, kafka has default values with partition 1 and replication 1. In case you wish to create custom topic use one of the following command

```
$ <kafka_path>/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topi
```

*NOTE: To make things easier created two scripts in directory `/MyDocs/Scripts/` `jobsExec` and `kafkaExec`. `jobsExec` executes the example as shown above and requires **four parameters** `<kafka_path>` `<flink_path>` `<project_path>` `<csv_path>` to execute (open script to check example with parameters). `Kafka exec` is used to show all the different **default topics** and requires **one parameter** `<kafka_path>`*