DEBS 2012 Solution: Source Definition

In die original solution, there was a protobuf base server. This server is no longer available. To process the data in any case we provide a simple example with a csv file:

```
#PARSER PQL
#QUERY
gchSource := TIMESTAMPORDERVALIDATE({
                  debug=true,
                  debugmode=0
                },
                CSVFILESOURCE({
                    source = 'GrandChallengeDataPoint',
                    delimiter = '\t',
                    filename = '${WORKSPACEPROJECT}/allData.txt',
                    dateformat="yyyy-MM-dd'T'HH:mm:ss.SSSSSSSzzz",
                    ///options=[['scheduler.delay','1000']],
                    schema=[
                      ['ts','StartTimestampString'],
                      ['index','long'],
                      ['mf01','integer'],
                      ['mf02','integer'],
                      ['mf03','integer'],
                      ['pc13','integer'],
                      ['pc14','integer'],
                      ['pc15','integer'],
                      ['pc25','long'],
                      ['pc26','long'],
                      ['pc27','long'],
                      ['res','long'],
                      ['bm05','Boolean'],
                      ['bm06','Boolean'],
                      ['bm07','Boolean'],
                      ['bm08','Boolean'],
                      ['bm09','Boolean'],
                      ['bm10','Boolean'],
                      ['pp01','Boolean'],
                      ['pp02','Boolean'],
                      ['pp03','Boolean'],
                      ['pp04','Boolean'],
                      ['pp05','Boolean'],
                      ['pp06','Boolean'],
                      ['pp07','Boolean'],
                      ['pp08','Boolean'],
                      ['pp09','Boolean'],
                      ['pp10','Boolean'],
                      ['pp11','Boolean'],
                      ['pp12','Boolean'],
                      ['pp13','Boolean'],
                      ['pp14','Boolean'],
                      ['pp15','Boolean'],
                      ['pp16','Boolean'],
                      ['pp17','Boolean'],
                      ['pp18','Boolean'],
                      ['pp19','Boolean'],
                      ['pp20','Boolean'],
                      ['pp21','Boolean'],
                      ['pp22','Boolean'],
                      ['pp23','Boolean'],
                      ['pp24','Boolean'],
                      ['pp25','Boolean'],
                      ['pp26','Boolean'],
                      ['pp27','Boolean'],
                      ['pp28','Boolean'],
                      ['pp29','Boolean'],
                      ['pp30','Boolean'],
                      ['pp31','Boolean'],
                      ['pp32','Boolean'],
                      ['pp33','Boolean'],
                      ['pp34','Boolean'],
```

```
                    ['pp35','Boolean'],
                    ['pp36','Boolean'],
                    ['pc01','Boolean'],
                    ['pc02','Boolean'],
                    ['pc03','Boolean'],
                    ['pc04','Boolean'],
                    ['pc05','Boolean'],
                    ['pc06','Boolean'],
                    ['pc19','Boolean'],
                    ['pc20','Boolean'],
                    ['pc21','Boolean'],
                    ['pc22','Boolean'],
                    ['pc23','Boolean'],
                    ['pc24','Boolean']
                ]
            }
        )
    )
```

In this case, the csv file is found in the workspace folder.

Remark:

- The input file does not contains long values for the time stamp but a String date format. The format of the input string is given with the option: dateformat.
- The input file contains some time stamps out of order. So the TimestampOrderValidate Operator is used to suppress these values.

In the original version, we used google protobuf, which is not supported anymore

We provide two solutions. The first one reads its input by providing a Google Protobuf server.

Solution with Protobuf

To use this solution, the following features need to be installed:

- de.uniol.inf.is.odysseus.debs2012.feature
- de.uniol.inf.is.odysseus.wrapper.protobuf.feature
- de.uniol.inf.is.odysseus.monolithic.feature

```
#PARSER PQL
#QUERY
gchSource := RECEIVE({
    SOURCE = 'GrandChallengeDataPoint',
    DataHandler = 'Tuple',
    TRANSPORT='ProtobufServer',
    OPTIONS=[
        ['type','debs.challenge.msg.CDataPoint'],
        ['port','9999'],
        ['BaseTimeUnit','NANOSECONDS']],
    SCHEMA=[
        ['ts','StartTimeStamp'],
        ['index','long'],
        ['mf01','integer'],
        ['mf02','integer'],
        ['mf03','integer'],
        ['pc13','integer'],
        ['pc14','integer'],
        ['pc15','integer'],
        ['pc25','long'],
```

```
        ['pc26','long'],
        ['pc27','long'],
        ['res','long'],
        ['bm05','Boolean'],
        ['bm06','Boolean'],
        ['bm07','Boolean'],
        ['bm08','Boolean'],
        ['bm09','Boolean'],
        ['bm10','Boolean'],
        ['pp01','Boolean'],
        ['pp02','Boolean'],
        ['pp03','Boolean'],
        ['pp04','Boolean'],
        ['pp05','Boolean'],
        ['pp06','Boolean'],
        ['pp07','Boolean'],
        ['pp08','Boolean'],
        ['pp09','Boolean'],
        ['pp10','Boolean'],
        ['pp11','Boolean'],
        ['pp12','Boolean'],
        ['pp13','Boolean'],
        ['pp14','Boolean'],
        ['pp15','Boolean'],
        ['pp16','Boolean'],
        ['pp17','Boolean'],
        ['pp18','Boolean'],
        ['pp19','Boolean'],
        ['pp20','Boolean'],
        ['pp21','Boolean'],
        ['pp22','Boolean'],
        ['pp23','Boolean'],
        ['pp24','Boolean'],
        ['pp25','Boolean'],
        ['pp26','Boolean'],
        ['pp27','Boolean'],
        ['pp28','Boolean'],
        ['pp29','Boolean'],
        ['pp30','Boolean'],
        ['pp31','Boolean'],
        ['pp32','Boolean'],
        ['pp33','Boolean'],
        ['pp34','Boolean'],
        ['pp35','Boolean'],
        ['pp36','Boolean'],
        ['pc01','Boolean'],
        ['pc02','Boolean'],
        ['pc03','Boolean'],
        ['pc04','Boolean'],
        ['pc05','Boolean'],
        ['pc06','Boolean'],
        ['pc19','Boolean'],
        ['pc20','Boolean'],
        ['pc21','Boolean'],
        ['pc22','Boolean'],
        ['pc23','Boolean'],
        ['pc24','Boolean']
        ]
    }
)
```

The source definition is done with a Receive operator. This means, the source sends its information to Odysseus.

The Data is transported by a Google Protobuf handler. Opposed to many other approaches, this handler combines Transport handler and Protocol handler.

Further information for the Protobuf Server is given with the options:

```
    OPTIONS=[
        ['type','debs.challenge.msg.CDataPoint'],
        ['port','9999'],
        ['BaseTimeUnit','NANOSECONDS']],
```

The type is the compiled version of the protobuf specification file given by the DEBS organisators. Is must be provided inside a special bundle.

Port if the port on which der Server is started on the machine where the query is executed. In a monolithic version this would be on localhost, in a server version this would be on the Odysseus server.

Odysseus can read the timestamps from the input schema. If no further information is given, the unit of the timestamp is assumed to be milliseconds. With BaseTimeUnit the real time unit can be set.

Finally, the schema definition represents the internal format of the data inside of Odysseus, i.e. there are tuples with the given schema.